universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

„Ritmo y estructura de la comedia áurea: posibilidades y límites del análisis digital automático del teatro español clásico“

verfasst von / submitted by

Lic. Mag. Fernando Sanz-Lázaro

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der Philosophie (Dr.phil.)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 792 236
Dissertationsgebiet lt. Studienblatt / field of study as it appears on the student record sheet:	Romanistik / Romance Studies
Betreut von / Supervisor:	PD Mag. Dr. Wolfram Aichinger

Esta tesis ha sido amparada por los proyectos de investigación *El Calderón Cómico* (FWF P 29115) e *Interpretation of Childbirth in Early Modern Spain* (FWF P 32263-G30), dirigidos por el Dr. Wolfram Aichinger, y *Sound and Meaning in Spanish Golden Age Literature* (FWF P 32563), dirigido por el Dr. Simon Kroll.

Este trabajo se compuso con XeLaTeX usando los tipos de letra CMU Serif y DejaVu Sans para el cuerpo del texto, `CMU Typewriter` para los listados de código fuente y *Latin Modern Math* para las fórmulas, todos bajo licencia libre; la portada incluye el tipo de letra comercial Arial de acuerdo con las directrices de la Universidad.

© 2023 Fernando Sanz-Lázaro

Bajo licencia Creative Commons Attribution 4.0 International (CC BY 4.0)
Puede encontrar los términos de la licencia en:
https://creativecommons.org/licenses/by/4.0/legalcode.es

Impreso en Viena

A Carlotta y Mateo

Agradecimientos

Esta tesis ha sido posible gracias al concurso de un sinnúmero de personas, que, lo sepan o no, han puesto su granito de arena —paladas, alguno—. Sirvan estas líneas al menos como un mínimo reconocimiento a todos ellos, ya que jamás podré pagarles como verdaderamente merecen.

Para comenzar, doy las gracias a Wolfram Aichinger, mi director de tesis, amigo y maestro, por no dejarme perder de vista lo que de verdad importa, empezando por el mismo título del trabajo y acabando con aquel curso de literatura que impartió durante el primer semestre de 2014 —y mío en la Universidad de Viena— en el Auditorio A de este viejo hospital general, donde entendí hacia dónde ir.

Vaya un agradecimiento muy especial para Simon Kroll por acogerme en su equipo, motivar esta disertación y tomarse un tiempo impagable en revisar el manuscrito con la minuciosidad del editor crítico que es, pero también por todo lo que he aprendido junto a él y por estar siempre alerta para señalar nuevos caminos por explorar.

Gracias a Álvaro Cuéllar, por el torbellino de ideas que ha traído a Viena consigo, por redescubrir *La francesa Laura* y porque, en el breve tiempo que hemos trabajado juntos, ha inspirado no pocas cosas y contribuido a mejorar muchas otras, incluyendo su lectura crítica del manuscrito. Hago una acotación aquí —*pun intended*— para dar también las gracias a Clara Monzó, calderonista de pro y *beta tester* sobrevenida. La tesis debe mucho a ambos.

Por supuesto, estoy obligado para con todos los investigadores cuyo trabajo ha sentado los cimientos de este —me alzo a hombros de gigantes—, algunos de los cuales he tenido el honor y el placer de conocer en persona a lo largo de estos años: Hanno Ehrlicher, Antonio Sánchez Jiménez y Ramón Valdés Gázquez, por su amabilidad y, sobre todo, por las ideas que he tomado prestadas de los trabajos de cada uno; Markus Ebenhoch, por su lectura y comentarios al primer capítulo teórico; Alejandro García Reidy, por sus detallados reportes de error (¡con solución incluida!); Fernando

Rodríguez-Gallego, por todas sus conversaciones con Simon que han terminado en mi código; Jos Sagüés †2014, por sacarme del aula a las tablas; María Victoria Curto, Isabel García Adánez, Gaston Gilabert, Laura Hernández-Lorenzo, Achim Hermann Hölter, Jörg Lehmann, Borja Navarro-Colorado, Mario Rossi, Christoph Strosetzki y Álvaro Torrente.

Los proyectos Sound and Meaning in Spanish Golden Age Literature, Interpretation of Childbirth in Early Modern Spain y El Calderón Cómico han contribuido a que pudiera dedicarme a tiempo completo a menesteres académicos hasta el término de esta tesis. No ha de faltar tampoco mi reconocimiento a mis compañeros en ellos y del departamento: Sabrina Grohsebner, Hannah Mühlparzer, Tamara Hanus, Sara Brankovic, Hannah Fischer, Kurt Kriz, Christian Standhartinger, Marie Radinger, Marie-Louise Fürsinn, Marie Stockinger, Emma Marcos, así como a todos aquellos con los que he tenido el gusto de trabajar y me han aportado nuevas cosas que llevar en mi bagaje. Mi deuda se extiende, claro está, al resto del Institut für Romanistik de la Universidad de Viena, que me ha visto crecer como hispanista.

Debo mi más profunda gratitud a Martina por todos sus sacrificios, sin los cuales no hubiera podido dedicar las inacabables horas que ha demandado esta empresa. Estoy asimismo obligado con la inquisitiva Carlotta y el siempre risueño Mateo, por todas los ratos con papá que les ha robado este trabajo. Vosotros tres dais sentido a todo. Es menester agradecer también a Vidal y Sonia su disposición para conseguirme rápidamente un sinfín de materiales a los que, de otra manera, quién sabe si habría echado el guante, además de la caza de erratas. Mención especial para Maricruz †2023, pues en aquellos libros gordos de Petete me topé inadvertidamente con la historia y los clásicos, sin los que hoy posiblemente no estaría escribiendo esto; vives en nuestra memoria. Gracias a Erika y Georg por su solicitud para echar siempre una mano con lo que sea. Y un recuerdo para los M.M.R..

A todos los que se me han pasado, que sois legión, también estoy en deuda con vosotros.

Índice general

Índice de figuras *ix*
Índice de cuadros *xi*
Índice de algoritmos *xiii*
Lista de abreviaturas *xviii*

1 Introducción **1**
1.1 Objetivos . . . 2
1.2 Hipótesis de trabajo, objetivo y preguntas de investigación . . . 6
1.3 Estado de la cuestión . . . 7
1.4 Metodología y estructura del trabajo . . . 11

I Fundamentos teóricos **15**
2 Humanidades digitales **17**
2.1 ¿Qué son las humanidades digitales? . . . 17
2.2 De *computación de humanidades* a *humanidades digitales* . . . 23
2.3 Datos . . . 35
2.4 Metodología *dirigida* y *asistida* por datos . . . 41
2.5 Filología digital . . . 43
2.6 Una digresión epistemológica . . . 46
2.7 Lectura distante . . . 52
3 Drama **57**
3.1 Literatura escenificada . . . 57
3.2 Aproximaciones al drama . . . 60
3.3 Dramatología . . . 62
3.4 Estructura del drama . . . 67
3.5 Ediciones... digitales . . . 83
3.6 Análisis dramático cuantitativo . . . 86
4 Fonología y fonética **93**
4.1 Por qué una aproximación fonológico-acústica . . . 93

4.2 Los sonidos de la lengua . . . 95
4.3 Fonética . . . 96
4.4 El sistema fonológico . . . 105
4.5 Neutralización . . . 111
4.6 Fonemas del español . . . 111
4.7 División silábica del español . . . 123
4.8 Transcripción grafémico-fonológica . . . 130
4.9 Correspondencia grafémico-fonológica . . . 134
5 Métrica 137
5.1 La lengua de los poetas . . . 137
5.2 Elementos métricos . . . 141
5.3 Ritmo . . . 144
5.4 Pausas . . . 152
5.5 Cómputo silábico . . . 154
5.6 Rima . . . 160
5.7 Agrupaciones de versos . . . 162

II Aplicación empírica 187
6 De la teoría a la práctica 189
6.1 Computador y papel . . . 189
6.2 Modelo de pruebas . . . 191
6.3 Estructura del modelo . . . 197
6.4 Módulos del programa . . . 198
6.5 Formatos de las fuentes bibliográficas . . . 202
6.6 Diferencias entre textos, textos y textos . . . 204
6.7 Peor escenario: PDF . . . 205
6.8 Escenario regular: archivos ofimáticos . . . 208
6.9 Escenario óptimo: XML-TEI . . . 211
6.10 Precauciones con la codificación . . . 217
7 Estructura digital del texto dramático 223
7.1 Texto dramático, datos dramétricos . . . 223
7.2 De texto plano a VED . . . 224
7.3 Excurso III: De VED a XML-TEI . . . 230
7.4 Estructurando los datos . . . 232
7.5 Implementación del proceso . . . 236
8 Transcripción automática 243
8.1 Sobre la transcripción automática . . . 243
8.2 Módulo transcriptor . . . 245

8.3 Métodos auxiliares de la transcripción . . . 259
8.4 Módulo silabeador . . . 264
9 Escansión automática 283
9.1 Introducción . . . 283
9.2 Separación por verso . . . 285
9.3 Escansión del verso . . . 290
9.4 Prosodia gramatical . . . 294
9.5 Ajustes métricos . . . 301
9.6 Taxonomía versal . . . 321
9.7 Verificación de la escansión rítmica . . . 321
9.8 Agrupación de versos . . . 323
10 Conclusiones 337
10.1 Recapitulación . . . 337
10.2 Respuestas a las preguntas de investigación . . . 342
10.3 Cuestiones abiertas y unas palabras finales . . . 348
Apéndices 353
A Resúmenes 353
A.1 Zusammenfassung auf Deutsch . . . 353
A.2 Abstract . . . 354
A.3 Resumen en español . . . 355
B Listados de código 357
B.1 Preprocesado . . . 357
B.2 Transcripción fonológica . . . 362
B.3 Escansión versal . . . 375
B.4 Agrupación estrófica . . . 391
B.5 Codificación XML-TEI . . . 396
C Ejemplos de uso 411
D Análisis dramático cuantitativo manual 417
Bibliografía 431
Fuentes primarias . . . 431
Fuentes secundarias . . . 437

Índice de figuras

2.1 Representación visual y cadena de caracteres. 39

4.1 Descripción de la onda sonora. 100
4.2 Onda compleja y las ondas simples que la componen. 101
4.3 Onda de [aː]. 102
4.4 Espectrogramas de [aː]. 103
4.5 Formantes F_1 y F_2 de las vocales (Hz). 104
4.6 Diptongo /eu/. 119
4.7 Construcción sílabas. 124

5.1 Verso 42 en una representación (Vega Carpio, 1982) de *La discreta enamorada*. 159

6.1 Código Python en el editor de texto Vim. 195
6.2 Equipo de desarrollo. 196
6.3 Estructura general. 197
6.4 Módulos. 200
6.5 Diálogo de búsqueda de LibreOffice.org. 209
6.6 Extensión de búsqueda *Alternative Find & Replace for Writer*. 210
6.7 Estructura de XML-TEI. 212
6.8 Transliteración de la variante ASCII ISO-8859-15 a Unicode. 219

7.1 Estructura de la configuración dramática. 234
7.2 Ejemplo de aplicación. 242

9.1 Esquema del proceso de escansión. 284
9.2 Perceptibilidad. 292
9.3 Vocales en el trapecio bucal. 293

D.1 Presencia en las distintas configuraciones. 421
D.2 Dominancia en la primera jornada. 428

Índice de cuadros

2.1 Escala de los jóvenes poetas según Akenside. 38
2.2 Datos estructurados. 39

4.1 Puntos de articulación por autor. 97
4.2 Formantes F_1 y F_2 de las vocales (Hz). 103
4.3 Fonemas vocálicos del español. 113
4.4 Rasgos acústicos de las vocales del español. 114
4.5 Diptongos. 116
4.6 Triptongos. 119
4.7 Fonemas y alófonos consonánticos del español. 121
4.8 Correspondencia grafémico-fonológica. 135

5.1 Clasificación del octosílabo de Navarro Tomás. 151
5.2 Estrofas. 172
5.3 Clasificación del endecasílabo. 175

7.1 Tabla de atributos dramáticos. 241

9.1 Tabla de atributos métricos. 320
9.2 Resultados en el corpus de Navarro-Colorado de 100 sonetos. 321
9.3 Discrepancias. 322
9.4 Ampliación de la tabla de atributos métricos. 331

D.1 Densidad escénica. 419
D.2 Distancia escénica. 422
D.3 Copresencia escénica. 423
D.4 Presencia escénica. 423
D.5 Distancia de Hamming. 424
D.6 Probabilidad P_{ij} de copresencia. 426
D.7 Valores λ_{ij}. 427

D.8 Copresencia escénica por parlamento. 429

Índice de algoritmos

1 Conversión de XML-TEI a texto. 216

2 Personajes que se van. 230

3 Conversión de VED a XML-TEI. 231

4 Cabecera para la tabulación. 237

5 Determinación del sexo. 237

6 Recorrido de las líneas de la obra (etiqueta). 238

7 Recorrido de las líneas de la obra (no encuentra etiqueta). 240

8 Definiciones previas de la biblioteca. 246

9 Cabecera del módulo de transcripción. 246

10 Módulo de preprocesado. 252

11 Módulo de reordenación. 255

12 Método de transcripción fonológica. 256

13 Caso de ⟨y⟩. 257

14 Caso de ⟨g⟩. 258

15 Caso de ⟨y⟩. 259

16 Creación de variables para sílabas y palabras. 260

17 Manejo de diptongos. 261

18 Transcripción fonética. 262

19 Sustituciones fonológico-fonéticas. 263

20 Transliteración IPA-SAMPA. 263
21 Cabecera del módulo silabeador. 265
22 Procesamiento de las excepciones. 267
23 Procesamiento de latinismos. 270
24 Método silabeador. 272
25 Separación de núcleos silábicos. 274
26 Distribución silábica de consonantes. 276
27 Bloque de comprobaciones. 277
28 Asignación final del resto. 279
29 Localizador de acento prosódico. 280

30 Escansión de los versos uno a uno. 286
31 Escansión de los versos uno a uno. 287
32 Actualización de la tabla. 289
33 Declaración de constantes y clase auxiliar. 291
34 Clases auxiliares. 293
35 Clase `Frase`. 294
36 Preprocesado. 295
37 Búsqueda de acentos prosódicos del verso (I). 296
38 Búsqueda de acentos prosódicos del verso (II). 297
39 Búsqueda de acentos prosódicos del verso (III). 299
40 Búsqueda de acentos prosódicos del verso (IV). 300
41 Búsqueda de acentos prosódicos del verso (V). 300
42 Tratamiento de la conjunción *y*. 301
43 Ajustes métricos. 302
44 Búsqueda de vocales en contacto (I). 303
45 Búsqueda de vocales en contacto (II). 304
46 Evaluación de la idoneidad de las sinalefas potenciales. 306

47 Ajuste de sílabas esperadas. 307
48 Reducción de lista de palabras a lista de sílabas. 308
49 Ajustes métricos (I). 308
50 Ajustes métricos (II). 309
51 Discriminación de núcleos vocálicos. 310
52 Identificación del patrón rítmico. 311
53 Ajuste *post hoc* del metro. 312
54 Ubicación de diéresis potenciales. 313
55 Aplicación de diéresis. 314
56 Preferencia de las diéresis. 315
57 Comprobación de esdrújula ante hemistiquio. 315
58 Reajuste en función del hemistiquio. 316
59 Sinalefas. 316
60 Ajuste silábico. 318
61 Aplicación de las sinalefas. 319
62 Ajuste de la posición. 319
63 Agrupación de versos. 324
64 Búsqueda de estrofas. 325
65 Definiciones de estrofas. 326
66 Búsqueda de estrofas. 328
67 Asignación de la letra del esquema de la rima. 329
68 Búsqueda de patrones asonantes. 330
69 Segunda pasada de correcciones. 335

Lista de abreviaturas

ADSO Análisis Distante del soneto castellano de los Siglos de Oro (XVI y XVII)

AFI Alfabeto Fonético Internacional

ASCII American Standard Code for Information Interchange

BERT Bidirectional Encoder Representations from Transformers

CSV Comma Separated Values

DOC Microsoft Word Document

ESPRIT European Strategic Programme on Research in Information Technology

FAIR Findability, accesibility, interoperability, and reusability

GNU GNU is Not Unix

GPL General Public License

GRISO Grupo de Investigación Siglo de Oro

HTML Hyper Text Markuup Language

ISO International Organization for Standardization

ODF Open Document Format for Office Applications

ODT Open Document Text

PDF Portable Document Format

PLN Procesamiento del lenguaje natural

RAE Real Academia Española

regex Regular expression

SAMPA Speech Assessment Methods Phonetic Alphabet

SGML Standard Generalized Markup Language

TEI Text Encoding Initiative

VED Visually-Encoded Drama

XML-TEI eXtended Markup Language-Text Encoding Initiative

XML eXtended Markup Language

Er zweifelte nun nicht, dass diese Gedichte ihm seinen Weg zum Theater noch mehr bahnen, und ihn bald seinem Ziele näher bringen würden.

Karl Philip Moritz, *Anton Reiser*

Introducción 1

Hemos de morir. Esta obviedad tiene implicaciones profundas no solo en nuestra forma de entender el mundo y a nosotros mismos, sino en cómo y hasta dónde podemos enfrentarnos a los problemas que nos encontramos durante la vida. Poco importan nuestras intenciones, creencias o deseos cuando llega el momento postrero: somos impotentes ante esa inapelable realidad. Esto hace que las aproximaciones a empresas de cierta envergadura hayan de ceñirse a un improrrogable límite temporal. Cuán factible es llevarlas a término, en muchos casos, depende de comprender la naturaleza perecedera de la existencia y adaptar nuestro propósito a la inevitabilidad del destino. Con frecuencia, no queda otra elección que resignarse a una solución de compromiso.

Los estudios filológicos no son una excepción a esta regla universal, como ya notó Bloom (1995, p. 29). El crítico apelaba precisamente la mortalidad del hombre para defender el canon occidental, arguyendo que «there is only so much time, and time must have a stop, while there is more to read than there ever was before». En efecto, no podemos dedicar una lectura minuciosa a cada una de las obras que componen un corpus de cierta extensión, por lo que debemos conformarnos con intentar aprehender la esencia del conjunto en una reducidísima fracción de sus elementos[1].

Podemos tomar la literatura universal, o la nacional de un país, un género, un período, una lengua, incluso un tema dado en un periodo concreto y en un solo lugar: a mayor o menor escala, volvemos a toparnos con el mismo obstáculo, como en un gráfico fractal en el que un patrón reaparece recursivamente hasta el infinito cada vez que nos aproximamos. Igual ocurre con el teatro del Siglo de Oro, para el que los recursos disponibles difícilmente alcanzan a cubrir siquiera una esquirla de lo escrito en aquella época tan fructífera; considerar todos y cada uno de los versos de centenares de obras

[1] De ahí la necesidad de seleccionar con cierto criterio, «we possess the Canon because we are mortal and also rather belated» (Bloom, 1995, p. 29).

se antoja una tarea como la de aquel párvulo que intentaba vaciar el Mediterráneo con su pequeña concha ante el desconcertado sabio de Hipona.

1.1 Objetivos

El sonido y el ritmo de las palabras arrastran carga semántica en no pocas ocasiones, sea o no de manera intencionada. Esto es particularmente cierto en las obras teatrales del Siglo de Oro, en las que determinado tipo de verso se presta más fácilmente a ser empleado en determinadas coyunturas. Los dramaturgos de la época no dejaron de señalarlo, como no podía ser de otra forma, y el paradigma lo encontramos escrito de la pluma del mismo Lope de Vega (1609/2006, vv. 305-312):

> Acomode los versos con prudencia
> a los sujetos que va tratando.
> Las décimas son buenas para las quejas;
> el soneto está bien en los que aguardas;
> las relaciones piden los romances,
> aunque en octavas lucen por extremo;
> son los tercetos para cosas graves,
> y para las de amor, las redondillas.

No ha de llamar a asombro que la crítica contemporánea también haya reparado en ello, sin olvidar las menciones recientes a la relación que parecen mostrar circunstancias y personajes de tipos muy concretos con sonidos (Kroll, 2019b, 2020; Domínguez Caparrós, 2005, p. 36) y cadencias (Sánchez Jiménez, 2017) particulares, si bien la comedia áurea no tendría un simbolismo tan definido como el de las antiguas representaciones rituales (Aichinger, 2012, p. 122). Los grandes dramaturgos auriseculares acomodan el metro a la fábula con un cuidado exquisito para propiciar un acompañamiento sonoro adecuado a la acción (Aichinger, 2015, pp. 48-60). Sin embargo, establecer una correlación a ciencia cierta requiere analizar caso por caso un considerable número de situaciones dramáticas, lo que implica, asimismo, la obligación de escandir un sinnúmero de versos, de achicar el mar con una minúscula concha.

Por fortuna, la tecnología ayuda a contrarrestar estas carencias, si bien dentro de sus posibilidades. Aunque el computador[2] se limita a ejecutar una serie de instruc-

[2]Hay división de opiniones en cuanto al término más adecuado para designar la máquina que en inglés se bautizó como *computer*. Mientras el español americano opta por la versión anglosajona, de forma indistinta en su variante masculina *computador* o femenina *computadora*, la variedad peninsular, a semblanza del *ordinateur* del vecino francés, se inclina por *ordenador*. Antonio Vaquero Sánchez,

ciones relativamente sencillas[3], lo hace a una velocidad prodigiosa y sin acusar el cansancio que, tarde o temprano, termina por abatir al más afanoso de los humanos. Esta extensión técnica de nuestras capacidades la han venido usando la lingüística computacional y la de corpus con resultados excepcionales, pero también ha llegado a ese otro lugar difuso que se ha dado en llamar humanidades digitales.

No aspiramos a demostrar aquí la —por otra parte, evidente— genialidad de Calderón para evocar emociones mediante la escenificación de pasiones ocultas (Bidwell-Steiner, 2013, p. 451) ni tampoco a explicar su singularidad con un estudio cuantitativo. Sí pretendemos, en cambio, ser capaces de extraer datos que reflejen categorías textuales y que estos posibiliten analizar diferencias cuantitativas. En último término, deseamos emplear herramientas prestadas de la lingüística en la creación de otras nuevas que faciliten al crítico —al humano— la identificación e interpretación de esas divergencias. Es el filólogo, conforme a sus conocimientos de la disciplina, quien decide qué observar, la forma de hacerlo y cómo entenderlo, pero delega en la máquina la extracción y clasificación de los datos. Nuestra labor es, pues, «enseñarle» al computador qué debe encontrar, cómo ha de llevarlo a cabo y de qué manera tiene que presentar los resultados. La disponibilidad de un enorme volumen de datos cuantificables obtenidos así da pie a descubrir patrones. A grandes rasgos, esto es lo que se ha dado en llamar *lectura distante* (Moretti, 2000a), sobre la que hablaremos más en detalle en el siguiente capítulo.

La lectura distante, empero, demanda datos estructurados, textos aptos para el tratamiento digital. Asimismo, estos deben obedecer a criterios filológicos pertinentes: si son inapropiados, invalidarían los resultados de cualquier análisis, por impecable que fuera la metodología empleada. Por lo tanto, la lectura distante requiere, en primer lugar, un corpus con la calidad óptima requerida por la investigación (Calvo Tello, 2019) y, en segundo, que este se halle dispuesto para extraer información precisa, veraz y relevante.

En el caso del drama español clásico, nos enfrentamos a un doble reto: por un lado, necesitamos textos anotados[4] en virtud de sus características dramáticas. Por otro, al

pionero de la computación en España y coautor de uno de los más rigurosos glosarios de terminología informática publicados (Vaquero Sánchez y Joyanes Aguilar, 1985) en español, entiende que ordenar es solo una pequeña parte de los problemas susceptibles de ser computables, por lo que considera inadecuado el calco del francés. Atendiendo a las razones que expone (Vaquero Sánchez, 1999, pp. 9-11), optaremos aquí por el término *computador*.

[3]La eficacia de la electrónica depende al final del buen hacer del encargado de dictar dichas instrucciones.

[4]Aquí y en adelante uso el calco del inglés *anotar* —de cierto predicamento en el ámbito de la lingüística— para referirme, en general, a enriquecer un texto con metadatos, mientras que relego el tal vez más adecuado término *etiquetar* para el sentido más concreto de marcar con una etiqueta un segmento textual.

estar los parlamentos versificados, debería trabajarse con datos que tengan también en cuenta las particularidades líricas de los textos. Por lo tanto, de embarcarnos en el análisis filológico digital del verso teatral, deberíamos satisfacer primero esos requisitos antes de emprender la labor.

Esta tesis doctoral surge en el marco del proyecto Sound and Meaning in Spanish Golden Age Literature[5], encabezado por Simon Kroll, como respuesta a la doble necesidad planteada arriba. Entre las muchas ocupaciones del proyecto, se encuentra la de llevar a cabo diversos tipos de análisis de textos teatrales. Particularmente, trata de hallar relaciones entre las características rítmicas de los versos y varios de los aspectos estructurales y semánticos de las obras (Kroll y Sanz-Lázaro, 2022). Para poder comparar grandes cantidades de versos teniendo en cuenta su función en la estructura dramática, se requería un corpus de piezas del género apto para el análisis digital de esas propiedades. En esta tesis abordamos la cuestión de cómo obtenerlo. Lo hacemos mediante el desarrollo teórico y analítico de los principios rectores de métodos y herramientas (Sanz-Lázaro, en prensa) que hemos concebimos para ello. Alcanzaremos esa meta tomando prestadas técnicas de la lingüística computacional y de corpus, que trasladaremos a los estudios teatrales áureos. Este trabajo se ocupa del examen obras dadas para extraer y organizar sus datos, mientras que son responsabilidad del proyecto la selección de los componentes del corpus y los experimentos que se valen de él.

Adicionalmente, lo aquí propuesto encuentra utilidad en otros terrenos, como, por ejemplo, la ecdótica. Durante el desarrollo del trabajo, nos percatamos de que el análisis automático de una obra produce efectos reconocibles si el texto de origen presenta inconsistencias como sílabas adicionales, disposición errónea u omisión de versos o incluso el intercambio de líneas. Si bien esta tarea puede resolverse mediante una lectura atenta, no cabe duda de que reconocer tales irregularidades con apenas un vistazo supone una ayuda en el examen del texto y su transmisión[6]. Confiemos en que la situación haya cambiado en los últimos tiempos y no venga a cuento repetir aquello de que «las "nuevas tecnologías" están pasando de largo por la Filología Española sin que la vieja dama, en lo que a la edición crítica se refiere, se inmute», como se lamentaba Ramón Valdés (2104b, p. i) no hace siquiera una década.

En realidad, si no nos dejamos deslumbrar por el brillo que desprende toda novedad antes de que el tiempo la cubra de una venerable pátina, vemos que las aspiraciones de esta tesis beben directamente de la idea primigenia de la filología, por mucho que ofrezca nuevos medios para abordar cuestiones eternas. Friedrich Nietzsche, en una

[5]Sound and Meaning en adelante.

[6]Esta observación se la debo Álvaro Cuéllar y Simon Kroll.

lección inaugural celebrada en la Universidad de Basilea en 1869, defendía que la filología es

> tanto un trozo de historia, como un trozo de ciencias naturales, como un trozo de estética: historia, en tanto que quiere aprehender en renovadas imágenes las manifestaciones de las individualidades nacionales determinadas, la ley imperante en el devenir de los fenómenos; ciencias naturales, en tanto que anhela explorar el instinto más profundo del hombre: el instinto del lenguaje; estética, finalmente, porque, de entre un conjunto de Antigüedades, eleva la llamada «clásica» con la aspiración y el propósito de desenterrar un mundo ideal sepultado y confrontarlo al presente como espejo de los modelos clásicos y eternos. (1869/1982, pp. 249-250, traducción propia)

Así pues, este trabajo parte de la *estética* de los textos porque pretende encontrar patrones poéticos y dramáticos que los identifiquen. Se basa para ello en las *ciencias naturales*, pues es en la lengua donde buscaremos esos patrones. La síntesis de esto faculta para observar la *historia*, hallar leyes que ofrezcan nuevas perspectivas en la descripción del teatro del Siglo de Oro. En ese sentido, nuestro empeño último es poner herramientas a disposición de otros mejores, no tanto para hacer progresar la ciencia filológica, sino para facilitar su progreso. Dicho de otro modo, aspiramos a aliviar al filólogo de escandir millones de versos y relacionar miles de diálogos[7] para que pueda dedicar las horas así ganadas a interpretar la evidencia.

Valga la prevención de que los análisis de lectura distante que, entre otros usos, pretendemos facilitar con este trabajo no sustituyen a los métodos tradicionales, sino que ofrecen una perspectiva complementaria. De la misma manera que la placa de Petri del microbiólogo tiene su ámbito de utilización y no suple al microscopio, la lectura distante tampoco reemplaza el escrutinio minucioso de los textos. Ni existe un algoritmo capaz de siquiera aproximarse al detallismo del lector humano ni existe humano con la velocidad de cálculo de un computador. En definitiva, empleémonos a fondo con las lecturas que lo requieran —y, a ser posible, también lo merezcan—, sin renunciar a una visión general precisa que comprenda el mayor número de elementos del conjunto.

[7] Además, la recolección de datos requiere una sustancial inversión de tiempo y recursos (Ehrlicher y Lehmann, 2019, p. 41), ambos exiguos en la investigación, lo que hace difícilmente viable la extracción manual de la información dramática y métrica de un corpus de cierta envergadura.

> Digital Humanities is an extension of traditional knowledge skills and methods, not a replacement for them. Its distinctive contributions do not obliterate the insights of the past, but add and supplement the humanities' long-standing commitment to scholarly interpretation, informed research, structured argument, and dialogue within communities of practice. (Burdick y col., 2012, p. 16).

1.2 Hipótesis de trabajo, objetivo y preguntas de investigación

Partimos de una hipótesis sencilla e intuitiva: es posible descomponer la obra teatral considerando sus partes constitutivas y organizarla como datos estructurados, tanto a un nivel superficial dramático como en la profundidad de la forma poética de cada uno de sus versos. De esta manera, se aplica la premisa recursivamente: una vez descompuesta la pieza en sus partes primarias, deberíamos poder hacer lo mismo con cada uno de sus componentes. Finalmente, asumimos que esto puede efectuarse automáticamente dado un conjunto de reglas adecuado.

Por lo tanto, el objetivo es de este trabajo es establecer dichas reglas, definir un procedimiento formal para extraer la información del texto y organizarla de tal modo que pueda ser usada directamente en análisis digitales. Responderemos a las siguientes preguntas de investigación para alcanzar esa meta:

i. ¿Cómo automatizar el análisis de la estructura dramática?

ii. ¿Cómo escandir automáticamente el verso dramático?

iii. ¿Cómo organizar la información resultante en un corpus anotado?

La empresa plantea ciertas dificultades por el propio género literario del que se ocupa. Por un lado, los estudios de lectura distante tienden a concentrarse en la prosa, por lo que no consideran la particular estructura del texto dramático, lo que se hace más notable si este está en verso. La escansión métrica digital, por su parte, está enfocada en la lírica, por lo que suele abordar estrofas predefinidas, mientras que el teatro es polimétrico y no hay una correspondencia entre línea y verso necesariamente ni solución de continuidad delimitando formas poéticas adyacentes.

1.3 Estado de la cuestión

La década de 1980 vio dar los primeros pasos a la filología digital aplicada al teatro español del Siglo de Oro, pero esta no comenzó a adoptar una configuración definida hasta el siguiente decenio. Aún hubo que esperar otros diez años más para que empezara a considerarse realmente popular, ya entrado el siglo XXI. Durante este último milenio, han surgido una plétora de ediciones digitales, bases de datos y programas de análisis literario (Kroll, 2019a). La ecdótica digital áurea bebe del proyecto pionero dirigido por Marco Presotto que fructificó en una ambiciosa edición de *La dama boba* (Vega Carpio, 1613/2015b). Esta trabajo marcó el sendero que habría de seguirse en el futuro, si bien posteriores iniciativas se centran en aspectos concretos: aquella edición estaba concebida como un laboratorio de pruebas, por lo que, a pesar de ser deseable, resultaría muy costoso explorar un corpus grande de la misma manera[8].

El propio Presotto (2018, pp. 32-36) nombra algunas iniciativas loables. El grupo Prolope, por ejemplo, pone a disposición del público tanto el texto crítico como la edición de una selección obras del Fénix (Prolope, 2023), pero no solamente eso, sino que su forma de abordar la edición constituye «un instrumento imprescindible que ha significado un cambio de paradigma en los estudios dedicados al Siglo de Oro» (Presotto, 2023, p. 2). El grupo GRISO ofrece los textos de las ediciones de los autos de Calderón que ha publicado en Reichenberger. Moretianos (PROTEO, 2023), responsabilidad del grupo PROTEO, se dedica a la obra de Agustín Moreto. Asimismo, otros proyectos, como Cervantes Virtual (Fundación Biblioteca Virtual Miguel de Cervantes, 2021) o Canon 60 (TC/12 Red del Patrimonio Teatral Clásico Español [TC/12], 2014) garantizan el acceso a textos de calidad de diversas procedencias, como los producido por el GIC (GIC, 2023). Además, él mismo codirige Gondomar Digital junto a Luigi Giulani y Fausta Antonucci («Gondomar Digital. La colección teatral del conde de Gondomar», 2017).

Las bases de datos también han gozado de gran predicamento, tanto en sus variantes dedicadas a compilar y organizar obras en crudo, información sobre estas u otras cuyo objeto es la información metatextual, pero sin anotar directamente el texto. Bajo el paraguas ASODAT (Ferrer Valls, 2023), se encuentran, entre otras, algunas tan asentadas como Artelope (Oleza Simó, 2013), Calderón Digital (Antonucci, 2018), CLEMIT (Urzáiz Tortajada y col., 2023), DICAT («Grupo de investigación teatral DICAT», 2023), Digital Música Poética («Digital Música Poética», 2023), ETSO (Cuéllar y Vega García-Luengos, 2017), CATCOM (Ferrer Valls, 2013) o MANOS

[8] José Manuel Fradejas Rueda (2022b) pone en perspectiva histórica los métodos y formatos empleados hoy en la edición crítica digitalizada.

(Greer y García-Reidy, 2022). Gracias a la combinación de estas últimas, por ejemplo, ha podido seguirse la pista del ejercicio de la composición dramatúrgica de autores no tan tratados como los grandes (García-Reidy, 2019)[9]. De aparición más reciente, AuTeSO (AuTeSO, 2023).

Resultan mucho menos numerosos los trabajos que abordan el análisis del propio texto. No obstante, encontramos algunos como los que se ocupan de la estilometría (Cuéllar y Vega García-Luengos, 2017; Hernández-Lorenzo y Byszuk, 2022). Precisamente, el trabajo de Cuéllar y Vega García-Luengos (2023) se valió de esto, entre otras cosas, para la celebrada atribución lopesca de *La francesa Laura* (Vega Carpio, 2023). Estos se valen del módulo de estilometría Stylo (Eder y col., 2016) del lenguaje de programación R para realizar pruebas de autoría. Por la propia concepción de los análisis estilométricos tradicionales[10], el texto teatral áureo se coteja considerando solo sus parlamentos y, estos, como si de prosa se tratase. Esto, por otra parte, ha obligado a desarrollar métodos de tratamiento de las fuentes, lo que llevó a Cuéllar a componer sus propios modelos lingüísticos automatizar transcripciones paleográficas (Cuéllar, 2023). Otro tipo de información textual más allá del léxico no ha despertado tanto interés en el ámbito hispánico, si bien, recientes estudios orientados a la poética abren una puerta para futuras aproximaciones (Hernández-Lorenzo, 2022a).

Curiosamente, la descripción cuantitativa de la obra teatral cuenta con bases teóricas que se remontan atrás más de medio siglo. Solomon Marcus (1973), por ejemplo, ya elaboró en su día un método preciso para representar la presencia escénica de los personajes y sus relaciones. Esta metodología se engloba en lo que Romanska (2015, p. 446) denominó más tarde *dramétrica*, que se traslada directamente al modelo digital. Contando con los textos debidamente anotados, pueden resolverse hoy en pocos segundos trabajos que le habría llevado semanas a Marcus. Para el análisis del teatro áureo, sin embargo, esa aproximación presenta un inconveniente: toma como segmento básico la escena. El drama del Siglo de Oro carece de tal división, al menos de forma explícita. En todo caso, en este trabajo no dilucidaremos la cuestión, pero sí propondremos algunos medios con nuestros resultados para aligerar, hasta cierto punto, la labor de aquellos dispuestos a lidiar con ella.

[9]Además, habría que señalar bases de datos que, sin referirse directamente a la literatura española o siquiera a la temprana Edad Moderna, albergan información relevante para el estudio de las letras áureas. Valga para ilustrarlo la que recoge la actual ubicación de los fondos —hoy dispersos— de la biblioteca privada del romántico alemán Ludwig Tieck, quien, aparte de dramaturgo, fue un ávido coleccionista de sueltas de dramas áureos españoles («Ludwig Tiecks Bibliothek. Anatomie einer romantisch-komparatistischen Büchersammlung», 2018).

[10]Existen más aproximaciones que han demostrado una precisión semejante a los análisis estilométricos tradicionales basados en la frecuencia de uso de palabras. Ver Kroll y Sanz-Lázaro (2022 y 2023) y Plecháč y col. (2018).

Existen otros análisis dramáticos que no están sujeto a la misma indeterminación. Un ejemplo de ello lo ofrece Ilsemann (1998), a pesar de que no aborda el teatro español áureo, sino el inglés isabelino —la frecuencia y extensión de los parlamentos de varias piezas de Shakespeare—. Las variables que describe son comunes a todas las obras teatrales, cualesquiera que sean su lengua y época, por lo que estos estudios del drama constituyen un magnífico modelo para aplicar a otras lenguas, sea en sus métodos (Ilsemann, 2013) o sus herramientas (Wilhelm y col., 2013).

En otras lenguas encontramos aproximaciones similares, si bien más tímidas. El teatro alemán, por ejemplo, ha sido investigado de forma cuantitativa (Dennerlein, 2015) y, hoy en día, cuenta con herramientas libremente accesibles para obtener información dramétrica de los textos (Schmidt y col., 2019); solo muy recientemente se han hecho incursiones en el teatro español, precisamente en el ámbito del drama aurisecular (Ehrlicher y col., 2020; Lehmann y Padó, 2022).

Todos estos estudios, como hemos dicho, requieren fuentes anotadas con la información necesaria y estructurados en un formato adecuado. Unos sesenta textos se encuentran disponibles en la colección Canon 60 (TC/12, 2014) como páginas web, aparentemente creados a partir de una fuente en XML-TEI, que, no obstante, necesitan cierto tratamiento previo para estandarizarlos. Más ambicioso es el proyecto DraCor (Fischer y col., 2019), que compila diversos corpus teatrales en once idiomas. Aparte de los generales de las lenguas inglesa y española, dispone de secciones específicas para Shakespeare y Calderón, a la última de las cuales ha contribuido Sound and Meaning con obras producidas mediante los resultados expuestos en esta tesis. Al estar codificado en el estándar de anotación XML de textos literarios de la Text Encoding Initiative (TEI), facilita diferentes análisis dramáticos. El trabajo de Ehrlicher y col. (2020) —además, responsables en su mayor parte de la sección calderoniana CalDraCor—, por ejemplo, emplea este tipo de archivos como corpus. Todos ellos tienen en común la obligada necesidad de incrementar sus fondos al lento paso que obliga el procesamiento de los textos.

La escansión automática de poesía española obtuvo sus primeros resultados relevantes gracias al estudio desarrollado por Gervás (2000). Este toma las palabras del verso e intenta separar sus sílabas, bien utilizando un *diccionario*[11], bien mediante reglas cuando este no contiene una clave para la palabra en cuestión. De esta forma, obtiene el número de sílabas, una lista de estas, los acentos internos de las palabras

[11]Se denomina *diccionario* a un tipo de datos consistente en un conjunto de elementos, cada uno de los cuales consta de una *clave* y un *valor*. Se accede al valor mediante una referencia a la clave. Piénsese en un diccionario de consulta al uso: la definición de una voz es el valor del elemento y el lema al que alude, su clave. En el caso mencionado, la clave es una palabra y los datos la lista de sus sílabas.

y la rima del verso. No obstante, este método presenta algunos aspectos susceptibles de mejora. El primer problema es consecuencia de la identificación de acentos considerando solo la ortografía. Esto hace que determinantes átonos como *la* o *mi* sean considerados como tónicos. Por lo tanto, el primer verso del celebérrimo soneto de Quevedo «Amor constante más allá de la muerte», que es el ejemplo que ofrece Gervás, se interpreta como *ce rrár po drá mís ójos lá pos tré ra*. A efectos de la clasificación de Gervás, este verso no sería problemático, pues solamente considera endecasílabos puros de las cuatro clases principales, por lo que, en la práctica, ese heroico corto se clasificaría igual que uno largo. Sin embargo, cabría preguntarse cómo resolvería versos con acentos impropios, como los que aparecen en ese mismo soneto. El método de resolución de sinalefas también presenta inconvenientes, pues estas se abordan secuencialmente hasta que el verso produce el recuento silábico deseado. Si el número de sinalefas coincide exactamente con el de sílabas supernumerarias, la resolución no plantea dudas, pero si hay más sinalefas potenciales que efectivas, es posible que atacar el verso de izquierda a derecha emplace la sinalefa entre las sílabas equivocadas. Asimismo, Gervás reconoce en el artículo que su programa no provee un método satisfactorio para resolver ambigüedades derivadas de figuras como sinéresis o hiatos. Esta aproximación rinde una precisión del 88,73 % de acuerdo con el propio Gervás.

Tales carencias no se le pasaron por alto a Navarro-Colorado (2015, 2017), quien desarrolló una alternativa mucho más precisa orientada a los versos endecasílabos. En primer lugar, etiqueta gramaticalmente las palabras para decidir si llevan acento métrico o no. Seguidamente, procede a la división silábica y detección de la sílaba tónica. Se decide qué palabras acarrean acento prosódico en función de la etiqueta gramatical. Si el recuento de sílabas es igual a once, se da por resuelto el verso; si no, se aplica el módulo de ajuste y, si hay varias alternativas, el de desambiguación. Esta última se hace atendiendo a la frecuencia de las posibles alternativas en un corpus. De esta manera, solventa el problema mencionado con la distinción de palabras átonas y tónicas del que adolecía el analizador de Gervás. Por otra parte, las ambigüedades ya no se resuelven linealmente, sino mediante un método que ofrece una mejora significativa y alcanza un 94,44 % de precisión en endecasílabos (Marco y col., 2021, p. 51735).

A las iniciativas pioneras, hay que añadir la de Agirrezabal y col. (2017), que usa redes neuronales para mejorar la aproximación de Gervás hasta el 90,84 % de precisión. En los últimos años, se han añadido otras dos, Rantanplan (Rosa y col., 2020) desde una perspectiva clásica con un 96,22 % de aciertos (Marco y col., 2021, p. 51740) y Jumper (Marco Remón y Gonzalo Arroyo, 2021), que aplica un novedoso método

que prescinde de la silabación, con una precisión del 94,97 % (Marco y col., 2021, p. 51740). Durante las últimas correcciones de esta tesis se presentó ALBERTI (Rosa y col., 2023), un sistema de clasificación basado en BERT, cuyo módulo de predicción de patrones versales alcanza 91,15 % de precisión mediante redes neuronales.

A pesar de los excelentes resultados en verso fijo, la mezcla de versos presenta más complicaciones, pues el número de versos correctamente escandidos cae en Jumper hasta el 81,73 %, en Rantanplan hasta el 78,67 % y el 49,38 % con el analizador de Navarrro-Colorado (Marco y col., 2021, p. 51740).

1.4 Metodología y estructura del trabajo

Este trabajo comienza enfocando el objeto del estudio desde una perspectiva propiamente deductiva. Parte de las generalizaciones de diversas ramas de la teoría literaria para codificarlas formalmente y aplicar esta formalización a un corpus de prueba. A la luz de los resultados de la primera aproximación, afinaremos inductivamente las reglas en situaciones excepcionales. En otras palabras, si se dan casos concretos que se desvían de lo previsto, existe la posibilidad de modificar el modelo considerando esas instancias para que el analizador automático sea capaz de identificarlos y tratarlos en consecuencia durante futuros análisis.

Hemos dividido el trabajo en dos partes, cada una con cuatro capítulos distribuidos de forma paralela. La primera consiste en una presentación descriptiva de los aspectos teóricos. Provee al lector de un compendio de los conceptos requeridos para el desarrollo de la segunda parte. Esta, a su vez, consiste en la formalización algorítmica de los aspectos teóricos vistos y la descripción de los tipos de datos abstractos empleados para estructurar la información textual y metatextual. Ambas partes están organizadas según la misma disposición: en primer lugar, se define el marco y, dentro de este, se va refinando la aproximación desde los trazos más gruesos a las pinceladas más minuciosas. Así lo haremos tanto para presentar el fondo teórico y metodológico como para formalizarlo. Partimos de una perspectiva general y vamos aumentando el nivel de detalle: drama, métrica y fonología; y, en estas, a su vez, de lo general a lo particular.

Empezamos por ubicar el estudio dentro de las humanidades digitales. Este trabajo está orientado a filólogos, a los que suponemos familiarizados con la métrica, la fonología y el drama; sin embargo, es probable que no lo estén tanto con las humanidades digitales[12]. En previsión de diferentes grados de experiencias con lo digital, el capítulo

[12]Al menos, si contemplamos los planes de estudio filológicos de las universidades austriacas y españolas al momento de redactar estas líneas.

dedicado a esta relativamente nueva disciplina no solo aborda aspectos a tener en cuenta en la segunda parte del trabajo, sino que presenta una breve introducción a la historia y naturaleza de esta rama humanística. Una vez hechas las presentaciones, definimos los principios ontológicos que emplearemos. Esto es, determinaremos las categorías a buscar y, para ello, debemos conocer los entes susceptibles de ser objeto de estudio así como las interrelaciones que establecen entre sí. Tomaremos prestadas para ello algunos conceptos de la *teoría de la información* para concretar la idea de *datos* y sus diferentes tipos porque es la manera en que manejaremos las distintas categorías.

Una vez establecidas estas premisas, disponemos del marco de referencia en el que ubicar las partes concretas de la obra teatral y así identificar y organizar sus componentes. Para ello, en el siguiente capítulo de esta parte, nos referiremos al drama y la dramétrica para establecer los elementos estructurales susceptibles de ser materia de estudio cuantitativo; así, determinamos las entidades a caracterizar a la hora de elaborar el modelo formal y, más importante, cómo organizarlas. En la consecución de este objetivo, además, hemos de caracterizar no solo qué entidades deseamos aislar como abstracciones, sino también cómo se plasman estas en el texto impreso en la práctica o, en nuestro caso, en la edición digital. Aparte de las particularidades estructurales, emplearemos este capítulo para acercarnos a los aspectos textuales del corpus que tomamos como modelo. Consideraremos, pues, las convenciones editoriales de las ediciones modernas usadas y discutiremos la elección de estas.

Cuando seamos capaces de descomponer el texto dramático en sus partes, deberíamos centrarnos de manera particular en una de ellas: el verso. Sin embargo, el teatro no se compuso para ser leído, sino representado ante el público. Por lo tanto, analizar sus versos teniendo en cuenta tan solo su grafía, muy especialmente cuando, como en el caso que nos ocupa, se pretende proporcionar datos sobre su ritmo y sonido, resulta una solución no del todo satisfactoria. Por este motivo, hemos optado por abordar los versos desde una perspectiva fonológica, lo que tratamos en el tercer capítulo de la primera parte. Emplearemos la fonética articulatoria como marco teórico para representar los versos en la formalización, por lo que nos centraremos particularmente en ella, pero introduciremos algunas nociones de fonética acústica cuando la ocasión lo requiera, que resultarán de gran utilidad para que dirimir por la vía experimental ciertas ambigüedades.

En el último capítulo de la parte teórica, nos ceñiremos a las prescripciones establecidas por la métrica española. Por este motivo, tomaremos de las poéticas aquellas descripciones que permitan definir reglas en la segunda parte tanto para interpretar los versos como para agruparlos en unidades superiores, como estrofas o poemas.

La segunda parte de este trabajo se ocupa de formalizar los elementos teóricos presentados en la primera. No obstante, antes de entrar de lleno, comenzaremos por explicar algunos aspectos metodológicos y técnicos de la implementación del modelo de pruebas y las herramientas empleadas, así como por justificar la elección de algunas de estas, tales como el lenguaje de programación Python o unas aplicaciones informáticas auxiliares, en lugar de otras de funcionalidad equivalente. La decisión de no ubicar este capítulo al final es deliberada: en los subsiguientes no entraremos en el diseño material que usamos en el computador, sino que presentaremos algoritmos y remitiremos al apéndice para ver su materialización en código informático. Sin embargo, algunos de los aspectos del diseño *en papel* se entienden mejor conociendo las cortapisas impuestas por la programación para tratar textos digitales en un entorno real, con sus limitaciones y particularidades. Esto es, aunque el código de los apéndices es una plasmación de los algoritmos propuestos, estos se concibieron con la idea de ser trasladados a un entorno computacional. Asimismo bosquejaremos una visión general del mecanismo de análisis textual, de manera que se aprecie cómo se unifica lo expuesto por separado en los siguientes capítulos, cómo interaccionan entre sí los diferentes módulos descritos.

Comenzamos por analizar la manera de transformar en datos el texto dramático, tal y como lo encontramos en su edición digital corriente, para organizar la información constituyente de modo que acepte el tratamiento digital de manera sencilla, tan directa como sea posible. Con el texto así preparado, comenzaremos el análisis automático para identificar, extraer y organizar como datos *estructurados* la información susceptible de estudio cuantitativo. Este paso, si bien instrumental para nuestro propósito, ofrecería ya un resultado finalista si el objetivo es codificar la información textual de comedias sin enriquecerla con otra derivada de análisis adicionales, como pudiera ser el metro de cada verso. Los resultados se ven en las adiciones a CalDracor del proyecto Sound and Meaning, codificadas a XML-TEI mediante la implementación práctica de la aproximación que se presenta en los capítulos sexto y séptimo.

La escansión automática emplea textos transcritos fonológicamente, por lo que, antes de ponernos con ella, nos ocuparemos en el siguiente capítulo del módulo aparte que aborda la tarea. Para afrontar la transcripción fonológica, trataremos, por una parte, la transformación mediante reglas de los grafemas en fonemas y, por otra, las de división silábica. Asimismo, presentaremos los resultados de las pruebas realizadas para determinar la precisión del sistema.

Finalmente, acometeremos la escansión métrica automática. Comenzaremos dando una visión general del procedimiento. Seguidamente, iremos deteniéndonos en los módulos que la desarrollan para examinar su funcionamiento en detalle. Esto es, de

qué manera se aplica la teoría métrica mediante reglas lógicas y qué casos aconsejan introducir excepciones como respuesta a los resultados. En la evaluación y pruebas, emplearemos obras de la colección de autos de Calderón de la Barca editados por el GRISO (2020) y el corpus de sonetosADSO (Navarro-Colorado y col., 2016). Este último se ha empleado en los estudios más recientes como *gold standard* para la evaluación de la escansión automática de sonetos (Marco y col., 2021, p. 51735).

Al completar ambas partes, estaremos por fin en condiciones de volver a las preguntas de investigación arriba formuladas, discutirlas según las vicisitudes encontradas a lo largo del periplo que aquí comienza y plantear respuestas de la mejor manera posible.

Finalmente, hemos incluido cuatro apéndices. El primero contiene un resumen de la tesis en alemán para satisfacer los requerimientos de la Universidad, así como su traducción española e inglesa. El segundo apéndice ofrece una implementación de lo presentado en esta tesis al lenguaje de programación *real*, con el código Python empleado para someter a prueba la validez de nuestra aproximación. El tercero describe de forma sucinta el uso de los programas desarrollados a partir de las ideas aquí expuestas, tanto para sugerir aplicaciones posibles como para demostrar la sencillez de su uso. El último apéndice consiste en un análisis cuantitativo de la primera jornada de *La vida es sueño*, que ilustra mediante el ejemplo las demandas de potenciales estudios de esa índole.

Parte I

Fundamentos teóricos

Ces gens là serviront d'exemple terrible à tous ceux qui prennent le monde pour un théâtre où la force technique et le triomphe de cette force mènent librement leur jeu.

Jean-Luc Godard, *Alphaville, une étrange aventure de Lemmy Caution*

Humanidades digitales 2

2.1 ¿Qué son las humanidades digitales?

En esta primera parte, introduciremos los elementos teóricos de los que habremos de valernos en la segunda. Como es menester, hay que contextualizar la tarea que nos ocupa en el marco metodológico en el que se ubica. La travesía que en este puerto iniciamos transcurre por las aguas de las *humanidades digitales*. Es de recibo, pues, perfilar ese ámbito de actuación para navegarlo con garantías y el primer reto que se plantea es dilucidar la naturaleza epistemológica y ontológica de la disciplina. Después de todo, sus lindes constreñirán el trabajo y, precisamente por eso, debemos apurarlos cuanto sea posible.

La labor no se antoja sencilla. Las humanidades digitales son un campo en continuo y rápido desarrollo, una disciplina que se expande abrazando nuevos objetos de estudio a la vez que produce otros a resultas de los trabajos en curso. Por si fuera poco, los avances técnicos y metodológicos de los últimos años han acelerado el paso. A causa de esto, se complica proponer una definición válida y, con más razón si cabe, una que lo siga siendo ante los embates del tiempo en un entorno siempre cambiante. ¿Cómo abordar un concepto que abarca elementos tan diferentes a primera vista como los análisis cuantitativos asistidos por computador, la digitalización de documentos y, de la conjugación de ambos, los archivos digitales, visualizaciones de datos, sistemas de información geográfica, materiales multimedia y un sinfín de aplicaciones más?

Intentaremos acercarnos a la respuesta a lo largo de las siguientes páginas en la medida de lo posible, si bien con la prevención de que lo haremos desde una perspectiva filológica, por ser esta la que nos concierne. No obstante, rogamos al lector que no pierda de vista que cuanto exponemos es, por lo general, extrapolable a *textos* en el

sentido más amplio, tal y como entiende el término la semiótica, allende lo puramente lingüístico y, por lo tanto, aplicable a otras disciplinas humanísticas.

Quizás podríamos evitarnos problemas si atajamos la discusión aquí mismo confiando la respuesta a la fe y la esperanza, aceptando caritativamente la primera propuesta de Scholes y Wulfman: «If "*fiat lux*" worked for the designer of this universe, with light appearing in answer to its name, perhaps "*fiat studies digitales*" will work for this field. The magical power of names should be respected. *En arche en ho logos* indeed» (Scholes y Wulfman, 2008, p. 50). ¿Sería lícito afirmar que las humanidades digitales han estado latentes en un limbo ontológico a la espera de ser nombradas para empezar a existir?

Contra lo que pueda parecer a primera vista, la argumentación de Scholes y Wulfman no deja de tener cierto poso de verdad. Los estudiosos empezaron a valerse de lo que hoy se conoce por humanidades digitales como herramienta y no como método. Los primeros experimentos apenas suponían una diferencia cuantitativa con respecto a los estudios más tradicionales; conceptualmente, la velocidad en que se manejan registros manual y mecánicamente es un detalle irrelevante, con independencia de que, a efectos prácticos, suponga un cambio descomunal. Valga la metáfora, los corpus textuales eran una suerte de *Mare Nostrum* que podía navegarse con una galera o una nao para llegar a un mismo puerto, sin más distinción que el esfuerzo humano requerido por una y otra embarcación y la carga transportada. El punto de inflexión lo marca el aprovechamiento de la capacidad del barco para lanzarse a mar abierto en una larga travesía, atravesar el océano incluso. Bastaría la voz de un almirante ordenando traspasar las Columnas de Hércules y singlar hacia el poniente. Y, aun así, aunque el audaz marinero triunfara en su empresa, no tendría una ruta de las Indias, sino una osada aventura, al menos hasta que diera noticias del Nuevo Mundo a su regreso. Si no se conoce el destino, no existe ruta. Como cantó el poeta sevillano, «se hace camino al andar».

De la misma manera, podría decirse que los pioneros de la *humanities computing*[13] tampoco estaban abriendo una nueva vía, sino adaptando las herramientas modernas, ahora a su alcance, a necesidades sempiternas, usaban naves más grandes para transportar la misma mercancía a los mismos destinos, tal y como lo habían hecho sus predecesores hasta aquel momento, si bien de forma más eficiente. Sin embargo, los protohumanistas digitales más visionarios se percataron de que esa mejorada capacidad podía emplearse de otro modo, al igual que, en la navegación, el volumen de las modernas embarcaciones fue decisivo para alcanzar sin escalas costas más lejanas.

[13]Nos quedaremos con el anglicismo para la etapa histórica de la disciplina y usaremos *computación aplicada a las humanidades* como concepto general.

Era cuestión de tiempo que alguien intentara ponerlo en práctica para llegar más allá, hasta donde nadie lo había hecho antes. Como con los navegantes de antaño, según se iban transgrediendo los límites conocidos por la computación aplicada a las humanidades, se difuminaba también la naturaleza exclusivamente instrumental del computador y, además, con ello, la validez descriptiva de aquel nombre primigenio.

Dejando a un lado digresiones navales y metafísicas, lo cierto es que las definiciones más estrictas excluyen aspectos hoy asociados a las humanidades digitales, mientras que las más laxas tienden a serlo tanto que comprenderían cualquier cosa remotamente relacionada con lo digital o con las humanidades. ¿Es el candidato lo suficientemente humanístico, lo suficientemente digital? ¿Nos limitamos a la disciplina académica o incluimos su objeto de estudio? Podemos estudiar medios tradicionales digitalmente, pero también hacerlo tradicionalmente con medios digitales. Lo mismo que se aplican métodos computacionales al estudio de la literatura, puede estudiarse un videojuego desde una perspectiva estrictamente narratológica. Si hablamos de *humanities computing*, el límite parece claro, pero el término *humanidades digitales* da pie a las más diversas interpretaciones, algunas incluso mutuamente excluyentes.

Por lo tanto, resulta más sensato acercarse lentamente al concepto que nos ocupa antes de aventurar una definición, dando vueltas en espiral en torno a él para aproximarse en cada ciclo un poco más al punto central, a la esencia. Empezaremos por establecer qué hacen las humanidades digitales, lo que, según Burdick y col. (2012, pp. 17-20), es ocuparse fundamentalmente del *curación*, *análisis*, *edición* y *modelado* de datos. Con curación, se refieren a aquellas actividades dedicadas a la selección y organización de materiales, como podrían ser los archivos o las bibliotecas digitales. En el ámbito de las humanidades digitales, esto también implica un elemento interpretativo, pues aprovecha los recursos digitales para mejorar la transmisión del material y contextualizarlo. Dicho de otro modo, no tratar el objeto como un elemento autónomo, sino dentro de la red que constituye junto con otros objetos y con sus respectivos metadatos.

Añadimos asimismo una definición finalista, señalando su objeto. Las humanidades digitales se ocupan de textos —entiéndase, insistimos, en el sentido semiótico y no lingüístico, porque ahora no nos referimos específicamente a la filología, sino a las humanidades en general—, y estos textos no se limitan a los tradicionales, pues pueden adoptar nuevas formas gracias a la técnica. Nos referimos también al empleo de la tecnología digital para facilitar el acceso a información humanística tradicional, como hacen las ediciones digitales y las colecciones de documentos digitalizados, pero también al empleo archivístico para almacenar y catalogar materiales de un modo que sería imposible de no existir esta tecnología. Hay quien sostiene incluso que los actos

de creación artística asistidos por un computador forman parte de las humanidades digitales (Scholes y Wulfman, 2008, pp. 52-55).

No tendríamos problema en seguir añadiendo descripciones desde diferentes perspectivas, por ejemplo, mediante la enumeración de algunas propiedades singulares que poseen. Así, Unsworth (2002) sostiene que, por un lado, se trata de un modelo mimético del objeto que pretende estudiar, pero también una manera de razonamiento, así como un conjunto de compromisos ontológicos. Además, todo lo anterior está determinado por el equilibrio entre la necesidad de una computación eficiente y la comunicación humana.

De la misma manera, Drucker (2021, p. 2) llama la atención sobre las diferencias en los materiales, su procesamiento y la presentación de los resultados obtenidos como rasgos propios de las humanidades digitales. Recalca la importancia de los materiales porque adaptar las fuentes a un formato digital requiere una mediación que tiene implicaciones en el modo en que estos podrán usarse. Esto, precisamente, debemos considerarlo con extremo cuidado a la hora de elaborar la parte empírica de nuestro trabajo, pues la forma en que se dispongan los resultados determinará en gran medida el ámbito de las investigaciones capaz de valerse de ellos.

En cuanto a la función mimética, partiremos de la premisa de que podemos crear representaciones de un objeto, pero serán siempre incompletas. La mímesis ha de ser necesariamente una simplificación porque solo el propio objeto se representa a sí mismo con total precisión. Incluso si fuera posible hacer una copia exacta a nivel atómico de un objeto dado, la copia seguiría careciendo de propiedades esenciales del original, como su ubicación en el espacio y en el tiempo. Para las obras de arte, hablaríamos del *aura* de la pieza original (Benjamin, 1935/2013). A pesar de esta incompletitud, los modelos no dejan de ser una herramienta preciosa para el desarrollo de los humanos, que se han servido de ellos desde sus orígenes, milenios antes de que alguien pudiera siquiera soñar con la computación. Nuestro conocimiento de todos los aspectos del mundo se basa en modelos parciales que vamos desbastando, puliendo y perfeccionando a lo largo del tiempo.

Supongamos, por ejemplo, una lista de frecuencias de las palabras de una obra literaria. Tal lista no es el texto, pero se refiere a él en cierta manera y lo describe parcialmente. No se trata de una descripción completa —que solo el propio texto podría ofrecer—, sino que está sesgada: muestra únicamente la parte que hemos decidido representar. Conocemos las palabras y su frecuencia; sin embargo, ignoramos dónde aparecen y su posición en contexto respecto a las demás palabras.

Las diferentes formas de representación suponen modos de razonamiento derivados

de ellas. Unsworth apela a Minsky (1975), quien sostiene que, ante una situación nueva, uno toma de su memoria un *marco* — esto es, un escenario estereotípico— que recuerda y le es familiar, que adapta a la realidad mediante la alteración de detalles allí donde sea menester. Por lo tanto, si hemos de enfrentarnos a nuevas formas de representación, normalmente, construiremos un marco de referencia a partir de nuestra experiencia previa, y lo modificaremos en función de las características del nuevo sistema frente al que nos encontramos.

Volviendo a la lista de frecuencias, no resultaría muy descabellado aventurar una jerarquía de importancia según la lista. Podríamos incluso dejar solo las palabras de contenido léxico pleno y hacer inferencias semánticas al modo de una exégesis tradicional. Sin embargo, la representación jerárquica forzaría un proceso discursivo diferente al de la simple lectura. O, como lo formulan Burdick y col. (2012), la computación depende de la desambiguación en todas sus etapas, lo que contraviene la ambigüedad y presupuestos implícitos de las humanidades. El aspecto computacional demanda que los humanistas trabajen en los términos impuestos por las limitaciones tecnológicas. Paradójicamente, los trabajos humanísticos computacionales han modificado el proceso a su vez, por lo que se han ideado vías para introducir la ambigüedad y la contingencia en la informática.

Habíamos dicho que la representación y el modo de razonamiento obligan a aceptar un conjunto de compromisos ontológicos. Esto es, surgen nuevas entidades que han de tenerse en cuenta. Unsworth (2002) pone como ejemplo una edición según las directrices de la Text Encoding Initiative (TEI) de la *Divina Comedia* de Dante. En ella se marcan los cantos, estrofas y versos a la manera usual, pero también los nombres propios y epítetos, se distingue el origen de las fuentes entre históricas, mitológicas, bíblicas y literarias, los tipos de animal o persona, y otra metainformación. En el texto TEI se representan nuevas entidades, los nombres, y estos pertenecen a distintas categorías susceptibles de ser estudiadas. Los metadatos reflejan el texto, ofrecen un modelo cuyas entidades representan una parte de él.

¿Hasta qué punto debemos considerar *nuevas*[14] esas entidades? En el ejemplo, estas existen en la edición TEI, aunque no en la representación tradicional. Cabría aducirse que las fuentes siempre han tenido el mismo origen, lo que no deja de ser correcto. No obstante, en el caso del texto por sí mismo, el modelo no es una parte intrínseca de él, sino que es una construcción externa. Un erudito familiarizado con las fuentes

[14]Markus Ebenhoch, muy acertadamente, me hizo notar aquí la conveniencia de matizar que la novedad no es histórica, sino respecto al propio texto. Estos metadatos en poco se distinguen conceptualmente de las milenarias glosas. La diferencia estriba en el incremento astronómico de la capacidad para glosar el texto digitalmente que, como se verá en Apartado 2.7, puede dar pie a enfoques muy diferentes a los clásicos.

dantescas que leyera la *Comedia* por primera vez podría ir clasificando todas estas mentalmente según avanza en la lectura. Sin embargo, un modelo así construido es extrínseco al texto, está solo en la mente del lector y, además, de forma incompleta, pues ha de irse construyendo necesariamente al vuelo, al paso que la vista progresa, verso a verso, terceto a terceto. El aparato crítico, por otra parte, no es mimético en circunstancias normales, pero tiene potencial para serlo: imaginemos, que describimos el texto en notas según los criterios de marcado mencionados en el párrafo precedente. La suma de todas las notas a pie de página ofrecería una idea precisa de algunos aspectos del texto, como el uso de las fuentes o el tipo de personajes que aparecen y en qué momento.

Si vamos a poner a la máquina a estudiar las relaciones entre los nombres de los personajes de la *Divina Comedia*, necesitaremos tener en cuenta el modo de operación del ingenio electrónico para plantearle cuestiones no solo computables, sino que también puedan resolverse con eficiencia. Poco podemos hacer formulando un problema de complejidad exponencial cuya solución requiriera cientos de años[15]. La otra cara de la moneda es la necesidad de servir a la comunicación humana. De escasa utilidad es que el problema sea computable si el resultado no ofrece información manejable ni sirve de consuelo que algo sea teóricamente posible si la complejidad de su codificación para que lo entienda la máquina escapa del alcance del encargado de la tarea.

En el prefacio de su influyente libro, Schreibman y col. circunscriben los ámbitos potenciales de las humanidades digitales a una interpretación más estricta del término y limitan su espectro a la aplicación de métodos computacionales a las obras humanísticas y a los resultados de estos trabajos. Lo ponen de la siguiente manera:

> In the digital humanities, critical inquiry involves the application of algorithmically facilitated search, retrieval, and critical processes that, originating in humanities-based work, have been demonstrated to have application far beyond. Associated with critical theory, this area is typified by interpretative studies that assist in our intellectual and aesthetic understanding of humanistic works. It also involves the application (and applicability) of critical and interpretative tools and analytic algorithms [...] on those artifacts produced through processes associated with archival representation made available via resources associated with processes of publishing and the communication of results. (Schreibman y col., 2004b, p. xxv)

[15]En la práctica, incluso mucho menos podría llegar a ser un margen intolerable si lo imponen los plazos.

Desde otra perspectiva, Laura Hernández-Lorenzo (2022b) reconoce la importancia de la interdisciplinariedad, que es obligada al tomar prestadas herramientas y métodos hasta ahora ajenos, trabajar con grandes cantidades de datos —posibilitado por lo anterior— y adoptar una sintonía con los avances tecnológicos actuales, que establece un nexo tangible con el desarrollo de la sociedad. Llama la atención también sobre otras características más notables, todas ellas con una vertiente ética y otra metodológica que se retroalimentan mutuamente. Estas incluyen la forma colaborativa de trabajo y su internacionalización, propiciados por la difusión y utilización de métodos y código abierto. Pasamos, pues, del estudioso solitario aislado en su despacho al equipo que, además, puede encontrarse repartido por todos los puntos del orbe[16].

Vemos así que, cuando se habla de humanidades digitales, nos hallamos ante un vasto campo en el que se entrecruzan y mezclan las letras y la computación, las humanidades y las ciencias exactas. Se centra en fuentes primarias tradicionales, pero también en nuevos objetos producidos digitalmente. Comprende el archivo de esos materiales, así como la democratización de su acceso[17]. No se limita a la curación de los textos, sino que abraza su análisis digital. Su metodología parte de las humanidades, pero la aplica usando herramientas computacionales. Demanda estrecha colaboración entre investigadores, aunque no depende de la coincidencia geográfica del equipo. De la combinación de todo esto, surgen nuevas ontologías, que suscitan a su vez nuevas preguntas, cuya respuesta exige nuevos métodos. Por lo tanto, debemos cuestionar estos métodos en primer lugar si albergamos la intención de aplicarlo con garantías.

2.2 De *computación de humanidades* a *humanidades digitales*

Las humanidades digitales, incluso ya entrado el segundo milenio, se tienen por una aproximación novedosa en el ámbito de la filología. Sin embargo, aunque su aparición no se remonte a los inicios de la disciplina como rama de las ciencias humanas en los albores del siglo XIX —si bien podrían encontrarse precedentes, como que Ada Lovelace, hija del poeta romántico Lord Byron, o que la matemática Anabella Milbanke, planteara emplear la pionera máquina de cálculo de Charles Babbage para fines distintos a las ciencias exactas (Hernández-Lorenzo, 2022b, p. 153)—, su historia va ya para los tres cuartos de siglo. Fueron asimismo los requerimientos de la investigación

[16]Conocido es el problema de la despoblación rural en España (Esteban Rodríguez, 2022). Cabría plantearse si las facilidades técnicas podrían ayudar a revertir esta tendencia demográfica del nuevo milenio.

[17]Esto, hasta cierto punto, ha cristalizado en los principios FAIR (Wilkinson y col., 2016) y su acogida por la comunidad científica.

filológica los que dieron el impulso inicial a la *humanities computing* y la fuerza motriz que mantuvo en marcha la nueva metodología durante sus vacilantes primeros pasos. Aún hoy, se siguen malinterpretando con frecuencia las humanidades digitales como una rama de la lingüística aplicada, a pesar de que hace tiempo que forman parte de la caja de herramientas de aquellos que estudian campos tan diversos como las artes plásticas o la historia, la música o los medios de comunicación de masas. No entraremos más allá, puesto que es la investigación filológica la que nos motiva, pero emplazamos al lector curioso a completar lo aquí expuesto en otro lado (Burdick y col., 2012; Berry, 2011b; Rio Riande, 2014; Schreibman y col., 2004a).

Para encontrar el germen de las humanidades digitales debemos retroceder hasta la Italia de la década de 1940, donde, como cuenta Winter (1999), el joven Roberto Busa, hermano de la Compañía de Jesús, cavilaba sobre la metafísica de la *presencia* en Santo Tomás. Alrededor de 1946, comenzó a escudriñar el corpus tomístico en busca de los términos *praesens* y *praesentia*, solo para concluir que eran periféricos y, si acaso, con cierto grado de conexión con la preposición *in*. Busa iba anotando las ocurrencias en tarjetas de cartulina, de las que llegó a tener más de diez mil. Como es de suponer, mantener un índice manual de todas las palabras resultó ser al cabo del tiempo una tarea inasumible para el padre Busa y sus colaboradores.

El jesuita no cejó en su empeño. Bien al contrario, en 1949 se desplazó hasta los Estados Unidos para emprender un peregrinaje por las universidades de ese país en busca de una solución a su problema. Por una de esas jugadas de la fortuna, Busa llegó a la Universidad de Columbia precisamente cuando se hallaba allí Thomas J. Watson, cerebro de la empresa IBM a la sazón, que trataba de hacer presión en favor de la candidatura de Dwight D. Eisenhower para la presidencia de la institución académica. IBM ya había comenzado por aquel entonces a desarrollar los primeros computadores para fines comerciales; ese casual encuentro no pudo tener más felices consecuencias, ya que el tenaz jesuita italiano consiguió convencer a Watson para que pusiera a su disposición los medios que necesitaba.

Pese al escepticismo inicial de Watson en cuanto a la adecuación de las máquinas para la tarea, Busa creía en la idea de tomar herramientas desarrolladas originalmente para otros fines —como el comercio o los cálculos matemáticos en la investigación científica— y encontrarles otras aplicaciones. En la mente del italiano, esas «otras aplicaciones» significaban una sola cosa: un índice de concordancias de las obras completas de Santo Tomás de Aquino. De vuelta en Italia y ya con los medios técnicos a su disposición, el jesuita empezó por grabar el texto en tarjetas perforadas, que era el medio de almacenamiento de aquellas máquinas primitivas antes de que los soportes magnéticos tomaran el relevo en el panorama computacional. Una vez codificado

el corpus de esta manera, escribió un programa para encontrar las concordancias. Habían nacido las humanidades digitales.

Busa no se había equivocado. Aquel visionario consiguió aplicar a la filología herramientas que habían sido diseñadas con vista a los registros de datos que requiere el mundo empresarial y, de esta manera, fue el creador del primer generador automático de concordancias, que publicó en 1951: el *Index Thomisticus* (Busa, 1951). Este índice trascendió su propio fin, porque, con él, se sentaron las bases de una nueva metodología de análisis textual y, por extensión, de aproximarse a las disciplinas humanísticas. Se trataba de lo que se dio en llamar *humanities computing*.

Otras iniciativas reseñables para encontrar concordancias, en la misma línea que el proyecto inicial del padre Busa, fueron la del reverendo John W. Ellison con la versión estándar revisada de la *Biblia* en inglés, que se completó en 1957. En aquella ocasión, no se emplearon máquinas de IBM, sino computadores Univac, y la de Parrish y Painter, que abordaba la literatura reciente con el poeta americano decimonónico Matthew Arnold y el irlandés W. B. Yeats (Burton, 1981, pp. 6-8), fallecido apenas dos décadas antes.

Ese mismo año, encontró la muerte Guy Montgomery, sin haber completado su índice de palabras de John Dryden. Sus colaboradores continuaron el trabajo y, para ello, volvieron a recurrir a IBM. El resultado lo publicaron algunos años más tarde (Montgomery y Hubbard, 1957). Aquello permitió hacer apreciaciones cuantitativas respecto a la poesía de Dryden, como la recurrencia de los términos *care*, *fate*, *friend*, *head* y *mind*. Con esto, el equipo de Montgomery estableció los fundamentos de los diccionarios de frecuencias de palabras y rimas que hoy continúan en servicio (Burton, 1981, p. 4).

También de la misma época datan los primeros esfuerzos en torno a la traducción automática. Andrew Booth y Richard Richens fueron pioneros en este campo, aunque la idea ya había surgido en la década precedente.

> The suggestion that modem computing machines could be used for the purposes of translation originated with the present author. It arose because, in 1946, various new uses for automatic digital calculating machines were being considered and these ranged from the more obvious applications to problems of mathematics and physics, to philosophical problems such as the mechanization of human thought processes, the playing of games and the translation of language. (Booth, 1958, p. 88)

El primer experimento se llevó a cabo en 1951 y consistía en traducir oraciones en ruso formadas con un cuarto de millar de palabras. La prueba obtuvo los resultados que Booth y Richens habían previsto, pero también confirmó sus sospechas en cuanto a las deficiencias que podían esperarse. En cualquier caso, se consideró lo suficientemente prometedor como para que pudieran seguir profundizando en la traducción por computador. La investigación siguió adelante y, algunos años después, publicaron todos los detalles del primer algoritmo de traducción automática (Richens y Booth, 1955). Booth recibió una posición en la Universidad de Londres a mediados de la década de 1950. Allí estaba surgiendo un cierto interés por aplicar los métodos digitales a la obra de Platón, en gran medida gracias al deseo de Leonard Brandwood de desarrollar una manera de encontrar el contexto de cualquier palabra en una edición determinada. A resultas, se simplificó sobremanera el análisis de los ritmos (Brandwood, 1956, pp. 46-49). Finalmente, Brandwood decidió concentrarse en el léxico y empezó a perforar tarjetas en abril de 1960. Esto llevó a su equipo quince meses. Verificar las tarjetas y corregir los errores, preparar las cintas y ejecutar el programa, varios miles de horas (Burton, 1981, p. 6).

Al mismo tiempo que el equipo de Montgomery se dedicaba a Dryden, y Booth y Richens a la traducción, Harry Josselson y Howard Hyatt estaban usando también equipos de IBM para elaborar una cuenta de palabras rusas (Josselson, 1953) extraídas de prosa y drama de los grandes autores del siglo XIX. Su objetivo era determinar las frecuencias de las categorías gramaticales en esa lengua y su distribución temporal y respecto a los géneros literarios. No se trataba ni de un índice ni de una compilación de concordancias, pero supuso una gran contribución para el estudio de estas últimas. Introdujeron reglas para marcar las entradas en las tarjetas y nuevas maneras de validar y ordenar los datos (Burton, 1981, pp. 4-5).

Entretanto, el desarrollo y la innovación en la metodología computacional progresaba a pasos de gigante. En 1953, un equipo de investigación de la Armada de los Estados Unidos, a las órdenes de la entonces capitán de corbeta Grace Hopper, empezó a desarrollar FLOW-MATIC, un primer lenguaje de programación de alto nivel y presentó el prototipo inicial en 1955 (Hopper, 1978). Este avance supuso que ya no sería necesario programar los computadores en código de máquina, sino que, a partir de entonces, se podrían controlar usando instrucciones en inglés. La invención no podía ser más oportuna, pues IBM abría su negocio al público general también en 1953 con su modelo IBM 701. Aún quedaba prácticamente una década para que los lenguajes de alto nivel destronaran al código de máquina, el tiempo que se necesitó para que un público más general y sin experiencia previa en computación reemplazara a los científicos y técnicos en el perfil del usuario mayoritario.

Ajeno a lo que se estaba llevando a cabo en los laboratorios militares, el padre Busa continuaba con el método digital «tradicional», aunque los medios disponibles mejoraron sustancialmente durante la siguiente década. Esto le dio la oportunidad de expandir su empleo a otros textos y lenguas clásicas. En 1957, estableció el *Centro per l'Automazione dell'Analisi Letteraria* en Gallarate, dedicado principalmente al estudio de textos en griego y latín (Burton, 1981, p. 3). La flexibilidad de la metodología del jesuita se demostró con la indexación de los *manuscritos del Mar Muerto*, que habían sido descubiertos en los años en que Busa comenzaba su periplo digital. Los textos provenían de rollos en hebreo, arameo y nabateo, todos ellos idiomas semíticos con diferentes escrituras paleográficas sinistroversas (Busa, 2019).

Al poco de comenzar el decenio de 1960, el Centro de Gallarate contaba con una treintena de empleados. Para 1964, este contingente ya había duplicado su número. Seis de aquellas personas se dedicaban a tareas administrativas, catorce a la preparación de textos latinos, griegos y hebreos, cuatro a programar los computadores y las casi cuarenta restantes —más de la mitad del personal— estaban encargadas de perforar las tarjetas e inspeccionarlas visualmente (Busa, 1965).

Precisamente, Busa expuso esto durante la Literary Data Processing Conference, que se celebró en septiembre de 1964 en las instalaciones de IBM de Yorktown Heights, en el estado de Nueva York. Esta conferencia tuvo el honor de ser el primer evento académico sobre humanidades digitales. En sus actas, ya se bosqueja una panorámica bastante fiel a los problemas que afronta la disciplina y las preocupaciones que suscita: el interés en la lexicografía, la edición textual, la enseñanza de la lengua, y la estilística, aspectos metodológicos tales como la programación y los datos de entrada y de salida, y el reto de salvaguardar los textos electrónicos (Hockey, 2004, p. 7).

A pesar del tremendo éxito de aquel jesuita del Veneto, la computación seguía restringida a unos pocos centros de estudio, aunque continuaba avanzando con paso firme. Hasta bien entrada la década de 1970, solo una minoría tenía acceso a máquinas capaces de llevar a cabo los cálculos que requerían los estudios filológicos digitales. Los investigadores podían tener la idea y la técnica podía haber desarrollado una solución, pero eso no era ni muchos menos garantía de que pudiera llevarse a cabo. El coste de los equipos continuaba siendo prohibitivo y el espacio requerido para alojarlos, desmesurado. Aquellos ingenios electrónicos requerían grandes instalaciones y personal muy especializado. Además, codificar los textos seguía demandando mucha mano de obra. De ahí que las iniciativas dependieran siempre, en gran medida, de la disponibilidad de un centro de computación y grabadores de datos. Disponiendo de todo esto, además, no debía faltar tampoco la paciencia, porque el siguiente trabajo en la cola de espera solo se empezaba a procesar una vez acabado el anterior.

Tuvieron suerte en este sentido Mosteller y Wallace (1963), pues contaron con la ayuda técnica de un equipo de investigación de IBM dirigido por el recientemente fallecido pedagogo Albert E. Beaton. El objeto de su estudio eran unos textos de cierta importancia en la corta historia estadounidense, una colección de 77 ensayos breves que se habían publicado de manera anónima entre 1787 y 1788 en *The Federalist* para persuadir a los habitantes del estado de Nueva York de ratificar la recién redactada *Constitución de los Estados Unidos*. La autoría de casi todos los ensayos estaba establecida con solidez, pero doce aún permanecían disputados y, de tres escritos al alimón, se ignoraba la extensión de la contribución de cada autor.

La cuestión se venía discutiendo desde finales del siglo XIX. A ninguno de los posibles padres de los textos en disputa le faltaron paladines dispuestos a romper cuantas lanzas fuera menester para defender a su predilecto. No hace falta decir que, en muchos casos, los argumentos se basaban en impresiones personales de los exégetas. La propuesta de Mosteller y Wallace era mucho menos susceptible de ser criticada por su subjetividad: tomaron las palabras más y menos frecuentes y compararon su uso. Esto es, verificaron si algún autor usaba léxico poco común más de lo habitual, pero también cómo se distribuían los vocablos más típicos —tales como las preposiciones *by*, *from* y *to*—, ya que la frecuencia de las comunes permanece más o menos invariable cualquiera que sea el tema, al contrario de lo que ocurre con las palabras de mayor carga léxica.

A la luz de las pruebas realizadas, Mosteller y Wallace encontraron una alta probabilidad de que todos los textos disputados salvo uno hubieran salido de pluma de Madison, así como las contribuciones más extensas en dos de los tres ensayos escritos en colaboración. En la conclusión de su artículo, subrayan que sus «datos complementan de manera independiente los del historiador». Observaron asimismo —y de esto se siguen aprovechando hoy en día los más modernos estudios de atribución de autoría—, que la distribución de palabras frecuentes como elemento discriminador permanece estable a lo largo de décadas y entre diferentes géneros. Habían completado el primer estudio estilométrico de atribución de autoría basado en la *lectura distante*, concepto que se presentará más adelante.

Hay que mencionar que ya Alvar Ellegård (1962) se había adelantado al examinar las cartas que escribió Junius para criticar el gobierno del rey inglés Jorge III. No obstante, si bien Ellegård se había ayudado de la informática para llevar a cabo los cálculos estadísticos, tanto la recuperación de palabras como su cuenta y clasificación se realizó de forma manual (Hockey, 2004, p. 5).

Estos no fueron los únicos estudios estilísticos llevados a cabo durante aquellos años.

No en vano, algunas propuestas de análisis estilométrico basado en la frecuencia de las palabras se habían adelantado más de un siglo a los intentos digitales. La diferencia estriba en que esos primeros análisis no tenían manera de manejar los volúmenes de texto que la introducción del computador hizo posible tratar. Por ejemplo, de acuerdo con Hockey (2004, p. 5), ya en 1851 Augustus de Morgan había propuesto examinar la autoría paulina de las epístolas atribuidas al santo a partir del examen cuantitativo del léxico. Trae a colación Hockey también que, más de una centuria después, el clérigo escocés Andrew Morton tomó el testigo, pero este pertrechado de los adelantos técnicos del siglo XX. A partir de la frecuencia las palabras más comunes y algunos cálculos estadísticos relativamente sencillos, Morton (1965) concluyó que Pablo solo había escrito cuatro de las epístolas que comúnmente se le atribuyen.

También en las Islas Británicas, poco más o menos por aquellos mismos años que los otros pioneros de la filología digital, el medievalista Roy Wisbey empezó a interesarse por las posibilidades que ofrecían las concordancias para el estudio de textos en alto alemán medio. A resultas de su propia experiencia (Wisbey y Bott, 1968), comenzó a usar computadoras en Cambridge a comienzos del verano de 1960 para preparar índices, concordancias y material lexicográfico de textos germánicos medievales. Presentó una solicitud a su universidad para establecer un centro de computación apenas empezaba a entrar el otoño. El *Literary and Linguistic Computing Centre* de la Universidad de Cambridge se estableció formalmente en 1964.

Los equipos también los proporcionaba IBM —esto no ha de extrañar, pues la competencia apenas estaba empezando a despegar—. La iniciativa británica se cuidó muy especialmente de que las máquinas fueran capaces de leer e imprimir símbolos diacríticos. A pesar de ello, el trabajo seguía siendo complicado. Wisbey y Bott dan cuenta de que, una vez que el Centro había aceptado un proyecto, el filólogo y el programador debían de acordar cuál era el formato más adecuado para perforar las tarjetas. Esto implicaba evaluar cuidadosamente todos los signos que podían aparecer, así como potenciales fragmentos en lenguas diferentes a la principal del texto. Después de eso, había que transcribirlo usando dispositivos perforadores. Para facilitar la tarea, en lugar de hacerlo con la equipación genérica provista de todos los diacríticos, se aligeraba el mecanografiado empleando equipos ideados para diferentes idiomas modernos europeos.

El sistema se organizaba en torno a un programa de clasificación grabado en una cinta magnética. Mientras que la propia empresa fabricante de las máquinas proveía este programa para la mayoría de las aplicaciones comerciales, en Cambridge tuvieron que programar las máquinas ellos mismos para adaptarlo a sus necesidades particulares. A pesar de esto, descartaron la idea de un programa general de concordancias o un

índice de palabras. De hecho, se rechazaba cualquier proyecto que levantara sospechas de ser ya no irrealizable, sino «muy ambicioso». Debemos pensar que, como dijimos, solo se llevaba a cabo un trabajo cada vez y este podía tomar una gran cantidad de tiempo en terminarse, durante el cual, cualquier otro proyecto debía permanecer a la espera.

Los recursos se empleaban eminentemente para trabajos de carácter lexicográfico, centrados casi exclusivamente en textos europeos medievales —alemán, español, francés, inglés y neerlandés—. Contando únicamente los textos en alto alemán medio, la suma ascendía a tres cuartos de millón de palabras. Entre los proyectos que se llevaban a cabo en el Centro, estaban índices de concordancias de dos ediciones diferentes de un mismo texto, que proporcionaban ediciones duales; un diccionario de rimas del alemán, junto a un programa capaz de identificar pares de rimas en un texto; un generador de frases a partir de una gramática; un comparador de textos que encuentra pasajes divergentes. El fruto más jugoso de aquel árbol fue quizás el *Vollständige Verskonkordanz zur* Wiener Genesis (Wisbey, 1967).

En el continente, destacan las digitalizaciones que se llevaron a cabo en el *Centre de recherche pour un Trésor de la langue française*. El centro se estableció en la Universidad de Nancy en 1960 bajo el auspicio de su entonces rector Paul Imbs para crear un diccionario de la lengua francesa a la imagen del *Oxford University Dictionary*. La digitalización comenzó en 1963 con equipos Gamma 60 de la firma francesa Bull, pero, si bien la digitalización fue crucial en la composición de la obra, no se puso todo ese potencial a disposición del público hasta 1992, año en que se publicó la primera edición en CD-ROM[18] (Papelier, 2005).

La década de 1960 no solo fue productiva en el ámbito lexicográfico, sino también tremendamente prolífica en otros muchos aspectos de la computación aplicada a las humanidades. En 1963 se presentó TUSTEP, la primera herramienta para la edición crítica filológica de textos. Este programa, abreviatura de *TUebinger System von TExtverarbeitungs-Programmen*, fue ideado por un equipo dirigido por el teólogo y filólogo Wilhelm Ott en la Universidad de Tubinga (Hockey, 2004, p. 7). Hoy sigue en desarrollo activo, con su última versión (Zentrum für Datenverarbeitung, 2023) publicada en el año que se redactaron estas líneas y cuenta con una fiel comunidad

[18]No podemos dejar de mencionar que aquellas computadoras Gamma 60 entraron en el imaginario de la cultura popular por el papel que una de ellas representó en la película del recientemente fallecido Jean-Luc Godard *Alphaville*. Una máquina idéntica a las que, en aquel tiempo, ya estaban recopilando datos lexicográficos en Nancy, convenientemente rebautizada como Alpha 60 para la ocasión, encarnaba —en la medida en que una máquina es capaz de encarnar— al antagonista cibernético del hierático detective Lemmy Caution.

de usuarios[19].

La informática aplicada a lo humanístico, si bien alejada aún décadas de la madurez de las humanidades digitales, había alcanzado ya un punto de desarrollo notable, lo que propició también la reflexión sobre la propia disciplina y en el marco de esta. Esto fructificó en el nacimiento de la primera publicación científica dedicada específicamente a este ámbito. Se tituló *Computers and the Humanities* y su primer número vio la luz en septiembre de 1966. En su presentación leemos el siguiente desiderátum:

> We recognize the strong fear often expressed that machinery will destroy the intangibles of humane scholarship—the intuitive, subtle responses of the trained mind. Distinguished voices have already warned that a fascination with method may seduce us away from meaningful goals. Such warnings are well taken, and our editorial limits will constantly exclude technique for its own sake. We need never be hypnotized by the computer's capacity to count into thinking that once we have counted things we understand them. («Prospect», 1966)

Aquella declaración de intenciones sigue sonando actual más de medio siglo después. A nadie familiarizado con el tema le llamaría la atención encontrar tal descargo en las más recientes publicaciones, ya que en ellas no suelen faltar otros en idénticos o muy semejantes términos. El tiempo no parece haber hecho mella en la necesidad de los detractores más viscerales de las humanidades digitales de hacer críticas, fundadas o no, ni moderado la querencia de sus practicantes por abrir el discurso con prevenciones de tal índole —más por el deseo de apaciguar a aquellos que por llamar la atención sobre algo de tal obviedad que a pocos se les escapa—. Para bien o para mal, estos fenómenos siguen jalonando ambos lados del sendero por el que discurren disciplinas que, para buena parte de sus practicantes, se definen

> not just by their subject domains, but by a specific set of methods, emphasizing qualitative description of selected case studies. If we start from this assumption, it will look like a definitional error to describe new quantitative projects as an expansion of the humanities. At best, we will be able to say that research using numbers dilutes the distinctive character of the

[19]Valdés Gázquez (2104a, p. 39, n. 43) apunta a la causa que ha podido relegar este sistema a una posición secundaria en el mercado internacional: los manuales y documentación se encuentran exclusivamente en alemán.

> humanities; at worst, we will fear it as a Trojan horse subverting the humanities' true purpose, which is presumably to serve as a counterweight to science. (Underwood, 2019)

Volviendo a la relación de hitos, en la década de 1970, surgieron centros de computación asociados a academias europeas de diversas lenguas con el fin de facilitar el trabajo lexicográfico en la composición de diccionarios, como el *Instituut voor Nederlandse Lexicologie* en Leiden, dedicado al neerlandés (Hockey, 2004, p. 4). Los vagos nexos que relacionaban los dispersos esfuerzos investigadores tomaron cuerpo en forma de asociaciones científicas, creadas con el fin de promover el desarrollo de la *humanities computing*. Así, la *Association for Literary and Linguistic Computing* se conformó en 1977 y, un año más tarde, la *Association for Computers and the Humanities* (Hockey, 2004, p. 8).

Tal vez, el desarrollo más notable de la década no fue en lo que se podía hacer con las máquinas, sino en cómo llevarlo a cabo. Los investigadores habían tomado conciencia sobre la existencia de otros proyectos similares a los suyos y que esto llevaba no pocas ocasiones a la duplicidad de esfuerzos. Con esta premisa en mente, se desarrolló en Gran Bretaña la segunda revisión (Berry-Rogghe y Crawford, 1973) de COCOA (D. B. Russell, 1967), un programa cuya primera versión había presentado la Universidad de Cardiff a finales de la década anterior. La novedad conceptual de la nueva reinterpretación era que no estaba diseñada para cumplir un único propósito en un equipo determinado, sino que podía aceptar diferentes corpus y funcionar en equipos distintos. Esta necesidad de aprovechar el trabajo llevó a crear el Oxford Text Archive en 1976, una primera biblioteca digital dedicada a recopilar y mantener los textos digitales empleados en las investigaciones y ponerlos a disposición de otros investigadores. A la iniciativa de Oxford, hay que sumar otras colectivas, como el *Thesaurus Linguae Graecae* (TLG), a expensas de la Universidad de California Irvine, que más tarde se amplió con una base de datos de textos latinos (Hockey, 2004, pp. 8-9).

De la década de 1970, resultan reseñables asimismo dos aspectos relacionados indirectamente con *humanities computing*. Por una parte, la entrada en escena de los discos magnéticos como sustitutos de las cintas —que habían reemplazado a su vez a las tarjetas perforadas— para el almacenamiento de datos. Esto supuso que ya no fuera necesario procesar los datos de manera secuencial, lo que agilizó el empleo de los computadores. Piénsese que, hasta entonces, la información se almacenaba en cintas, cuyos principios y mecanismo no diferían mucho de las cintas de audio o de vídeo domésticas que todos conocemos. Por lo tanto, del mismo modo que no se requería

pasar toda la cinta de música sin perder la vista el contador de vueltas para llegar a la canción deseada y rebobinarla por completo para escucharla desde el principio, también era menester pasar de manera mecánica por toda la cinta de datos hasta llegar a los buscados. La irrupción del disco magnético, al igual que sucedió después con el disco compacto en la industria musical, eliminó de un plumazo esta limitación. Por otra parte, empezaron a impartirse cursos sobre computación aplicada a las humanidades e incluso se consideraron los lenguajes de programación como ejercicios de disciplina mental al modo que se había usado tradicionalmente el latín (Hockey, 2004, p. 9).

La siguiente década supuso una revolución en la difusión y alcance de la informática. Esta se popularizó gracias a que las máquinas habían reducido su precio y tamaño y, a la vez, aumentado su capacidad. Microcomputadores, como el Sinclair ZX Spectrum o el Commodore 64, entraron en muchos hogares y el IBM PC vio una implantación en el ámbito profesional como ningún otro computador lo había hecho antes (Fradejas Rueda, 2022b). Aquellos trabajos que, hasta hacía nada, eran prerrogativa de los pocos afortunados que contaban con el respaldo de un centro de computación, se hallaban ahora accesibles a cualquiera. Al mismo tiempo, los conocimientos técnicos necesarios para operar un computador y los recursos humanos para hacerlo se habían reducido a velocidad pareja a la de los costes. Ya no era necesario contar con un equipo de trabajadores para perforar tarjetas durante meses porque una sola persona podía introducir directamente los datos valiéndose del teclado; tampoco se requerían programadores especializados capaces de interactuar con la máquina en su propio código digital, sino que los lenguajes de alto nivel con los que venían equipados los computadores[20] franqueaban el paso a los legos.

Hay que señalar un hito que aconteció en 1987. Ese año se organizó una reunión en el Vassar College de Poughkeepsie para evaluar la posibilidad de definir un estándar de codificación humanística de textos. Hasta entonces, los intentos de unificar formatos habían sido infructuosos. Un año antes se había presentado un nuevo método de codificación a instancias de la ISO denominado SGML, siglas de *Standard Generalized Markup Language*. Este formato era ideal para codificar una gran variedad de textos, incluyendo metadatos, lo que abría la puerta no solo a la representación del texto, sino a la inclusión de aquella información adicional que requerían los investigadores (Hockey, 2004, pp. 12-13). Aquello fue el germen de la Text Encoding Initiative, más conocida por sus siglas TEI. La primera edición de sus directrices se publicó en

[20]Nótese que el usuario de los tres populares microcomputadores mencionados interactuaba por efecto con la máquina mediante comandos del lenguaje BASIC.

mayo de 1994 y varios proyectos las adoptaron poco después. Apenas había comenzado 1999, cuando la Universidad de Virginia y la Universidad de Bergen presentaron una propuesta conjunta al comité ejecutivo de TEI para la creación de la organización internacional TEI Consortium para mantener y desarrollar el estándar TEI. La iniciativa noruego-estadounidense fue aceptada y pronto se unieron al consorcio la Universidad de Brown y la Universidad de Oxford. La prioridad de la recién creada organización era presentar una versión XML de las directrices, lo que cuajó a mediados de 2002. Estas eran una mejora de las primigenias, sin cambios sustanciales salvo la propia codificación y la resolución de algunos errores que se habían detectado. En 2007 se revisaron a fondo las directrices y se presentó la versión actualizada que hoy sigue en uso (TEI, 2005).

De acuerdo con Rio Riande y González Blanco García (2015, p. 3), fue la década de 1990 la que vio las primeras aproximaciones a la computación aplicada a las humanidades en el mundo hispanohablante. Da fe de ello la obra de orientación eminentemente filológica de Marcos Marín (1994). Se lamenta (pp. 55-70) de la escasa atención que se le ha prestado a la innovación lingüístico-tecnológica desde las instancias oficiales, aunque reconoce varios proyectos surgidos al amparo de la largueza financiadora que propiciaron los fastos del quinto centenario del descubrimiento de América. Llama la atención sobre la necesidad de compilar un corpus del español estandarizado conforme a TEI (Marcos Marín, 1994, pp. 79-90). No fue a caer en saco roto, pues, en 1995 (Rojo, 2016, p. 197), se aprobaba la creación del Corpus de Referencia del Español Actual de la Real Academia Española (RAE), que se desarrolló a una velocidad inusitada (Pino, 1995; Pino y Santalla, 1996). Desde entonces, la disciplina no ha dejado de avanzar.

Alrededor de 2004, aconteció un cambio terminológico con implicaciones conceptuales profundas, para visibilizar un nuevo paradigma. Lo que se había dado en llamar *humanities computing* pasó a ser uno de los muchos aspectos de un concepto mucho más amplio que se bautizó como *digital humanities* o, como las conocemos en español, humanidades digitales. A la popularización del término contribuyó la influyente antología monográfica editada por Schreibman y col. (2004a). El motivo de esta transformación era prevenir la concepción de la disciplina como el mero empleo de herramientas digitales para acelerar el estudio tradicional. Se trataba ahora de expandirla para tratar con entidades digitales desde la perspectiva humanística y no con entidades humanísticas con herramientas digitales. Esto es, «they point toward a new way of working with representation and mediation, what might be called the digital 'folding' of reality, whereby one is able to approach culture in a radically new way» (Berry, 2011a, p. 1). Ironías del destino, ese mismo año también dejaba de

publicarse la histórica *Computers and the Humanities* (Crymble, 2021, p. 6), como queriendo ratificar así la nueva tendencia. Dicho de otra manera, con este término, la disciplina pretendía trascender su estatus de servicio complementario de apoyo y erigirse como una actividad intelectual por derecho propio, rigurosa y con un universo teórico por explorar (Hayles, 2011, p. 43).

2.3 Datos

En el curso de estas contextualizaciones metodológica e histórica, hemos nombrado en varias ocasiones el término *datos*, pero hemos eludido ofrecer una definición precisa, confiando en el buen entendimiento del lector, pues es una noción de corriente en un mundo cada vez más digitalizado. Llegados aquí, la idea intuitiva se antoja insuficiente y se hace necesario precisar a qué nos referimos cuando hablamos de datos porque el concepto se materializa de muy diversas maneras en función de su propósito.

Mientras que los datos son un concepto ubicuo en las ciencias sociales —las encuestas con las que se elaboran las estadísticas con las que nos desayunamos a diario, por ejemplo—, siguen resultando *rara avis* en las humanidades: su empleo dista de poder darse por supuesto. Los métodos que usaba la disciplina hasta hace poco casi en exclusiva no se prestaban a clasificaciones formalmente estables, por lo que las nuevas aproximaciones han traído consigo requerimientos metodológicos y estos, a su vez, interpretativos. Quizá la recolección de datos sea uno de los puntos que más han influido en esta tendencia, ya que requieren gran cantidad de tiempo y recursos (Ehrlicher y Lehmann, 2019, p. 41).

La extracción de datos es precisamente el pilar donde se sustenta el proceso, pues la fiabilidad del análisis depende de la calidad de los datos, como bien sugiere la máxima informática *garbage in, garbage out*. Inversamente, la calidad de los datos producidos condiciona su interpretación. Como extractores y organizadores de esos datos, no podemos ignorarlo. De nada sirven unos algoritmos impecables si producen datos en un formato ilegible y, al contrario, esos mismos algoritmos poco tienen que ofrecer si los datos de entrada son irrelevantes, erróneos o intratables. Veamos entonces cómo abordaremos la cuestión.

Una primera aproximación al concepto *datos* la hacemos definiéndolos como la «información, en cualquier forma, con la que opera un computador» (Butterfield y col., 2016, *s.v. data*), que no se aleja mucho de nuestra noción intuitiva del significado del término. Desde una perspectiva etimológica, *data* es el plural de *datum*, que es el

género neutro del participio pasado del verbo *dare*. O sea, 'dado' o, nominalizado, '[algo] dado'. En el siglo XVII se empezó a emplear en cálculos matemáticos para indicar presupuestos *dados* (*Online Etymological Dictionary*, 2022, *s.v. data*). Sin embargo, este uso se extendió en la segunda mitad del siglo XX para adquirir nuevas connotaciones acordes a los nuevos tiempos. De esta manera, en la década de 1960 algunos sectores concretos de la administración estadounidense ya definían datos como

> un término general empleado para denotar alguno o todos los números, letras y símbolos, o hechos que se refieren a o describen un objeto, idea, condición, situación u otros factores. Connota elementos básicos de información que pueden ser procesados o producidos por un computador. En algunos casos, se considera que los datos solo pueden expresarse de numéricamente, pero la información no está así limitada. (Bureau of the Budget, 1962, *s.v. data*, traducción propia)

Esto conduce al término *información*, que no es otra cosa que una «colección de hechos u otros datos, especialmente los derivados del proceso de datos» (Bureau of the Budget, 1962, *s.v. information*). Obviando la circularidad de la definición, parece que estamos ante dos conceptos relacionados, ¿pero cuál es esta relación? Una manera simple de entenderlo es hablar del «sentido que un ser humano concede a los datos» (National Telecommunication Information Administration, 1997, *s.v. information*, traducción propia). Tal vez una descripción más conveniente para nuestro propósito sea la que da la siguiente fórmula: $información = datos + estructura$ (Bradley, 2004, *s.v. data*), por lo que, despejando la ecuación, datos serían información sin estructura.

Los datos pueden ser *cuantitativos* o cualitativos. Cuantitativos, como su nombre indica, son los susceptibles de contarse que proporcionan información a través de su tratamiento matemático: el número de palabras de un texto; de estas, cuántas son léxicas y, de estas últimas, cuántas son adjetivos femeninos singulares, por ejemplo. Los datos *cualitativos*, por el contrario, son aquellos otros que, no siendo susceptibles de cuantificación, permiten hacer valoraciones interpretativas descriptivas: la calidad del texto, su ambientación, la profundidad psicológica de sus personajes... En el punto medio, tenemos datos categóricos si, por ejemplo, una palabra es léxica o un adjetivo femenino singular. Este tipo de clasificación depende en gran medida de la persona que juzga los datos. En cualquiera de los dos casos, hemos resuelto la ecuación anterior, si bien el primero y el último se estructuran mediante algoritmos formales y el segundo solo admite una interpretación holística.

Así pues, atendiendo a la posibilidad de definiciones categóricas, no debemos confundir lo digital con lo cuantitativo. Es cierto que el poder de cálculo del computador lo hace singularmente apto para contar y llevar a cabo operaciones aritméticas, pero recordemos por un momento su denominación *ordenador*: el computador ordena datos y los presenta de manera propicia para su interpretación cualitativa. Y, al contrario, la metodología cuantitativa no requiere necesariamente de asistencia digital[21], como ya habíamos visto antes cuando hablábamos de los precedentes históricos de la estilometría de nuestros días[22].

La cuantificación existe desde mucho antes de la existencia del computador digital. Si bien es innegable que se ha usado de manera cuando menos discutible, valga como ejemplo la tabla del médico y poeta Mark Akenside bajo seudónimo Musiphron (1746), en la que presenta una escala de poetas europeos de todas las épocas siguiendo el ejemplo de la que compuso Roger de Piles (1708, pp. 489-498) para pintores (franceses en su mayoría). Algunas de las magnitudes, como las que se refieren al colorido o el dramatismo suscitan la cuestión de cómo se cuantificaron (Cuadro 2.1). A pesar de todo, hay que reconocer que la subjetividad de estas valoraciones palidece ante la escala que propuso el poeta y músico Christian Friedrich Daniel Schubart para los *jóvenes* poetas alemanes, en la que consideraba aspectos como el *genio* o la *gracia* (Schubart, 1792)[23].

No todos los datos cuantitativos se organizan de igual manera. Se distinguen según su grado de estructuración, que, además, resulta ser un factor determinante del modo en que habremos de tratarlos. De esta forma, se habla de datos *estructurados* y *no estructurados*. La denominación de estos últimos surge como parte de la jerga especializada de las bases de datos para referirse a aquello que no podía guardarse en formato tabular, o sea, *estructurado*. Se denominan, pues, datos no estructurados a los que presentan ambigüedades clasificatorias que hacen inviable su procesamiento digital, aquellos que carecen de una clasificación lógica mediante compartimentaciones explícitas (Brackett, 2014, p. 15). Esto es, por ejemplo, una imagen o un texto digital. Pensemos en una obra de teatro del Siglo de Oro como la que corresponde al

[21]Hoy en día, la frontera se difumina por la omnipresencia del computador en todos los ámbitos. Como muestra de la dificultad para trazar una línea, considérese, por ejemplo, el reciente estudio de Aichinger (2023). En este, los datos se extrajeron manualmente, aunque el computador sirvió de herramienta para presentarlos mediante una hoja de cálculo y llevar a cabo en ella sencillas maguer repetitivas operaciones aritméticas. Cuando la diferencia, como en este caso, no es metodológica, sino instrumental, ¿cabría hablar de *análisis digital*? ¿Dónde ponemos la frontera?

[22]Tenga el lector esta prevención en cuenta y sepa disculpar el pequeño sacrificio de exactitud que haremos en este trabajo en aras de la conveniencia al emplear en ocasiones el término *cuantitativo* para referirnos específicamente a aquellos métodos asistidos por computador.

[23]Ver Spoerhase (2018) para profundizar en estos y otros ejemplos de escalas de este tipo en el siglo XVIII.

	Estructura	Patetismo	Expresión dramática	Caracterización	Gusto	Colorido	Versificación	Moral	**Estimación final**
Ariosto	0	15	10	15	14	15	16	10	13
Boileau	18	16	12	14	17	14	13	16	12
Cervantes	17	17	15	17	12	16	—	16	14
Corneille	15	16	16	16	16	14	12	16	14
Dante	12	15	8	17	12	15	14	14	13
Eurípides	15	16	14	17	13	14	—	15	12
Homero	18	17	18	15	16	16	18	17	18
Horacio	12	12	10	16	17	17	16	14	13
Lucrecio	14	5	—	17	17	14	16	0	10
Milton	17	15	15	17	18	18	17	18	17
Molière	15	17	17	17	15	16	—	16	14
Píndaro	10	10	—	17	17	16	—	17	13
Pope	16	17	12	17	16	15	15	17	13
Racine	17	16	15	15	17	13	12	15	13
Shakespeare	0	18	18	18	10	17	10	18	18
Sófocles	18	16	15	15	16	14	—	16	13
Spenser	8	15	10	16	17	17	17	17	14
Tasso	17	14	14	13	12	13	16	13	12
Terencio	18	12	10	12	17	14	—	16	10
Virgilio	17	16	10	17	18	17	17	17	16

Cursiva del original.

Cuadro 2.1: Escala de los jóvenes poetas según Akenside.

fragmento de la derecha de Figura 2.1 en un procesador de texto. El lector discierne una estructura clara en la disposición espacial de los elementos de la izquierda si conoce ciertas convenciones. Sabrá que las letras en versalita sin sangrar corresponden a una didascalia de personaje, el texto sangrado corresponde a parlamentos y cada línea se relaciona con un verso, excepto si están en cursiva, en cuyo caso se trata de una acotación. El cerebro ha procesado las marcas visuales y ha estructurado la información.

Desde el punto de vista del computador, sin embargo, el texto es algo como lo que aparece a la derecha de Figura 2.1. Ni siquiera eso, porque lo que vemos en pantalla no es sino una forma de representar una serie de grupos de bits, cada uno con la codificación adecuada para cada carácter, desde 01100001 para la letra *a* hasta

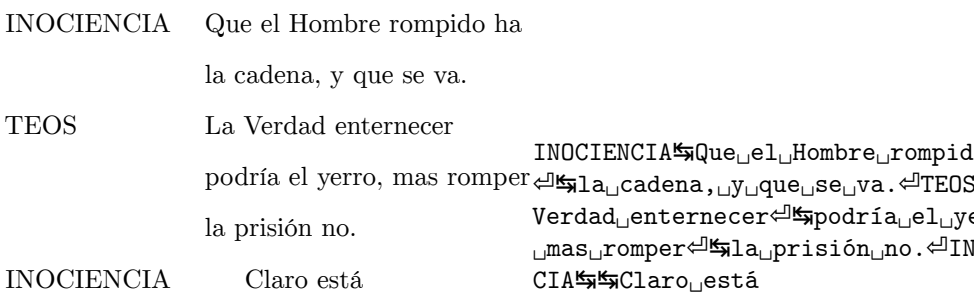


Figura 2.1: Representación visual y cadena de caracteres.

00001010 para el salto de línea. Incluso esta notación no deja de ser una abstracción porque, en realidad, tampoco son ceros y unos, sino estados de carga eléctrica o magnética. Olvidémonos de estas concreciones y volvamos a las representaciones abstractas. Letras, números, símbolos o espacios en blanco son todos un elemento de la misma clase. Si queremos dividir por palabras, por ejemplo, la máquina necesita ser instruida para reconocer los caracteres alfabéticos, encontrar los límites de cada agrupación (espacios, signos de puntuación, etc.) y ordenar las divisiones resultantes en una lista para construir así una estructura de orden superior.

Los datos estructurados son aquellos que están ordenados de acuerdo un patrón específico y sus elementos tienen relaciones susceptibles de ser representadas en formato tabular. En este caso, podría tomarse cada línea de texto, asignarle una fila, e ir añadiendo la información adicional conocida en nuevas columnas[24].

Línea	Personaje	Parlamento	Verso
Que el Hombre rompido ha	INOCENCIA	200	1030
la cadena, y que se va.	INOCENCIA	200	1031
La Verdad enternecer	TEOS	201	1032
podría el yerro, mas romper	TEOS	201	1033
la prisión no.	TEOS	201	1034
Claro está	INOCENCIA	202	1034

Cuadro 2.2: Datos estructurados.

Aun conteniendo exactamente los mismos grafemas, estos datos estructurados ya no son la fuente de Figura 2.1 porque, durante el proceso de categorización, hemos hecho juicios al considerar unos criterios y descartar otros (Drucker, 2021, p. 21). Como consecuencia, se ha perdido por el camino información: en el ejemplo, incluso con todos los grafemas intactos, ha desaparecido toda la composición tipográfica, desde el tipo de letra al espaciado. Por una parte, esto refleja lo incompleto del modelo y, por otra, pone de manifiesto que los datos son tan objetivos como lo haya sido su

[24]En el ejemplo, modernizamos la didascalia de personaje para tener un locutor normalizado si deseamos comparar con otras obras.

selección. En otras palabras, el límite de la objetividad de una investigación digital está limitado por la que sea capaz de alcanzar el propio investigador.

A las dos categorías de datos vistas, habría que añadir una tercera intermedia, a la que se denomina datos *semiestructurados*. En realidad, se considerarían como una modalidad diferente de datos estructurados, pues se definen como aquellos en los que el esquema estructural es intrínseca; son, por así decirlo, autodescriptivos. Por el contrario, en los datos estructurados en el sentido clásico, la estructura y los datos son entidades diferentes. Para ilustrarlo, recurriremos al formato semiestructurado por excelencia, XML, en conformidad con las directrices de TEI. El Listado 2.1, como vemos, presenta una disposición jerárquica en la que las etiquetas delimitan los niveles, los metadatos como el número de verso u otras de sus propiedades, como ser una parte de un verso compartido, se marcan como atributos de la etiqueta del texto al que conciernen[25].

Listado 2.1: Datos semiestructurados.

```
<sp who="#inocencia">
    <speaker>INOCIENCIA</speaker>
    <l n="1030">Que el Hombre rompido ha</l>
    <l>la cadena, y que se va.</l>
</sp>
<sp who="#teos">
    <speaker>TEOS</speaker>
    <l>La Verdad enternecer</l>
    <l>podría el yerro, mas romper</l>
    <l part="I">a prisión no.</l>
</sp>
<sp who="#inocencia">
    <speaker>INOCIENCIA</speaker>
    <l part="F">Claro está</l>
</sp>
```

En lo tocante a este trabajo, damos por sentado que el grueso de las obras que se hallan disponibles se encuentra como datos no estructurados, por lo que se hace necesario prepararlos previamente. En el futuro, tal vez deje de ser este necesariamente el caso, dado el creciente número de ediciones en XML-TEI, como las del proyecto DraCor (Fischer y col., 2019).

[25]En este caso, la didascalia de personaje no se normaliza, pero sí el identificador único. Esto lo veremos en más detalle en próximos capítulos.

2.4 Metodología *dirigida* y *asistida* por datos

Ya con una idea más precisa de a qué nos referimos cuando hablamos de datos, intentaremos una clasificación metodológica más fina según la forma de emplearlos. La parte práctica de este trabajo no basa sus resultados en el procesamiento o el análisis de los datos, sino que pretende producirlos como resultado de aplicar a los textos los conceptos teóricos vistos en estos primeros capítulos. No se trata tanto de definir en esta sección cómo nos conduciremos en la segunda parte del trabajo, sino de esbozar sus fines últimos, los posibles usos potenciales de los resultados. Solo así conoceremos qué es lo que debemos producir. No perdamos de vista que nos proponemos extraer y ordenar datos de textos teatrales y el interés teórico del proceso sirve a un fin ulterior: pretendemos que esa labor sea un medio que conduzca a resultados aplicables a la investigación. Siendo así, no podemos obviar este aspecto, ya que sería tanto como ignorar las necesidades de aquello a lo que, en último término, está orientado todo el esfuerzo.

¿Cómo pueden valerse las investigaciones humanísticas de este trabajo? La respuesta determinará los datos que debemos extraer y cómo los organizamos una vez extraídos. Contestarla pasa por especificar las metodologías predominantes para abordar la evaluación de los datos. Escobar Varela (2021, pp. 7-8) hace una primera distinción entre dos maneras de utilizarlos. Por un lado, define la metodología dirigida por datos[26] como aquella en que el computador procesa los datos de entrada y produce otros de salida, fruto de la lógica y representaciones formales. Esta metodología apuesta por producir resultados objetivos y reproducibles bajo las mismas condiciones, al modo de las ciencias naturales.

Según Escobar Varela, en las metodologías dirigidas por datos, estos se usan para responder preguntas mediante la creación de representaciones formales de una cuestión y la automatización del procedimiento para llegar a responderla. Además, los criterios con los que se evalúa el resultado se definen antes de comenzar el experimento. Esta aproximación presenta el inconveniente de requerir una simplificación que, necesariamente, ha de dejar de lado muchos aspectos del referente real. Volvemos al problema de la incompletitud del modelo. Sin embargo, concentrarse en puntos concretos permite evaluarlos a una escala imposible desde una aproximación holística. Podemos comparar el ritmo de millones de octosílabos en lugar de dos romances de forma fragmentaria. Esto, además, abre la puerta a descubrir aspectos contraintuitivos, presenta la ventaja epistemológica de dar pie a experimentos reproducibles y producir resultados falsables. Planteamos la pregunta de cuál es la probabilidad de

[26] *Data-driven methodology* en inglés.

que una hipótesis dada sea cierta; otros experimentos han de poder corroborar el resultado de forma independiente.

Las metodologías *asistidas por datos*[27] ofrecen al investigador información para especular e interpretar desde la subjetividad. Se presentan nuevas perspectivas para el problema, abren otras vías de pensamiento o refuerzan o debilitan las ya existentes, pero no dan respuestas. Dicho de otra manera, esta aproximación aporta un punto de vista sobre la cuestión y contribuye así a formar la opinión de quien ha de encontrar los resultados. Permite incluso evaluar el problema en otro marco de referencia. Estas metodologías se prestan especialmente para problemas que son incuantificables por su propia naturaleza. Aquí no se plantea una cuestión a resolver con los datos, sino que estos se usan para matizar la respuesta o plantear nuevas cuestiones (Escobar Varela, 2021, pp. 7-8). En este tipo, se encuentran, por ejemplo, aquellos estudios que recurren a bases de datos para encontrar ocurrencias de un caso particular y las analizan sin que la electrónica intervenga en la interpretación. Tales investigaciones tienen una influencia decisiva en las humanidades digitales, especialmente en cómo se definen los datos para ser interpretados.

Las investigaciones dirigidas por datos tienen la facultad de influir en su contrapartida asistida, en tanto que condicionan la preparación de los datos para un determinado uso, lo que orientará las posibilidades de consulta, pero también tiene la capacidad de producir resultados capaces de modelar qué nuevas preguntas han de plantearse. El caso contrario es igualmente cierto porque la selección de aspectos textuales de interés, la demanda de poner la atención en uno u otro particular, suele preceder al estudio digital. Esto fuerza elecciones ontológicas cuyas consecuencias trascienden el efecto inmediato, lo que se observa muy claramente, por ejemplo, con la ayuda de visualizaciones, a las que Moretti (2019, p. 86) concede la categoría de definitorias de la disciplina

> because visualization is never just visualization: it involves the formation of corpora, the definition of data, their elaboration, and often some sort of preliminary interpretation as well [...] What interests us is visualization as a practice, in the conviction that practices—what we learn to do by doing, by professional habit, without being fully aware of what we are doing—often have larger theoretical implications than theoretical statements themselves. Whether this has indeed been the case for digital humanities, is for readers to decide.

[27] *Data-assisted methodology* en inglés.

Dependiendo del proyecto, una de estas dos metodologías ofrece más posibilidades que la otra, pero también tienen demandas diferentes. Así, según Escobar Varela (2021, p. 12), la metodología dirigida por datos requerirá una definición precisa de los conceptos a observar que, en el caso de los estudios literarios, se traduce generalmente en elementos representados mediante valores discretos; por otra parte, estos valores deben aceptar una aproximación aritmética. Asimismo, los datos deberían poder acumularse para posibilitar la obtención de un corpus más grande a partir de la combinación de otros más pequeños. Naturalmente, esto último solo atañe a aquellos casos en los que tiene sentido, por ejemplo, para examinar la obra completa de un autor como suma de sus trabajos individuales.

De esta manera, tenemos dos formas entreveradas de abordar el objeto de estudio, ambas se influyen mutuamente y su síntesis traza las líneas directrices que señalan el devenir de la disciplina. ¿Pero cómo se aplica esto a la crítica literaria?

2.5 Filología digital

Como dijimos, cada rama de las humanidades aplica los métodos digitales en respuesta a sus necesidades particulares. Para usar una terminología adecuada, procesa un tipo determinado de datos. Para el filólogo, claro está, esos datos son los textos y, en nuestro caso particular, los textos literarios de un determinado género y una época concreta. Por otra parte, podríamos enfocarnos en otros aspectos, por ejemplo, en una caracterización puramente lingüística (Sanz-Lázaro, 2021).

En este sentido, los filólogos debemos considerarnos privilegiados porque, dentro de las humanidades digitales, nuestra disciplina es posiblemente la más antigua y la que más ha avanzado. La lingüística computacional y de corpus, el Procesamiento del lenguaje natural (PLN), las ediciones digitales, los programas de reconocimiento óptico de caracteres... La suma de todo ello conforma una sólida base donde se sostienen nuevos estudios de la rama. Para diferenciarlo de las humanidades digitales en su acepción más amplia, nos referiremos a estos aspectos concretos que nos conciernen con un término más específico: *filología digital* (Andrews, 2013).

Debemos tener en cuenta también que la filología digital no es un bloque monolítico, pues se ramifica en proyectos dedicados a la clasificación de textos, a su recopilación, al análisis y a la difusión (Bay-Cheng, 2017). En muchos casos, se trata de la extensión natural de los procedimientos tradicionales, al modo de las ediciones digitalizadas o, en los análisis, la aplicación de la tecnología a ideas que se venían gestando desde

mediados del siglo XIX, pero cuyo empleo continuó siendo muy trabajoso hasta el advenimiento y popularización de la informática (Hoover, 2013, p. 517).

Difícilmente le sorprenderá a alguien la afirmación de que la crítica literaria se remonta a los inicios de la producción textual. Las obras clásicas, sin ir más lejos, se compusieron en el marco de un campo literario dominado por una estética no solo descriptiva sino también prescriptiva. Las bondades de un texto se medían en función de su observancia de las normas tenidas por canónicas en cada momento histórico. Esto se tradujo asimismo en la aparición de métodos para evaluar el grado de fidelidad a la norma que, a su vez, servían también para comparar obras entre sí. Piénsese, por ejemplo, en la métrica en el griego clásico y la distribución de los pies no ya como criterio clasificatorio, sino como *norma* determinada por el género o determinante de este. Rommel (2004, pp. 88-89) señala que precisamente ese propósito comparatístico era el que tenían los estudios literarios asistidos por computador primigenios en las décadas de 1960 y 1970: buscar e identificar patrones y cadenas en los textos electrónicos. Y no olvidemos que una década antes ya habían empezado a florecer las listas de concordancias, que supusieron sin duda una gran ayuda para la lectura crítica de los textos que cubrían.

Continúa Rommel llamando la atención sobre la enorme ventaja que supone el empleo de la electrónica, incluso tomándola como una mera herramienta carente de capacidad analítica intrínseca. Esto es, como instrumento auxiliar en estudios asistidos por datos, como hemos descrito arriba. La máquina complementa la memoria del humano, pues constituye un archivo ordenado de ocurrencias según los criterios que se le indiquen. Al contrario que los archivos físicos, el medio digital es capaz de almacenar una cantidad prácticamente ilimitada de información sin ocupar espacio. Además, puede recuperar un volumen descomunal de esta antes de que un humano tenga tiempo siquiera de abrir las tapas del ejemplar impreso y buscar el índice [28]. Por otra parte, si la máquina sirve como archivero virtual para sacar registros, no es menos útil como copista. Recorre el corpus incansable y archiva ocurrencias a una velocidad vertiginosa sin importarle el tamaño del texto o el número de ejemplares a escrutar. Estos dejan así de ser un factor limitador del trabajo humano.

La capacidad extendida de análisis ha trascendido su esencia original a nuevas aproximaciones comparatísticas gracias a la disponibilidad masiva de fuentes primarias procesables. Esta nueva suerte de entender las humanidades digitales ya no supone solo un cambio cuantitativo respecto al volumen de texto y el tiempo empleado, sino

[28] Estrictamente hablando, sí ocupa espacio, aunque sea apenas los escasos 1,65 cm^2 de una tarjeta de memoria microSD. La capacidad tampoco es ilimitada, pero, a efectos prácticos, los dispositivos de almacenamiento modernos satisfacen con creces la demandas del corpus literario más exigente.

que la aplicación de la tecnología a la crítica puede adquirir una naturaleza cualitativa. Trasladamos aquí la apreciación de Moretti (2008, p. 117) sobre la diferencia entre la novela china y europea, «a story with a thousand characters is not like a story with fifty characters, only twenty times bigger: it's a different story». No se trata de aplicar sin más nuevas aproximaciones que, incluso estando imbuidas de valores humanísticos, se rigen por métodos equiparables a los de las ciencias naturales (Potter, 1989, p. xxix), sino de explotar las posibilidades que abre la disponibilidad de observar detalles en conjuntos masivos de textos literarios.

El obstáculo con que se encuentra esta nueva metodología reside en la limitación del conjunto de fenómenos textuales que pueden ser objeto de un análisis digital productivo, sea cuantitativo o cualitativo. La causa es que tales fenómenos requieren asociarse con características superficiales del texto que reconozca el computador, por lo general mediante la identificación de ciertos patrones. A pesar de que los computadores tienen una extraordinaria capacidad para ese tipo de análisis, su aporte se reduce a la resolución de problemas textuales operando al servicio de la intuición y perspicacia humana.

Las implicaciones metodológicas están claras, la mayoría de los estudios, tanto guiados como asistidos por datos, se ubican dentro del marco teórico de la estilística. Esto es, los textos serían constructos estéticos que causan una respuesta en el lector mediante el empleo de ciertos elementos en su estructura superficial. Estos llegan a ser tan sutiles que no se reconocen individualmente a simple vista, pero la suma de todos ellos determina la impresión global que provoca el texto. Sin embargo, el computador detecta estos microelementos e identifica patrones en su distribución, lo que puede ayudar a comprender cómo se alcanza dicho efecto estético (Rommel, 2004, pp. 89-90).

Tales implicaciones metodológicas presentan una contrapartida para Rommel (2004, pp. 90-91), quien sostiene que esto lleva a simplificaciones injustas de lo que son los estudios basados en la computación, como que estos son poco más que «contar palabras»[29]. Esto, a su vez, influencia de manera negativa las aproximaciones subsiguientes abordadas desde este punto de vista. Irónicamente, los estudios literarios que emplean herramientas digitales tienden a poner el acento en aspectos teóricos y metodológicos, tales como qué constituye un *texto* y como la definición de tal concepto determina su análisis e interpretación. Resulta de crucial importancia la manera

[29]Al respecto, Álvaro Cuéllar me llamó la atención sobre el prólogo del libro digital de José Manuel Fradejas (2022a), un detallado manual para iniciar a los filólogos en el análisis estilístico con R titulado *Cuentapalabras*: el título es un dardo alusivo a que «los grandes de la materia filológica han desdeñado estos análisis como meros *cuentapalabras* que no superan en nada las técnicas filológicas y de la hermenéutica, pues no tienen interés alguno en aquello que sea empírico y pueda refutar sus apreciaciones».

en que la crítica tradicional trata las características formales del texto y el papel de la electrónica en profundizar en ellas o abrir nuevas vías de entender el objeto de estudio.

A pesar de que aún quede mucho camino por recorrer, Rommel (2004, p. 93) considera que, desde al menos la década de 1970, las computadoras han proporcionado a la filología digital acceso prácticamente ilimitado a textos electrónicos de calidad, programas que dan la opción al filólogo de establecer los términos en que realiza la investigación —y no al revés— y equipos de gran capacidad de proceso y almacenamiento capaces de llevar a cabo las tareas propuestas.

Con todo, para que aplicar la computación a la filología sea factible, es conveniente ceñirse a una serie de buenas prácticas que garanticen tanto resultados productivos —ni triviales ni inválidos— como que los métodos y las herramientas no roben el foco al objeto de la investigación. Potter (1991, pp. 427-428) adelanta los siguientes ocho puntos para abordar los estudios literarios con una metodología digital.

1. Considerar las particularidades lingüísticas del texto.
2. Tener en cuenta la lógica del análisis del discurso.
3. Basar el experimento en la teoría (y tener la preparación para poder criticarla si fuera necesario).
4. Definir cómo la cuantificación pone a prueba la teoría.
5. Evitar las computadoras hasta tener un plan de investigación definido.
6. Diseñar una pequeña base de datos para poner a prueba puntos específicos.
7. Reducir la fase computacional al mínimo.
8. Describir el trabajo de una forma clara y replicable.

2.6 Una digresión epistemológica

Resulta difícil hablar de filología digital sin referirse a los métodos cuantitativos. Sin ellos, la disciplina queda despojada de uno de sus principales valores. A cambio, debemos observancia a dos clases de principios. Por una parte, los ontológicos, que llevan a considerar los tipos de fenómenos textuales que se dan y las analogías que estos presentan según sus características contables. Por otra parte, hemos de prestar

atención a los principios analíticos, que llevan a decidir qué conjuntos de propiedades son comparables, qué métodos se prestan para la comparación y cómo deben interpretarse los resultados. Incluso si nos limitamos a contar elementos textuales, es necesario definir el significado de *contar* y de *elementos textuales*: qué ha de contarse, por qué y cómo. Hay que definir también una topología con el corpus, las categorías en las que este se organiza y las reglas que las relacionan tanto con el corpus como entre sí. Finalmente, se necesita un marco analítico para explicar la manera en que las medidas de los datos textuales en una topología muestran diferencias entre sus categorías (Gavin, 2022). En este trabajo, nos concentramos en la ontología, dejando que nuestros resultados sean el objeto de la parte analítica.

Llegados a este punto, no se le escapará al lector que navegamos entre dos aguas, a un lado las del constructivismo y a otro las del positivismo. Por una parte, aquellos que consideran el texto literario como un constructo subjetivo, cada una de cuyas interpretaciones es tan válida como el lector o la escuela exegética estimen oportuno; por otro, los que sostienen que no existe interpretación aceptable más allá de la físicamente verificable. Nosotros, sin embargo, intentaremos abrirnos paso entre Escila y Caribdis.

García Barrientos (2020, p. 15) plantea las dificultades que encuentran los estudios literarios a resultas de no desarrollarse como una *sucesión* a la manera que lo hacen las ciencias naturales. Pone como ejemplo la *Poética* de Aristóteles, que aún hoy sigue siendo una lectura imprescindible. Más allá de su valor en la *historia* de la estética teatral, es fundamental y válida como teoría *viva*. Concluye que la poética está por ello más cercana a los clásicos literarios que a los científicos, en tanto que los últimos solo admiten la lectura como historia, pero no como ciencia. El estudio de la literatura, al igual que otras disciplinas humanísticas, no admite la simple reducción a un compendio de hechos objetivos. Coincidimos, pues, con Cameron en su afirmación de que la mayoría de escuelas de crítica literaria

> do not seek to construct new empirically based knowledge, but are rather concerned with explication and interpretation – enterprises which have little to do with the teleological narrative of scientific 'progress'. Granted, philologists might 'make progress' towards elucidating the meaning of ancient texts which were previously obscure because of gaps in our linguistic or historical knowledge, but critics do not 'make progress' towards the 'true' interpretation of a Keats sonnet; they merely offer different readings. Unlike a new scientific theorem, moreover, a new reading of a text does not automatically supersede all previous interpretations. Rather

> it is of interest to the extent that it reveals additional meanings in a text, proposes hitherto unnoticed connections between texts, or foregrounds themes in texts which resonate with the concerns of a particular moment. (Cameron, 2011, p. 67)

Aquí, Cameron responde a las tesis de Gottschall (2010), quien sostiene en la primera parte de su libro que la crítica literaria académica ha caído en la irrelevancia social; achaca esto a deficiencias metodológicas que impiden el progreso de la disciplina de una forma acumulativa como en las ciencias naturales. Critica asimismo la división entre las humanidades y otras disciplinas, ve en esta separación la causa que ha mantenido a la crítica literaria ajena a los métodos y teorías que han desarrollado otros campos. En su lugar, arguye Gottschall, los estudios literarios se han enrocado en un castillo de naipes teórico, cuyas asunciones se sostienen mutuamente de manera circular, pero sin fundamento empírico: «it's turtles all the way down», podríamos resumirlo tomando prestada la conocida anécdota de Bertrand Russel (Hawking, 1995); en el mejor de los casos, matiza Gottschall, la circularidad acaba en el argumento de autoridad sentenciado en su momento por alguno de los grandes popes de la disciplina. A estas deficiencias, añade, hay que sumar una perversión en la actitud, pues considera que la misión de buena parte de las humanidades no es hacer avanzar el conocimiento, sino practicar alguna variante de activismo, por lo que su ejercicio académico es un mero medio para la consecución de objetivos ajenos a la búsqueda de la verdad.

Cabe criticarle a Gottschall que tome tan solo una parte muy concreta de la academia anglosajona para representar un campo tan amplio como dispar. En concreto, dirige sus diatribas contra aquellos reductos en los que el *constructivismo duro* es una ideología política que frisa la fe religiosa. Deja de lado las posiciones más moderadas, como las tradiciones críticas europeas, precisamente aquellas con los pies bien puestos en el suelo. El terreno de la ecdótica, por ejemplo, sigue relativamente libre de la dogmática posmoderna. Al crítico se le exige una mínima justificación textual para airear sus filias y fobias. No deja de asistirle la razón a Gottschall, empero, en lo que respecta a la tendencia de las humanidades a elaborar marcos teóricos autorreferenciales blindados contra la falsación, y de ahí que convivan más o menos felizmente Aristóteles con el último grito en teorías críticas identitarias.

Parece sensata la afirmación de Cameron de que no hay una *verdadera* interpretación de Keats sino distintas lecturas. No obstante, habría que plantearse si todas ellas son igualmente válidas si admitimos que nuestro objeto de estudio es el texto — en su sentido más inclusivo, sin entrar en si desde el formalismo buscamos en él la

literariedad o el estilo desde la estilística—. Pongamos por caso un ejemplo exagerado —aunque nos tememos que perfectamente posible—: podría deconstruirse el poema «To the autumn» hasta concluir que se trata de una oda a la primavera. Por peregrina que pudiera resultar tal lectura a primera vista, poco se diferencia cualitativamente de interrogarlo para encontrar en él agravios sociales reales o imaginarios propios de nuestro tiempo. Para curarse en salud, podría incluso blindarse la conclusión con un marco teórico refractario a la refutación o, mejor aún, hacer del método una trampa saducea, de manera que cualquier intento de rebatir el argumento confirmara su validez.

No cabe duda de que las dos interpretaciones pueden ser una contribución a la encomiable causa de sendos exégetas hipotéticos. Por el contrario, en lo que respecta a la compresión de la obra literaria, no parece grande su aportación, si bien puede ser reveladora sobre el intérprete y su tiempo. Esto es, ninguna de las dos lecturas ficticias propuestas aspira a desentrañar la realidad estética del texto, sino imponerle los prejuicios personales del crítico: el texto no es un fin en sí mismo, es un medio[30].

Estas dos lecturas espurias, sin embargo, serían igual de válidas que una tercera que encuentre en los versos de Keats una evocación bucólica del otoño en el Hampshire rural basándose en la evidencia textual y biográfica. En este sentido, no podemos más que reconocer que Gottschall pone el dedo en la llaga cuando señala que tanto la primacía del compromiso ideológico sobre el epistémico como la renuncia a la demostración empírica comprometen la credibilidad de las disciplinas humanísticas en su conjunto por acción u omisión. Tomamos prestadas las palabras de García Barrientos (2020, p. 23): «las ciencias humanas, en general, y la teoría literaria, en particular, deberían compartir con las ciencias propiamente dichas, por lo menos, los principios de racionalidad y honradez intelectual».

Es indiscutible que existen lecturas rigurosas desde el compromiso ideológico, en las que el texto tiene prevalencia sobre la imposición de la doctrina subyacente al método. Esto es, aquellas que interpretan la evidencia estética presente en la obra literaria a la luz de unas ideas, en lugar de embutir estas en la obra literaria para juzgarla, incluso a despecho de la evidencia. De esta manera, encontramos crítica literaria abiertamente política —valgan como ejemplo Benjamin o Lukács— y, sin embargo, impecable si basa su razonamiento en los valores estéticos del texto, que interpreta observando unas premisas ideológicas particulares sin imponérselas.

[30]Resulta irónico que el tecnicismo usado sea precisamente *interrogar* porque, en buena parte de los casos donde se usa, bien podría añadírsele «bajo tortura» sin forzar un ápice el significado, tal es la crudeza con la que se retuerce el sentido para, diríamos, extraer las confesiones.

La situación se complica cuando el marco teórico niega el conocimiento objetivo, como es el caso de aquellas escuelas que adoptan el escepticismo radical. Russell, con muy buen tino, comenta que este,

> while logically impeccable, is psychologically impossible, and there is an element of frivolous insincerity in any philosophy which pretends to accept it. Moreover, if scepticism is to be theoretically defensible it must reject all inferences from what is experienced; a partial scepticism, such as the denial of physical events experienced by no one, or a solipsism which allows events in my future or in my unremembered past, has no logical justification, since it must admit principles of inference which lead to beliefs that it rejects. (B. Russell, 1948/1976, p. 9)

Huelga decir que la frívola falta de sinceridad que menciona el filósofo inglés es, precisamente, lo que lleva a las escuelas escépticas radicales a dudar de todos los presupuestos y marcos teóricos salvo uno en concreto: el suyo propio. En efecto, el escepticismo convenientemente selectivo de una escuela se traduce en un sistema autorreferencial, pues cuestiona todo fuera de él. Por lo tanto, es invulnerable a la refutación externa. En tanto que autojustificatorio, tampoco lo es desde dentro. El sistema es válido porque el sistema dice que es válido.

Por otro lado, tampoco faltan sistemas no necesariamente autorreferenciales que, sin embargo, plantean argumentos que no admiten refutación, por lo que caerían en otro tipo de trampa epistemológica. No obstante, su proceso de razonamiento resulta coherente dentro de su marco porque parten de evidencia para llegar a las conclusiones, válidas o no. Popper (1963/1989, p. 35) lo ejemplifica con dos casos, uno de un hombre que empuja a un niño al agua para ahogarlo y otro que sacrifica su vida intentando salvarlo. Ambos comportamientos tienen explicaciones análogas, tanto en términos freudianos como adlerianos. El primer caso sería para Freud un complejo de Edipo reprimido sublimado; el segundo, diría Adler, la necesidad de probar la valía por un complejo de inferioridad. Las observaciones de ambos psicoanalistas son correctas y sus explicaciones perfectamente plausibles dentro de la lógica interna de sendos sistemas.

Este tipo de razonamiento no acepta refutación porque ello exigiría una imposibilidad lógica: demostrar que los hombres de los ejemplos no padecían complejos. No abogamos con esto por descartar el psicoanálisis, como tampoco rechazamos métodos hermenéuticos por pecar, a su manera, de lo mismo que los dos psicoanalistas austriacos. Es innegable que las observaciones de Freud son agudas y certeras sin que

la explicación que les atribuye a resultas de su marco teórico cambie eso una jota, y las plasma en una prosa sublime que, por sí misma, ya le habría ganado los laureles. Es más cuestionable empero la superioridad de un método hermenéutico especulativo como explicación última del texto.

Así pues, creemos que acierta Escobar Varela (2021, pp. 35-40) al optar por el realismo crítico como tercera vía. Arguye que la irreconciabilidad de ambas posturas solo tiene sentido en una epistemología de suma cero, esto es, si una forma de entender la realidad excluye a la segunda. Este no es el caso porque, como bien indica Cameron, la literatura admite infinidad de lecturas que se suman una a otra y añaden matices adicionales a cómo se entiende el texto.

Los nuevos métodos cuantitativos no son una excepción. Una vez aceptado que en la obra de arte confluyen aspectos mensurables e inconmensurables, tiene poco sentido limitar las vías de estudio, renunciar a medir aquello que lo admite o despreciar cuanto sea incuantificable. Por el contrario, se obtiene una representación más compleja del objeto combinando ambas aproximaciones. No se trata de reemplazar un método de análisis literario por otro, sino de ir sumando nuevas perspectivas que proporcionen una imagen polifacética del texto, de combinar diferentes puntos de vista para ofrecer un reflejo más preciso y detallado, «instead of saying that the humanities are besieged and giving up ground, we could truthfully say that these disciplines are discovering new missions and new ways to understand culture» (Underwood, 2019).

Por lo tanto, adoptaremos una postura escéptica en el sentido popperiano (Popper, 1935/2002), no hacia la ontología, sino hacia la epistemología. Aceptamos la existencia de una realidad externa objetiva, pero solo concedemos validez a la teoría sobre esta como una explicación incompleta y provisional. Incompleta, en tanto que se limita a describir aquellos aspectos que conocemos hasta el momento. Provisional, en cuanto que puede ser reemplazada si se encuentra una justificación más adecuada a los mismos fenómenos, como el modelo celeste copernicano sustituyó al ptolomeico.

No renunciaremos, empero, a la subjetividad, ya que, como dijimos, la excepcionalidad de la literatura —y, por extensión, del arte— radica en que buena parte del hecho artístico escapa a la cuantificación. Únicamente podemos aproximarnos a este lado inconmensurable acumulando apreciaciones subjetivas que muestren diferentes facetas del objeto de nuestra atención. Recordemos la historia de *Rashomon*, en la que un bandido, la esposa de un samurái asesinado, un leñador y el espíritu del difunto ofrecen cuatro versiones contradictorias del homicidio. Desconocemos la realidad objetiva y no llegaremos nunca a conocerla. Son ejecuciones pictóricas de un mismo motivo desde ángulos variados, cada lienzo pintado con mayor o menor destreza, más

preciso en unos detalles de la naturaleza que imita y menos a otros, unos con querencia hiperrealista y otros abstractos. Ninguna obra es completamente fiel al modelo y no hay dos obras iguales, ni siquiera aquellas compuestas sobre el mismo caballete. Sin embargo, la suma de todos ellos nos acerca al modelo real, incluso más allá su aspecto externo. En la película de Kuroshawa, este modelo natural son los hechos a esclarecer, en la crítica literaria, es la estética del texto.

Nuestro propósito es, por lo tanto, facilitar la recopilación de pruebas forenses que complementen las declaraciones de los testigos del crimen de *Rashomon*, de dotar al delineante de papel milimetrado y plumilla para hacer una proyección diédrica ortogonal del motivo que inspira a los pintores. Se trata, en definitiva, de añadir otro modelo —imperfecto e incompleto, como el resto— que represente otros aspectos diferentes del texto literario.

2.7 Lectura distante

Este trabajo pretende dotar a la filología digital de una herramienta de extracción de información que pueda ser usada en estudios tanto asistidos como guiados por datos. Por una parte, ha de facilitar las observaciones en el texto y sus metadatos según los criterios que el investigador estime relevantes y refinar sus pesquisas mediante la combinación de esos principios. Esto es, los datos deben servir como un archivo donde buscar y clasificar la información de la mejor manera posible. Por otra parte, los datos producidos han de poder tratarse digitalmente para realizar análisis digitales masivos y extraer las conclusiones de los resultados. Dicho de otro modo, debemos proporcionar el material con el que se lleven a cabo estudios guiados por datos. En el ámbito de la filología digital hay que hablar entonces de la *lectura distante*[31].

El filólogo italiano Franco Moretti acuñó el término como respuesta a la *close reading*, como la de la escuela del Nuevo Criticismo. Arguye Moretti (2000a, p. 57) que esta última está limitada por la necesidad de circunscribirse a un corpus extremadamente reducido. Dedicar tanto tiempo a unos pocos textos implica la aceptación tácita de que solo un puñado de obras importa realmente. Añade que esta lectura tradicional es prácticamente un ejercicio teológico: el tratamiento solemne de unos pocos textos elegidos tomados muy seriamente. Contra esto, propone Moretti un «pacto con el diablo»: si sabemos cómo leer los textos, intentemos aprender cómo no hacerlo: la lectura distante,

[31] *Distant reading* en inglés.

> where distant [...] *is a condition of knowledge*: it allows you to focus on units that are much smaller or much larger than the text: devices, themes, tropes—or genres and systems. And if, between the very small and the very large, the text itself disappears, well, it is one of those cases when one can justifiably say less is more. If we want to understand the system in its entirety, we must accept losing something. We always pay a price for theoretical knowledge: reality is infinitely rich; concepts are abstract, are poor. But it's precisely this 'poverty' that makes it possible to handle them, and therefore to know. This is why less is actually more. (Moretti, 2000a, pp. 57-58).

En otras palabras, la diferencia conceptual entre la lectura distante y la lectura atenta no procede tanto de la extensión del corpus como de la escala del análisis y las observaciones que permiten hacerla. Para nuestro propósito, esto se traduce en el escrutinio de elementos mínimos, como el ritmo del verso, para encontrar patrones macroscópicos en sus combinaciones (Kroll y Sanz-Lázaro, 2023, p. 343, n. 2).

Moretti (2017) recurre a la máxima de Schleiermacher «entender al autor mejor que él mismo» para definir crítica textual, única justificación de la existencia de los críticos, y reconoce la genialidad del alemán por hallar el origen de tal facultad en la ignorancia del crítico. Este, desconocedor del discurso que ha llevado al autor hasta el texto, se ve forzado a considerar una multitud de detalles de los que el autor no sería consciente. El obstáculo para la lectura distante que ve Moretti en Schleiermacher es que, para el último, el objeto de la crítica es comprender el texto en su individualidad conforme a su intención, mientras que un corpus carece tanto de individualidad como de intención.

Complementaremos a Moretti con un comentario de Friedrich Schlegel (1804-1806/ 1957, p. 107, traducción propia) en términos sorprendentemente similares[32]: «La crítica no es realmente otra cosa que la comparación del espíritu y las letras de una obra, lo cual se trata como infinito, como absoluto e individuo. Crítica significa comprender un autor mejor de lo que él mismo se ha comprendido»[33].

Además de lo postulado por Schleiermacher, Schlegel aporta un nuevo factor que resulta muy conveniente para nuestra empresa, a saber: la crítica como comparación

[32] No entraremos aquí en si la hermenéutica de Schleiermacher bebió del filólogo, pero animamos al lector a profundizar en un interesantísimo caso de transtextualidad. Ver, por ejemplo Patsch, 1966, p. 436 nn. 13-15.

[33] «Kritik ist eigentlich nichts als Vergleichung des Geistes und des Buchstabens eines Werks, welches als *Unendliches*, als Absolutum und Individuum behandelt wird – *Kritisieren* heißt einen Autor besser verstehn als er sich selbst verstanden hat».

entre la forma externa y el significado profundo del texto. Nos tomaremos aquí la licencia de forzar el sentido de manera anacrónica[34] para llevárnoslo a nuestro terreno; adoptaremos en su literalidad la afirmación del filósofo. No hace falta decir que, de esta manera, nos encontramos precisamente con el objeto de los estudios de lectura distante.

Si retomamos la conocida analogía del pudin con la que Wimsatt y Beardsley (1954, pp. 4-5) ilustraban la falacia intencional, los métodos digitales no proporcionan una crítica gastronómica del plato, sino la receta o, mejor dicho, las recetas exactas de cada uno de las comidas que se han servido en un restaurante. Hallamos así una fórmula para expresar la distinción entre el pudin de un determinado local y los equivalentes de la competencia. Evidentemente, esto no proporcionará la experiencia sensual que se obtiene mediante la degustación, pero sí puede contribuir a explicar las sensaciones placenteras que se experimentan al saborearlo. Este método, claro está, solo reconoce los ingredientes que se encuentran en el plato; nada dice de los sentimientos del cocinero o cómo quería este que supiera su creación; afortunadamente, todos juzgamos un pudin —al menos en privado— sin otro criterio que nuestro paladar. Lo que tampoco tendremos es una descripción precisa que permita reproducir el toque especial de los grandes chefs, esa combinación de saber apreciar los detalles y obrar en consecuencia, la variable oculta que rompe el determinismo, aquello que marca la diferencia entre el prosaico condumio y la obra de arte gastronómico, entre el escritor y el literato.

La crítica digital, apunta Moretti (2017, pp. 6-7), está reemplazando significados con patrones, pero también interpreta estos a partir de palabras. La ausencia de contradicción radica en que el término *significado*[35] se interpreta de modos diferentes, y Moretti lo explica valiéndose de los términos alemanes empleados por Schleiermacher *sentido* y *significado*[36]. Al final, se trata del significado del elemento en contexto o aislado. Ahí es donde entra en escena la interpretación: encontrar el sentido del texto más allá del significado de sus componentes individuales. Para ello, es imprescindible considerar también aquello que el autor conoce y no explicita.

Moretti (2003, p. 67) ilustra la lectura distante por analogía con la historiografía antigua, que solo tenía ojos para los eventos extraordinarios y desdeñaba lo consuetudinario. El cambio de paradigma que propició la Escuela de los Annales llevó a considerar la *gran masa de hechos*. Se trata, pues, de poner el foco literario sobre esta y no en la singularidad de las grandes obras canónicas. Pone como ejemplo la

[34]Anacrónica solo en cierto modo. No olvidemos que la exégesis numerológica es anterior a Schlegel, como, por ejemplo, la practicada por ciertas escuelas cabalísticas.

[35]En el original *meaning*.

[36]En el alemán original de Schleiermacher *Sinn* y *Bedeutung*, respectivamente. Moretti los traduce al italiano como *senso* y *significato* y al inglés como *significance* y *meaning*.

literatura británica del siglo XIX, de la cual, un mínimo corpus de apenas unas doscientas novelas sería ya generoso, incluso notablemente mayor de lo que hoy podríamos considerar como el canon de la época. A pesar de todo, esto no representa ni una centésima parte de las obras que llegaron a publicarse en aquel periodo. Para leerlas todas, necesitaríamos leer una novela cada día durante más de un siglo. Sin embargo, pretendemos entender toda esta producción con un puñado de casos individuales que, para más inri, han entrado en el canon precisamente por su excepcionalidad. Concluye diciendo que no estamos hablando de la suma de casos individuales sino de un complejo sistema colectivo que debe intentarse aprehender como tal.

En definitiva,

> you reduce the text to a few elements, and abstract them, and construct a new, artificial object. A model. And at this point you start working at a 'secondary' level, removed from the text: a map, after all, is always a look from afar—or is useless, like Borges's map of the empire. Distant reading, I have called this work elsewhere; where distance is however not an obstacle, but a specific form of knowledge: fewer elements, hence a sharper sense of their overall interconnection. Shapes, relations, structures. Patterns. (Moretti, 2004, p. 94)

La lectura distante sigue dependiendo de la atenta para obtener esos pocos elementos necesarios para construir el modelo, si bien esta se delega. La lectura distante por sí misma ofrece una solución teórica a un problema que, a efectos prácticos, solo sirve para terminar trasladándolo. Pensemos en la literatura como un vasto campo en el que el canon constituye una porción mínima del total, insuficiente para describirlo; explicar el hecho literario mediante su canon sería como tratar de reducir una extensa pradera de pastos a unas pocas flores de brillantes colores que se destacan entre la hierba. La lectura distante propone tomar elementos mínimos de las caracterizaciones hechas por terceros: de las flores vistosas, pero también de las menos; plantas, hierbas, piedras, charcos... incluso —si se me permite— de evidencias *post-hoc* de presencia bovina. Con todo, esto no elimina el sesgo de la crítica literaria. Buscaríamos los componentes mínimos de lo que hay en el prado de nuestra analogía en el catálogo de una floristería; esto es, nos limitaríamos a unas pocas flores, ignorando las menos vistosas así como todo lo demás, lo que Margaret Cohen (1999, p. 23) denominó «the great unread». Resultaría poco factible repartir el trabajo de forma satisfactoria: no solo requeriría coordinar un exorbitante número de personas, sino que, además, el reparto jamás sería justo, en tanto que apenas hay una reducida cantidad de obras

maestras en comparación con la inmensidad de otras, no solo menores y mediocres, sino también insufribles.

Debemos, pues, allanar el camino a la lectura distante, de manera que, aprovechando que nos concentramos en aspectos simples concretos, sea la máquina la que se encargue de escudriñar los textos en pos de ellos. Evitemos llevarnos a engaño: la automatización no elimina por completo la necesidad de cierta intervención manual en el análisis porque los textos han de digitalizarse en primer lugar y prepararse para que el computador sea capaz de interpretarlos; no obstante, incluso teniendo esto en cuenta, los recursos temporales y humanos se reducirían varios órdenes de magnitud. Lo que nos proponemos no es otra cosa que encargar al computador la búsqueda de los elementos mínimos en piezas teatrales áureas, como Moretti (2000b) puso a sus alumnos a hacer lo propio en novelas policíacas. Claro está que, además, lo hacemos con la intención de multiplicar el número de obras analizadas mientras reducimos el tiempo a una fracción; de lo contrario, nuestro propósito práctico carecería de sentido.

Prodigios verán los hombres
en tres actos, y ninguno
a su representación
faltará por mi descuido.
Calderón, *El gran teatro del mundo*

Drama 3

3.1 Literatura escenificada

El teatro es ilusión. Personas de carne y hueso *interpretan* papeles y, al hacerlo, encarnan —en el sentido literal de la palabra— personajes que no existían más que sobre el papel hasta que se abre el telón y se crea el mundo ilusorio del teatro, cuya efímera tangibilidad se esfuma en la nada en el mismo momento que cae por última vez la cortina. El teatro, según Aichinger (2013, pp. 99-100), posee un carácter mágico que propicia una transformación vital de intensidad extraordinaria provocada por la literatura. Nos hace experimentar una versión de la vida aumentada, hasta el punto que olvidamos estar en un patio de butacas y, durante el tiempo que dura la representación, aceptamos como auténtico cuanto ocurre sobre las tablas. Olvidamos que estamos ante una realidad escenificada; no solo eso, sino que *deseamos* olvidarlo; lo hacemos a sabiendas y de buena gana. El teatro no es un engaño al uso porque el espectador es consciente en todo momento de que la *realidad* que se le está ofreciendo deja de serlo al terminar la función; incluso el montaje más realista rebosa artificio[37]. No obstante, el espectador está más que dispuesto a dejar de lado su escepticismo, acepta contemplar esa nueva realidad en sus propios términos. Esta suspensión de la incredulidad es necesario para alcanzar la catarsis aristotélica y es, por lo tanto, un elemento esencial del teatro. No obstante, dejemos aquí la sala de la recepción literaria para, valga la metáfora, echar un vistazo al género dramático desde el foro.

Una característica definitoria del drama, aquella que lo diferencia de otros géneros literarios mayores, ya la señaló Aristóteles en su *Poética* al conceder que, de la tragedia —podríamos decir que por extensión también de otras formas dramáticas—, «el más

[37]En el teatro del Siglo de Oro, los costes de la puesta en escena de autos sacramentales se disparó entre 1601 y 1631 a causa de una espectacularización del evento (Ehrlicher, 2008, p. 201).

importante de estos elementos es la estructuración de los hechos; porque la tragedia es imitación, no de personas, sino de una acción y de una vida, y la felicidad y la infelicidad están en la acción, y el fin es una acción, no una cualidad» (Arist. *Poet.*, 1450a15-18). No solo es crucial, nos atrevemos a decir, sino que esta estructura de los hechos constituye el drama mismo, pues su esencia última es la acción de los personajes. Al contrario que la narrativa y, aún más, la lírica, donde las descripciones, pensamientos o reflexiones reclaman para sí una función articuladora en la estructura interna de la obra, el drama se fundamenta en acciones.

Tal vez lo exponga Platón con más claridad cuando dice que la tragedia surge al retirar las palabras del poeta de la composición narrada, para dejar sin otro acompañamiento las intervenciones de los personajes que en ella toman parte (Plat. *Rep.*, 393a-394c), esto es, los parlamentos. Dicho de otra manera, la fuerza motriz de la historia está en los diálogos de los personajes y no en la voz narrativa explícita del poeta. No quiere decir esto que la voz del narrador desaparezca por completo. Recordemos, por ejemplo, el comienzo de la película *Bande à part*. El actor Sami Frey va conduciendo un descapotable por una carretera a las afueras de París. A su lado está sentado Claude Brasseur. Ambos conversan sobre un asunto que se traen entre manos, posiblemente un negocio algo turbio que entraña cierto riesgo. Ven a una chica conocida sobre una bicicleta. En un momento de silencio, se oye una voz superpuesta que dice algo como: «Mi historia comienza aquí, dos semanas después de conocer a Odile, Franz lleva en su coche a Arthur para enseñarle la casa». Tenemos una voz extradiegética que narra, en efecto, pero, su intervención no determina que estemos viendo a Franz llevando a Arthur a la casa. Ambos reconocen a Odile antes de la intervención del narrador. La película ha avanzado sin necesidad de este y sigue avanzando con independencia de su participación; el narrador no puede sino limitarse a glosar la acción que están llevando a cabo los personajes. En la obra narrativa, por el contrario, la voz narratorial tiene la facultad de establecer los hechos[38] sin necesidad de la intervención directa de los personajes.

El poeta —el buen poeta— se retira además estilísticamente, porque, si se aviene a las exigencias del decoro, ha de hacer hablar a sus personajes en observancia de lo que su naturaleza ficcional demanda:

> La avenencia del estilo y el carácter o sujeto interno se conoce como decoro o adecuación del estilo al contenido. El decoro es, en general, la *voz ética* del poeta, la adaptación de su propia voz a la voz del personaje o al tono

[38]No entraremos en si debemos fiarnos del narrador para no complicar innecesariamente el ejemplo. Sobre esto en el Siglo de Oro, ver Ehrlicher (2021).

> vocal que demanda el sujeto o el ambiente. Y, al igual que el estilo está en su más pura forma en la prosa, está el decoro en el drama en su más pura forma, donde el poeta no aparece en persona. El drama se describe, desde nuestro punto de vista actual, como *epos* o ficción absorbido por el decoro. (Frye, 1957/1971, pp. 268-269, traducción propia)

Esta predominancia de las acciones de los personajes tiene una implicación adicional: la historia avanza más o menos en tiempo real a medida que se suceden las intervenciones. Al contrario que en una obra narrativa, en la que el tiempo se ralentiza o acelera a voluntad, la obra teatral requiere sincronía. Podemos detener el tiempo mientras Segismundo declama el celebérrimo soliloquio de *La vida es sueño*, ¿pero qué ocurriría si un hipotético personaje entrara a interrumpir las disquisiciones metafísicas del prisionero? La mínima interacción nos hubiera obligado a poner a correr nuevamente el reloj. En este sentido, tal vez sean los soliloquios los parlamentos menos dramáticos, los más cercanos a la lírica, en tanto que expresan la interioridad subjetiva de un narrador, «la pronunciación monológica de un *yo*» (Kayser, 1948/1992, p. 191, traducción propia, énfasis añadido). Una buena porción de los apartes se interpreta de una forma similar, como una breve interrupción del curso de la historia. Sin embargo, en cualquier caso, no son estos recursos definitorios del género, ya que este acepta la creación sin recurrir a ellos. Por lo tanto, hemos de entenderlos más como una contingencia que como necesidad.

Tampoco se le escapó esto a Hegel, quien, más de dos milenios después, afirma que «la necesidad del drama es, en efecto, representar las acciones y relaciones humanas en el tiempo presente para la conciencia representativa y, por consiguiente, manifestaciones lingüísticas de personas que representan la acción» (1835/2016, 3, C, III, I, a, traducción propia). Aquí tenemos un matiz fundamental, la *manifestación lingüística*[39] como acción. En efecto, en las obras teatrales aparecen acotaciones más o menos detalladas, aunque, incluso el dramaturgo más minucioso deja en gran medida la proxémica a un elemento mediador entre el texto y el espectador, normalmente el actor[40]. Sin embargo, en condiciones normales, la esencia textual de la acción del personaje llega al receptor sin haber sido alterada, pues el actor transmite el texto, si bien añadiendo matices propios. En otras palabras, el dramaturgo proporciona el armazón lingüístico sobre el que el actor construye la edificación interpretativa, donde emplaza las nuevas piezas sin modificar el andamiaje. Se trata sin duda de un ejercicio de transmedialidad (Türschmann, 2013, pp. 181-182), en el que el texto se

[39] *Sprachliche Äußerung* en el original.

[40] Este elemento puede ser un sistema complejo, por ejemplo, el que forman el director de escena y el actor.

combina con la de la actuación en un flujo intermedial del que surge la totalidad del hecho dramático.

No importa si presenciamos la representación de una tragedia de Eurípides o de una novísima pieza experimental, la naturaleza del texto como un discurso en tiempo real mediado por el actor no varía en lo concerniente al espectador. Esta y no otra es la naturaleza de la obra teatral. Así debemos abordarla si aspiramos a analizarla en sus propios términos.

3.2 Aproximaciones al drama

El drama tiene una naturaleza dual, ya que se observa desde una perspectiva escénica o una textual, sin perjuicio de que el texto sea parte constituyente de la puesta en escena. La disputa se remonta, al menos, a la polémica que sostuvieron los miembros del Círculo de Praga, Zich y Veltrusky. El primero defendía una tesis *escenocéntrica*, que la obra dramática solo existía en tanto que se representa, mientras que el segundo contraargumentaba desde una posición *textocéntrica*, argüía que el texto es una obra artística completa en sí misma que, además, determina el resultado de la representación (Schaeffer, 1995, pp. 612-613). Esta diferencia ya estaba presente en el Siglo de Oro y poetas de la talla de Calderón así lo hicieron constar (Ehrlicher, 2023, p. 122).

Schaeffer (1995, p. 614) señala que el teatro se presta al estudio desde una perspectiva antropológica, semiótica, de análisis del discurso y mimética. El primer enfoque retoma la hipótesis aristotélica del origen religioso del drama. El problema que presenta esta aproximación es la falta de evidencia documental o arqueológica que la avale, por lo que hoy se tiende a considerar la representación ritual más como una de las múltiples formas que adopta el teatro que como su germen de las artes escénicas.

La aproximación semiótica (Schaeffer, 1995, pp. 615-616) toma el teatro como un polisistema compuesto de un conjunto de sistemas de signos visuales, verbales y sonoros. Este planteamiento presenta la dificultad de tener que segmentar el sistema en unidades de significado mínimas para establecer un *código*. El teatro occidental —al contrario que, por ejemplo, el *nô* japonés—, carece de un sistema de unidades proxémicas aisladas con significado. Resulta llamativo que otro aspecto aparentemente asequible a la segmentación, las interacciones entre diálogo y música, se haya desarrollado tan poco —tal vez por el carácter interdisciplinar del fenómeno, que requeriría aunar la crítica teatral y la musicología—. Entendido como texto, la obra teatral escapa a los principios del análisis narratológico, pues no cuenta una historia, sino

que es una progresión dinámica de interacciones lingüísticas. En este sentido, resulta mucho más adecuada la segmentación en unidades deícticas referidas a los actantes que el análisis narratológico.

Con todo, el aspecto lingüístico está en el mismo centro del teatro, textual o escenificado. Este lenguaje, además, no es el natural, sino su representación artística: está guiado por una aspiración estilística y responde a las necesidades comunicativas teatrales y no a las de los actos de comunicación corrientes (Schaeffer, 1995, p. 617). Hablamos, pues, de una *doble enunciación*[41] en los actos de habla teatrales, ya que los emite el personaje y están dirigidos a su interlocutor en escena, pero, por otra parte, son también el discurso autorial dirigido al espectador.

> Todo discurso en el teatro tiene dos emisores en la enunciación, el personaje y el escritor (como tiene dos receptores, otro personaje y el público). Esta ley del doble emisor de la enunciación es un elemento capital del texto teatral: es aquí donde se encuentra la falla inevitable que separa al personaje de su discurso y le impide constituirse como verdadero emisor de sus palabras. Cada vez que habla un personaje, no lo hace en solitario, y el poeta habla al mismo tiempo por su boca; de aquí el dialogismo constitutivo del texto teatral. (Ubersfeld, 1977/1996, p. 105, traducción propia)

Al contrario que, por ejemplo, una homilía —teatralizada y sujeta a códigos escénicos convencionales (Ebenhoch, 2015, pp. 318-319)—, el teatro es ajeno a la transmisión inmediata. Si el dramaturgo interpreta sus propias líneas, actúa, representa necesariamente el papel de un personaje, mientras que el predicador no encuentra impedimento para hablar con su propia voz.

El análisis de la pieza teatral como un texto destinado a la representación propone entenderlo según cómo actúa en la audiencia mediante su escenificación. Los diálogos no se abordan desde la crítica literaria ni el análisis del discurso de una conversación no ficcional, sino como medio dramatúrgico. Se considera la tensión de un texto que no únicamente debe ser recitado sino representado dentro del contexto dramático.

Finalmente, considerando la función mimética del texto, esto es, desde la poética clásica, se observan dos aspectos principales: el estudio del conflicto y el del discurso dramático. En cuanto a lo primero, los personajes se distinguen de sus funciones dramáticas. Consideremos, por ejemplo, los movimientos de los *actantes*, cada uno de los cuales posee un dominio constituido sintácticamente por los movimientos que le

[41] *Double énonciation* en francés.

pertenecen y semánticamente por las máximas de acción pertinentes en su dominio sintáctico (Schaeffer, 1995, pp. 618-620). El segundo aspecto lo propone García Barrientos, quien lo denomina *dramatología*, inspirado por la *narratología* de Genette; lo desarrollaremos en la siguiente sección.

3.3 Dramatología

El concepto de *dramatología* es una propuesta de García Barrientos como alternativa a lo que él denomina pensamiento «débil», como una reacción a la terminología deliberadamente vaga que conduce a una lógica deficiente cuando esta se emplea para enlazar proposiciones (García Barrientos, 2020, p. 40). Trataremos de condensar aún más en esta sección lo que García Barrientos ya sintetiza en *Anatomía del drama*, bajo el pertinente subtítulo *Una teoría fuerte del teatro*.

El primer presupuesto de la dramatología tiene raíces aristotélicas, pues toma prestados los dos modos de mímesis descritos en la *Poética*:

> En efecto, con los mismos medios es posible imitar las mismas cosas unas veces narrándolas (ya convirtiéndose hasta cierto punto en otro, como hace Homero, ya como uno mismo y sin cambiar), o bien presentando a todos los imitados como operantes y actuantes. (Arist. *Poet.*, 1448a19-24)

Sin embargo, García Barrientos matiza que, cuando Aristóteles describió los modos miméticos, no había forma posible de conocer que, un par de milenios más tarde, de manera simultánea, unos hermanos de Besançon, por un lado, y un inventor estadounidense por otro, construirían sendos ingenios para proyectar imágenes en movimiento. El *medio* de imitación y no el *modo* es lo que crea la analogía del cine con el teatro y lo distingue de la narración. En efecto, si consideramos el modo, el cine comparte con la narración su carácter mediado, no por la voz del narrador sino por la cámara. Ambos elementos se interponen entre el receptor y la ficción, mientras que, en el teatro, ese elemento intermedio está ausente, el acceso del receptor al mundo de ficción es directo. Se trata, pues, de una representación más objetiva, en tanto que no interviene la subjetividad del elemento mediador.

En lo concerniente a la forma en el análisis automático de textos teatrales, debemos tener muy en cuenta que la *inmediatez* dramática afecta de igual manera al texto escrito que a la representación escénica, a pesar de que, por razones obvias, esta sea menos apreciable en el primero. Sin embargo, esta inmediatez modal encuentra un

reflejo estructural externo evidente en el texto, en la superposición de dos subtextos diferenciados, uno descriptivo y otro reproductivo:

> Llama sobre todo la atención que en un drama «escrito» discurran dos textos uno junto al otro: por un lado, el texto secundario, esto es, la información sobre dónde acontece la historia representada en cuestión, en qué tiempo, etc., quién está hablando y, quizás también, qué está haciendo, etc.; por otro lado, el texto principal[42] mismo. Este último consta exclusivamente de frases, que son *pronunciadas* «realmente» por los personajes representados. Mediante la especificación del personaje correspondiente, las frases del texto principal adquieren una suerte de «comillas». Tanto estas frases mismas como los personajes que se indica que están hablando en cada momento, así como las circunstancias mismas, devienen en lo *representado* a través de los elementos del texto secundario, pertenecen al «nivel concreto» de este. Pero las frases pertenecientes a este nivel son justamente frases y, por lo tanto, constituyen un nuevo nivel, a saber: el de las circunstancias y los destinos de aquellos de los que trata el discurso en las frases pronunciadas. (Ingarden, 1931/1972, p. 220, traducción propia, énfasis en el el original)

Para García Barrientos, los términos abstractos *texto principal* y *secundario* encuentran su forma concreta en castellano en *diálogo* y *acotación*. Convenimos con que puede hacerse esta equivalencia *grosso modo*, aunque, en nuestro caso, trataremos de hilar más fino. La manera en que necesitamos estructurar el texto, el personaje y las instrucciones escénicas, incluso siendo ambas indicaciones del dramaturgo, corresponden a dos entidades diferenciadas: la primera es metainformación para la clasificación de los diálogos en el texto completo, ya que es un atributo de estos, mientras que la segunda no. La acotación no afecta siempre a un solo nivel: se refiere a un parlamento, además de la evidente didascalia de personaje, si aquel, por ejemplo, se introduce con la acotación «aparte» o solo a una porción de este si llega una vez iniciado el parlamento, por lo que afecta únicamente esa parte del parlamento o el modo de escenificarlo. La acotación puede afectar a toda la escena (p. ej., «Éntranse todos»), al acto («Tocan chirimías y corrése la cortina. Aparecen...») o a la obra completa («Fin»), o incluso una parte no dialogada («Desenvaina»).

Esto, como vimos en el capítulo anterior, obliga a tomar una decisión ontológica. Por ejemplo, podríamos considerar los parlamentos no según quién los pronuncia sino si

[42]En el original se emplea *Haupttext* para el texto principal y *Nebentext* para el secundario.

indican apartes, líneas cantadas o leídas. Tal clasificación implicaría una organización diferente a la hecha —por personaje— y determinaría los tipos de análisis que podrían llevarse a cabo con tales datos. No obstante, seguimos topándonos con que las acotaciones constituyen una clase abierta, lo que obligaría a considerar apenas un subconjunto mínimo de ellas, las que indiquen una cualidad del parlamento, y tratar al resto como una entidad diferente única. Si bien podría enfocarse así el texto desde otro punto de vista, nos devolvería a la situación original en lo que toca a los segmentos afectados por las acotaciones. Por este motivo, resultaría más productivo crear una estructura de datos que defina una entidad textual cerrada que adopte valores de dos categorías mutuamente excluyentes, diálogo y acotación, cuya pertenencia la determina que el texto forme parte o no de un parlamento.

Aquí se da una cuestión que nos atañe, se trata de dilucidar quién pronuncia cada parte del texto. En cuanto a los diálogos, es evidente que corresponden a cada personaje. ¿Pero qué pasa con las acotaciones? Si, como dijimos, no resulta conveniente emplearlas para clasificar otras entidades en función suya, deberíamos ordenarlas como partes textuales en una categoría alternativa a los diálogos. Después de todo, en el siglo XVII forman ya una estructura (Monzó, 2020, p. 562). El autor ha retrocedido del todo, no es factible, como en la narratología, ubicar la presunta voz autorial en el mundo ficcional de la narración. Los apartes de la obra, sin entrar en dónde en concreto, no se pronuncian en el universo diegético. No se trata de una descripción de un suceso del que tiene conocimiento el narrador, sino de una prescripción de cómo ha de desarrollarse un evento aún por suceder en un mundo que es ajeno a la voz. Al contrario que en el drama, el universo del narrador épico es ficcional, se halla en el mundo diegético o en uno de orden superior que lo contiene, pero no por ello menos ficcional. En la representación, por el contrario, la voz autorial no existe y discurre paralela al texto dramático sin intersección alguna.

Esto sigue siendo cierto tanto si oímos una voz narradora extradiegética, como en el ejemplo de *Bande à part* como si se recita un prólogo como en las tragedias clásicas; en el primer caso, la voz corresponderá a un nuevo personaje sin cuerpo, VOZ tal vez, cuya mecánica en nada se diferencia de la del resto de actores, aunque no compartan espacio ficcional. En el segundo caso, estamos no solo fuera de la diégesis, sino del universo ficcional. La declamación acontece en el universo no-ficcional o en uno ficcional diferente al de la historia.

García Barrientos (2020, p. 49) sostiene la tesis de que las acotaciones no son de «nadie», como contrapunto a Ubersfeld (1977/1996, p. 18), quien atribuye este subtexto al dramaturgo. Arguye la incapacidad funcional, si bien materialmente posible, de enunciar el *yo* en la acotación, que distingue del paratexto alegando que aquella, al

contrario que el paratexto, sí tiene un autor. Encontramos evidencias tangibles de esto en los propios textos: Monzó (2021, pp. 93-94) habla, por ejemplo, de un caso de conciencia autorial en las acotaciones, en las que un escritor quiere hacerse oír, pero va reduciendo con cada nueva obra el ímpetu inicial por prescribir su criterio escénico. Si no es tan tajante como para hablar de *voz autorial*, sí reconoce la existencia de una *voz dramatúrgica* en las acotaciones.

Consideremos ahora las indicaciones del director escénico: estas son en el drama de una naturaleza semejante a las indicaciones del comediógrafo, en tanto que externas a la representación, aunque determinantes de lo que en ella sucede. La diferencia estriba en la fijación escrita de las acotaciones y la naturaleza efímera de las órdenes del director. A pesar de ello, ambas funcionan en el drama de la misma manera. El director de escena usará con toda probabilidad la segunda persona del imperativo para dirigir a los actores, o la tercera persona en subjuntivo cuando sea preciso que se traslade una orden. En cualquier caso, no interviene el *yo*, pues el director no está en escena, de la misma manera que tampoco lo está la voz de las acotaciones en el universo donde se desarrolla el diálogo.

Suavizaremos aquí la tesis. Las acotaciones no las pronuncia nadie, sí, pero nadie en el subconjunto del hecho teatral que es el universo representado, porque esa voz no existe allí. No obstante, resulta más difícil aceptar un *nadie* absoluto y sin matices, sea solo por no negarle la existencia al dramaturgo y al sufrido director teatral. Nos inclinaremos preventivamente, por el momento, por Ubersfeld y Monzó.

Señala García Barrientos (2020, pp. 51-56), asimismo, las diferencias semióticas elementales entre la obra escrita y la representada. Mientras que la primera se desarrolla en dos etapas diferenciadas, una de producción, en la que el *autor* produce la *obra*, y otra de consumo, en la que la *obra* es leída por el *lector*, la obra representada es un proceso único en el que el *actor* representa para el *público*, de manera que los procesos de creación y recepción constituyen una única experiencia intersubjetiva compartida por público y actor. Esto no supone únicamente la ausencia en el proceso del *autor*, sino de la obra literaria. Al contrario que el libro, que admite infinitas lecturas de un texto inmutable, la representación es única e irrepetible. Eso la emparenta con la música en vivo o el espectáculo circense, mientras que coloca al texto dramático junto al cine o la grabación musical, que pueden copiarse sin límite y verse, oírse o leerse tantas veces como se desee.

Sobre esto precisamente ya había reflexionado Walter Benjamin en su conocido ensayo sobre la reproductibilidad del arte en nuestro tiempo. Este, además, concreta

la obra en la máquina[43], lo que agrega al término matices semánticos relacionados con la complejidad de la mediación. Este mecanismo es el que, de forma metafórica, presentan las escenas de las obras narrativas, donde el autor fija su *cámara* discursiva en aquellos detalles que quiere resaltar y nos oculta otros que prefiere escamotearnos, o la música grabada, que es el resultado de una mezcla particular de la pista de cada instrumento.

> Así, la ejecución del actor está sometida a una serie de exámenes ópticos. Esta es la primera consecuencia de que la del actor se presente a través del equipo técnico. La segunda consecuencia radica en que el actor cinematográfico, al no presentar él mismo su ejecución al público, pierde la posibilidad reservada al intérprete escénico de adaptarla al auditorio durante la representación. Esto se traduce en un carácter de examinador, que no se altera en lo más mínimo por el contacto personal con el actor. *El público solo se identifica con el actor en tanto que lo hace con la máquina. Este asume así su carácter: examina.* (Benjamin, 1935/2013, pp. 37-38, traducción propia, énfasis en el original)

Como consecuencia, es imposible revertir el proceso de la producción teatral. La representación no admite reducción a forma escrita y tampoco acepta ser narrada, ni verbalmente ni de forma visual con una cámara. La esencia del teatro es la actuación: mientras que un encuentro deportivo puede celebrarse en ausencia de público sin detrimento de su naturaleza, el teatro lo es solo en la medida que existe la comunicación intersubjetiva entre el actor y el público. «El teatro necesita al público para ser; las demás actuaciones lo requieren para *ser actuaciones*» (García Barrientos, 2020, p. 59, énfasis en el original).

A pesar de todo, nuestro análisis toma el texto escrito como objeto. Debemos, por lo tanto, considerar la manifestación escrita a la luz de lo expuesto no como la obra, sino una descripción de esta. El texto es, pues, la guía inmutable a partir de la cual se construyen las obras teatrales, sin que ninguna de ellas sea exactamente igual a otra. Retomando lo que tratábamos en el capítulo anterior, el texto es un modelo de la representación y, como tal, parcial e incompleto. De esta manera, describiremos diferentes aspectos de la obra teatral según la parte que consideremos.

De acuerdo con los tipos propuestos por la dramatología, encontramos textos lingüísticos y no lingüísticos. Entre los primeros distinguiríamos orales, como una grabación

[43] *Apparat* en el original.

de los diálogos, y escritos, como las acotaciones. Entre los segundos, podríamos encontrar textos icónicos, como un *storyboard*, y simbólicos, como la partitura de la música. Se clasifican también según si son de referencia verbal, como los diálogos, o no verbal, como la escenografía, así como entre transcriptivos o reproductivos, como el diálogo (sea escrito o grabado), y descriptivos, como las acotaciones (García Barrientos, 2020, p. 64). Para nuestro estudio, nos interesan los textos lingüísticos escritos, de referencia verbal, tanto transcriptivos como descriptivos, como objeto y medio respectivamente.

El modelo dramatológico distingue tres categorías en los textos dramáticos:

- texto escénico
- texto dramático
- texto diegético

El *texto dramático* es un texto lingüístico de referencia verbal y no verbal, reproductivo y descriptivo, cuyo ejemplo por antonomasia es la propia obra dramática escrita; el *texto escénico* representa aspectos no dramáticos que han de tenerse en cuenta en la representación, como, por ejemplo aquellos que describen los libretos de luces o de tramoya; el *texto diegético* o *fábula* es el argumento de la obra textualizado (García Barrientos, 2020, p. 65). Debemos notar que las divisiones no son completamente estancas, ya que resulta habitual incluir en el texto dramático elementos del texto diegético o del texto escénico.

3.4 Estructura del drama

El libreto de una pieza teatral moderna suele presentar una distribución en actos, subdivididos en escenas. Es responsabilidad del director escénico la organización interna de estas divisiones externas. Esta forma de segmentación es —en cierto modo— moderna, pues su redescubrimiento tras el lapso medieval se lo debemos al humanismo renacentista y su interés por Séneca y otros comediógrafos clásicos. La disposición externa responde a una necesidad práctica: el director de escena ha de conocer los personajes que hay sobre el tablado en cada momento durante la representación (Kayser, 1948/1992, p. 170). ¿Qué pasa entonces, por ejemplo, con buena parte del teatro áureo en la que el escritor no señala las escenas? En ese caso, aparte de la organización interna, también las divisiones externas menores reacaerían sobre el director de escena.

Empecemos, pues, por definir la estructura del drama como obra literaria. En un primer nivel hacemos la clasificación tradicional entre *texto* y *paratexto*. En este último encontraríamos elementos como el prólogo y los elogios o, en las ediciones modernas, la introducción y notas. El *texto*, por su parte, lo constituirían los dos subtextos de la obra teatral, esto es, el *diálogo* y las *acotaciones*. El diálogo es el componente verbal y las acotaciones el no verbal. Entre estas últimas, distinguimos las acotaciones *espaciales*, que atañen al decorado; *temporales* para hacer indicaciones sobre el ritmo; *sonoras* para señalar las intervenciones musicales o ruidos dentro y *personales*. Estas últimas, a su vez, se dividen entre nominativas, paraverbales, corporales, psicológicas y operativas (García Barrientos, 2020, pp. 75-79).

Monzó (2019, pp. 75-76), por su parte, propone acotaciones de *loci scœnici*, que comprenden las salidas y entradas; *de caracterización del personaje*, con indicaciones de vestuario, dramáticas o de utilería; *escenotécnicas*, sobre tramoya y mobiliario; *de técnica actoral*, que guiarían la proxémica, kinésica, gestos y voz y, por último, acotaciones *de efectos especiales*, incluyendo efectos visuales y sonoros, luminotecnia, música y otros.

En nuestro trabajo, examinaremos con detenimiento los diálogos, pero nos valdremos de las acotaciones para obtener información dramática no verbal. Adelantamos que las sonoras nos ayudarán a determinar el metro, mientras que las personales nominativas y operativas esclarecerán la configuración escénica[44].

El drama se divide en segmentos mayores que constituyen los *actos*, que suele corresponder a una unidad espacio-temporal con los límites definidos tanto en el texto como en su representación. En los actos distinguimos instancias de «una acción escénica ininterrumpida que tiene lugar en un espacio y tiempo determinados» (Ruano de la Haza, 2007, p. 91), denominados *cuadros*. Estos se identifican atendiendo a diferentes marcadores. Algunos de ellos son escénicos, como la situación de que no queden personajes sobre el escenario, que cambie la ubicación o el tiempo de la acción, que se revele un nuevo decorado, o de otro tipo, como que ocurra un cambio estrófico (Ruano de la Haza, 1994, pp. 291-292). El primer tipo de cuadro, que implica un cambio en la configuración de los personajes, se ha denominado tradicionalmente *escena*.

Los actos están asociados a la ficción dramática, pero los consideraremos como componentes estructurales externos a efectos prácticos, aprovechando los elementos paratextuales que señalan sus límites. Esta demarcación explícita no se da necesariamente

[44]En febrero de 2023, Clara Monzó propuso en Graz una taxonomía más exhaustiva para las acotaciones del drama áureo, sobre lo que propone en su tesis doctoral. Convendría tener en cuenta esta u otra taxonomía igualmente descriptiva tanto para una hipotética extensión de este trabajo como para definir las etiquetas necesarias para trasladarlo a ediciones XML-TEI.

en todos los segmentos —véanse, por ejemplo, los cuadros y escenas sin marcar—, si bien es la norma para los de orden superior, por lo que haremos uso de ella y segmentaremos atendiendo al criterio del editor; en ausencia de marcas, la división está sujeta a elementos no siempre apreciables en la estructura propiamente textual, lo que requiere una lectura atenta e, incluso así, las divisiones no siempre están exentas de controversia. Nos centraremos, por lo tanto, en aquellos elementos visibles de la superficie del texto. Recordemos que este trabajo no aspira a resolver el problema la segmentación implícita, sino aportar datos y herramientas que contribuyan, entre otras cosas, a echar luz sobre él, a contribuir a los análisis con metainformación dramática y métrica de los versos. Tengamos en cuenta esta limitación del alcance no solo como prevención, sino como sugerencia para potenciales usos de sus resultados.

Acto

El acto se define como una división externa, mas solamente hasta cierto punto, dado que, en su encarnación clásica es consustancial a la historia. En efecto, este segmento como división dramática engloba acciones que suceden en una unidad espacial (Kayser, 1948/1992, p. 170). Este concepto de *unidad espacial* se ha querido respaldar a menudo apelando a Aristóteles u Horacio —al menos desde Trissino—, a pesar de que la evidencia textual de ello es cuestionable. Lo atribuye Corneille (1660/1862, p. 117) a una extrapolación espuria de la unidad tiempo a la distancia que se cubre en ella. En efecto, aconseja Lope que se componga el acto «procurando, si puede, en cada uno / no interrumpir el término de un día» (Vega Carpio, 1609/2006, vv. 213-114). Del paso de la unidad temporal a la espacial encontramos rastros al final de la primera parte del *Quijote*, en medio de las invectivas del cura y el canónigo contra las tendencias literarias de moda, donde el primero alude a las comedias y, entre otros defectos, les achaca la siguiente falta:

> ¿Qué diré, pues, de la observancia que guardan en los tiempos en que pueden o podían suceder las acciones que representan, sino que he visto comedias que la primera jornada comenzó en Europa, la segunda en Asia, la tercera se acabó en África, y así se hubiera hecho en todas las cuatro partes del mundo? (Cervantes Saavedra, 1605/2016a, p. 624)

Resulta irónico que el intransigente cura cervantino se refiera a los actos como *jornadas*[45], ya que esta innovación se define, precisamente, por la ruptura con las unidades

[45]Entenderemos *jornada* en lo sucesivo como un caso particular del acto. En cualquier caso, nos referimos a conceptos y no a sus denominaciones particulares, por lo que, por ejemplo, los veintiún

clásicas. Luis Alfonso de Carvallo ilustra esta característica de la siguiente manera en uno de sus diálogos:

> Jornada es nombre italiano; quiere decir cosa de un día, porque *jiorno* significa el día. Y tómase por la distinción y mudanza que se hace en la comedia de cosas sucedidas en diferentes tiempos y días, como si queriendo representar la vida de un santo, hiciésemos de la niñez una jornada, de la edad perfecta otra, y otra de la vejez. A estas jornadas llaman los latinos *actos* (Horatius in *Poetica*) y tiene cinco cada comedia. (Carvallo, 1602/1997, p. 261, énfasis en el original)

Fausta Antonucci (2017, p. 112) trae a colación a Juan de la Cueva, quien ofrece una razón de peso para este cambio: «Huimos la observancia que forzaba / a tratar tantas cosas diferentes / en término de un día que se daba» (en Sánchez y Porqueras Mayo, 1972, p. 144). En efecto, la jornada dota al drama español de una flexibilidad inusitada porque libera al poeta de la necesidad de embutir la acción en el corsé temporal prescriptivo y, por lo tanto, limitar la acción o poner en riesgo la verosimilitud.

Cuando el cura del *Quijote* vuelve a la carga, nótese que en esta nueva ocasión lo hace arguyendo, nada menos, aquello que usa Carvallo como elemento definitorio de la jornada: «Porque, ¿qué mayor disparate puede ser en el sujeto que tratamos que salir un niño en mantillas en la primera cena [*sic*] del primer acto, y en la segunda salir ya hecho hombre barbado» (Cervantes Saavedra, 1605/2016a, p. 624). No embargante la prescriptiva que esgrime nuestro «hombre docto graduado en Sigüenza»[46], nos quedaremos con Carvallo.

Sea como fuere, la comedia nueva acaba con la preceptiva temporal. Lo hace deliberadamente, consciente de los preceptos que contraviene, pero también de la imperiosa necesidad de hacerlo. Los espectadores del siglo XVII que acuden a ver comedias son exigentes y demandan una fábula que no se condensa de sol a sol. Lope, buen conocedor de su público, no solo lo asume, sino que lo lleva a gala.

> No hay que advertir que pase en el período
> de un sol, aunque es consejo de Aristóteles,

autos de la *Celestina* —una obra de las que más bebió la literatura aurisecular— los entenderíamos como *escenas* a pesar de su nombre.

[46] La ironía de la frase que emplea Cervantes para caracterizar a su cura ha hecho correr ríos de tinta. La materia la tocamos también nosotros en otro sitio (Sanz-Lázaro, 2022a, 2022c) tangencialmente, aunque no motivados por el clérigo cervantino, sino por la propia ciudad del Doncel y cómo había llegado a adquirir tal «prestigio» la universidad menor allí emplazada.

> porque ya le perdimos el respeto
> cuando mezclamos la sentencia trágica
> a la humildad de la bajeza cómica.
> Pase en el menos tiempo que ser pueda
> si no es cuando el poeta escriba historia
> en que hayan de pasar algunos años,
> que estos podrá poner en las distancias
> de los dos actos, o si fuere fuerza
> hacer algún camino una figura,
> cosa que tanto ofende a quien lo entiende,
> pero no vaya a verlos quien se ofende.
> ¡Oh, cuántos de este tiempo se hacen cruces
> de ver que han pasado años en cosa
> que un día artificial tuvo de término,
> que aun no quisieron darle el matemático!
> Porque, considerando que la cólera
> de un español sentado no se templa
> si no le representan en dos horas
> hasta el jüicio desde el Génesis
> yo hallo que si allí se ha de dar gusto
> con lo que se consigue es lo más justo. (Vega Carpio, 1609/2006, vv. 189-210)

Volviendo a Carvallo, este menciona que la comedia latina tiene cinco actos, como también son cinco en los que se dividen el drama alemán, francés e inglés tradicionalmente. Sin embargo, las comedias clásicas del teatro ibérico, tanto español como portugués, presentan una particularidad: están repartidas en tres actos. Como suele ocurrir con las cosas de éxito, a este formato no tardaron en salirle numerosos padres. Juan de la Cueva reclama para sí la invención de la siguiente manera:

> A mí me culpan de que fui el primero
> que reyes y deidades di al tablado,
> de la comedia traspasando el fuero:
> que el un acto de cinco le he quitado,
> que reducí los actos en jornadas,
> cual vemos que es en nuestro tiempo usado. (cit. en Sánchez y Porqueras Mayo, 1972, p. 143)

Sin embargo, si prestamos oídos a Cervantes, tal vez deberíamos atribuir el mérito al insigne alcalaíno.

> Y esto es verdad que no se me puede contradecir, y aquí entra el salir yo de los límites de mi llaneza: que se vieron en los teatros de Madrid *Los tratos de Argel*, *La destrucción de Numancia* y *La batalla naval*, donde me atreví a reducir las comedias de tres jornadas de cinco que tenían. (Cervantes Saavedra, 1615/2016b, pp. 94-95).

No dudamos de que ambos tuvieran la osadía de hacerlo, aunque, siendo tanto el uno como el otro aficionados al teatro, podríamos cuestionar la buena fe de sus afirmaciones —no lo haremos—. Tanto Lope de Vega (1609/2006, vv. 215-217) como Caramuel (1668/2011) atribuyen la feliz invención a Critobal de Virués. En cualquier caso, los tres irían a rebufo de la *Comedia florisea* de Avendaño, que presentaba esta división apenas había llegado al mundo el capitán. Y a todos ellos se les habría adelantado Antonio Díez al disponer así el *Auto de Clarindo* (Kayser, 1948/1992, p. 172). Tal cambio externo tendría como consecuencia que la *fábula* estructurada en *presentación*, *intensificación*, *clímax con peripecia*, *declinación* y *desenlace*[47] (Freytag, 1894, pp. 102-122) habría de repartirse en *presentación*, *nudo* y *desenlace*. Nos inclinamos empero a pensar que la innovación de Freytag consiste en hilar fino el nudo al describir la intensificación de la acción hasta el punto de inflexión climático, a partir del cual aquella comienza a declinar.

A despecho de nuestros pioneros ibéricos, la estructura en tres partes *prótasis*, *epítasis* y *catástrofe* no es en realidad innovación española —o al menos no lo es como un ideal—, pues ya la habían descrito —si tal vez más respecto a la fábula que a la forma externa— Elio Donato en sus comentarios a Terencio, así como Evanthius en el siglo IV en un tratadillo que se incorporó a los textos del primero. Además del prólogo, ambos gramáticos latinos distinguían tres partes en la actuación que le seguía.

> La comedia se divide en cuatro partes: *prólogo*, *prótasis*, *epítasis* y *catástrofe*. El prólogo es el primer discurso, que los griegos llamaban πρόλογος, esto es, las palabras que preceden a la verdadera recitación de la historia. Los hay de cuatro clases: συστατιχὸς, comendaticio, que preconiza la obra o al poeta; ἀναφοριχὸς, relativo, por el que se denosta al rival o se

[47] En el alemán original *Einleitung*, *Steigerung*, *Höhepunkt mit Peripetie*, *Fallen der Handlung* y *Lösung*.

> elogia al público; ὑποθετιχὸς, argumentativo, exponiendo el argumento de la historia; μιχτὸς, mixto, conteniendo todas las anteriores. Entre prólogo y prologio hay una diferencia, de acuerdo con algunos: es prólogo cuando el poeta se exculpa o se recomienda la historia al público. Sin embargo, es prologio cuando se dice algo sobre el argumento. La prótasis es el primer acto de la historia y el inicio del drama, en el que se desarrolla parte del argumento, pero otra parte se reserva para mantener la expectación de los espectadores. La epítasis es el desarrollo y progreso de las dificultades, por así decirlo, el nudo del enredo. La catástrofe es el retorno de las cosas hasta un final divertido, una vez que los personajes toman conciencia de los eventos. (Evanthius, ca. 350/1838, p. xviii, traducción propia, énfasis en el original)

Podríamos pensar, sin embargo, que más que una recuperación del formato en tres partes, el teatro español lo reinventa al plasmar la estructura implícita en la forma externa propia del género. No solo por cuanto Cervantes no apela a la autoridad de los clásicos, como convenía hacerse en estos casos para dar más empaque a una *innovación*, sino porque las fuentes clásicas se refieren estrictamente a una división interna. Se afirma en la introducción atribuida a Donato de la terenciana *Adelfos* que «esta obra, como otras de su género, tiene necesariamente cinco actos» (Donatus, ca. 350/1840, pp. 3-4, traducción propia). A pesar de esto, vuelve a reivindicar la segmentación terciaria: «los doctores de antaño los separaban [...], el prólogo es más calmado, se ocupa más de justificarse a sí mismo que de atacar al adversario; la *prótasis* es turbulenta; la *epítasis* ruidosa y la *catástrofe* suave» (Donatus, ca. 350/1840, p. 4, traducción propia), lo que concuerda con la máxima horaciana:«No tenga menos ni sobrepase los cinco actos / la obra, que quiere ser demandada, vista y repuesta» (Hor. *Ars*, vv. 189-190; traducción propia).

Esta falta de correspondencia exacta entre la segmentación externa y la interna no se le pasó por alto a Caramuel, quien delimitaba claramente ambas divisiones. Nótese que, sin embargo, habla de catástasis en lugar de catástrofe y no de una división cuaternaria en la que la catástasis es el segmento que precede a la catástrofe.

> Sé que las comedias de los antiguos se repartían en muchos actos y los actos en escenas, mas toda esa cantidad de escenas parece superflua, dado que toda comedia tiene —y eso mismo pensaban los escritores antiguos— tres partes: *prótasis*, *epítasis* y *catástasis*, pudiendo cada una corresponder a un acto. (Caramuel Lobkowitz, 1668/2011, 289-290, énfasis en el original)

En lo que respecta a la obra fijada por escrito, contamos con la inestimable labor de la ecdótica para discernir las divisiones más complejas de la obra. Sin embargo, resulta muy conveniente para nuestro propósito que todas las ediciones de textos dramáticos, críticas o no, se hayan ceñido tradicionalmente a unas convenciones de composición tipográfica precisas, aunque se han ido refinando a lo largo del tiempo. De esta manera, en mucha mayor medida que otros géneros, en el drama pueden identificarse incluso visualmente la estructura de la obra y, al nivel más alto, estarían los actos o *jornadas*.

Con esto, tendríamos las divisiones mayores de la comedia. Sin embargo, es posible —al menos en teoría— dividir estos segmentos globales en unidades menores atendiendo a diferentes criterios, como veremos a continuación.

Cuadros

Como decimos, los cuadros se determinan de varias maneras. En su forma clásica, se atienen a criterios escénicos; esto es, son lo que se ha venido denominando como *escenas*. Debemos cuidarnos, sin embargo, de tomar estas divisiones según «el modelo pseudoclásico de las escenas a la francesa, cuya inadaptación manifiesta al objeto teatral áureo no impide que continúe, aún en nuestros días, su maléfica influencia» (Vitse, 1998, p. 46). Por consiguiente, entenderemos la escena de la comedia nueva no como un recurso poético, sino como un constructo crítico —una manifestación más de la sempiterna propensión humana por encasillar, que nos lleva no pocas veces a la pareidolia— o escénico, para satisfacer las demandas organizativas de la función.

No en vano, la escena en su sentido clásico es una subdivisión externa del acto que responde a una necesidad práctica real. La segmentación facilita el control de los personajes que hay en cada momento sobre las tablas durante la representación. Esto cobra una especial importancia en montajes de compañías pequeñas o cuando pocos actores han de asumir varios papeles en la misma pieza (Kayser, 1948/1992, p. 170). Un cambio de decorado —o, en los teatros de comedias áureos, mudar una *apariencia*— o incluso correr un telón requerirían asimismo cierto control. Sería, no obstante, más discutible la influencia del criterio si el acto se ciñe al precepto de la unidad de tiempo.

Hay que advertir asimismo de que ninguno de estos criterios reclama para sí la predominancia absoluta en la determinación de la estructura. Los criterios escénicos no carecen de imprecisiones, mientras que los sonoros suelen resultar demasiado exiguos. Por lo tanto, ninguno de ellos resulta autosuficiente porque todos requieren el apoyo

de elementos estructurales internos, apelando a la línea temporal, el espacio o la acción. Ni que decir tiene que estos son, con harta frecuencia, hipotéticos, en el mejor de los casos (Oleza Simó, 2010, p. 127).

De cualquier manera, la segmentación como respuesta a una necesidad del montaje entra en conflicto con otra exigencia también de orden práctico. Dice Donato que «en un intento por retener a un espectador mal dispuesto, el dramaturgo latino minimiza las divisiones, temiendo seguramente que el espectador, aburrido al final de un acto o incitado a irse, rechace el resto de la comedia y se levante» (Donatus, ca. 350/1840, pp. 3-4, traducción propia). Los contemporáneos del crítico latino encontrarían una equivalencia directa en el público barroco, al menos en lo concerniente al nerviosismo del respetable ante el escenario vacío. Contravenir los gustos de la audiencia no es cosa que deba tomarse a la ligera: incluso el *Arte nuevo* previene sobre la posibilidad y recomienda evitarla en la medida de lo posible.

> Quede muy pocas veces el teatro
> sin persona que hable, porque el vulgo
> en aquellas distancias se inquïeta
> y gran rato la fábula se alarga;
> que, fuera de ser esto un grande vicio,
> aumenta mayor gracia y artificio. (Vega Carpio, 1609/2006, vv. 240-245)

Este temor al público lleva a menudo a eslabonar segmentos sin solución de continuidad, lo que produce una liminaridad difusa y, si a las palabras de Donato hemos de atenernos, también confusa. Este escribe lo siguiente al respecto, aunque hay que prevenir de que el comentarista no se refiere a las escenas, sino a los actos:

> Es difícil discernir con claridad las divisiones de los actos en las obras latinas, pues están ocultas [...]. Pero merece la pena aprender —si bien con dificultad— a discernir cómo y con qué criterio es posible entenderlas y distinguirlas. Empezaremos diciendo que un personaje que ha salido a escena cinco veces no puede salir otra vez más. Con frecuencia nos engañamos pensando que un personaje, por estar en silencio, ha entrado en escena, cuando espera sin hablar en el proscenio el momento de dar su réplica. Resulta, por lo tanto, necesario observar con cuidado dónde y cuándo se vacía el escenario de todos los personajes, de manera que se escuche al coro o el flautista. Cuando vemos eso, reconoceremos ahí el final de un acto. (Donatus, ca. 350/1838, p. 5, traducción propia)

Esta concepción deliberadamente imprecisa de las divisiones hace que, en la práctica, resulte muchas veces dificultoso discernirlas en el montaje[48]. También ocurre esto en el texto escrito, donde suele dejarse la tarea al buen entendimiento del lector lo que se dispone sobre el tablado a discreción del director escénico.

Lo que es cierto para separaciones explícitas como los actos, lo es con más razón para las implícitas. Existen ciertas convenciones que el público o el lector han de tener en cuenta para analizar la estructura externa de la obra e interpretarla de manera correcta. Si hablamos de actos en el teatro latino, Donato vuelve en nuestra ayuda, si bien resulta solo de utilidad en una versión preceptiva ideal del segmento.

> Con frecuencia confunde al lector que un personaje pronuncie la última línea de un acto y la primera del siguiente sin percatarse de que ha entrado [de nuevo], lo cual notan los expertos de inmediato por la secuencia de sucesos y el lapso de tiempo. Puede pasar, pues, que un personaje salga y entre cuando creemos equivocadamente que no se ha movido del proscenio. Un personaje puede salir a escena cinco veces, pero no decimos que deba hacerlo, se trata de dejar claro que no puede salir una vez más. En la tragedia se permite y es usual salir menos veces. (Donatus, ca. 350/1838, p. 5, traducción propia)

En la práctica, el aviso de Donato se toparía con un contratiempo que imposibilitaría su aplicación al teatro áureo. La división implícita de la jornada en escenas da lugar a que los personajes tengan oportunidad de entrar y salir un número indefinido de veces en cada acto. De esta manera, la segmentación del drama latino según Donato estaría más emparentada, al menos en este sentido, con la hipotética escena de la comedia nueva que con la jornada, y es en ese ámbito donde encontraría su aplicación.

En cuanto al reflejo de las escenas en el texto escrito, no nos acompaña la misma suerte que con los actos. El teatro del siglo XVII suele carecer de marcas específicas para señalar estas divisiones en las ediciones impresas, aunque se encuentran manuscritos

[48]Marcus (1973, pp. 327-328) llama la atención sobre un particular que introduce cierto grado de subjetividad interpretativa en el asunto. Saca a colación la *opera prima* de Schiller (1781/2001) *Die Räuber*. En un alarde pasional de *Sturm und Dränger*, el poeta mata a una serie de personajes uno detrás de otro en las tres últimas escenas de la obra: Spiegelberg, Franz, Schweizer, el conde de Moor y Amalia. Marcus considera un cambio de configuración dentro de la escena, dado que el personaje está ausente —aún de cuerpo presente—. No da indicaciones Marcus respecto al caso de que, dado que Spiegelberg muere en el mismo lugar en el que sucede la última escena de la obra, bien podría considerar el directo de escena volver a poner al actor allí tendido. De acuerdo con su interpretación de las configuraciones, cabe suponer que no habría que considerarlo como un personaje sino como parte del decorado. ¿Y los personajes que, como los bandidos en la obra, duermen roncando sonoramente, y *entran en escena* al ser despertados por la detonación de un arma? ¿Y si descansaran en silencio?

en los que se indican de forma explícita, por ejemplo, mediante una línea trazada entre las escenas.

Como indicamos, la obra teatral se divide también según criterios no escénicos; entre estos se cuenta la segmentación de la obra según el metro de los parlamentos (Vitse, 2010, p. 20). Aunque no pretendemos definir aquí un método para encontrar estas divisiones ni considerarlas, pues requeriría evaluar factores internos, que exceden las intenciones de este trabajo, sí aspiramos a facilitar en la medida de lo posible la tarea. En otras palabras, el resultado de este estudio ha de proveer una vía para señalar los cambios estróficos y presentar los resultados de la manera más neutral posibles. Es ya tarea del exégeta interpretarlos y determinar cuáles de esos cambios marcan una división más allá de lo puramente métrico. No entramos en los criterios que se consideren pertinentes a la hora de utilizar esta información. De ahí que no consideremos segmentar los textos en cuadros, sino organizar nuestros resultados de manera que ayuden a hacerlo. Esto es, no seremos nosotros los que hagamos estudios de segmentación, pero necesitamos conocer el tema y adaptar nuestro trabajo sus necesidades para producir resultados útiles.

Por otra parte, suscita hoy pocas dudas que hay una relación entre la naturaleza semántica de los pasajes el su metro de una buena composición teatral. El *Arte nuevo* ya da buena cuenta de ello cuando aconseja adecuar los versos a la materia de la que se ocupan.

> Acomode los versos con prudencia
> a los sujetos de que va tratando.
> Las décimas son buenas para quejas;
> el soneto está bien en los que aguardan;
> las relaciones piden romances,
> aunque en octavas lucen por extremo;
> son los tercetos para cosas graves
> y para las de amor, las redondillas. (Vega Carpio, 1609/2006, vv. 305-312)

Esto parte ya del proceso compositivo, el poeta con buenos hábitos de trabajo no escribiría de ordinario la fábula en versos a vuela pluma, sino que dispondría primero sus partes en prosa, para versificarlas después: «El sujeto elegido escriba en prosa / y en tres actos de tiempo le reparta» (Vega Carpio, 1609/2006, vv. 211-212), lo que implica interrupciones en el planteamiento de la versificación. Dicho de otra manera, dos segmentos en prosa contiguos tienen cada uno versificación propia, sea coincidente con la otra o no. Aunque, como recuerda Fausta Antonucci (2010, p. 78),

la segmentación métrica ya se había abordado con anterioridad (Williamsen, 1978; Dixon, 1985 y 1994; Vitse, 1985/1996 y 1988), esta cobró una especial relevancia teórica a partir de un artículo del último sobre el *Don Juan* (Vitse, 1998).

En este estudio, Vitse arguye que el metro tiene una función estructurante y, para ello, se sitúa «preferentemente en el texto escrito, con especial atención al texto en el momento de su escritura, y con clara y consciente postergación de la representación». Así se plantea «por qué, en un momento dado cambia un poeta de comedias la forma métrica hasta entonces utilizada por él» (1998, p. 49), no considerando las formas sino «el fenómeno del cambio en sí» (p. 50). De sus observaciones, colige Vitse que el cambio estrófico no es solamente un elemento opcional del paso de un cuadro a otro, sino una característica necesaria (p. 52).

Advierte, empero, de que, siendo el cambio estrófico condición necesaria, no es suficiente para determinar un cambio de cuadro. De ahí que no solo proponga el metro como elemento delimitador de las *macrosecuencias*, sino que identifique su función estructurante de las *microsecuencias* de las que se componen las grandes unidades. Esto es, *secuencias englobadoras* y *secuencias englobadas*. Va incluso un paso más allá y hace la siguiente afirmación sobre *El burlador de Sevilla*:

> A nivel de las microsecuencias, pues, y en este momento de nuestro análisis, no le pertenece la primacía al elemento espacial, que vendría confirmado eventualmente por algún que otro corte métrico interno. No, ahora, sobre todo cuando estalla entre ellas un conflicto de poder divisorio, quien manda es la métrica, y es su vasalla la repartición espacial configurada por los desplazamientos, hacia fuera o hacia dentro del escenario, de los personajes. (Vitse, 1998, p. 53)

Arguye que la comedia nueva se estructura en torno a la acción, a la que se supeditan los aspectos espaciales y temporales; las marcas divisorias entre distintos bloques de acción se señalan mediante el metro, que coincide o no con cambios espacio-temporales (Vitse, 1998, p. 55). En consideración a sus observaciones, Vitse propone el siguiente modelo interpretativo (pp. 58-59):

1. Ha de partirse de la premisa de que el dramaturgo emplea la pluralidad métrica tanto para fines estéticos como estructurales.
2. Con esto en mente, han de jerarquizarse las formas métricas englobadas y englobadoras en el texto.

3. Una vez sistematizado, se identifican las unidades métricas básicas o *secuencias*.

4. Con apoyo de los criterios espaciales, temporales, escénicos y escenográficos, se establece la distribución definitiva en macrosecuencias monométricas, macrosecuencias polimétricas sencillas y macrosecuencias polimétricas complejas.

La distinción entre los dos tipos de macrosecuencia polimétrica se halla en la concurrencia o no del resto de criterios, así como de otros indicadores externos para distinguir los límites de cada macrosecuencia fuera de toda duda. Esto aconseja una postura más conciliadora, como la propuesta de Crivellari (2013, p. 82) de una interacción hermenéutica entre criterios. En la práctica, parece una aproximación sensata, como reconoce (Casariego Castiñeira, 2018, pp. 38-46).

En *La cena del rey Baltasar* de Calderón, encontramos un ejemplo práctico de en qué nos afecta esto y las limitaciones de la automatización de los análisis digitales. Sánchez Jiménez y Sáez (2013, pp. 39-54) segmentan el texto en una macrosecuencia inicial (vv. 1-1224) y otra final (vv. 1225-1574). En la segunda, distinguen tres divisiones estructurales: una de silvas pareadas (vv. 1225-1322) con una copla englobada (vv. 1313-1316), una con un romance en *e-o* (vv. 1329-1380) orlada por sendas sextillas (vv. 1323-1328 y 1381-1386) y una última con un romance en *e-o* (vv. 1387-1574). Incluso conociendo los límites de las secuencias, el crítico debe escrutar la evidencia interna para determinar que la última sextilla pertenece al mismo segmento que el penúltimo romance, pero a uno diferente que el último.

Y, al contrario, el último segmento de la primera macrosecuencia (vv. 1074-1224), salvo unos pocos versos englobados cantados, es un romance en *e-o* y, por lo tanto, métricamente indistinguible del primero de la segunda. Considerar los cambios escénicos puede no resultar suficiente; vayan como muestra los últimos versos (vv. 1190-1224) de la secuencia del sueño (vv. 1074-1224), que ocurren tras despertar Baltasar. Estos comparten ubicación con la macrosecuencia siguiente, diferente del espacio onírico de la anterior. Atendiendo meramente al metro, espacio y vacíos escénicos, esos versos deberían integrarse en el segmento siguiente. Sin embargo, en cuanto a contenido, no cabe duda de que es un epílogo al sueño, por lo que tendría poco sentido hacerlo. Apenas contamos con la salida de la Idolatría como marca divisoria; así, careciendo del juicio sobre la evidencia interna que aporta el criterio humano, sería harto complicado determinar cuándo señala un cambio de secuencia la salida de un personaje a escena.

A esto debemos añadir el lugar que ocupan las partes cantadas en el esquema. Gilabert (2014, p. 84) aboga por reconocer la existencia de macrosecuencias musicales, a

las que atribuye una función estructural y capacidad de englobar otras partes menores, sean cantadas o declamadas. Aquí tenemos menos opciones porque solo podemos señalar divisiones si hay un cambio métrico, como en los versos ordinarios, pero no en el caso de música instrumental, que requeriría un escrutar las acotaciones con más atención de la que aquí podemos ofrecer. No obstante, el análisis automático y clasificación de las acotaciones podría ser un interesante campo a explorar en el futuro.

En resumen, a pesar de que no se establece una distinción explícita en los textos áureos suficiente para llevar a cabo la segmentación, es posible realizarla atendiendo a algunas características. Entre estas, se cuentan algunos elementos de orden externo, como el vacío escénico y el cambio estrófico, por lo que un texto etiquetado con dicha metainformación podría dividirse de forma automática según una u otra característica o una combinación de ambas, lo que reduciría el proceso manual a cotejar los elementos externos para verificar la idoneidad de las divisiones. El vacío escénico presenta algunas dificultades porque la automatización debe guiarse por las siempre parcas acotaciones. Esto es, identificar una configuración escénica a partir de una acotación o el escenario vacío llevando la cuenta de la salida de los personajes del escenario. Resultaría mucho más difícil considerar formas más sutiles de señalar la nueva secuencia, como la desaparición de un personaje. El cambio estrófico, por el contrario, se observa directamente con independencia de la estructura interna o del detallismo del poeta para describir la escenografía, se trata de un hecho objetivo y explícito. Nuestra labor es, pues, extraer esta información en la medida que lo admita el texto y presentarla de la mejor manera para determinar las lindes de los bloques constituyentes de cada obra dramática.

Unidades mínimas

Del mismo modo que los cuadros componen actos, aquellos también están constituidos por unidades menores. Estas, claro está, varían en función del criterio observado para segmentar la obra. De esta manera, si nos ceñimos al auditivo, consideraremos el verso como unidad mínima del segmento, mientras que, si nos decidimos por dividir considerando las configuraciones escénicas, la unidad textual mínima será el parlamento. Esta última, además, es un elemento común a todos las obras teatrales, en verso o en prosa, y prerrogativa del género. Su modalidad escrita se vale de un elemento compositivo visual para señalar tales divisiones, en lugar de los signos ortográficos con los que se denota el discurso directo en otros géneros. Dado que trabajamos con textos escritos, nos limitaremos a describir estas unidades en el texto fijado, con especial énfasis en las ediciones modernas de las que nos valdremos.

Lo primero que hay que hacer notar es que las piezas teatrales tienen como fin la representación, por lo que los apoyos visuales tienen una finalidad práctica concreta y no responden solo a consideraciones estéticas. Una composición que defina cada elemento ayuda al actor en la lectura escenificada —y, para nuestra fortuna, también a nosotros en el proceso de descomponer el texto en sus elementos constituyentes—. No llama a asombro, pues, que estas divisiones aparezcan de forma explícita incluso en los autógrafos.

Si tomamos, por ejemplo, el autógrafo de *La dama boba* (Vega Carpio, 1613), vemos que Lope configura el manuscrito según una disposición precisa: una columna amplia a la derecha con los parlamentos distribuidos en una línea por cada verso, a su izquierda aparece una columna menor con las didascalias de personaje subrayados. Las acotaciones suelen aparecer en una columna más a la izquierda (fol. 2v), ocupando las tres columnas y orlados por líneas de división entre cuadros (fol. 4r). No faltan inconsistencias, como versos fuera de las posiciones asignadas (fol. 5r). Los dramaturgos de la época no suelen diferir sustancialmente de este modelo convencional. Así, apreciamos en un autógrafo de Calderón de *El secreto a voces* (Calderón de la Barca, 1642) los mismos elementos, con pequeñas variaciones. Aquí vemos los parlamentos en una columna central principal, y las didascalias de personaje en otra más estrecha a la izquierda, subrayadas en ocasiones, mientras que otras acotaciones aparecen en el margen derecho.

Las ediciones impresas de la temprana Edad Moderna reúnen ya todos los elementos tipográficos necesarios para analizar la estructura de la obra dramática. Por ejemplo, en *La dama boba* (Vega Carpio, 1617), encontramos el cuerpo del texto dividido en dos columnas, con la indicación de la jornada de cuerpo entero. En cada una de las columnas se observan tres niveles textuales. Los parlamentos aparecen en redonda alineados a la izquierda y a cada línea le corresponde un verso. Las acotaciones van centradas en cursiva y las didascalias de personaje se sangran a la francesa en la primera línea del parlamento, con frecuencia como una abreviatura del nombre del personaje. Los versos compartidos difieren de la convención actual en que no se sangra la continuación en el siguiente renglón, sino que el la didascalia para señalar el personaje continuador del verso, así como la primera línea de su parlamento, se añaden a la última línea del parlamento precedente, a continuación del inicio del verso si el espacio da lugar a ello. Las imprentas respetan este mismo modelo en ediciones posteriores, como se aprecia, verbigracia, en *El secreto a voces* (Calderón de la Barca, 1795) impreso a finales del siglo XVIII, más de cien años posterior a la mencionada de Lope. Se repiten los mismos elementos sin apenas alteraciones.

Las ediciones modernas, como decimos, adoptan estas convenciones con pequeñas

variaciones, si acaso con algunas mejoras y añadidos de unos pocos elementos para facilitar la lectura, como la habitual disposición escalonada de las réplicas en los versos compartidos (Arellano, 2007, p. 67), en lugar de la línea única. A veces, se añaden otras indicaciones ausentes en los manuscritos y ediciones clásicas, como las acotaciones editoriales. Estas van entre corchetes o paréntesis en algunas impresiones modernas para distinguirlas de las propias del texto. También se diferencia entre la acotación externa de carácter independiente y la interna[49], que afecta a un parlamento o una parte de este, que suele anteponerse a la primera línea de las que modifica en el mismo nivel de sangrado que el resto de líneas del parlamento.

Con estas prevenciones, consideramos entidades clasificatorias concretas, empezando por las metatextuales, esto es, autor para el comediógrafo, género —por ejemplo comedia, auto sacramental o entremés— y subgénero —de capa y espada, de figurón, etc.—. En cada una de estas clases, se incorporan varias piezas teatrales. Por su parte, encontramos divisiones textuales, la mayor de las cuales es la jornada o acto en obras de cierta extensión y género. Por debajo de esta, se hallarían los cuadros que, como hemos dicho, no se indican de manera explícita. Por el contrario, otros elementos de orden inferior son discernibles visualmente en la composición de la página. Se distinguen los parlamentos, las líneas que los componen, así como un atributo, el *locutor*[50] que pronuncia cada uno. Como advertimos, las líneas tienen una correspondencia biunívoca con los versos en ediciones antiguas. Por el contrario, el locutor no es en ellas un atributo único del parlamento ni la línea en su totalidad elemento constitutivo, pues el verso compartido tiene dos locutores y cada una de sus partes pertenece a un parlamento distinto. En las ediciones modernas, por el contrario, esto no se cumple para los versos compartidos, que se dividen en tantas líneas como en parlamentos se distribuyan.

Las acotaciones también se distinguen en la tipografía y composición. Dependiendo de su posición relativa respecto al parlamento, se dividen en internas o externas. La acotación externa estaría al mismo nivel que el parlamento, mientras que la interna se subordinaría a él. El parlamento, por su parte, se caracteriza por el locutor que lo pronuncia y su posición en la serie, y está compuesto de líneas y, ocasionalmente, de acotaciones internas. Las líneas se caracterizan también por el verso al que pertenecen, que, como dijimos, en las ediciones modernas suele corresponderse con la línea, pero no necesariamente. A pesar de ello, estas ediciones facilitan en gran medida

[49]No confundir con las *acotaciones implícitas* (Monzó, 2019, pp. 38-40). Cuando nos referimos a acotaciones en este trabajo, sean internas o externas, lo hacemos a las explícitas.

[50]Hablamos aquí de *locutor* y no de didascalia de personaje porque ya no nos referimos a la acotación del texto, sino a un atributo de las categorías abstractas que definimos.

la identificación visual, al señalar con niveles adicionales de sangrado las réplicas y contrarréplicas en los versos que se extienden por varias líneas.

3.5 Ediciones… digitales

Al abordar digitalmente los textos áureos, nos enfrentamos a diversos contratiempos y, contra lo que pudiera parecer, la escasez de modelos lingüísticos para analizar la variedad diacrónica es el menor de ellos. No debe llevar esto a sorpresa, porque salvar ese obstáculo no requiere más que tomar una edición moderna con ortografía normalizada. La ortotipografía, como sabemos, es una convención relativamente moderna que, en las obras auriseculares, depende en buena medida de los gustos y recursos instrumentales del cajista. Por fortuna, esto no es sino una inconveniencia menor, ya que la finalidad del teatro es —no lo perdamos nunca de vista— llevarse a escena. La voz, como es evidente, no transmite la ortografía.

No quiere decir esto que haya unanimidad en la ecdótica, si bien una escuela opta por modernizar las grafías, siempre que esto no afecte a la pronunciación (Arellano, 2007, p. 36), hay críticos que permanecen más fieles a la grafía original. Diferencias aparte, ambas aproximaciones suelen ser respetuosas con el consonantismo del texto y conducirían a una misma transcripción fonética moderna. Superadas las vacilaciones del siglo XVI, el reajuste de las sibilantes está consumado a efectos prácticos en el periodo que nos ocupa (Lapesa, 1981/2008, secs. 91-93), por lo que una modernización ortográfica no debería alterar la esencia de la representación escénica.

Donde encontramos un problema con la modernización es en parte de las vocales. Calderón, por ejemplo, escribe en sus autógrafos *cay*, que se vuelve *cae* en ediciones modernas. Esto da lugar a ciertas complicaciones. Partimos de la premisa de que, como veremos más adelante, las vocales altas /i/ y /u/ hacen diptongo fonológico (ver 4.6) en interior de palabra, por lo que *cae* sería bisilábica, al contrario que *cay*, que sería monosilábica. En principio, el texto moderno sin advertencia al respecto nos llevaría a dos silabeos diferentes. A nivel fonético podríamos unificar las sílabas porque la versión moderna no forma diptongo. Si optamos por esta, encontraríamos un nuevo contratiempo de orden suprasegmental. Tanto el diptongo como la reducción son decrecientes. Por lo tanto, para que pudiera haber una sinalefa con una hipotética vocal en la siguiente palabra —que resultaría en triptongo—, esta debería tener una apertura igual o menor que la semivocal o vocal reducida precedente (Navarro Tomás, 1957). De esta suerte, *cae* podría unirse, además de con palabras comenzando con vocal alta —fonema que se reduciría—, también con otras que comiencen por *e*, mientras que *cay* solo aceptaría vocales altas. Más complicado lo tiene el editor para

satisfacer por igual ortografía y pronunciación en la equivalencia /i/-/e/ en palabras como *fénix* o *fácil*, como ocurre, por ejemplo, en posición de rima en un romance en *e-e* de la *La vida es sueño*.

Los signos de puntuación son un universo en sí mismos. Partimos de manuscritos con una puntuación deficiente las más de las veces (Arellano, 2007, pp. 49-63). De aquí, pasamos a una tipografía que varía según las veleidades y equipo de cada cajista, inconsistentes entre imprentas, entre libros de la misma imprenta e incluso, en ocasiones, entre páginas del mismo libro (Blecua, 1983/2001, pp. 142-143; Carreño, 2019, p. 99). Gracias a la normalización ortográfica, las ediciones modernas han podido superar estas carencias, así como otras de índole acentual. Aquí es ya el criterio del editor el que fija la puntuación conforme a la norma y las variaciones son deliberadas para reflejar la interpretación del texto.

Para ilustrar cómo afecta al ritmo una elección editorial aparentemente mínima, tomaremos prestado un ejemplo de Arellano (2007, p. 64) en el que trae a colación las formas *Marinatarasca* y *marina tarasca*. Más allá del significado, la acentuación prosódica de *Marinatarasca* sería `----+-` mientras que la de *marina tarasca* sería `-+--+-`. No nos cabe duda de que las ediciones críticas modernas aventajan a históricas, en tanto que las primeras atienden a un criterio filológico y no dependen de la intuición del cajista o la disponibilidad de tipos. No podemos sino subrayar la importancia de emplear ediciones bien curadas para obtener los mejores resultados posibles.

Las ediciones modernas tampoco están exentas de inconvenientes. La multiplicidad de formatos digitales, estilos y convenciones hace que extraer la información requiera un proceso distinto entre ediciones, incluso si, a simple vista, las diferencias son mínimas. Hemos hablado ya de la disposición del texto, de la estructura dramática ideal fijada por escrito. Queda, por lo tanto, indicar cómo se plasma esta disposición en las ediciones digitales que queremos examinar. No es esta una cuestión excesivamente compleja en lo que se refiere al propio texto, ya que el trabajo de enjundia ya lo han llevado a cabo otros para llegar hasta allí. Las reglas del juego ya las pactaron dramaturgos e impresores con los lectores en tiempo inmemorial, por lo que poco queda que añadir en ese sentido. El texto en sí, gracias a la encomiable labor de los editores, llega hasta nosotros curado con esmero. El interés para nuestro trabajo radica en la estructura de la información que queremos extraer, así como otra metatextual que produciremos a partir de aquella.

Debemos mencionar en primer lugar algunos aspectos sobre el formato interno de la edición digital del texto dramático. No es esta una cuestión trivial, pues la forma en que se organiza la información determina tanto la manera de encarar la tarea como

los usos que se le podrá dar al producto resultante. Para ello, debemos considerar estas ediciones como un conjunto organizado de datos, tal y como los describimos en el capítulo anterior. En este sentido, debemos desembarazarnos de la abstracción con la que vemos el libro electrónico, dejar de entenderlo como una emulación digital del libro de papel y reducirlo a sus partes: el lector de libros electrónicos es un computador equipado con una pantalla en la que representa símbolos —el texto— conforme a las instrucciones que encuentra en un archivo informático —el libro electrónico—. Por lo tanto, el computador opera con los datos del libro electrónico representándolos en la pantalla.

Aquí es donde las diferencias entre datos estructurados y no estructurados adquieren protagonismo. Nuestras fuentes serán principalmente ediciones dispuestas como datos no estructurados. Esto es, el libro electrónico, cuya disposición visual reproduce la del libro y que podemos incluso imprimir de manera que el producto final no diferiría cualitativamente de un libro ordinario. La máquina conoce detalles visuales del texto, pero permanece ajena a su estructura interna: sabe que debe colocar la primera letra de la didascalia de personaje en unas determinadas coordenadas espaciales y la segunda a su derecha; la línea inicial del parlamento ha de ubicarse en tal posición, y las sucesivas, a la misma en el eje horizontal pero añadiendo un desplazamiento hacia abajo. Estas tres partes son exactamente equivalentes para la máquina, ya que interpreta la posición geométrica como un valor arbitrario. Esto es, la información no está organizada jerárquicamente, lo que dificulta su análisis. Nótese cómo el cerebro humano organiza la información visual para deducir la estructura textual. La tarea consiste, por lo tanto, en enseñar al computador a interpretarla de forma análoga a como lo hacemos cuando leemos la página.

Los textos críticos se presentan también como datos estructurados, o, mejor dicho, semiestructurados. El formato más en boga y que se considera estándar al tiempo de redactar estas líneas es XML-TEI, que ya introdujimos en el capítulo anterior. En este caso, la organización no es estrictamente visual, sino que se encuadra en una jerarquía lógica señalada con etiquetas. De este modo, la etiqueta que señala el texto contendría otras divisiones, por ejemplo, los preliminares y el cuerpo del texto. Estas contendrían otros elementos, como los actos, que, a su vez, estarían formados de escenas, y estas, de elementos menores, como parlamentos y acotaciones. Siguen siendo factibles otras unidades, tanto por encima como por debajo. Así, es posible agrupar los versos en unidades métricas (por ejemplo, englobando los que pertenecen a un romance en una etiqueta al efecto), o indicar información adicional dentro de la jerarquía del parlamento para marcar, verbigracia, la estructura métrica.

De la misma manera, se incluye otra información metatextual, mediante la que se

insertaría un aparato crítico al uso, tanto en forma de notas usando la etiqueta apropiada en la jerarquía a la que concierna o de, por ejemplo, unos preliminares, mediante la creación de otra estructura aparte y al mismo nivel que el cuerpo texto. Esta organización es ideal porque, por un lado, existen herramientas para transformarla a un formato visual tradicional para su lectura y, por otro, la jerarquización de la información facilita su procesamiento digital. Lamentablemente, no abundan las ediciones XML-TEI y, entre estas, las que contienen información complementaria metatextual son los menos. Así pues, cobra más importancia nuestra labor de tomar los textos en el primer formato y extraer la información para disponerlos en el segundo u otro equivalente.

3.6 Análisis dramático cuantitativo

Dinu (1968, p. 34) y Marcus (1973, p. 316) coinciden en destacar la labor que requiere llevar a cabo los cálculos envueltos en los análisis dramétricos, pues una pieza de N personajes tiene 2^N posibles configuraciones escénicas. La primera jornada de *La vida es sueño*, por ejemplo, con apenas diez personajes, adoptaría 1.024 valores. Llegados al incremento exponencial, el análisis asistido por computador se torna crucial (ver Capítulo D para un ejemplo de análisis dramétrico manual). Las magnitudes envueltas, si bien pueden obtenerse a mano, requieren recopilar datos y hacer operaciones sencillas para obtener los valores de cada una. En una comedia, tendríamos que ir unidad por unidad[51] anotando los personajes presentes. A partir de eso, se llevarían a cabo los procedimientos aritméticos necesarios para calcular las medidas relativas para cada par de ellos. Mediante la combinación de estos resultados, obtendríamos los valores absolutos para cada personaje y otras medidas complejas. Si bien una jornada requiere un tiempo relativamente breve y no supondría mayor problema para, por ejemplo, repetir el proceso con las otras dos jornadas y analizar la variación de las configuraciones escénicas a lo largo de la obra si —como es nuestro objetivo— automatizamos la extracción de datos y disponemos estos de forma tal que puedan ser usados por el programa informático que realice las operaciones, obtener los valores de toda la obra sería, literalmente, cuestión de segundos.

Si bien tal análisis daría lugar a observaciones interesantes, no variaría sustancialmente de la aproximación tradicional. Después de todo, nos habríamos acercado al texto mediante la lectura de una sola pieza; las ventajas que aportan las herramientas digitales, pese a ser notables, no alteran la esencia del método. Cambiemos el enfoque y pensemos en términos de lectura distante. ¿Y si analizáramos la variación de las

[51] Macrosecuencia, secuencia, escena, cuadro...

configuraciones escénicas de las comedias de Calderón según su género? La extracción de datos y el cálculo no diferiría en su grado de complejidad, pero ahora requeriría un tiempo notablemente mayor porque obligaría a repetir el mismo proceso para cada jornada de cada una de las piezas. ¿Y si queremos comparar autos sacramentales con comedias? ¿O el corpus de Calderón con el de Lope? No es necesario volver a decir que la posibilidad teórica de llevar algo a cabo, con harta frecuencia, se da de bruces en la práctica con los siempre escasos recursos humanos y temporales.

De acuerdo con Ilsemann (1998, pp. 15-16), nos encontramos con dos vías de investigación opuestas. En una dirección se hallaría la búsqueda heurística de diferencias en un corpus textual completo, como, por ejemplo, la caracterización tipológica de un género. Pensemos para ilustrarlo en las divergencias entre una comedia y una tragedia si consideramos parámetros tales como el número de personajes en escena, el tipo de personaje o la longitud de los parlamentos. En el teatro del Siglo de Oro, además, podríamos sumar a las variables puramente teatrales otras métricas. Cabría especular, por poner el caso, que, si la interpolación de consonancias en tiradas asonantes aparece de manera habitual en los pasajes cómicos, una frecuencia baja de este recurso en la obra bien podría sugerir un género más serio.

De la misma manera, se establecen criterios diferenciadores dentro de una sola obra dramática. Podría suponerse que los personajes se caracterizan mediante las relaciones cuantitativas de *dominancia escénica* (Capítulo D). Ilsemann advierte que el peso de tales relaciones lo confiere la recepción subjetiva, por lo que el análisis cuantitativo, en este caso, serviría para objetivizar tal interpretación. Por este motivo, una formulación deficiente de las hipótesis del análisis por computador conduce con frecuencia a resultados triviales. Sería un pobre aporte, por ejemplo, contar los parlamentos de *El burlador de Sevilla*, comparar los resultados de los personajes y concluir de la abrumadora dominancia de don Juan Tenorio que se trata de un personaje protagonista. De la misma manera, tampoco diría nada nuevo comprobar que la densidad escénica es superior en las comedias, donde la claridad de la trama y las tipologías reconocibles de personaje desempeñan un papel primordial.

Resultaría tal vez más productivo, creemos, emplear estas medidas para comparar valores semejantes. Esto es, contrastar la dominancia escénica entre los protagonistas de dos obras, la del protagonista por género o la densidad escénica entre los corpus de comedias de varias plumas. La obviedad, en esos casos, se diluiría al pasar de la clasificación categórica mayor al sutil continuo de los detalles que, ahora sí, escapan fácilmente al escrutinio del lector corriente. La mente humana ha evolucionado para clasificar el mundo a su alrededor en grandes grupos, conforme a unos pocos patrones

más o menos abstractos. El computador, por el contrario, es capaz de trabajar con infinidad de detalles concretos para llevar a cabo su clasificación; esto es el fundamento de, sin ir más lejos, los análisis estilométricos para la atribución de autoría.

En cualquier caso, debemos proporcionar dos cosas: la segmentación de la obra y los personajes que intervienen en cada segmento. Para lograrlo, debemos seguir caminos diversos dependiendo del criterio tomado para hacer las divisiones. Como norma común, los límites de un acto son los de uno o dos de sus segmentos. Si atendemos a un criterio sonoro, debemos dividir el texto según los metros de los versos que lo componen. Si pretendemos dividir en escenas, dado que no suelen indicarse explícitamente como con los actos, debemos hacer el seguimiento de los personajes que hay sobre las tablas en cada momento. Para esto, nos valdremos de las acotaciones personales operativas[52]. La escena VI[53] de *La vida es sueño* acaba de la manera que vemos en el ejemplo 1[54]

(1) *(Éntranse todos. Antes de que se entre el* REY *sale* CLOTALDO, ROSAURA *y* CLARÍN*, y detienen al* REY.)

Esto es, se vacía el escenario (*éntranse todos*) y salen (*sale*) Clotaldo, Rosaura y Clarín, que toman a Basilio con ellos cuando no ha desaparecido del todo, pero ha abandonado el proscenio (*antes de que se entre*). Por lo tanto, el programa debe señalar un cambio de nueve —si consideramos dos soldados junto a Astolfo, dos damas de Estrella y dos acompañantes en la comitiva real— a cero personajes, y de cero a cuatro. A continuación, en presencia de Rosaura y Clarín, conversan Clotaldo y el rey y, tras el cuarto parlamento de este, aparece la acotación de (2) al pie de su parlamento. Al estar en singular, habremos de entender que solo le afecta a él. Quedan así Clotaldo, Rosaura y Clarín en solitario ante el espectador.

(2) BASILIO. No te aflijas Clotaldo,
si otro día hubiera sido, 875
[...]
descuidos vuestros, perdono.
(Vase.)

[52]Para encuadrar algunas de estas acotaciones aquí según la clasificación de García Barrientos (2020, pp. 76-79) asumiremos que *estar presente* es una acción.

[53]Según las divisiones de la edición de Ciriaco Morón (Calderón de la Barca, 1636/2019).

[54]Morón se guía aquí por la edición de Madrid (Calderón de la Barca, 1636a). En la de Zaragoza (Calderón de la Barca, 1636b) se lee en su lugar: «*Vanse todos y queda el* REY. Sale[n] CROTALDO [*sic*], CLARÍN, ROSAURA, *y gente*».

Insistimos: la división en escenas responde en muchos casos a una cuestión interpretativa, por lo que, al extraer los elementos del texto debemos presentar la evidencia sin hacer juicios valorativos, pero ofreciendo información objetiva que facilite hacerlos. Para nuestro trabajo, esto significa llevar la cuenta de personajes en escena y señalar cuando se ha producido un posible vaciado del escenario. La interpretación de estos datos queda al buen criterio del filólogo. Aceptará los vacíos como divisiones tal cual o, si es sensato, cotejará estas marcas con la información interna. La misma prevención concierne a la segmentación métrica. Aunque marquemos los cambios, únicamente la observación en detalle es suficiente para discernir los segmentos englobantes de los englobados.

No debemos quedarnos en los segmentos, por lo que profundizaremos un nivel más en la estructura dramática. Ilsemann (1998, p. 11) defiende que el parlamento es la unidad mínima natural del texto dramático apta para el análisis digital. Las unidades subordinadas no tienen sentido en este tipo de análisis. La palabra o el verso no constituyen unidades dramáticas por sí mismas. No quiere decir esto que no lo sean en otros ámbitos —hay estudios estilométricos que toman, por ejemplo, la palabra y el verso como unidades mínimas (Kroll y Sanz-Lázaro, 2022)— ni que sean irrelevantes en el análisis dramático. Al contrario, estas subunidades caracterizan el parlamento, que definimos en función de sus elementos constitutivos.

Las unidades mínimas estructurales y sus componentes presentan una situación muy distinta a la de los segmentos. Al contrario que estos, parlamentos, líneas, versos y palabras están claramente definidos, por lo que establecer sus límites no es una cuestión interpretativa. En lo que a nuestro trabajo concierne, esto se traduce en identificar tales componentes de manera directa sin temor a incurrir en apreciaciones subjetivas que introduzcan un sesgo en los datos producidos.

Dicho de otro modo, para las escenas debemos limitarnos a presentar la información extraída, con el fin de facilitar el ulterior trabajo de segmentación; asimismo es posible presentar los segmentos métricos, pero la distinción entre macro y microsecuencias requiere considerar otros factores que no se infieren por completo a partir de los elementos textuales. Los versos, líneas y parlamentos, por el contrario, son identificables y mensurables, por lo que, una vez procesado el texto, el investigador puede aplicar métodos estadísticos sin necesidad de un paso interpretativo intermedio. De esta forma, las medidas basadas en estos componentes son las que mejor se prestan a la lectura distante, ya que permiten procesar un gran número de piezas teatrales y comparar valores dramétricos con una intervención humana mínima.

Ilsemann propone valorar los siguientes indicadores:

1. El número de letras que constituyen el parlamento
2. El número de palabras que constituyen el parlamento
3. El número de parlamentos de un personaje
4. El parlamento más largo del personaje

Asimismo, propone considerar esos valores respecto al total del personaje y del texto. Esto, claro está, es el punto de partida, ya que los resultados de un personaje por sí mismo no dicen nada fuera del contexto que provee el resto de apreciaciones de conjunto, sea respecto a otro personaje, a un grupo de personajes o a la totalidad del elenco.

Tampoco debemos perder de vista que el teatro áureo está en verso. Esto aporta un componente adicional a la caracterización de los parlamentos que responde a otros criterios, como el número de versos, el número de sílabas o el tipo de metro empleado. Podemos incluso apelar a la prosodia y describir el parlamento según el número de acentos o los patrones rítmicos de sus versos.

Lo ideal sería, no obstante, poder combinar observaciones sobre los parlamentos con las configuraciones escénicas y otros elementos dramáticos para obtener valores tales como parlamentos ajenos en presencia o en ausencia del personaje, número de apartes en el parlamento[55], parlamentos ajenos desde la muerte del personaje, parlamento más largo de ese personaje, media de palabras por personaje o los porcentajes de parlamentos y palabras de un personaje respecto al total respectivo (Ilsemann, 1998, pp. 12-13). Donde Ilsemann contempla palabras, nosotros también consideraremos otros elementos definitorios del parlamento, como, por ejemplo, los versos.

Cuantos más detalles del texto identifiquemos, obtengamos y dispongamos de la manera adecuada, más posibilidades hay de hacer uso de ellos para considerar la pieza teatral desde una perspectiva similar a la que hemos ofrecido u otra que explote los datos recopilados de manera diferente.

Debemos desarrollar la segunda parte del trabajo con esto presente si queremos ser capaces de proporcionar resultados útiles más allá de la mera demostración teórica. Esto es, hemos de instruir al computador para componer cuadros de información

[55]Cabría plantearse si han de distinguirse diferentes categorías de elementos en el texto dramático según si contribuyen al avance de la fábula o tienen carácter metadiscursivo. Volvemos aquí una vez más a los fundamentos que habían planteado Aristóteles o Hegel sobre la naturaleza de los géneros literarios y las características distintivas de cada uno.

aptos para ser interpretados digitalmente y realizar con ellos operaciones[56]. La elección de los elementos es vital porque no solo determinan la posibilidad de obtener resultados o contribuyen a optimizarlos, sino que los aspectos de la obra teatral que afloraren dependen de la elección de las partes observadas. Repetimos: aunque escape de nuestro propósito, nuestra labor es proporcionar suficientes elementos de juicio para llegar a un buen puerto, pero, incluso si proveemos los medios, de poco servirán si se carece del criterio filológico para emplearlos e interpretarlos con sensatez.

[56]Considérense operaciones similares a las presentadas en Capítulo D, pero extrapolando a un corpus masivo de piezas teatrales lo hecho en ese ejemplo para una sola jornada.

Confórmase también con el latín en el *a.b.c*, aunque difieren en esto, que la lengua castellana tiene una *j* larga que vale por *gi*, y tiene una que nosotros llamamos *cerilla*, la cual hace que la *c* valga por *z*; tiene más una *tilde* que en muchas partes puesta sobre la *n* vale tanto como *g*.

Juan de Valdés, *Diálogo de la lengua*

Fonología y fonética 4

4.1 Por qué una aproximación fonológico-acústica

La fonología y la métrica son disciplinas íntimamente ligadas. Después de todo, la rima no es otra cosa que la repetición de ciertos sonidos, las sílabas métricas son aquellas con núcleo vocálico puro y el ritmo es producto de la prosodia. Una aproximación a la métrica no puede partir del texto escrito, sino de su recitación o declamación (Saavedra Molina, 1944, p. 85). Después de todo, «en la poesía, la materia prima es la sustancia fonética, el plano de la articulación» (García Adánez, 2005, p. 96). Teniendo esto en cuenta, no es de extrañar que tratadistas de renombre se hayan ocupado de uno y otro campo de manera complementaria (Quilis, 1984/2013, 1993/2019; Navarro Tomás, 1991, 2004) y que, además, al hacerlo traspasaran en ambas direcciones las líneas divisorias para ilustrar un área con ejemplos de la otra. Así pues, incluir aspectos de la fonología en este trabajo no es sino una demanda del vínculo natural que la une a la poética.

En concreto, son dos las razones que llevan a tal decisión metodológica. La primera de ellas es de orden práctico: el trabajo de analizar fonológicamente resulta inexcusable después de todo, incluso aunque sea de forma implícita. Poco importa lo mucho que intentemos disimularlo y hacerlo pasar por otra cosa. Por ejemplo, asumimos como algo natural que *cantaba* y *lava* tienen rima consonante y, sin darle muchas más vueltas, concluimos que *ciencia* tiene asonancia en *e-a*; no en *i-a*, ni en *i-i* ni tampoco en *e-i*: en *e-a*. Sin embargo, tras estas aparentes obviedades se esconde un proceso que no es trivial, a pesar de que nuestra mente lo haya llevado a cabo sin dar tiempo a reflexionar sobre cómo se ha hecho. En el primer caso, hemos ignorado los grafemas ⟨b⟩ y ⟨v⟩ y hemos evaluado ambos como el fonema /b/[57]. En el segundo,

[57]Usaremos la notación del Alfabeto Fonético Internacional (AFI).

hemos descartado los sonidos semiconsonánticos representados por el grafema ⟨i⟩ y nos hemos quedado con los núcleos vocálicos. De igual manera habríamos procesado dígrafos o grafemas que representan distintos fonemas en función del contexto. En resumen, hemos leído fonológicamente la ortografía y, para conseguirlo, hemos tenido que seguir todos esos pasos, conscientes de ello o no.

A la máquina, como es de suponer, hay que indicarle explícitamente cómo ha de realizar cada una de esas tareas, lo que acaba abriendo la puerta a dos aproximaciones alternativas: por una parte, podríamos abstraernos de la fonología y abordar el proceso desde una perspectiva ortográfica, como es lo habitual en la escansión automática del verso español. Esta forma lleva las más de las veces hasta un resultado correcto, a pesar de representar una idealización del verso aparentemente desconectada de su esencia eufónica. Sin embargo, esto no evita tener que recurrir en último término a la fonología, incluso si, como en el ejemplo anterior, lo disimulamos bajo subterfugios.

En realidad, estos análisis ortográficos, como ya sabemos, consisten en establecer pautas de equivalencia entre sonido y grafía para poder aplicar reglas fonológicas al texto escrito. Por lo tanto, si hemos de valernos de normas fonológicas después de todo, hagámoslo explícitamente. Esto redundará positivamente en la claridad expositiva, al facilitar una división más precisa del trabajo según los aspectos lingüísticos que se tratan. Como contrapartida, tal decisión requiere invertir tiempo, pero esto queda compensado con creces por la agilidad que confiere al resto del proceso. Como beneficio adicional, aporta un transcriptor fonológico cuyos posibles usos se extienden más allá del propósito de este trabajo.

Por otra parte, la ortografía presenta ambigüedades deliberadas, pues la Real Academia prefiere una representación inadecuada a comprometer la consistencia interna de su sistema ortográfico. Dado que las ediciones modernas se ciñen a las prescripciones académicas, resulta imposible distinguir solo con el texto escrito el hiato en palabras como *guion* o *actuar*. Al emplear la transcripción fonológica distinguiendo entre vocales y vocoides, tal ambigüedad desaparece. Esta ventaja se extiende luego en el análisis métrico a los diptongos impropios formados mediante sinalefa.

La segunda razón que aconsejaba recurrir a la fonología —más bien, a la fonética acústica— es que algunos aspectos fonéticos no se indican en el texto y tampoco responden a un patrón generalizable. Es más, la pronunciación cambia dependiendo del gusto personal, de la forma de entender el parlamento, quizá de las indicaciones del director de escena... Los problemas generales que plantea Sánchez Jiménez (2017, p. 12) para el metro lírico se hacen más presentes, si cabe, en el teatro, donde las tiradas de versos provienen de más de una garganta las más de las veces. Un ejemplo

concreto lo ofrecen las sinalefas, en tanto que el acento y, con él, el rasgo silábico de uno de los fonemas, se desplaza en determinadas circunstancias[58]. Esto, si bien no altera el patrón rítmico del verso, determina el núcleo vocálico de la sílaba. A final del verso en rima no habría problema, ya que los núcleos vocálicos se deducen de aquella, pero en interior o a final de verso no rimado, crearía ciertas complicaciones. En cualquier caso, se hace evidente la necesidad de abordar la diversidad de realizaciones.

¿Cómo salimos del atolladero? La autoelicitación queda descartada por el indudable sesgo que introduciría. Resta, por lo tanto, afrontar la cuestión escuchando grabaciones de versos dudosos. Sin embargo, resulta difícil para el oído poco entrenado. Diversos factores pueden llevar a una percepción que no responde a las características físicas del sonido. No pocas veces, lo percibido, incluso por la misma persona que produce el sonido, no responde a sus propiedades. Nuestra propuesta es, retomando lo dicho en el primer capítulo, abstraernos del objeto de estudio y recurrir a un modelo. Así pues, en este presentamos los aspectos fonológicos que servirán para caracterizar los versos en la segunda parte, así como otras aproximaciones complementarias de las que nos valdremos para resolver dudas de forma tan objetiva como las circunstancias lo permitan.

4.2 Los sonidos de la lengua

Cuando oímos un verso, antes incluso de tener ocasión de pensar en su significado, ya somos conscientes de su sonoridad. No solo en los versos teatrales, sino también en la lírica, incluso leída. La exégesis semántica requiere un esfuerzo consciente, mientras que percibimos aun sin proponérnoslo la eufonía de las palabras hasta en la lectura más casual. El elemento fundamental que compone esa musicalidad del texto son sus sonidos. Para analizar el verso propiamente, habremos de abordarlos considerando esos sonidos. Para ello, necesitamos un sistema con el que describirlo de un modo adecuado. Existen diferentes maneras de construir tal sistema considerando diversos aspectos constitutivos, sea de acuerdo con la producción de los sonidos, elementos acústicos o su percepción. En tal caso, hablaremos de *fonética*. Podemos también considerar los sonidos atendiendo a su función en el sistema comunicativo lingüístico y, entonces, trataremos la *fonología* (Quilis, 1993/2019, p. 23).

Fonología, pues, es la subdisciplina lingüística que se ocupa de los sonidos conforme a la función que estos desempeñan en una lengua concreta. Los sonidos, desde esta

[58] Precisamente introduje este aspecto tratando de identificar el núcleo silábico de sinalefas en casos donde había vacilaciones en los informantes, al hallar que entre dos oyentes había discrepancias sobre la vocal en la que creían percibirlo en determinados casos.

perspectiva, sirven para diferenciar significados mediante oposición y están determinados por la función que cumplen en el sistema lingüístico de esa lengua. Esto es, los sonidos sirven para diferenciar entre las diferentes *palabras* mediante variantes combinatorias en su articulación. El contraste entre determinados rasgos distintivos confiere valor a estas variantes. Estos elementos carecen de significado por sí mismos, pero permiten diferenciar construcciones lingüísticas de orden superior. El inventario de todas las oposiciones diferenciadoras constituye el sistema fonológico de una lengua (Jakobson, 1962, p. 231).

La fonología estudia los elementos fónicos del lenguaje, lo que la emparenta con la fonética. Sin embargo, a la manera de la gramática, se dedica a ello considerando su función en el sistema lingüístico. Comparte también características con la estilística porque refleja un estilo personal o colectivo en la expresión y puede tener un enfoque sincrónico o diacrónico (Alarcos Llorach, 1964, p. 35). De esta forma, por ejemplo, si tomamos los sonidos oclusivos velares [g] y [k], la sonoridad del primero lo distingue del segundo desde una perspectiva fonética articulatoria. Lo relevante de esta distinción es que tiene consecuencias semánticas, lo que, en nuestro ejemplo, sirve para reconocer la diferencia semántica entre *gata* y *cata*.

4.3 Fonética

Fonética articulatoria

La *fonética articulatoria* se refiere a la descripción del comportamiento de los órganos fonadores con respecto a la producción de los sonidos. A nivel articulatorio, Quilis (1993/2019, pp. 28-56) distingue dos rasgos privativos (consonántico-vocálico y sonoro-sordo) y dos graduales (*modo de articulación* y *punto de articulación*). Mientras que todos los autores coinciden en describir los mismos modos de articulación y rasgos privativos, no ocurre así con los puntos de articulación, lo que conduce a diferentes agrupaciones dependiendo de la precisión con la que definan cada posición.

En cualquier caso, necesitaremos describir el sistema fonológico completo del español para poder representar los versos observando todas las características que se infieren del texto escrito. Esto es, hemos de describir las propiedades fonológicas distintivas de cada sílaba y algunos de sus elementos prosódicos. En primer lugar, queremos representar los núcleos de vocales puras como elemento central de la sílaba, pero sin descuidar los sonidos vocálicos no silábicas o vocoides, por ser estas cruciales en el ajuste métrico con sinalefas y, excepcionalmente, sinéresis. Las consonantes aportan otros matices, como es el caso de las aliteraciones, en las que, precisamente,

la necesidad del sonido frente al texto escrito cobra mayor relevancia que nunca. Los rasgos prosódicos son asimismo importantes porque confieren al verso su patrón rítmico. La ortografía por sí sola no es suficiente para resolver este problema, ya que los acentos prosódicos no pueden desligarse de aspectos semánticos de la palabra, en ocasiones muy sutiles.

Quilis (1993/2019) señala siete posiciones: *bilabial*, *labiodental*, *dental*, *interdental*, *alveolar*, *palatal* y *velar*. Navarro Tomás (2004, sec. 12) añade a estas las *bilabiovelares*, que incluirían sonidos semiconsonánticos. Canellada y Madsen (1987), por su parte, tienen en cuenta cinco: *labial*, *dental*, *alveolar*, *palatal* y *velar*. Aquí labial y labiodental, así como dental e interdental se unificarían en sendas macrocategorías. Alarcos (1964) reduce los órdenes a cuatro: labial (donde se reúnen las realizaciones labial y labiodental), dental (los sonidos con realización interdental, dental y alveolar), palatal y velar.

Quilis	Navarro Tomás	Canellada y Madsen	Alarcos
Bilabial Labiodental	Bilabial Labiodental	Labial	Labial
Dental Interdental Alveolar	Dental Interdental Alveolar	Dental Alveolar	Alveolar
Palatal	Palatal	Palatal	Palatal
Velar	Velar Bilabiovelar	Velar	Velar

Cuadro 4.1: Puntos de articulación por autor.

Sintetizando las descripciones anteriores, tenemos sonidos consonánticos labiales, esto es, aquellos en los que no interviene la lengua directamente, sino que se pronuncian mediante la articulación de los labios; de ambos, en los sonidos bilabiales, o de uno solo en combinación con las piezas dentales frontales, en el caso de los labiodentales. La lengua sí interviene en la producción de los sonidos dentales, de forma que esta se aproxima a los dientes incisivos. No obstante, aquí también caben clasificaciones de segundo orden diferenciando las piezas dentales que toman parte. Si la lengua se emplaza sobre los incisivos superiores, hablamos en propiedad de un sonido dental, y de un sonido interdental si esta se ubica entre las superiores y las inferiores. Los sonidos alveolares son aquellos en los que la lengua se aproxima o toca la cresta alveolar, con una realización laminar en español. Los sonidos palatales se producen al actuar el cuerpo de la lengua contra el paladar duro. Finalmente, los sonidos velares requieren aproximar el dorso de la lengua al velo del paladar. En combinación con los labios, producirían un sonido *bilabiovelar*.

Según el modo de articulación, distinguiremos entre consonantes *oclusivas*, *nasales*, *fricativas*, aproximantes, *africadas*, *laterales* y *vibrantes* (Canellada y Madsen, 1987, pp. 33-35). Los sonidos oclusivos son aquellos en los que se interrumpe por completo el flujo de aire que atraviesa el tracto vocal. En los nasales, también se contiene el flujo de aire en el tracto vocal, al modo de los sonidos oclusivos, pero el descenso del velo del paladar deja que el aire escape sin constricciones por el conducto rinofaríngeo, que hace de resonador. Las consonantes fricativas son aquellas que se pronuncian haciendo fluir el aire a través del estrechamiento en el tracto fonador producido por la aproximación de dos elementos articuladores. Las aproximantes se asemejan a las fricativas, aunque con un cierre menor, de modo que hay autores que las incluyen dentro de aquellas (Alarcos Llorach, 1964, p. 161; Navarro Tomás, 2004, sec. 76), aunque otros especifican y distinguen los sonidos fricativos como *de resonancias altas* y los aproximantes como fricativos *de resonancias bajas* (Quilis, 1993/2019, p. 258). Las africadas se generan en dos fases, primero una oclusiva y después una fricativa. Las laterales requieren el cierre completo de la parte central de la lengua a la vez que permiten el flujo por uno o los dos lados de esta. Las vibrantes se producen mediante una secuencia de cierre momentáneo y apertura. Las vibrantes y las laterales se agrupan como *líquidas*, mientras que en el grupo de las fricativas se distinguen las *sibilantes*, o aquellas en que el estrechamiento adopta forma de canal.

Excurso I: fonética acústica

En el grueso de nuestro trabajo prima la fonología articulatoria, pues interesa como herramienta descriptiva para caracterizar las propiedades métricas del verso escrito. La fonética es apenas un complemento con el que matizar algunos aspectos relativamente menores. Por otra parte, no empleamos su modalidad acústica en ningún momento para describir un verso. Sería poco apropiado porque resulta difícil conocer a ciencia cierta las frecuencias exactas de las ondas y la duración de cada segmento. Por el contrario, las descripciones categóricas más o menos laxas que ofrece el modelo articulatorio se prestan para tratar los sonidos como idealizaciones. ¿Cuál es entonces el propósito de este subapartado?

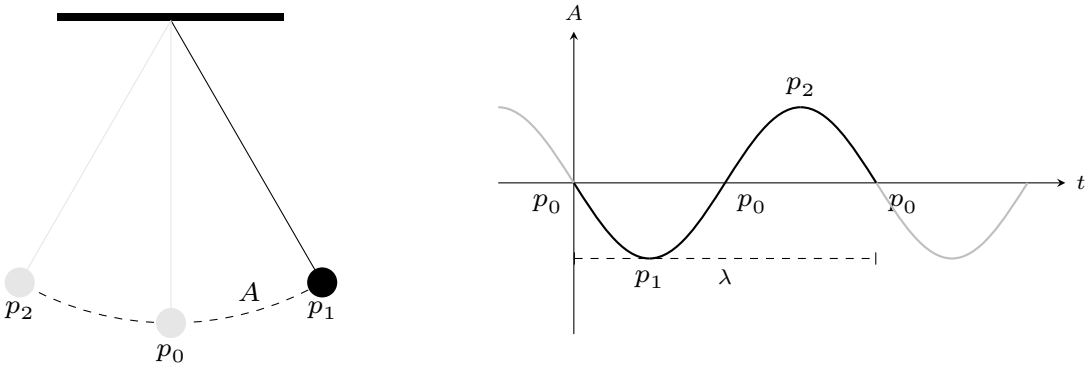
Se trata de recorrer el sendero en el sentido contrario. Cuando la representación idealizada es ambigua, necesitamos elegir una entre todas las realizaciones potenciales. El problema es que la teoría no cubre este aspecto, en tanto que no responde a una propiedad privativa de la lengua y, en muchos casos, tampoco del dialecto o siquiera del idiolecto, sino que la variación responde a menudo a las connotaciones mínimas que se quieran imbuir al verso, a la caracterización del personaje o cualquier otra causa imprevisible. Para decidirnos por una solución, considerando que nuestra elicitación

podría estar sesgada, decidimos observar la pronunciación de terceros con español como lengua materna. Nos concentraremos en unos pocos rasgos superficiales, ya que solo necesitamos identificar uniones vocálicas en este estudio, de manera que podamos precisar en qué sílabas se dan ajustes silábicos en el verso y qué vocales son nucleares. Esto es, dónde se rompe un diptongo o donde se fusionan dos vocales y de qué manera. Por otra parte, usamos este recurso en aquellas situaciones excepcionales en las que las indicaciones teóricas son insuficientes para deducir de forma apropiada la resolución.

El sonido, tal y como lo percibimos, es el resultado de la recepción por el aparato auditivo de las *ondas sonoras* transmitidas por el aire[59] y la decodificación que hace el cerebro de las señales nerviosas que le envía el oído con la información captada. Las ondas sonoras son el resultado de un movimiento vibratorio en un cuerpo. En el caso del sonido, este movimiento vibratorio lo hacen las partículas del aire, al modo de las bolas de un péndulo de Newton (Figura 4.1a). Para describir la onda, imaginaremos un péndulo: en reposo, este se encuentra en su posición de equilibrio p_0, pero, si le aplicamos una fuerza, oscilará perdiendo velocidad angular (ω) hasta que esta se anula en el punto de oscilación máxima p_1. La distancia lineal entre este y el de equilibrio se denomina amplitud (A). Desde el punto de oscilación máxima, el péndulo invierte el sentido y va aumentando ω hasta que alcanza su máximo en p_0. A partir de ahí, vuelve a ralentizarse hasta llegar a p_2, donde $\omega = 0$, y, de nuevo, vuelve a cambiar el sentido de su trayectoria. El tiempo (t) que tarda el péndulo en completar el *ciclo* p_0- p_1-p_0-p_2-p_0 se denomina *periodo* de vibración (λ) o longitud de onda. La magnitud inversa representa el número de ciclos por unidad de tiempo, se denomina *frecuencia* (f). Este movimiento se traduce en una onda sinusoidal (Figura 4.1b), donde los puntos más alejados de la posición de equilibrio son las crestas y los valles de la onda, que es precisamente la que toma el movimiento de las partículas de aire en la transmisión del sonido.

La *intensidad* (I) con la que percibimos el sonido en un punto es la cantidad de energía que atraviesa en sentido perpendicular la unidad de superficie colocada en ese punto por unidad de tiempo. En el sistema internacional esta magnitud se mide en $J/m^2 \cdot s$ o w/m^2; es proporcional al cuadrado de la amplitud y de la frecuencia. Es usual emplear también el *decibelio*, una medida logarítmica definida por la intensidad de un sonido determinado (I) y una umbral ($I_0 = 10^{-12} w/m^2$), tal como se define en

[59] El sonido se transmite en otros medios aparte del aire, pero, en tanto que lo relevante es el sonido de la lengua hablada, obviaremos otras posibilidades posibles para concentrarnos en la transmisión aérea, por ser esta la normal.



(a) Péndulo. **(b)** Onda simple.

Figura 4.1: Descripción de la onda sonora.

la ecuación 4.1.

$$\beta = 10 \cdot \log \frac{I}{I_0} \tag{4.1}$$

Los fundamentos de la onda simple son la base conceptual para abordar el sonido lingüístico. Este es siempre una combinación de varias de estas ondas porque ni el aparato fonador humano es capaz de producir un sonido «puro» ni el medio de transmisión que encontraremos en circunstancias normales provee las condiciones para una propagación uniforme. Por lo tanto, el sonido lingüístico es una *onda compleja* o *compuesta*, resultante de la adición de un número dado de ondas simples (Figura 4.2). La distribución de las amplitudes de cada frecuencia simple de una onda compleja constituye el *espectro* del sonido.

Las ondas compuestas son *periódicas* o *armónicas* y *aperiódicas* o *inarmónicas*. En el primer caso, la frecuencia de cada una de las ondas simples es un múltiplo entero de la onda simple de frecuencia más baja, que denominamos *frecuencia fundamental*. La onda correspondiente se conoce como *primer armónico* o tono fundamental. Al ser el resto de armónicos múltiplos del primero, la onda compleja se repite con la misma frecuencia que la onda simple de frecuencia más baja. En Figura 4.2 esta corresponde a la onda *a*. Las ondas *b* y *c* serían, respectivamente, el primer y el segundo armónico de aquella. Este tipo de onda lo tienen todos los sonidos vocálicos. En el caso de las ondas aperiódicas, como las resultantes de producir [f] o [s], la proporcionalidad no se cumple, por lo que el espectro del sonido no presenta la regularidad del tipo anterior. La impresión auditiva registrada de la frecuencia fundamental se denomina *tono* y determina la *entonación* a nivel oracional. El número y la cualidad de los armónicos, por su parte, configuran el *timbre* (Quilis, 1993/2019, p. 94; Clegg y Willis, 2018, p. 93).

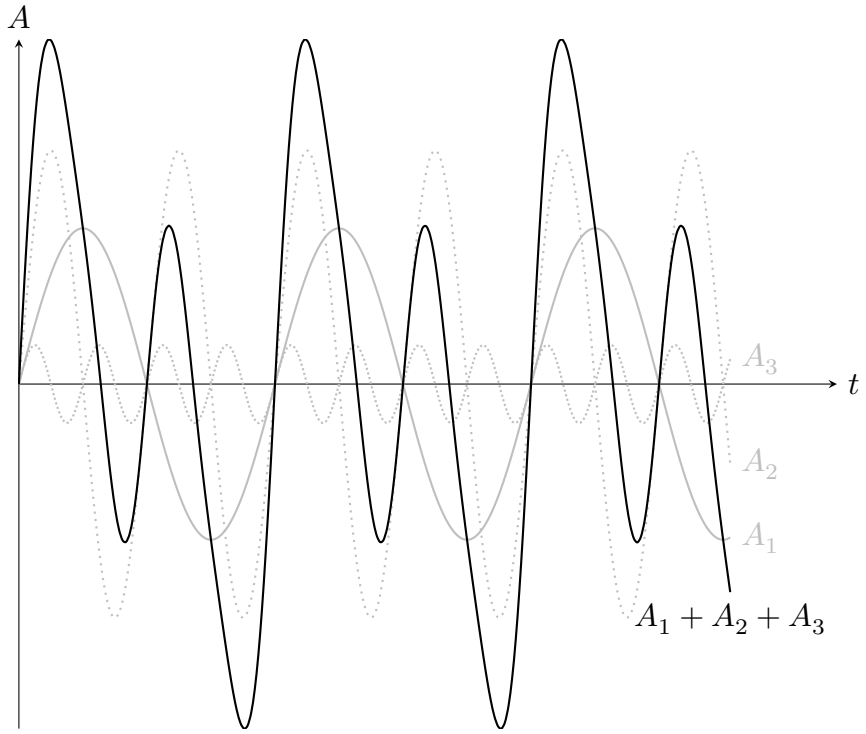


Figura 4.2: Onda compleja y las ondas simples que la componen.

Llamamos *resonancia acústica* al fenómeno en el que un cuerpo vibrante comunica esta vibración a otro cuerpo cercano. Se da cuando la frecuencia de la vibración del primer cuerpo coincide con la frecuencia natural de vibración del segundo (*resonador*). La extensión del rango de frecuencias efectivas del resonador se denomina *ancho de banda*. Aprovechando esta característica, es posible emplear el resonador como un *filtro* acústico, al amplificar solo determinadas frecuencias.

El sonido lingüístico se produce de tres maneras diferentes, tal como indica Quilis (1993/2019). Los sonidos sonoros son resultado del flujo de aire al atravesar la glotis puesta en vibración. Otros sonidos se originan al paso de la corriente de aire por constricciones, tal y como sucede en las fricativas. Por último, la obstrucción del paso a través del conducto vocal produce una compresión del aire que, al ser liberado, provoca una explosión, como en el caso de las consonantes oclusivas.

El tracto vocal es un tubo cerrado en un extremo y de apertura regulable en el otro. Esto es, en uno de los dos finales de este canal, se encuentra la laringe, que bloquea la salida del aire; en el otro, se hallan dos salidas, la boca y las fosas nasales, cuya apertura o cierre cambia el hablante a discreción, pudiendo cerrarla completamente o graduar la apertura mediante el movimiento de la lengua o los labios. Las cavidades actúan asimismo como resonadores, cuyas frecuencias de respuesta se alteran desplazando los órganos articulatorios, de forma que estos, en función de su posición,

filtren determinadas frecuencias y amplifiquen otras.

Los picos del espectro del sonido se denominan *formantes*. Se ha observado experimentalmente que la resonancia encuentra un rango de frecuencias de transmisión efectiva en el tracto vocal, con independencia del origen del sonido. Esta frecuencia se encuentra muy próxima al máximo del espectro del sonido íntegro, por lo que, aunque sean diferentes desde un punto de vista conceptual, la frecuencia de resonancia y la del formante pueden ser consideradas equivalentes a efectos prácticos (Fant, 1960/1970, p. 20).

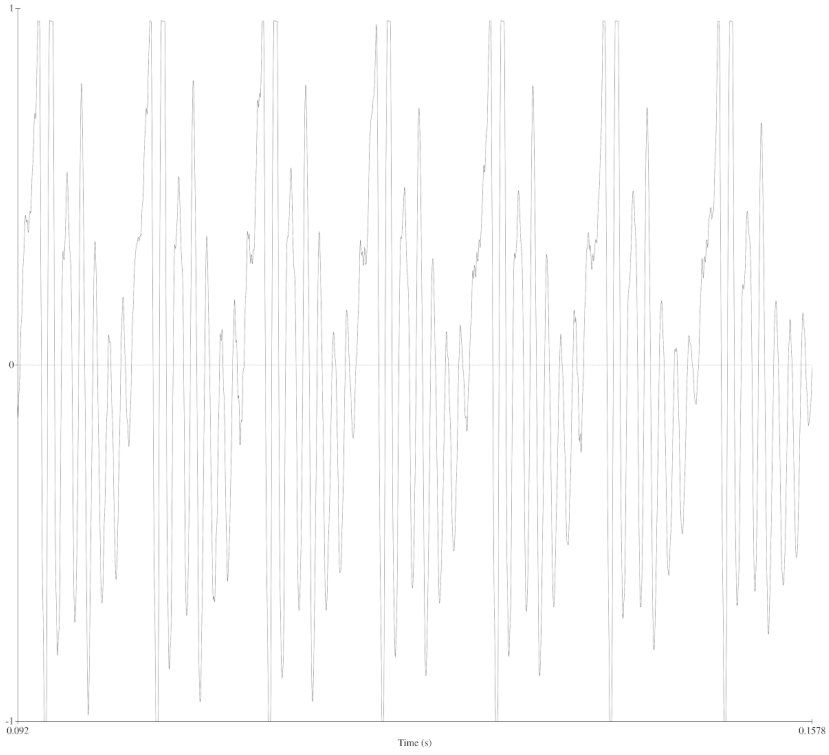
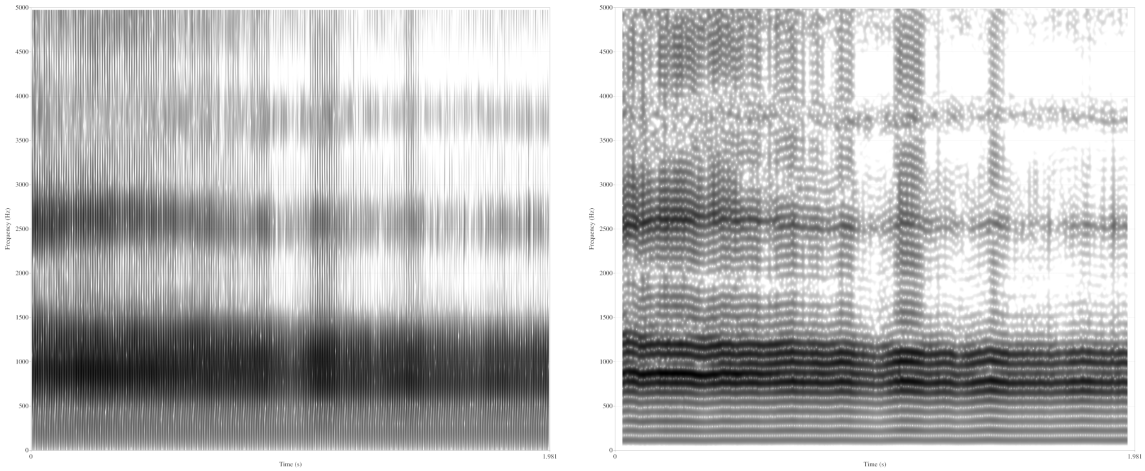


Figura 4.3: Onda de [aː].

Con esto, emprendemos la tarea pertrechados de los medios técnicos apropiados (Boersma y Weenink, 2022): analizar un *sonograma* (Figura 4.4), que es el gráfico que representa las variaciones de la frecuencia en el eje de ordenadas a lo largo del tiempo en el de abscisas. El sonograma se filtra según un determinado rango de frecuencias para examinar diferentes aspectos del sonido. Así, un sonograma *de banda estrecha* como el de Figura 4.4b es más adecuados para observar los armónicos de la onda, más difíciles de discernir en uno *de banda ancha* como el de Figura 4.4a, que, por el contrario, ofrece una perspectiva más amplia del sonido (Clegg y Willis, 2018, pp. 95-96). Las franjas más oscuras corresponden a los formantes en el sonograma de banda estrecha o a sus agrupaciones en el de banda ancha. En el ejemplo, que representa el sonido /a/, la frecuencia fundamental se encuentra alrededor de los 108,65

Hz, el primer formante F_1 tiene una frecuencia ocho veces mayor, el segundo F_2 diez veces mayor, el tercero F_3 veinticuatro veces mayor y el cuarto F_4 veinticinco veces más.



(a) Banda ancha. **(b)** Banda estrecha.

Figura 4.4: Espectrogramas de [aː].

Estos formantes, de forma especial el primero y el segundo, facilitan la distinción entre las vocales. El primer formante F_1 viene definido por la manera de articulación, ya que la frecuencia de este tiene una relación proporcional con la apertura bucal y, por el contrario, cuanto más cerrada se encuentre la boca, tanto más bajo será el formante. El segundo formante lo determina el lugar de articulación, de manera que cuanto más anterior sea la articulación, más alta será la frecuencia y viceversa, cuanto más posterior sea el sonido, más baja la frecuencia de F_2. De esta manera, los valores de F_1 de las vocales españolas serían aproximadamente los representados en la Tabla 4.2 (Clegg y Willis, 2018, p. 101).

Vocal	$\mathbf{F_1}$	$\mathbf{F_2}$
[i]	315	2174
[e]	501	1895
[a]	762	1300
[o]	501	966
[u]	315	836

Cuadro 4.2: Formantes F_1 y F_2 de las vocales (Hz).

La relación con el trapecio articulatorio se hace más evidente si disponemos estos valores en un gráfico. Como muestra Figura 4.5, la posición de los sonidos según las frecuencias de sus dos primeros formantes está relacionada con el trapecio articulatorio.

Para interpretar sonidos consonánticos no basta con identificar el modo y lugar de articulación, sino que una descripción precisa requiere conocer el estado de las cuerdas vocales; esto es, hallar el grado de intervención de este órgano porque su vibración determina si la consonante es *sonora* o *muda*.

Las oclusivas se identifican por presentar un periodo de ausencia de energía durante la fase productiva, en el que no hay expulsión de aire. En las fricativas, la energía se extiende por el gráfico a lo largo del tiempo de producción. Las africadas están caracterizadas por la producción en dos fases, una primera oclusiva seguida de otra de liberación africada. Las consonantes nasales y laterales, conocidas también como *sonorantes*, presentan formantes al modo de los sonidos vocálicos, aunque estos no son tan intensos. Las vibrantes se distinguen por las interrupciones de la onda sonora, poseyendo varias las múltiples y una las simples (Clegg y Willis, 2018, pp. 101-102).

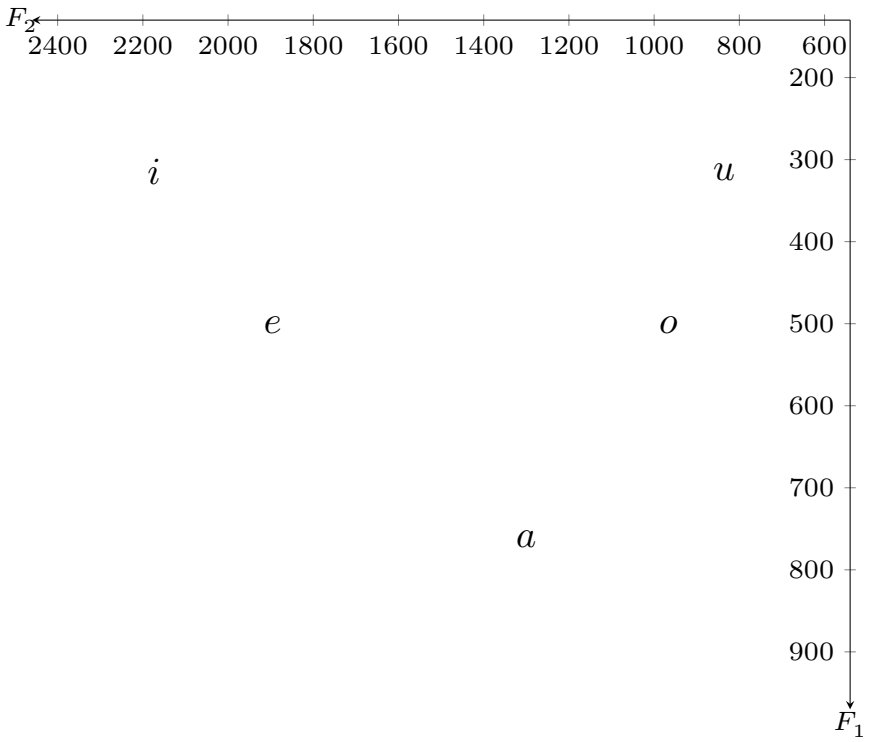


Figura 4.5: Formantes F_1 y F_2 de las vocales (Hz).

La posición de la articulación varía con el modo en que esta se produce. Para las oclusivas, hay que observar las transiciones entre formantes de las vocales en contacto con la consonante, ya que estas tienen formas propias para cada sonido. En el caso de las fricativas, hay que tener en cuenta además la energía acústica empleada en su producción (Clegg y Willis, 2018, p. 102). Determinar la sonoridad de la consonante presenta menos dificultades. La vibración de las cuerdas vocales se aprecia en el

sonograma de banda ancha, que muestra estrías verticales en las sonoras, de las que carecen las sordas (Clegg y Willis, 2018, p. 102).

4.4 El sistema fonológico

La unidad fonológica mínima es el *fonema*, que Jakobson (1962, p. 231) define como «un conjunto de propiedades sonoras concurrentes usadas en una lengua dada para distinguir palabras de significado distinto». Esto es, los fonemas deben servir al hablante para discriminar entre una palabra y otra según el principio de pertinencia; esto es, lo que distingue entre las propiedades esenciales y las contingentes en la diferenciación de significados. En palabras de Martinet (1965/1970, p. 45), «le principe de pertinence nous permet de distinguer ce qui, dans chaque langue ou chaque usage, est essentiel parce que distinctif, et ce qui est contingent, c'est-à-dire déterminé par le contexte ou diverses circonstances».

Un fonema, sin embargo, no corresponde a un sonido único, ya que admite varias realizaciones fonéticas con diferentes sonidos. Estas dependen del estilo de habla o del contexto fonético concreto en el que aparecen. La diferencia entre sonidos responde a factores externos y no influye en la distinción de significados, por lo que esos sonidos no son sino variantes de un fonema dado (Jakobson, 1962, p. 231) o *alófonos*. Así, los sonidos [β] y [b] son realizaciones diferentes del mismo fonema /b/, y se selecciona una u otra dependiendo del contexto fonético donde se produzca el fonema. Por lo tanto, aunque *abocar* se pronuncie [aβoˈkaɾ] y *embozar* [emboˈθaɾ], la diferencia no se percibe como un sonido diferente, por lo que hablamos de un único fonema /b/.

Esto obliga a establecer criterios para distinguir entre un fonema y una realización alternativa de este. Trubetzkoy (1939, pp. 42-46) propone para ello las siguientes cuatro reglas:

I «Cuando dos sonidos de la misma lengua se dan exactamente en el mismo contexto fónico y pueden intercambiarse sin provocar una diferencia en el significado intelectual de la palabra, esos dos sonidos son solo dos variantes facultativas de un único fonema» (Trubetzkoy, 1939, p. 42, traducción propia).

II «Cuando dos sonidos se dan exactamente en la misma posición fónica y no pueden ser intercambiados sin alterar el significado de la palabra o hacerla irreconocible, estos dos sonidos son realizaciones de dos fonemas distintos» (p. 44; traducción propia).

III «Cuando dos sonidos acústica o articulatoriamente emparentados de una lengua no se dan nunca en el mismo contexto fónico, se consideran como variantes combinatorias del mismo fonema» (pp. 44-45; traducción propia).

IV «Dos sonidos, aunque cumplan las condiciones de la regla III, no deben ser considerados variantes del mismo fonema en la lengua en cuestión si estos pueden darse en posiciones adyacentes, esto es, como grupo fónico de una secuencia de sonidos en aquellas posiciones en las que, también, uno de los sonidos también puede aparecer aislado» (p. 46; traducción propia).

Estas reglas se ven mejor mediante casos prácticos. Verbigracia, la regla I se aplica a los sonidos [h] y [s] del español meridional como alófonos de /s/. Vemos que [ˈkasah] y [ˈkasas] son formas intercambiables de realizar *casas*. El ejemplo que dimos arriba de /ˈgata/ y /ˈkata/ es representativo de la regla II. La regla III alude a casos como el de los alófonos [d] y [ð] del fonema /d/, pues el primer sonido únicamente aparece detrás de pausa, consonante nasal o lateral, mientras que el segundo lo hace en el resto de casos, no haciéndolo en los que afectan al primero. Así, tenemos [esˈpalda] y [esˈpaða]. La regla IV es relevante en ejemplos como el de la semiconsonante [w] y la semivocal [u̯] como variantes combinatorias de /u/, ya que, además de cumplir la regla III, no aparecen en posición contigua a [u].

Los sonidos lingüísticos, empero, en muy contadas ocasiones aparecen aislados. Por el contrario, suelen ser el resultado de la combinación articulada de una secuencia. Esto no es impedimento para que se dé el caso opuesto. Lo que en apariencia son dos fonemas individuales contiguos pueden no serlo, sino constituir solamente uno. Por ello, se hace necesario distinguir entre fonemas simples adyacentes y fonemas complejos constituidos de una serie de sonidos. Trubetzkoy propone las siguientes seis reglas para hacerlo:

I «Solo puede ser considerado como realización de un fonema individual la unión de sonidos cuyos componentes no se reparten entre dos sílabas en la lengua en cuestión» (Trubetzkoy, 1939, p. 50, traducción propia).

II «Solo puede ser considerado como realización de un fonema individual una unión de sonidos producida por un movimiento articulatorio homogéneo o a través de la disolución progresiva de un complejo articulatorio» (p. 51; traducción propia).

III «Solo puede ser considerado como realización de un fonema individual una unión de sonidos cuya duración no sobrepase la duración de la realización de otros fonemas de la lengua en cuestión» (p. 53; traducción propia).

IV «Una unión de sonidos potencialmente monofonemática —esto es, que cumple las reglas I, II y III— debe ser considerada realización de un único fonema si se trata como un fonema individual[60], esto es, si se da en aquellas posiciones en las que grupos fonemáticos no pueden ocurrir en la lengua en cuestión» (p. 53; traducción propia).

V «Una combinación de sonidos que cumpla las reglas de I a III debe ser considerada la realización de un fonema si esto produce simetría en el sistema fonológico»[61] (Trubetzkoy, 1969, p. 59, traducción propia).

VI «Si un componente de una unión fonemática potencial no pude ser evaluada como variante combinatoria de algún otro fonema de la lengua en cuestión, la unión de sonidos debe ser interpretada como un fonema individual» (Trubetzkoy, 1939, p. 56, traducción propia).

Podemos aclarar estos puntos con algunos casos ilustrativos. Vemos que en el fonema /ʧ/ del español hay una obstrucción que se libera para deshacerse luego en la fricativa. Sin embargo, sus componentes [t] y [ʃ] no se dividen entre dos sílabas como sí ocurre en otros idiomas como el alemán. Compárense las palabras de sendas lenguas *hucha* y *Lautschrift*, pronunciadas respectivamente como /ˈu.ʧa/ y /ˈlaʊt.ʃrɪft/. De esta manera, atendiendo a las reglas I y II, [tʃ] constituirían un único fonema complejo /t͡ʃ/ en español pero dos fonemas simples /t/ y /ʃ/ en alemán. El equivalente a estos fonemas germanos lo tenemos en castellano con /ks/, la pronunciación de *x*. En *facsímil*, por ejemplo, se distribuye en dos sílabas como /fak.ˈsi.mil/. Con respecto a la regla III, se ve en *preeminente* que no se interpreta como /preːmi.ˈnen.te/, pues contravendría la homogeneidad silábica del español —sin olvidar que la cantidad no es contrastiva en esta lengua—. Sobre la regla IV, Trubetzkoy propone el ejemplo de palabras alemanas como *Zwei* o *Pflanze*, que parecen contravenir la regla de que el alemán solo permite en ataque los segmentos de tres consonantes [ʃtr], [ʃpl] y [ʃpr]; de esta manera, parece obligado evaluar los grupos [ts] y [pf] en ataque como fonémicos.

Partiendo de lo expuesto, recabamos el inventario de fonemas de una lengua dada. Para ello necesitamos, además, determinar el *contenido fónico* de cada uno de ellos, esto es, sus propiedades distintivas. Dicho de otra manera, se requiere encontrar aquellas características comunes a todas las variantes de un mismo fonema (Trubetzkoy, 1939, p. 59). Por ejemplo, si describimos el fonema /d/ como plosivo, sería adecuado

[60]La tautología es del original: «Eine potentiell monophonematische Lautverbindung (d.i. den Forderungen der Regel I—III entsprechende) Lautverbindung muß als Realisation eines einzigen Phonems gewertet werden, wenn sie als Einzelphonem behandelt wird [...]».

[61]He recurrido a otra edición (Trubetzkoy, 1969) porque la regla V no aparece en el texto original (Trubetzkoy, 1939), pasando de la IV a la elaboración de la V, pero sin llegar a enunciarla.

para su realización [d], pero excluiría otras realizaciones tan válidas como el aproximante [ð]. Por el contrario, resultaría inadecuado describirla solo como dental porque, aunque ambos alófonos lo cumplen, también lo hacen realizaciones de otros fonemas, como [t] y [θ]. Es necesario, por lo tanto, indicar los rasgos relevantes: en el caso de /d/, dental sonoro.

El nivel fonológico constituye el material principal que conforma los cimientos sobre los que se asienta la métrica. Sin embargo, no hay que dejar de lado del todo algunos aspectos de la lengua que hunden sus raíces en la fonética, de forma muy especial en el territorio de las vocales, donde la propiedad silábica del fonema representa un papel protagonista en la medida del verso. Debemos, pues, dilucidar el modo en el que el contexto fónico afecta a los fonemas individuales si esto altera la sustancia métrica del verso. Debemos hacer notar que esta disolución de las lindes entre los niveles fonético y fonológico no es extraña a la disciplina. Tanto Quilis (1993/2019) como Alarcos Llorach, por ejemplo, clasifican hasta cierto punto los fonemas según criterios fonéticos. Este último lo justifica arguyendo que

> hay que tener presente la diferencia entre fonología y fonética, a pesar de que en la práctica fonológica se utiliza la nomenclatura fonética por simplicidad. Esta está basada en las características articulatorias de los sonidos, aunque haya entre sus términos algunos puramente impresionísticos; por ser bien conocida, ofrece ventajas para la comprensión el aprovecharla en fonología. (Alarcos Llorach, 1964, p. 54)

Una vez establecido el marco en el que definiremos el sistema fonológico y hechas las debidas prevenciones, queda por definir una serie mínima de rasgos identificativos u oposiciones. Estos rasgos no solo deben distinguir a los términos de la oposición, sino que tienen que representar una cualidad común a ambos (Trubetzkoy, 1969, pp. 68-87). Esto servirá para organizar el repertorio de todos los fonemas disponibles en una lengua en un sistema fonológico.

Atendiendo a la relación con el sistema fonológico, se distingue entre *oposiciones bilaterales* y *multilaterales*[62], y *oposiciones proporcionales* y *aisladas*. Llamamos *oposiciones bilaterales* a aquellas en las que el *tertium comparationis* es particular de los dos términos y no se da en otro término del mismo sistema fonológico. Las *oposiciones multilaterales*, por su parte, son aquellas en las que la base de la comparación se da en otros términos. Quilis (1993/2019, p. 36) ejemplifica la relación bilateral con los fonemas /k/-/x/ del español, cuyas propiedades compartidas (orales, velares,

[62]También se conocen como *unidimensionales* y *pluridimensionales*, respectivamente.

sordas) no aparecen juntas en ningún otro fonema, de la misma manera que /e/-/i/ (vocales, sonoras, anteriores). Como ejemplo de multilateralidad, Quilis cita /b/-/d/, ya que sus rasgos comunes (consonantes, oclusivas, orales, sonoras) se dan también en /b/-/g/ o /d/-/g/.

En las *oposiciones proporcionales*, la relación que se da entre sus términos se encuentra en los de otra oposición, mientras que los términos de las *oposiciones aisladas* tienen una relación única en el sistema fonológico. De esta manera, /p/-/b/ es proporcional porque la misma relación se da en /t/-/d/ y /k/-/g/, mientras que la oposición /r/-/l/ es aislada, ya que la relación no aparece en ningún otro par del sistema.

Según las relaciones entre los términos, las oposiciones son *privativas*, *graduales* o *equipolentes*. Las *oposiciones privativas* son de naturaleza binaria y se caracterizan por la presencia o ausencia de una marca. El término que presenta la marca se denomina *término marcado* y, de forma análoga, el que no la presenta se llama *no marcado*. Por ejemplo, si consideramos el rasgo sonoridad, en la oposición /p/-/b/, /b/ será el término marcado (sonoridad: +) y /p/ el no marcado (sonoridad: −). Por el contrario, las *oposiciones graduales* presentan la distinción como un grado no binario. La apertura de las vocales es un ejemplo de este tipo de oposiciones. Así, /i/ es más cerrada que /e/, que a su vez es más cerrada que /a/. Las *oposiciones equipolentes* son aquellas cuyos miembros son equivalentes. Esto es, no han de ser consideradas ni privativas ni graduales, por ejemplo /p/-/k/ o /i/-u/.

Las oposiciones fonológicas también se caracterizan mediante su poder distintivo en *constantes* o *neutralizables*. Las primeras se realizan siempre sin que tenga relevancia la posición que ocupen, mientras que las segundas no se dan en ciertas posiciones. En este último caso, la posición donde se mantiene la distinción se denomina *de pertinencia*, mientras que aquella en la que se neutraliza se llama *de neutralización*. Por ejemplo, /s/-/θ/ serían constantes en *español de Castilla*[63] porque no pierden capacidad distintiva según su posición, como se ve en *casa* /ˈka.sa/ y *caza* /ˈka.θa/ o *hez* /eθ/ y *es* /es/. Por el contrario, /r/-/ɾ/ solo funcionan en posición prenuclear, de forma que mientras que *perro* /ˈpe.ro/ y *pero* /ˈpe.ɾo/ son distintas fonológicamente, /baɾ/ y /bar/ son dos formas de pronunciar la misma palabra *bar*.

Los rasgos distintivos varían según desde qué perspectiva fonética se aborde la fonología. Quilis (1993/2019, p. 44) menciona la *fonética sincrónica*, *diacrónica*, *articulatoria*, *acústica*, *auditiva* y *psicológica*. Este estudio se vale ante todo de la fonética articulatoria como marco descriptivo del sistema fonológico. Sin embargo, dadas las

[63]Por comodidad, me referiré al español de Castilla para denotar las variedades peninsulares septentrionales que se acercan a lo que, con todas las precauciones, podríamos llamar *español estándar*.

dificultades que presentaría examinar físicamente a los hablantes para controlar la producción del sonido, nos apoyaremos en la fonética acústica para contrastar muestras de sonidos mediante la observación de sus espectrogramas. Esto permite, por un lado, obtener de manera sencilla las muestras como archivos de audio y, por otra parte, examinarlos a fondo con el *software* adecuado[64]. Así pues, habremos de establecer estos marcos fonéticos descriptivos.

Los rasgos distintivos en se dividen en *prosódicos* e *intrínsecos* o *inherentes*. Todos los fonemas poseen estos últimos, pero los primeros son exclusivos de los fonemas constituyentes de núcleo silábico (Quilis, 1993/2019, p. 111).

Desde un punto de vista acústico, se distinguen diferentes rasgos. Alarcos Lorach (1964, p. 85) propone *vocálico/no vocálico* y *consonántico/no consonántico*, que se basan en el formante generador de la onda sonora y la presencia o ausencia de obstáculo en el canal bucal. Permiten discriminar entre fonemas vocales, consonantes, líquidos y glotales; *denso/difuso*, basándose en la proporción de las cavidades bucofaríngeas; *grave/agudo*, junto a los rasgos precedentes, revela la ubicación de los fonemas, *bemolizado/normal* y *sostenido/normal*, que dependen del timbre del resonador bucal y distinguen series paralelas de sonidos con timbre modificado; *nasal/oral*, según se emplee la nariz o la boca como cavidad resonadora; *tenso/flojo*, en función de la tensión de los órganos fonadores, que indica propiedades prosódicas de las vocales y distingue entre consonantes fuertes y débiles; *sonoro/sordo*, que viene determinado por la vibración de la glotis y actúa en combinación con el rasgo precedente; *continuo/interrupto* y *estridente/mate*, que dependen de si el ataque es abrupto o hay una interrupción en el flujo del aire, así como de la aproximación de los órganos fonadores y *recursivo/infraglotal*, determinado por la existencia de cierre glotal al final del sonido y diferencia el modo en que la glotis interviene en la expulsión del aire.

Quilis (1993/2019, pp. 165-167), por su parte, distingue para los sonidos vocálicos entre *compactos* y *no compactos*, siendo *difuso* y *no difuso* propiedades de los últimos. Asimismo, desdobla grave/agudo en *grave* y *no grave*, pudiendo aquellos que se hallan en el segundo caso subdividirse en *agudo* y *no agudo*. Además, hace distinciones más finas, añadiendo *denso/no denso*, *difuso/no difuso*, *grave/no grave* y *agudo/no agudo*.

[64] En nuestro caso, Praat (Boersma y Weenink, 2022).

4.5 Neutralización

El fenómeno de la neutralización consiste en la desaparición del contraste fonémico en ciertos entornos fonológicos, bien por la pérdida de una característica propia del fonema o por la adquisición de otra ajena a este.

Alarcos Llorach (1964, pp. 97-99) describe dos tipos de neutralización, una *condicionada por el contexto* y otra *exigida por la estructura*, que es *interna* del sistema. Mientras que la primera ocurre como respuesta a fonemas cercanos, como en el caso de una consonante nasal ante otra consonante, la segunda la determina la posición, como en las vibrantes a final de sílaba. La neutralización condicionada es *disimilativa* si el fonema pierde su carácter distintivo por la influencia de otro con un rasgo distintivo semejante o *asimilativa* si lo pierde por contacto con otros fonemas carentes de este. La neutralización interna, por su parte, se divide entre la *centrífuga* y la *reductiva*. La neutralización centrífuga es aquella que ocurre en los límites de palabra o sílaba, mientras que la reductiva es la que se da en todas las sílabas menos en la culminante o acentuada.

Este suele darse con mayor frecuencia en la distensión silábica de final de sílaba (Alarcos Llorach, 1964, p. 180), lo que, a efectos prácticos, reduce su alcance a las consonantes que suelen darse en tal posición. En español, la neutralización es especialmente notable en coda. En lo que a nuestro estudio respecta, la asimilación en coda en la rima podría determinar la diferencia entre consonancia y asonancia. Esto, sin embargo, solo sucedería en casos tan excepcionales que podemos pasarlo por alto. No obstante, hemos de mencionarlo como una posibilidad teórica.

4.6 Fonemas del español

Sonidos vocálicos

Desde las gramáticas más primitivas de Grecia y la India, los estudiosos del lenguaje han divido los sonidos entre consonantes y vocales. Esta clasificación no es arbitraria, ya que se ha comprobado que hay diferencias articulatorias que, a su vez, tienen consecuencias en el plano acústico. Las vocales se producen por una acción articulatoria fuerte, en la que se alcanza una máxima apertura, mientras que una acción articulatoria débil produce el efecto contrario (Quilis, 1993/2019, pp. 141-143). En lo que concierne a este trabajo, los sonidos vocálicos son fundamentales, ya que con ellos se contarán las sílabas. Por esta razón debemos identificarlos y caracterizarlos. Sin embargo, aunque todos los núcleos silábicos son vocálicos, no todas las vocales son

necesariamente silábicas, por lo que será preciso, en ese caso, no solo caracterizarlas, sino identificar sus propiedades físicas para contar con criterios para resolver casos dudosos.

Podemos caracterizar los sonidos vocálicos, al igual que los consonánticos, por el modo y por el punto de articulación. El modo de articulación se refiere a la apertura que deja la lengua en la cavidad bucal, mientras que el punto de articulación, al contrario que las consonantes, no es respecto a cierres o puntos de contacto, sino a la región palatal a la que se aproxima el dorso lingual. Por el modo de articulación, distinguimos entre vocales altas, medias y bajas.

La lengua castellana se caracteriza por la homogeneidad del repertorio vocálico en todas sus variedades dialectales. Así se ha descrito tradicionalmente y la fonética moderna tampoco ha encontrado más vocales a nivel fonológico, por lo que limita a describir las diferentes realizaciones fonéticas de los fonemas. En la conciencia lingüística del hablante existen tan solo cinco vocales: *a*, *e*, *i*, *o* y *u*. En este caso, sí existen alófonos que dependen sobre todo del contexto fónico del segmento dado (Navarro Tomás, 1946, p. 32). Las realizaciones de /a/, /e/ y /o/ son siempre vocálicas puras y son capaces de formar núcleo vocálico, mientras que, como detallamos abajo, /i/ y /u/ se realizan como vocales puras cuando son tónicas o cuando son átonas, pero no están en contacto con otra vocal (Alarcos Llorach, 1964, pp. 152-153).

Si nos situamos en el trapecio bucal, el fonema /i/ es articulatoriamente *alto* y el más *anterior* del repertorio español. Es acústicamente *vocálico*, *no consonántico*, *no compacto*, *difuso*, *no grave* y *agudo*. Presenta dos realizaciones en distribución complementaria. La primera es la oral regular [i] (⟨piso⟩ [ˈpiso]). La segunda es la nasalizada [ĩ], cuando /i/ se encuentra entre pausa y consonante nasal (⟨instante⟩ [ĩnsˈt̪ãn̪te]) o entre dos consonantes nasales (⟨mimo⟩ [ˈmĩmo]) (Quilis, 1993/2019, pp. 168-169).

También localizada articulatoriamente en una posición anterior, aunque a una altura *media*, se encuentra el fonema /e/. Acústicamente, se describe como vocálico, no consonántico, no compacto, *no difuso*, no grave y agudo. Tiene también dos realizaciones, una oral regular [e] (⟨peso⟩ [ˈpeso]), y otra nasal [ẽ] cuando /e/ se encuentra entre pausa y consonante nasal (⟨encono⟩ [ẽŋˈkono]) o entre dos consonantes nasales (⟨memo⟩ [ˈmẽmo]) (Quilis, 1993/2019, p. 169).

El fonema vocálico más *central* y *bajo* es /a/. Acústicamente, es vocálico, no consonántico, *compacto*, *denso*, no grave y *no agudo*. También /a/ presenta dos alófonos en distribuciones complementarias, uno nasal [ã] cuando se halla entre una pausa y una consonante nasal (⟨antena⟩ [ãn̪ˈt̪ena]) y otra regular [a] (⟨casa⟩ [ˈkasa]) en el resto (Quilis, 1993/2019, p. 169).

El fonema /o/ es medio y *posterior*. Conforme a sus propiedades acústicas, es vocálico, no consonántico, no compacto, no difuso, *grave*. De la misma manera, /o/ tiene dos realizaciones, una oral regular [o] (⟨vaso⟩ ['ßaso]) y otra nasal [õ], cuando /o/ se encuentra entre pausa y consonante nasal (⟨oncología⟩ [õŋkoloˈxia]) o entre dos consonantes nasales (⟨nona⟩ ['nõna]) (Quilis, 1993/2019, p. 169).

Por último, el fonema /u/ es articulatoriamente alto y posterior. Se caracteriza acústicamente por ser vocálico, no consonántico, no compacto, difuso, grave. Al igual que el resto de vocales, /u/ también tiene dos realizaciones en distribución complementaria, una oral regular [u] (⟨huso⟩ ['uso]) y otra nasal [ũ], cuando /u/ se encuentra entre pausa y consonante nasal (⟨ungüento⟩ [ũŋˈgwe̤nto]) o entre dos consonantes nasales (⟨mundo⟩ [ˈmũn̪do]) (Quilis, 1993/2019, p. 170).

Esto se resume en que, considerando la apertura, el español tiene dos vocales *altas* o *cerradas*, aquellas en que la lengua se aproxima al paladar duro o al paladar blando. Estos son los fonemas /i/ y /u/. También cuenta con dos vocales *medias*, en las que la lengua se separa de la bóveda del paladar para posicionarse en medio de la cavidad bucal. Estas son /e/ y /o/. El español solo posee una vocal *baja* o *abierta*, esto es, cuando se produce una apertura máxima al alejarse la lengua del paladar en el mayor grado colocándose en la parte inferior de la cavidad bucal. Esta es /a/ (Quilis, 1993/2019, pp. 146-147).

Según el punto de articulación, el español dispone de vocales anteriores, centrales y posteriores. Las vocales anteriores, aquellas que se producen cuando la lengua se halla en la parte frontal de la cavidad bucal, son /i/ y /e/. En español solo aparece /a/ como vocal media, en la que el dorso de la lengua se halla en la región del paladar medio. Las vocales posteriores del español, en las que el posdorso de la lengua se acerca a la región posterior de la cavidad bucal, son /u/ y /o/ (Quilis, 1993/2019, pp. 147-148). En lo relativo a sus cualidades acústicas, todas las vocales del repertorio fonológico del español son *vocálicas* y *no consonánticas*.

	Anterior	**Central**	**Posterior**
Cerrada	i		u
Media	e̬		o̬
Abierta		ä	

Cuadro 4.3: Fonemas vocálicos del español.

Así pues, la lengua española cuenta con cinco vocales que se distinguen entre sí por oposición: la anterior cerrada /i/, la anterior media /e/, la central abierta /a/, la posterior cerrada /u/ y la posterior media /o/. Todas ellas aparecen indistintamente en posición tónica (1) y átona (2).

Rasgos	i	e	a	o	u
Vocálico/no vocálico	+	+	+	+	+
Consonántico/no consonántico	-	-	-	-	-
Compacto/no compacto	-	-	+	-	-
Difuso/no difuso	+	-	-	-	+
Grave/no grave	-	-	-	+	+
Agudo/no agudo	+	+	-	-	-

Cuadro 4.4: Rasgos acústicos de las vocales del español.

Aunque desde una perspectiva fonológica, determinados alófonos de la lengua estándar representan fonemas por sí mismos en algunos dialectos, no nos ocuparemos de esas variedades diatópicas; nos centraremos en la estándar. Esto es, no consideraremos que, por ejemplo, en algunas variedades meridionales, la cualidad vocálica tiene valor fonológico y la apertura de /e/ es contrastiva. En esas variedades, [ˈpaðɾɛ] no es equivalente a [ˈpaðɾe] porque, mientras que esta última realización se estaría refiriendo al singular ⟨padre⟩, la primera lo haría al plural ⟨padres⟩. Por el contrario, en el español estándar [ɛ] y [e] son alófonos del fonema /e/, por lo que ambas realizaciones corresponderían a ⟨padre⟩ (Alarcos Llorach, 1964, pp. 148-150; Quilis, 1993/2019, pp. 173-176). Por esto, sintetizamos el sistema vocálico ideal en los cinco fonemas /a/, /e/, /i/, /o/ y /u/, sin contemplar variaciones no distintivas en la cualidad.

(1) /ˈdado/ *dado*
/ˈdedo/ *dedo*
/ˈdido/ *Dido*
/ˈdodo/ *dodo*
/ˈdudo/ *dudo*

(2) /daˈto/ *dató*
/deˈxo/ *dejó*
/diɾiˈxjo/ *dirigió*
/doˈljo/ *dolió*
/duˈdo/ *dudó*

A consecuencia de lo expuesto, los fonemas /a/, /e/, /i/, /o/ y /u/ se relacionan biunívocamente con los grafemas ⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩ y ⟨u⟩, salvando las excepciones de los dígrafos *gu-* y *qu-*. En esos casos, se trata de una sola unidad cuyos componentes no tienen realización fonológica propia independiente. Sin embargo, esto no es suficiente en el análisis métrico porque la cuenta de vocales silábicas es lo que determina en último término el cómputo silábico del verso.

Dicho de otra manera, únicamente los fonos silábicos son relevantes en el cómputo de sílabas del verso con independencia de su función fonológica. Por ejemplo, la palabra ⟨hueco⟩ se analizaría fonológicamente como /ˈue.ko/. Si no atendemos al valor silábico de las vocales, tendríamos tres sílabas alrededor de los núcleos /u/, /e/ y /o/. Sin embargo, este no es el caso, pues solo son silábicas /e/ y /o/. Aquí, el fonema /u/ se une a /e/ como semiconsonante no silábica, formando un diptongo en [ˈwe.ko]. De la misma manera, ⟨hoy⟩ se transcribiría /oi/, a pesar del diptongo en el que /i/ se realiza como una semivocal [o̯i̯]. Si vamos a ocuparnos de las sílabas, bien podemos añadir vocoides no silábicos a nuestro repertorio, las semivocales [u̯] e [i̯] y las semiconsonantes [w] y [j] como realizaciones en posición no vocálica de ⟨u⟩ e ⟨i⟩, respectivamente. Como veremos más adelante, en métrica, esta propiedad no se circunscribe a los diptongos naturales del español, sino que, estando dos vocales cualesquiera en contacto, una de ellas pierde su condición silábica en determinadas circunstancias y altera el número de sílabas del verso.

Respecto a las diferencias fonéticas en la realización de los alófonos vocálicos, estas no influyen en la constitución métrica esencial del verso ni en apoyos secundarios, como la armonía vocálica. No obstante, la ausencia de relevancia métrica no implica que no aporten matices al verso. Cabe suponer que diferentes sonidos confieren un tono particular a las palabras. De ser así, habría que considerar en qué medida estaría esto provocado por la vocal o por el entorno fónico concreto que determina un alófono dado.

Secuencias vocálicas

Una secuencia de dos o más vocales se da en una sola sílaba, lo que llamamos diptongo o un triptongo, o distribuida en sílabas diferenciadas, lo que constituye un hiato. En el primer caso, tenemos palabras como ⟨ay⟩ /ai/ y ⟨buey⟩ /buei/, y en el segundo ⟨ahí⟩ /aˈi/.

Los diptongos en español suelen aparecer como la recategorización de un hiato (Chitoran y Hualde, 2007). El fenómeno surge de forma natural en el habla rápida y descuidada cuando se encuentran dos vocales. En este caso, se funden en un único sonido si se trata de dos instancias del mismo fonema, mientras que, uno de ellos deviene en no silábico si son dos vocales distintas. De esta manera, ⟨creer⟩ se realiza como [kɾeɾ] y ⟨avión⟩ como [aˈʃjon][65].

[65]Además de este fenómeno natural, debemos mencionar otro artificial de la misma índole consistente en la prescripción institucional de diptongos arbitrarios contra el uso habitual. Así, palabras como ⟨cruel⟩, cuya forma bisilábica es la común y aceptada en el solar natal de la lengua castellana (Quilis, 1993/2019, pp. 185-186), obligarían a considerar excepciones en el sistema de representación

Se ha discutido si los diptongos tienen un carácter monofonemático, como defiende Navarro Tomás (1946, pp. 13-14). Se apoya en las reglas I, II y III de Trubetzkoy (1939, pp. 42-45) para determinar la naturaleza fonemática de un sonido, pero la postura contraria (Alarcos Llorach, 1964, § 96) parece tener más peso, ya que se incumplen los puntos I y II, que son propios de un sonido monofonemático. Si tenemos en cuenta los requisitos que también propone Trubetzkoy (1939, pp. 50-52) para verificar la naturaleza monofonemática de un sonido, hemos de descartar la tesis de Navarro Tomás.

A nivel fonológico, solo hablamos de diptongos propios cuando es una vocal cerrada la que pierde el carácter silábico. De esta forma, el español cuenta con un repertorio de seis diptongos crecientes (vocal$^{+\text{CERRADA}}$ vocal$^{-\text{CERRADA}}$), siete decrecientes (vocal$^{+\text{CERRADA}}$ vocal$^{-\text{CERRADA}}$) y dos homogéneos (vocal$^{+\text{CERRADA}}$ vocal$^{-\text{CERRADA}}$). En el último caso, la primera vocal pierde la naturaleza silábica y es la segunda la que hace de núcleo (Quilis, 1993/2019). Cabe mencionar la particularidad de que, en caso de que la semiconsonante [w] sea el primer alófono prenuclear de la sílaba, es corriente la anteposición de un alófono de /g/ (p. ej., ⟨hueso⟩ /ˈgwe.so/).

	Tipo	**V$_1$**	**V$_2$**	**Ejemplo**	**t. fonológica**	**t. fonética**
1	↗	/i/	/e/	*tiene*	/ˈtie.ne/	[ˈt̪je.ne]
2	↗	/i/	/a/	*patria*	/ˈpa.tɾia/	[ˈpa.t̪ɾ̪ja]
3	↗	/i/	/o/	*vio*	/ˈbio/	[ˈbjo]
4	↗	/u/	/e/	*bueno*	/ˈbue.no/	[ˈbwe.no]
5	↗	/u/	/a/	*cuatro*	/ˈkua.tɾo/	[ˈkwa.t̪ɾ̪o]
6	↗	/u/	/o/	*superfluo*	/su.ˈpeɾ.fluo/	[su.ˈpeɾ.flwo]
1	↘	/e/	/i/	*ley*	/ˈlei/	[ˈlei̯]
2	↘	/a/	/i/	*aire*	/ˈai.ɾe/	[ˈai̯.ɾe]
3	↘	/o/	/i/	*soy*	/ˈsoi/	[ˈsoi̯]
4	↘	/e/	/u/	*Europa*	/eu.ˈɾo.pa/	[eu̯.ˈɾo.pa]
5	↘	/a/	/u/	*aula*	/ˈau.la/	[ˈau̯.la]
6	↘	/o/	/u/	*bou*	/ˈbou/	[ˈbou̯]
1	⟶	/i/	/u/	*viuda*	/ˈbiu.da/	[ˈbju.ð̪a]
2	⟶	/u/	/i/	*fui*	/ˈfui/	[ˈfwi]

Cuadro 4.5: Diptongos.

Hay que hacer notar, no obstante, que algunas de estas combinaciones son muy infrecuentes, o incluso ajenas a la naturaleza de la lengua española, como es el caso del diptongo [ou̯]. Este aparece exclusivamente en préstamos y nombres propios asimilados de otras lenguas, como el topónimo *Salou*.

propuesto por la Academia de querer amoldar el modelo al referente real. En lugar de eso, se optó por tratar de adaptar la lengua a su modelo.

La mayoría de los tratadistas (Navarro Tomás, 2004; Quilis, 1993/2019; Alarcos Llorach, 1964) no reconocen los diptongos decrecientes entre vocales cerradas salvo como una particularidad dialectal. Navarro Tomás (2004, sec. 149) indica que es un arcaísmo, pero que se conserva en algunos puntos septentrionales de España, particularmente en Asturias y llega a decir que «un poeta no podría hoy como en otro tiempo, emplear *cuida* en rima con *muda* o *duda*, sino con *mida*, *vida*, etc.» (Navarro Tomás, 2004). Lo cierto es que en casi todas las variedades se da el diptongo /ui̯/, aunque circunscrito a un conjunto muy concreto de palabras, como *muy* o *Ruy* (Calderón Rivera, 1991). Sin entrar en más consideraciones sobre el uso contemporáneo del español, debemos tener en cuenta estos diptongos en el análisis de textos auriseculares, como bien nota Bello (1835/1981, p. 94) e ilustra el ejemplo 3.

(3) Prodigios verán los hombres
en tres actos, y ninguno
a su representación
faltará por mi descuido.
Y pues que ya he prevenido,
cuanto al teatro, presumo

(Calderón, *El gran teatro del mundo*, vv. 225-230)

La palabra *descuido* se pronunciaría /desˈkwi.do/ en una lectura moderna, por lo que tendríamos una asonancia en i-o. Sin embargo, este no es el caso, pues la asonancia es en u-o, como *ninguno* y *presumo*. Por lo tanto, debemos volver a evaluar el diptongo como decreciente /desˈkui̯.do/, tal y como se concibió en la composición de la obra. En este caso, nos orientamos por el contexto de la rima, pero esta facilidad no existe en el interior del verso. El problema se complica cuando hallamos que el propio Calderón diptonga las mismas vocales de forma creciente (4).

(4) Narciso Que siempre... Pero ¿qué ruido
es este?
Bato Que el corzo herido

(Calderón, *Eco y Narciso*, vv. 2480-2481))

Efectivamente, aquí ha de pronunciarse *ruido* como /ˈrwi.do/ para que rime con *herido*. Aunque la pronunciación con hiato /ruˈi.do/ tampoco es ajena al español —y, desde luego, de ningún modo lo es al teatro del Siglo de Oro (Calderón, *La vida es sueño (primera versión)*, v. 100)—. No es el caso de este verso porque, de hacerse, estaríamos ante nueve sílabas. Si nos encontramos ante una licencia poética del poeta

o de un uso vacilante de un elemento propio del repertorio fonológico del español áureo, es una cuestión de respuesta incierta. Hallarla requeriría un estudio más profundo del que podemos ofrecer aquí sin desviarnos del propósito de este trabajo. A la vista de esto, asumiremos que se dan diptongos decrecientes entre vocales cerradas en posición de rima, cualquiera que sea la razón que lleve al dramaturgo a emplearlos.

Por otra parte, se plantea la cuestión contraria: resulta evidente que la generalización de Navarro Tomás (2004, sec. 66), que también contempla Quilis (1993/2019, pp. 178-181), no se cumple en todos los casos. En concreto, el diptongo /eu/, que se realizaría como decreciente [eu̯] atendiendo a la escala de perceptibilidad, como en *deudo*, se pronuncia también como creciente [e̯u] con la vocal alta /u/ como núcleo vocálico si el diptongo es en sílaba inacentuada, de manera que en habla vulgar llega incluso a la elisión total de /e/, como, por ejemplo, en *Lute* como diminutivo de *Eleuterio*.

Volvamos la vista atrás por un momento a lo que vimos en la sección sobre fonética acústica. Recordemos que para los primeros formantes de /e/ estaban en torno a $F_1 = 501Hz$ y $F_2 = 1895Hz$ y los de /u/ cerca de $F_1 = 315Hz$ y $F_2 = 836Hz$. Hemos tomado los formantes del diptongo en una grabación de la palabra *eufemismo* tomada de un informante al que no se le explicó el propósito del experimento. Hemos marcado en el gráfico las tres partes de la enunciación: correspondientes a /e/, /u/ y una transición entre ambos sonidos con $F_2 \approx 1500\,Hz$. Asimismo se indica con una línea donde se registra mayor intensidad. Comprobamos que la intensidad comienza a notarse de forma aproximada a la mitad del tiempo de /e/, alcanza su pico en el periodo de transición y desciende en /u/. Asimismo, el tiempo de /e/ es sensiblemente menor que el de /u/. Esto lleva a pensar que, en este caso, la descripción de Navarro Tomás es acertada. No obstante, por lo ya dicho, seguimos albergando reservas sobre su generalización. Sea como fuere, considerando que la declamación dramática implica una pronunciación cuidada, la daremos por buena en lo que a este estudio concierne.

Más allá de los límites de la palabra, la métrica demanda una nueva incursión en el terreno de la fonética porque el diptongo es un mecanismo de ajuste silábico que se da de manera constante en los versos en la unión entre dos palabras cuando la primera acaba en una vocal y la siguiente empieza también con otra. En este caso, una de las dos vocales pierde su calidad vocálica al unirse a la otra. Hablamos aquí de una sinalefa, y el accidente requiere la reevaluación del verso entero, en tanto que afecta al cómputo silábico, como veremos un poco más en detalle más adelante.

Desde una perspectiva acústica, estas propiedades son observables y cuantificables. De acuerdo con Milner (2021), las vocales puras y los vocoides del español presentan

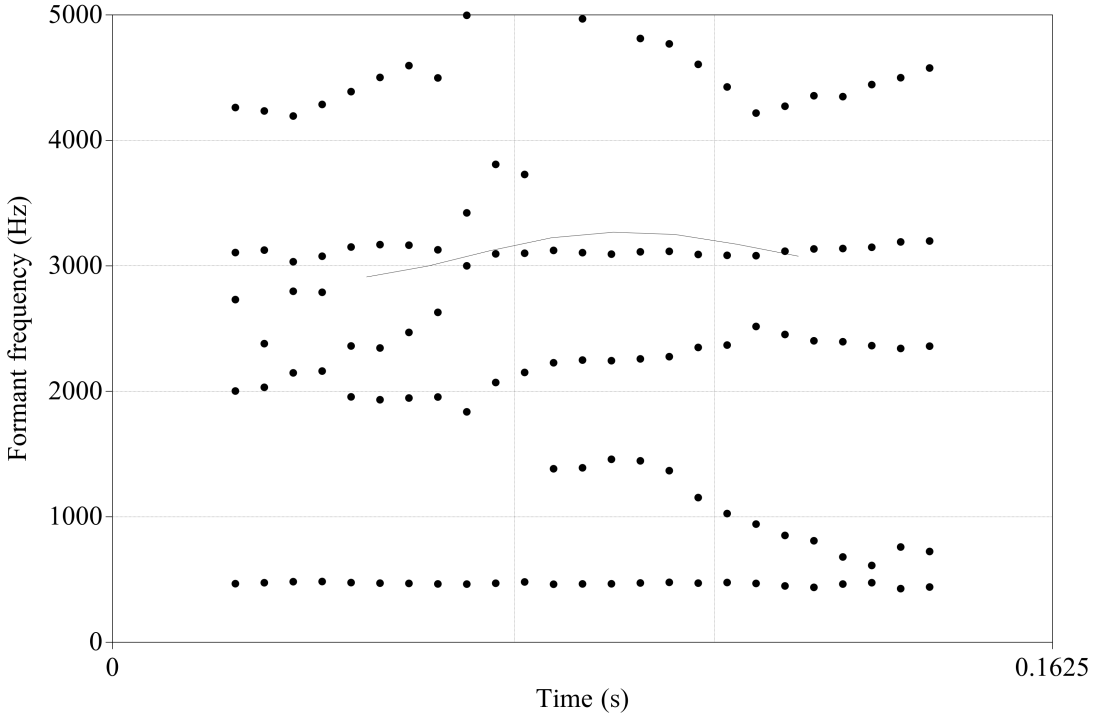


Figura 4.6: Diptongo /eu/.

diferencias acústicas en su primera y segunda frecuencia fundamental, que son más bajas en las vocales puras. De esta manera, el F2 de [i̯] y [u̯] es más alto que el de [i] y [u], respectivamente. Se da asimismo una transición entre formantes y amplitud a lo largo del segmento, lo que acorta la duración del diptongo respecto al hiato. En caso de sinalefa en el verso, se observa de esta manera si el diptongo impropio se realiza como creciente o decreciente.

	Tipo	**Ejemplo**	**t. fonológica**	**t. fonética**
1	/iai/	*estudiáis*	/estuˈdiais/	[estuˈðjai̯s]
2	/iei/	*vieira*	/ˈvieira/	[estuˈðjei̯s]
3	/ioi/	*hioides*	/ˈioides/	[ˈjoi̯ðes]
4	/iau/	*«riau»*	/riau/	[rjau̯]
5	/uai/	*averiguáis*	/aberiˈguais/	[aberiˈɣwai̯s]
6	/uei/	*santigüéis*	/santiˈgueis/	[santiˈɣwei̯s]
7	/uau/	*guau*	/guau/	[wau̯]

Cuadro 4.6: Triptongos.

Los triptongos son más escasos, y surgen de la combinación de un diptongo creciente y otro decreciente. En español solo hay siete tipos (Canellada y Madsen, 1987). El resultado es un núcleo vocálico precedido de un sonido semiconsonántico y seguido de uno semivocálico. Este núcleo solo lo constituyen los fonemas /a/, /e/ y /o/. Los sonidos no vocálicos los forman los fonemas /i/ y /u/, con la particularidad de que no

se dan las combinaciones /ieu/, /iou/, /uoi/, /ueu/ y /uou/. Dicho de otra manera, las siete combinaciones fonéticas posibles son: [jai̯], [jei̯], [joi̯], [jau̯], [wai̯], [wei̯] y [wau̯].

Así pues, la fonología no resulta suficiente por sí misma para caracterizar la sílaba métrica en las uniones vocálicas. Para conocer su naturaleza, es necesario identificar el núcleo y, en caso de secuencia vocálica, localizar qué fonema es el preponderante, ya que los otros son irrelevantes en estas circunstancias.

Sonidos consonánticos

Aunque no de tanta importancia como las vocales, ya que no afectan a la cuenta silábica, las consonantes desempeñan también un papel esencial. En primer lugar, la consonancia de la rima versal depende de ellas. En segundo, las reglas de división silábica las tienen en cuenta, por lo que su caracterización es necesaria a para deslindar sílabas. Sus propiedades físicas, por otra parte, interesan aquí en tanto que las distinguen de las vocales; las posibles realizaciones alternativas (p.ej. /tɾans.atˈlan.ti.ko/, /tɾan.satˈlan'ti.ko/ o incluso la realización americana /tɾan.saˈtlan.ti.ko/) se resuelven mediante reglas, que asumiremos sin entrar en otras discusiones que sobrepasarían el ámbito de este trabajo. Dejamos la puerta abierta a diferentes alternativas en la formalización de estos presupuestos teóricos para adaptar el análisis a un habla particular.

El repertorio del español dispone de tres fonemas oclusivos sordos con oposición por el punto de articulación. El labial /p/, el dental /t/ y el velar /k/ (Quilis, 1993/2019, pp. 194-195). De estos fonemas, /t/ y /d/ tienen realización laminar (predorsal) y [t̪] y [d̪] dentoalveolar (Martínez Celdrán y col., 2003, p. 257). A los fonemas sordos se oponen respectivamente los sonoros /b/, /d/ y /g/. No obstante, hay que hacer notar que el rasgo oclusivo no es pertinente en las últimas, ya que se realizan por norma general con los alófonos aproximantes [β̞], [ð̞] y [ɣ̞] (Navarro Tomás, 2004, secs. 79-132). La realización oclusiva ocurre después de una consonante nasal o de pausa. Asimismo, en el caso de /d/, también se da el alófono oclusivo después de una consonante lateral. Los fonemas /k/ y /g/ son pospalatales cuando anteceden a /i/ o /e/, por lo que su realización es [k̟] y [g̟] o [ɣ̟] (Canellada y Madsen, 1987, pp. 20-21).

Las oposiciones por sonoridad no son pertinentes en posición posnuclear, por lo que la oposición /p/-/b/ se neutraliza a /B/ con rasgo común de labialidad, /t/-/d/ a /D/ con rasgo común de dentalidad y /k/-/g/ a /G/ con rasgo común de velaridad (Quilis, 1993/2019, p. 205).

En cuanto a las consonantes nasales, todos los fonemas de este tipo en el repertorio español son sonoros. Quilis describe tres fonemas: uno bilabial /m/, otro lingualveolar

/n/ y uno linguopalatal /ɲ/, que se oponen en posición silábica prenuclear con una sola realización [m], [n] y [ɲ] respectivamente (Quilis, 1993/2019, p. 225) y se neutralizan en el archifonema /N/ en coda. No obstante, Clegg y Willis (2018, pp. 331-337) añaden a las realizaciones descritas por Quilis para /m/ la realización alternativa [ɱ] ante una consonante labiodental y para /n/, [n̪] ante un sonido interdental [n̪̪] ante uno dental, y las palatalizaciones [n̠] ante un sonido palatal y [ŋ] ante uno velar.

Modo de articulación	**Punto de articulación**									
	Labial		**Dental**			**Palat.**	**Velar**			
	Bil.	**L.-den.**	**Den.**	**Interd.**	**Alv.**		**Vel.**	**Bil.-alv.**		
Oclusiva	p b		t d				k g			
Nasal	m	[ɱ]		[n̪]	[n̪̪]	n	ɲ	[ŋ]		
Africada							t͡ʃ [d͡ʒ]			
Fricativa		f			θ	s		x		
Aproximante	[β̞]				[ð̞]		ʝ [j]	[ɣ̞]	[w]	
Lateral				[l̪]	[l̪̪]	l	ʎ			
Vibrante múlt.						r				
Vibrante simp.						ɾ				

Cuadro 4.7: Fonemas y alófonos consonánticos del español.

Cuando estos tres fonemas están en distensión silábica y preceden a una consonante, incluso entre palabras, su punto de articulación se asimila al de la consonante siguiente. De esta manera, por ejemplo, si la consonante nasal va seguida de una labial, se articula en posición labial (⟨invento⟩ /inˈbento/ [imˈbento]). En tal posición, el punto de articulación deja de ser pertinente y todas las nasales se neutralizan al archifonema /N/ (/iNˈbento/). Este mismo fenómeno sucede en posición final absoluta (Alarcos Llorach, 1964, pp. 180-182). Estos rasgos han de darse necesariamente porque la neutralización solo se da en fonemas muy cercanos (Quilis, 1993/2019, pp. 228-231). Martínez Celdrán (2004, p. 51) hace notar al respecto que /ɲ/ no debería considerarse salvo desde una perspectiva diacrónica porque solo se da en posición inicial en español moderno.

El español dispone de cuatro fonemas fricativos /f/, /θ/, /s/ y /x/ y uno africado /t͡ʃ/, así como un sonido aproximante sonoro /ʝ/. Debemos añadir a estos los alófonos aproximantes de /b/, /d/ y /g/ arriba mencionados. Esta colección se refiere al español estándar, pues existen dialectos en los que no se da la oposición entre los fonemas /θ/ y /s/, por lo que optan por /s̄/ en todos los casos, lo que se conoce como *ceceo* o por /s/, en cuyo caso hablamos de *seseo* (Clegg y Willis, 2018, p. 286).

El fonema /f/ es labiodental sordo y su realización normal es [f], que se produce al obstruir el paso del aire colocando los incisivos sobre el labio inferior. Podemos decir que tiene una distribución única con el alófono [f] (Quilis, 1993/2019, p. 246). El fonema /θ/ es interdental o linguinterdental, con el ápice lingual colocado sobre los

incisivos (Quilis, 1993/2019, p. 247). Ambos fonemas se sonorizarían ante consonante sonora como [v] y [ð], respectivamente, aunque, el caso de /f/ es muy raro por la escasez de casos de este fonema en posición de coda.

El fonema /s/ tiene varias realizaciones para distintos dialectos. En Castilla, la realización es apicoalveolar (Quilis, 1993/2019, p. 248). Este fonema se sonoriza habitualmente ante una consonante sonora y su realización es [s̬] (Torreblanca, 1978; Quilis, 1993/2019, p. 251), pero puede no ser el caso, especialmente si el hablante es una mujer (Muñiz Cachón y Cuevas Alonso, 2012). Llama la atención que Quilis (1993/2019, p. 251) indique que el fonema /s/ suele perderse ante /r/, por lo que ⟨dos reales⟩ se pronunciaría [ˈdo reˈales] y, en una pronunciación muy cuidada, con una /r/ asibilada [ˈdo ɹeˈales]. Navarro Tomás (2004, sec. 107) no va tan lejos y, valiéndose de los mismos ejemplos, propone [ˈdos̬ ɹeˈales], con una elisión total para casos muy particulares, al igual que hacen Canellada y Madsen (1987, p. 23).

El fonema /ʝ/ tiene realización *aproximante* en todos los contextos salvo después de pausa, nasal o lateral, en cuyo caso se presenta como una africada [ʧ͡ʝ̞] (Martínez Celdrán y col., 2003). No debe confundirse con el alófono [j] de /i/, que muestra diferencias acústicas y articulatorias significativas. Según Martínez Celdrán (2004, pp. 205-208), estas se resumen en que [j] es más breve y con cierta frecuencia apenas un sonido de transición que solo existe acompañado de una vocal nuclear y no se da en ataque sino siempre en posición nuclear, mientras que el sonido [j̞] tiene una amplitud menor, de forma especial en F_2, y solo se da en ataque. A las características de este alófono habría que añadir la explosión con la que comienza el africado [ʧ͡ʝ̞]. Tampoco debe confundirse con el sonido lateral /ʎ/ (Clegg y Willis, 2018, p. 287), presente en el habla cuidada.

El fonema /x/ es velar en el español de Castilla, con un alófono [χ] antes de /o/ y /u/ en todas sus realizaciones (Martínez Celdrán y col., 2003). Por su parte, el africado /t͡ʃ/ es linguopalatal sordo.

En lo que respecta a las *consonantes líquidas*, encontramos dos laterales con sus respectivos alófonos y dos vibrantes, una simple y otra múltiple. Las laterales en español son /l/ y /ʎ/, aunque esta última está en vías de desaparición y predomina únicamente en las variedades septentrionales ibéricas y en las tierras altas americanas, siendo lo habitual en el sur de España y América que se neutralice a /ʝ/. Las laterales se caracterizan por presentar formantes atenuados debido al cierre central, que, junto a las transiciones vocálicas —menos abruptas, en cualquier caso, que para las consonantes nasales—, indican el lugar de articulación.

El fonema /l/ es lateral lingualveolar y se realiza como [l], salvo en algunas posiciones silábica posnucleares (Quilis, 1993/2019, pp. 309-310). Cuando precede a /θ/, se desplaza hasta una posición interdental (⟨colza⟩ [kol̪θa]), antes de /t/ o /d/ es linguodental (⟨caldo⟩ [kal̪̪do]) y ante palatal se realiza linguopalatalizado (⟨el ñu⟩ [el̪ ɲu]) (Clegg y Willis, 2018, pp. 341-250). El fonema /l/ se da en la posición inmediatamente anterior o posterior al núcleo vocálico, mientras que /ʎ/ solo aparece en posición prenuclear en palabras patrimoniales. Esto hace cuestionable una neutralización a /L/ de las laterales desde una perspectiva sincrónica.

En castellano encontramos una consonante vibrante simple /ɾ/ y una múltiple /r/. Se producen colocando el ápice de la lengua contra los alvéolos para generar una o varias oclusiones. Ambas son apicoalveolares y no tienen más que un alófono. La vibrante simple solo aparece en interior de palabra en posición prenuclear entre vocales (⟨hora⟩ /ˈoɾa/) o entre /p/, /t/, /k/, /b/, /d/, /g/ o /f/ y vocal (⟨protráctil⟩ /pɾoˈtɾaktil/, ⟨crabrón⟩ /kɾaˈbɾon/ y ⟨grecofrancés⟩ /gɾekofɾanˈθes/) (Quilis, 1993/2019, p. 330). La oposición entre /r/ y /ɾ/ solo es pertinente fonológicamente en posición intervocálica (⟨perro⟩ /pero/-⟨pero⟩ /peɾo/), por lo que en todas las demás posiciones se neutraliza al archifonema /R/ (Alarcos Llorach, 1964).

Al contrario que con las vocales, la caracterización fina de las consonantes sí tiene consecuencias en el metro. Dicho de forma más precisa, la neutralización fonémica podría afectar a una parte tan importante como es la rima. Si bien resultaría difícil hallarlo, es una posibilidad teórica real. Por ejemplo, ⟨Abraham⟩ y ⟨Guzmán⟩ serían /abraˈaN/ y /guθˈmaN/ y, por lo tanto, formarían rima consonante.

4.7 División silábica del español

Características generales

La sílaba española consiste en un núcleo vocálico obligatorio, opcionalmente precedido de un ataque o seguido de una coda, ambos consonánticos (Figura 4.7). El núcleo consta generalmente de una sola vocal como /o/ en ⟨ocaso⟩ [o.ˈka.so], pero puede ir unido a vocales débiles en forma de vocoide, precedido de una semiconsonante como en ⟨idiota⟩ [i.ˈðjo.ta] o seguido de una semivocal como en ⟨hoy⟩ [oi̯], lo que constituiría sendos diptongos, o incluso orlado a ambos lados, como en ⟨dioico⟩ [ˈðjoi̯.ko], en cuyo caso hablaríamos de triptongo. Por su parte, tanto coda como ataque, cuando se dan, están compuestos por entre una y dos consonantes. Se denomina sílaba libre a aquella que consta de un núcleo vocálico en solitario o acompañado de uno o dos sonidos

consonánticos en ataque, y trabada a la que tiene más de un sonido consonántico en coda (Navarro Tomás, 1946, pp. 46-47).

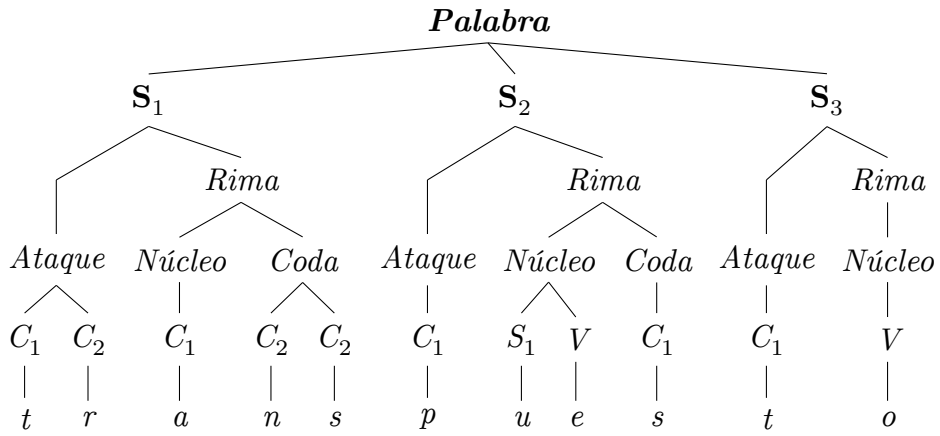


Figura 4.7: Construcción sílabas.

De acuerdo esto, tenemos nueve combinaciones posibles del núcleo con hasta dos consonantes en ataque o en coda, distribuidas según lo mostrado en el ejemplo 5.

(5)

V	*y*, *o*,
CV	*la*, *de*,
VC	*un*, *él*
CVC	*mar*, *ven*,
CCV	***bra***·*sa*, *la*·***cre***
VCC	***obs***·*ce*·*no*, ***ins***·*pi*·*ra*·*ción*
CCVC	***bron***·*ce*, *im*-***pren***·*ta*
CVCC	***pers***·*pec*·*ti*·*va*, ***cons***·*cien*·*te*
CCVCC	***trans***·*mu*·*tar*

Cualquier consonante puede ocupar la posición de ataque en solitario. Sin embargo, si hay dos consonantes, no todas las combinaciones son posibles. En posición de coda ocurre lo mismo, con la restricción adicional de que determinadas consonantes no se dan jamás en dicha posición.

Asimismo, aparece como máximo una sola consonante a final de palabra en español moderno, al contrario que en otras variedades diacrónicas donde se dan casos como *puent*, *muert* (Navarro Tomás, 1946, p. 49). No obstante, las pocas coincidencias relativamente recientes que se encuentran en el CORDE nos remiten a documentos de la primera mitad del siglo XVI, por lo que, considerando asimismo que el registro escrito es más conservador que la lengua hablada, asumimos que el español del Siglo de Oro se comporta en este sentido como la variedad moderna.

División entre vocales

En la división vocálica podemos distinguir los casos vocal$^{+\text{ALTA}}$ vocal$^{+\text{ALTA}}$, vocal$^{+\text{ALTA}}$ vocal$^{-\text{ALTA}}$, vocal$^{-\text{ALTA}}$ vocal$^{+\text{ALTA}}$ y vocal$^{-\text{ALTA}}$ vocal$^{-\text{ALTA}}$. Exclusivamente el último de los cuatro casos forma siempre un hiato. Todos los otros casos constituyen diptongo o hiato (Ríos Mestre, 1999). De esta forma, ⟨crear⟩ será necesariamente bisilábica /kɾeˈaɾ/, pero tenemos ⟨lío⟩ /ˈli.o/ [ˈli.o], ⟨lio⟩ /liˈo/ [liˈo] o ⟨vio⟩ /bio/ [ßjo] y ⟨voy⟩ /ai/ [aɪ̯] o ⟨ahí⟩ /a.ˈi/ [a.ˈi], y ⟨fui⟩ /fui/ [fwi] o ⟨hui⟩ /u.ˈi/ [/u.ˈi/]. Aunque la división silábica se infiere de la ortografía la mayoría de las veces, este no es siempre el caso, como vemos, por ejemplo, en ⟨criado⟩ /kɾi.ˈa.do/ [kɾi.ˈa.ðo] y ⟨aciago⟩ /a.ˈθia.go/ [a.ˈθja.ɣo].

De acuerdo con Quilis (1993/2019, pp. 184-186), dos vocales átonas forman diptongo si al menos una de ellas es alta. La secuencia vocal$^{+\text{ALTA}}$ vocal$^{-\text{ALTA}}$ o vocal$^{-\text{ALTA}}$ vocal$^{+\text{ALTA}}$ forma diptongo cuando la vocal alta es tónica. El resto de los casos suele formar diptongo, salvo en los siguientes casos descritos por el autor:

Verbos en *-iar*: *agriar*, *aliar*, *ampliar*, *arriar*, *ataviar*, *autoviar*, *aviar*, *ciar*, *confiar*, *congloriar*, *contrariar*, *criar*, *desafiar*, *desaviar*, *descarriar*, *desconfiar*, *desvariar*, *desviar*, *enfriar*, *enviar*, *espiar*, *estriar*, *expiar*, *extraviar*, *ferroviar*, *fiar*, *gloriar*, *grafiar*, *guiar*, *hastiar*, *istriar*, *liar*, *malcriar*, *piar*, *porfiar*, *recriar*, *reenviar*, *resfriar*, *triar* y *vaciar*. Previene, no obstante, que no afecta el futuro y al condicional porque, mientras que en *fiar*, *fía* o *fio* una de las vocales es tónica, en *fiaré* o *fiaría* son ambas átonas.

Verbos en *-uar*: *acentuar*, *actuar*, *atenuar*, *cansuar*, *conceptuar*, *contextuar*, *continuar*, *descontinuar*, *desgraduar*, *deshabituar*, *desvirtuar*, *estatuar*, *evaluar*, *exceptuar*, *extenuar*, *fluctuar*, *ganzuar*, *graduar*, *habituar*, *individuar*, *infatuar*, *insinuar*, *menstruar*, *perceptuar*, *perpetuar*, *puar*, *puntuar*, *ruar*, *situar*, *tatuar*, *tumultuar*, *usufructuar* y *valuar*, y se exceptúan los verbos terminados en *-cuar* [kwaɾ] y *-guar* [gwaɾ], que forman diptongo en todos los casos y, por el mismo motivo que los verbos en *-iar*, el futuro y el condicional.

Verbos en *-uir*: *afluir*, *argüir*, *atribuir*, *circuir*, *concluir*, *constituir*, *construir*, *contribuir*, *derruir*, *destituir*, *destruir*, *diluir*, *disminuir*, *distribuir*, *esmuir*, *estatuir*, *excluir*, *fluir*, *huir*, *imbuir*, *incluir*, *influir*, *inmiscuir*, *instituir*, *instruir*, *intuir*, *irruir*, *muir*, *obstruir*, *ocluir*, *prostituir*, *recluir*, *reconstituir*, *reconstruir*, *redargüir*, *refluir*, *rehuir*, *restituir*, *retribuir* y *sustituir*. La excepción del condicional y el futuro aplica aquí también.

Adjetivos en *-uoso(s)*/*-uosa(s)*: *suntuoso*, *virtuoso*, *sinuoso*.

Casos particulares y sus flexiones: *acuoso*, *anual*, *arriero*, *bienio*, *biombo*, *boquiancho*, *cliente*, *crianza*, *cruel*, *desconfianza*, *diablo*, *diálogo*, *dieciocho*, *ferrovial*, *fianza*, *gorrión*, *guión*, *hiato*, *maniobra*, *miasma*, *piano*, *prior*, *santuario*, *Sión*, *tiara*, *triángulo*, *trienio*, *truhán* y *ventiocho*. Navarro Tomás (2004, sec. 68) añade *ruido*, *suave*, *viaje* y *viuda*.

Secuencias /ˈi.u/, /ˈu.i/, /iˈu/ o /uˈi/: *jesuita*, *circuito*, *fortuito*, *casuística* y *diurno*.

A esto debemos añadir los verbos en *-eír* (Ríos Mestre, 1999, sec. 6.8.2.1.2.2), como *desleír*, *engreír*, *freír*, *refreír*, *reír*, *sofreír* y *sonreír*. Se volvería a excluir el futuro y el condicional. Ríos Mestre (1999, sec. 6.8.2.1.2.3) alude asimismo a los verbos en *-oír*, pues considera inoportunas las salvedades del futuro y el condicional.

Canellada y Madsen (1987, pp. 51-52) señalan asimismo *ahilarse*, *baulero*, *buhonero*, *criador*, *huidizo*, *miopía*, *prohibir*, *reunión*, *reiré* y *semiesfera*. Observamos que estos son bien derivados de palabras sin ambigüedad ortográfica, de las excepciones descritas o resultado de la prefijación, a excepción de *buhonero*, *miopía* y *prohibir*.

No hemos considerado prefijos acabados en vocal como *re-* o *tri-* (por ejemplo, *rehilamiento* y *triángulo*), que podrían tender a formar hiato en algunas circunstancias, como un habla muy lenta, dado que la literatura al respecto ha resultado ser muy escasa. Sin embargo, incluimos un archivo de excepciones en la formalización donde forzar el hiato en casos particulares para modificar el comportamiento del módulo que realiza la silabación. Las entradas correspondientes llevan una marca que las identifica como comentarios y no como reglas, por lo que no se tienen en cuenta; esta marca se borra discrecionalmente cuando se desea hacer el hiato.

División entre consonantes

Podemos distinguir los siguientes casos: una consonante intervocálica, grupo de dos consonantes intervocálicas, tres o más consonantes intervocálicas, que Quilis (1993/2019, pp. 367-369) resuelve de tres maneras diferentes.

V-C-C-V: El caso más simple es el de una consonante intervocálica. Dado que el español tiene tendencia a las sílabas abiertas, dicha consonante se agrupará con la sílaba siguiente.

V-C-C-V: Si las consonantes son una agrupación de consonante del tipo *muta cum liquida*, esto es bilabial o labiodental (/p/, /b/, /f/) o linguovelar (/g/, /k/) seguida de una líquida (/ɾ/, /l/) o una linguodental(/d/, /t/) seguida de vibrante (/ɾ/), nos encontramos ante una unidad inseparable que forma la coda del núcleo silábico siguiente (*probar*, *plomo*, *broma*, *blanco*, *francés*, *flanco*,*gracia*, *glosa*, *craso*, *claro*, *drama*, *trama*). Cualquier otra combinación se reparte entre la primera y la segunda sílaba.

V-C-C-C-(C)-V: Para que se dé una combinación de tres consonantes, los dos últimos fonemas deben formar una unidad inseparable como las descritas en el párrafo anterior o bien las dos primeras deben ser una agrupación del tipo /Ns/ o /Bs/ (⟨constar⟩ /konsˈtaɾ/, ⟨obstar⟩ /obsˈtaɾ/). Si hay cuatro consonantes seguidas se trata de una conjunción de ambos casos (⟨abstracción⟩ /abs.tɾakˈθion/), por lo que se reparten dos en cada núcleo.

Contacto entre vocales de diferentes palabras

El principio fundamental de la reducción vocálica a grupos silábicos reza que, según la fonética, dos vocales cualesquiera en contacto admiten reducirse siempre a una sola sílaba, por lo que lo harán si no hay razones gramaticales o de otra índole que lo impidan (Navarro Tomás, 2004, sec. 136). Canellada y Madsen (1987, p. 55) hacen una descripción detallada de las circunstancias que concurren en los límites de palabra cuando hay dos vocales en contacto y cómo se resuelven estas uniones. Según los autores, se producen cuatro tipos de unión: *hiato*, *reducción*, *diptongación* y *elisión*. Salvo en el hiato, hablaremos en estos casos de *sinalefa*. Llamamos sinalefa a la unión en una única sílaba de dos vocales adyacentes ubicadas en posición final e inicial de sendas palabras contiguas. Pueden unirse de esta manera hasta cinco vocales distribuidas a lo largo de hasta tres palabras, como, por ejemplo, en ⟨repudio a Eugenia⟩ [reˈpu.djo̯a̯e̯u̯ˈxe.nja]. Para que ocurra una sinalefa han de cumplirse cuatro condiciones:

1. La pronunciación debe ser rápida.
2. No hay enfatizaciones.
3. Si intervienen más de dos vocales, entre dos vocales abiertas solamente puede haber otra abierta.

Si no se cumplen estas condiciones, lo normal es que se produzca un hiato /boi̯.aˈka.sa/ o, si acaso, que se redistribuyan las fronteras silábicas /bo.ja̯ˈka.sa/. Cuando sí concurren todas las condiciones necesarias, la resolución de la sinalefa varía en función de las partes que la constituyen.

De esta manera, si se encuentran dos vocales iguales, estas se funden en una. Si el resultado tiene más cantidad, hablaríamos de reducción (6a). Por el contrario, si la cantidad es regular, hablaremos de elisión (6b). La realización según uno u otro recurso depende sobre todo de la velocidad de la pronunciación. Además, si una de las vocales es tónica, la unión también será tónica (Navarro Tomás, 2004, secs. 138-139).

(6) ⟨va a verlo⟩
   a. /ˈbaːˈbeɾ.lo/
   b. /ˈbaˈbeɾ.lo/

Si las vocales son diferentes, una de las dos pierde su condición silábica, pero conservar el timbre vocálico. En este caso, se trataría de una reducción (7). Cuál de ambas vocales hace de núcleo y cuál se reduce depende de varios factores. Si ambas son átonas, el elemento de la segunda posición tiene precedencia (8). Si una de las dos partes es tónica y la otra no, se prefiere la vocal tónica para hacer de núcleo (9). En cualquier otro caso, la vocal más abierta precede a la más cerrada como posible núcleo (10) (Navarro Tomás, 2004, secs. 140-152).

(7) ⟨se acerca⟩ /se̯aˈθeɾ.ka/
(8) ⟨parece huraño⟩ /paˈɾe.θe̯uˈɾa.ɲo/
(9) ⟨la hebra⟩ /ˈla̯e.bɾa//
(10) ⟨va este⟩ /ˈba̯es.te/

En el último caso, hay que tener además en cuenta la escala de perceptibilidad, que, según Navarro Tomás (2004, sec. 25), se establece según el grado de apertura bucal, por lo que /a/ sería la vocal más perceptible, seguida de /o/ y de /e/ y las menos perceptibles serían /i/ y /u/. Asimismo, al igual que ocurre entre vocales iguales, si una de las partes de la unión es tónica, la sílaba resultante también lo será (2004, sec. 142).

Si la vocal que pierde su calidad de núcleo silábico es /i/ o /u/, también lo hace con su condición vocálica, de forma que ⟨mi amigo⟩ se pronunciaría /mjaˈmi.go/ en ese caso. Además, si la vocal cerrada es tónica, el acento se desplaza a la siguiente vocal. Esto es, en ⟨salí ayer⟩ /saˈljaˈja̯eɾ/, el acento recaería en [a]. Esto se confirma también en la poesía (Esgueva Martínez, 2004).

La elisión total, aunque no es ajena a la lengua hablada —especialmente en expresiones hechas como **sabusté* o **sacabao*—, se considera propia de registros vulgares. En cualquier caso, dado que, en las representaciones dramáticas de las que nos ocupamos, el habla es intencionalmente cuidada —se declama—, esta situación no debería suponer un inconveniente. Podemos asumir que el dramaturgo habría indicado explícitamente si alguna pronunciación ha de ser deliberadamente inadecuada, como, por ejemplo, cuando un rústico habla sayagués. Aquí, claro está, se hace patente la necesidad de una buena edición que enmiende el exceso de celo corrector de algún cajista.

Si la agrupación es de más de dos vocales, la reducción no es siempre posible. Para que esto suceda, deben estar agrupadas de menor a mayor perceptibilidad si la última vocal es nuclear (11), y viceversa si es la primera (12). Si la vocal nuclear ocupa una posición central, el grado de perceptibilidad de las vocales no nucleares debe reducirse en ambas direcciones (13), de modo que, si la secuencia vocálica continúa, la vocal menos perceptible represente la división silábica.

(11) ⟨espíritu o aliento⟩ /esˈpi.ɾi.tu̯o̯aˈljen.to/

(12) ⟨halla o ubica⟩ /ˈa.ʎao̯u̯ˈbi.ka/

(13) ⟨vengo a esperar⟩ /ˈven.go̯ae̯s.peˈɾaɾ/

Puede darse en este último caso la ambigüedad de que la vocal reducida liminar se agrupe indistintamente con el primer núcleo (14a) o con el segundo (14b).

(14) ⟨la he asado⟩

a. /ˈla̯e.aˈsa.do/

b. /laˈe̯aˈsa.do/

Conjunciones *y*/*e* y *o*/*u*

La conjunción *y* se une a las siguientes vocales observando unas normas (Canellada y Madsen, 1987, p. 59; Navarro Tomás, 2004, sec. 49). Cuando aparece en posición interconsonántica, se pronuncia como /i/ de relajación variable. Cuando va entre una consonante y una vocal, se realiza como semiconsonante en diptongo con el fonema vocálico siguiente y se pronuncia /j/, salvo que vaya precedida de /s/ o /z/, en cuyo caso la pronunciación será consonántica y se unirá en ataque como /ʝ/. En la situación inversa, si va precedido de una vocal y seguido de una consonante, diptonga con el núcleo vocálico precedente en calidad de semivocal como /i̯/. La última posibilidad

es que se encuentre entre dos vocales, en cuyo caso tendrá calidad consonántica y se unirá en ataque a la sílaba de su derecha como /j̯/.

Las conjunciones *e*, *o* y *u* suelen unirse con una consonante final de la primera palabra y diptongan con la vocal inicial de la segunda (Canellada y Madsen, 1987, p. 60). Entre vocales, suele producirse un hiato con la primera palabra y un diptongo con la segunda tanto para *e* como para *o* (Canellada y Madsen, 1987, p. 60). Lo mismo señala Bello (1835/1981, pp. 109-111), que llama la atención sobre lo áspera que se hace al oído castellano la ligadura de dos vocales por *o*. No obstante, Bello indica que la conjunción *e* funciona de manera opuesta, uniéndose a la vocal precedente.

4.8 Transcripción grafémico-fonológica

Ríos Mestre (1999, sec. 2.3) describe tres procedimientos de transcripción: por reglas, si se toma como entrada los grafemas para transformarlos, por diccionario, si se toman unidades léxicas, y la pronunciación por analogía, basada en la inteligencia artificial. Los sistemas basados en reglas son los menos costosos de construir porque se fundamentan en la sistematización de la teoría lingüística. Estos se modifican de manera sencilla y la corrección de errores es rápida. Como contrapartida, son lentos, y su velocidad disminuye en función del número de reglas por lo que, a mayor precisión, mayor lentitud. La transcripción por diccionarios es más exacta además de veloz, pero, por su propia naturaleza, solo admite transcribir palabras previamente almacenadas. Independientemente de esto, decidirse por uno u otro método requiere, además, tener en cuenta otro aspecto fundamental: la relación entre grafema y fonema de la lengua en cuestión.

El sistema de reglas se ha empleado tradicionalmente en la conversión de texto a habla, con buenos resultados (Pérez y Vidal, 1991; López-Gonzalo y col., 1993; Rodríguez Crespo y col., 1998), pero no siempre conduce a una resolución óptima de todos los casos si la ortografía no da lugar a la interpretación fonética libre de ambigüedad. En esos casos, las irregularidades demandan procedimientos específicos. Entre estos, señalaremos la regularización de las excepciones ortográficas o las listas de palabras (1999, sec. 2.3). Esta última aproximación usaría la transcripción mediante diccionario como apoyo para la transcripción por reglas.

En cuanto a la formalización de las reglas, Ríos Mestre (1999, sec. 2.4) se basa en Chomsky y Halle (1968) para enunciar cuatro tipos de reglas de transformación. Primero, habría reglas biunívocas, cuando un grafema se transcribe con un único fonema, por ejemplo, ⟨t⟩ → /t/. El siguiente tipo sería la correspondencia dos a uno, en el

caso de un dígrafo, por ejemplo ⟨qu⟩ → /k/. También se da el caso contrario, con una correspondencia uno a dos, cuando a un grafema le corresponden dos fonemas, como en el caso de ⟨x⟩ → /ks/. Finalmente, existe la posibilidad de una correspondencia uno a cero, cuando a un grafema, como ⟨h⟩, no le corresponde fonema alguno.

Las reglas simples de reescritura pueden formar un sistema booleano complejo, con la derivación mediante operaciones unarias como la negación, o binarias como la combinación de dos reglas entre sí. Estas, a su vez, se aplican de forma recursiva. Asimismo, resulta conveniente bajo ciertas circunstancias valerse de definiciones, en lugar de secuencias y reglas *globales*, que consideren estados anteriores de la derivación (Ríos Mestre, 1999, sec. 2.4)

De acuerdo con la descripción de van Leuwen (1989), tenemos un texto de entrada y uno de salida. El primero es grafémico y el segundo es su transcripción fonética. A los caracteres de estos textos los denominamos *segmentos*, y la herramienta para operar en ellos es la regla lingüística. Cada segmento se traducirá conforme a su contexto según la forma propuesta por Chomsky y Halle (1968):

$$F \longrightarrow C \;/\; L_R$$

En la entrada tenemos un foco F que se reescribe como C si F va precedido contextualmente por L y seguido de R. Hugo van Leuwen (1989, p. 86) da como ejemplos de reglas la asignación directa de fonemas a grafemas, la inserción de límites, modificación de raíces, supresión de elementos y generalizaciones fonológicas.

Llamaremos *primitivas* a los segmentos, rasgos o etiquetas que forman una unidad mínima. De este modo, tenemos primitivas como, por ejemplo, el segmento *a* para la letra a, el rasgo ⟨+VOC⟩ para unidades vocálicas o la etiqueta ⟨+TON⟩ para señalar el *prosodema* acento. Una primitiva por sí misma determina un patrón simple, pero también es posible combinar varias primitivas entre sí para definir patrones complejos. Para describir estos, aplicamos operaciones sobre las primitivas que notamos con *operadores*. Estos son *concatenación*, *alternancia*, *opción*, *complementación* y *simultaneidad*.

La *concatenación* es una operación binaria que se emplea para definir un patrón *a* seguido de forma inmediata de otro *b* y se representa con una coma. Para señalar el segmento *a* seguido de *b*, usaremos la siguiente expresión.

a, b

La *alternancia* equivale a la operación lógica de *disyunción*, representada mediante la inclusión en llaves[66]. De este modo, para representar un segmento que adopte los valores a o b, lo haremos de la siguiente manera.

$$\{\begin{smallmatrix}\mathrm{a}\\ \mathrm{b}\end{smallmatrix}\}$$

Con la *opción* se especifica si se dan o no instancias de un patrón dado y cuántas veces se repite este en cada instancia. La representamos mediante paréntesis. Para indicar instancias de entre i y j veces el segmento a, se hará de la siguiente manera.

$$\mathrm{a}(i,j)$$

La *complementación* es una operación unaria que equivale a la *negación lógica* y la representamos mediante el símbolo de prima. Describimos cualquier segmento distinto de a con la expresión que sigue.

$$'\mathrm{a}$$

La *simultaneidad* equivale a la conjunción lógica y lo representamos encerrando en corchetes sus componentes. Si queremos indicar, por ejemplo, segmentos distintos de a con rasgo vocálico y etiquetados como acentuados, lo expresaremos de la siguiente forma.

$$\begin{bmatrix}\langle +\textsc{voc}\rangle\\ \langle -\textsc{ton}\rangle\\ '\mathrm{a}\end{bmatrix}$$

Si se dan los patrones especificados, se lleva a cabo una de las siguientes acciones: *asignar segmento*, *modificar rasgo*, *asignar etiqueta*.

El procedimiento más habitual es la asignación de segmento. Consiste en dar segmentos de salida a los segmentos de entrada. Supongamos que tenemos las siguientes reglas de transformación para el grafema c:

$$\mathrm{c} \longrightarrow \theta \;/\; _ \begin{bmatrix}\mathrm{e}\\ \mathrm{i}\end{bmatrix}$$

$$\mathrm{c},\mathrm{h} \longrightarrow \text{ʧ} \;/\; _\langle +\textsc{voc}\rangle$$

$$\mathrm{c} \longrightarrow \mathrm{k} \;/\; _ \begin{bmatrix}'\mathrm{h}\\ '\mathrm{e}\\ '\mathrm{i}\end{bmatrix}$$

Tomemos la palabra *descachace*. Se recorren los segmentos de la palabra de uno en uno y se aplican sucesivamente las tres reglas definidas arriba. El resultado de aplicar

[66] En lugar de emplear una llave para cada elemento, como hace van Leuwen, hemos optado por simplificar la notación para múltiples operandos para representar de manera sencilla y sin ambigüedad relaciones lógicas como $p \wedge (q \vee r)$.

estas reglas de transcripción a cada uno de los segmentos se ve en las tablas de transformación. La flecha señala el segmento tratado en cada uno de los pasos, por lo que esta se va desplazando a medida que se evalúan nuevos elementos de la entrada.

entrada:	d	e	s	c	a	c	h	a	c	e
				↑						
Salida 1:	d	e	s	k	a	c	h	a	c	e
						↑				
Salida 2:	d	e	s	k	a	ʧ		a	c	e
									↑	
Salida 3:	d	e	s	k	a	ʧ		a	θ	e

Como se ve, no se aplica regla alguna en los tres primeros segmentos, pues no se cumplen las condiciones necesarias. En el tercer segmento, la evaluación de la primera y la segunda regla es negativa, por lo que tampoco se aplica una acción. La tercera regla, por el contrario, sí es pertinente, de modo que obtenemos la transformación que se observa en Salida 1. En el sexto segmento no se aplica la primera regla, pero sí la segunda, mientras que la tercera tampoco se cumple. Cuando llegamos al penúltimo segmento, aplicamos la primera regla, que es la única relevante de las tres.

Valiéndonos de la modificación del rasgo es posible establecer generalizaciones. Lo ilustraremos con el ejemplo de la sonorización de /s/. Sabemos que /s/ se sonoriza ante una consonante sonora, de modo que se realiza con el alófono [z] en ese caso. Esto se define con la regla siguiente.

$$\text{s} \longrightarrow \langle +\text{son} \rangle \; / \; _ \left[\begin{smallmatrix} \langle +\text{cons} \rangle \\ \langle +\text{son} \rangle \end{smallmatrix} \right]$$

Finalmente, pasamos a etiquetar el segmento si es necesario. Por ejemplo, señalamos el prosodema acento de una vocal con tilde ortográfica añadiendo una nueva primitiva al de tipo etiqueta de la siguiente manera:

$$\left\{ \begin{smallmatrix} \text{á} \\ \text{é} \\ \text{í} \\ \text{ó} \\ \text{ú} \end{smallmatrix} \right\} \longrightarrow \langle +\text{ton} \rangle \; / \; _$$

De esta manera, combinando primitivas mediante operadores y aplicando reglas de transformación de manera secuencial, vamos transponiendo los elementos del texto para producir la transcripción fonológica.

4.9 Correspondencia grafémico-fonológica

Fonemas

Ya tenemos un método de representar las transformaciones requeridas para representar un texto grafémico según su equivalente fonológico. Queda, por tanto, definir las primitivas a las que se aplicaran las transformaciones. Proponemos dos tipos: el segmento, que se corresponderá con un grafema o un dígrafo y la etiqueta ⟨+TON⟩, que corresponde al prosodema acento.

Respecto a las primitivas segmentales, la ortografía de la Real Academia Española señala que la correspondencia grafema-fonema es prácticamente unívoca en el español contemporáneo, tal como lo hemos representado en Cuadro 4.8. Esta correspondencia, aun siendo poco ambigua si la comparamos con otras lenguas como, por ejemplo, el inglés, dista de ser perfecta (Real Academia Española y Asociación de Academias de la Lengua Española [RAE], 2010b, pp. 87-199). Por lo tanto, se hace inevitable considerar una por una ciertas excepciones.

Existen correspondencias unívocas, como son las de las vocales y las consonantes ⟨b⟩, ⟨d⟩, ⟨f⟩, ⟨h⟩, ⟨j⟩, ⟨k⟩, ⟨l⟩, ⟨ll⟩, ⟨m⟩, ⟨n⟩, ⟨ñ⟩, ⟨p⟩, ⟨q⟩, ⟨s⟩, ⟨t⟩, ⟨v⟩, ⟨x⟩ y ⟨z⟩. Asimismo, los dígrafos ⟨ch⟩ y ⟨ll⟩ equivalen a sendos fonemas únicos. Hay que notar que el grafema ⟨h⟩ es mudo salvo cuando forma parte del dígrafo ⟨ch⟩, en cuyo caso no se considera individualmente. Lo mismo sucede con el grafema ⟨l⟩ —equivale al fonema /l/— cuando aparece duplicado. En ese caso, se considera como un diírafo que representa el fonema /ʎ/.

El grafema ⟨c⟩ se transcribe /k/ cuando antecede a ⟨a⟩, ⟨o⟩, ⟨u⟩ o consonante, y /θ/ cuando precede a ⟨e⟩ o ⟨i⟩. El grafema ⟨g⟩ corresponde a /g/ ante ⟨a⟩, ⟨o⟩, ⟨u⟩ o consonante líquida. Delante de ⟨e⟩ e ⟨i⟩, se pronuncia /x/, salvo que se interponga ⟨u⟩ entre la consonante y la vocal, en cuyo caso, el grafema ⟨u⟩ será mudo y ⟨g⟩ se pronunciará /g/. El fonema ⟨q⟩ representa el fonema /k/ y aparece siempre acompañado del grafema mudo ⟨u⟩ y solo ante ⟨e⟩ e ⟨i⟩. El grafema ⟨r⟩ equivale al vibrante simple /ɾ/, salvo a comienzo de palabra, si aparece duplicado o detrás de consonante líquida o nasal. El grafema ⟨w⟩ no se da más que en unas pocas voces extranjeras y su pronunciación varía en función de la de su lengua de origen. La pronunciación habitual de ⟨x⟩ es /ks/, pero se conserva la histórica /x/ en algunos nombres propios, en particular topónimos como *México*, *Texas* u *Oaxaca*, y /s/ a comienzo de palabra, como en *xenón* o *xilografía*. Finalmente, ⟨y⟩ representa el fonema vocálico /i/ en posición nuclear (semivocálico si es el constituyente no silábico de un diptongo decreciente) o el consonántico /ʝ/ si es prenuclear.

Grafema	**Fonema**	**Ejemplo**
⟨a⟩	/a/	*ama*
⟨b⟩	/b/	*boba*
⟨c⟩	/k/, /θ/	*casa*, *cresta*, *cirio*
⟨ch⟩	/ʧ/	*choza*
⟨d⟩	/d/	*dado*
⟨e⟩	/e/	*eso*
⟨f⟩	/f/	*foco*
⟨g⟩	/g/, /x/	*gato*, *gemelo*, *grueso*
⟨gue⟩, ⟨gui⟩	/ge/, /gi/	*guerra*, *guiso*
⟨h⟩	∅	*hado*
⟨i⟩	/i/	*izar*
⟨j⟩	/x/	*jota*
⟨k⟩	/k/	*kilo*
⟨l⟩	/l/	*losa*
⟨ll⟩	/ʎ/	*llanto*
⟨m⟩	/m/	*metro*
⟨n⟩	/n/	*nada*
⟨ñ⟩	/ŋ/	*ñoño*
⟨o⟩	/o/	*oso*
⟨p⟩	/p/	*popa*
⟨qu⟩	/k/	*queso*
⟨r⟩	/r/, /ɾ/	*ropa*, *enroque*, *cara*
⟨rr⟩	/r/	*carro*
⟨s⟩	/s/	*sosa*
⟨t⟩	/t/	*topo*
⟨u⟩	/u/	*uso*
⟨v⟩	/b/	*vaso*
⟨w⟩	/b/, /u/	*watio*, *Winston*
⟨x⟩	/ks/, /x/, /s/	*excelente*, *México*
⟨y⟩	/i/, /j/	*ay*, *ya*
⟨z⟩	/θ/	*zapato*

Cuadro 4.8: Correspondencia grafémico-fonológica.

Prosodemas

Para transcribir texto de manera adecuada debemos asimismo conocer también la prosodia de las palabras. El español, empero, es una lengua de acento libre, esto es, el acento no tiene una posición fija, al contrario que las lenguas de acento fijo, como el francés, ni determina la acentuación del contexto silábico, como sucede con el latín. Por lo tanto, encontramos el acento prosódico en la última, penúltima, antepenúltima o, en casos muy excepcionales, incluso en la sílaba anterior a esta (RAE, 2010b, p. 24).

Afortunadamente, la ortografía del español provee reglas sencillas, precisas y con pocas excepciones para ubicar las sílabas tónicas de las palabras escritas observando las prescripciones académicas (RAE, 2012, pp. 39-40). Podemos establecer reglas si

asumimos que un texto está redactado con una ortografía adecuada. Las más sencillas de identificar son las esdrújulas y sobreesdrújulas porque todas llevan acento gráfico (*esdrújula*, *déjamelo*). Son agudas aquellas palabras que terminan en *n*, *s* o vocal y la última sílaba lleva tilde (*Calderón*). También son agudas aquellas que no tengan tilde alguna y no terminen en las letras indicadas o terminan en *n* o *s* (*merced*) que no preceden de forma inmediata a una vocal sino a una consonante (*vermuts*), o terminan en *y* (*jersey*). Son llanas aquellas palabras que llevan tilde en la penúltima sílaba y terminan en consonante distinta de *n* o *s* (*superávit*), en dos consonantes (*bíceps*) o en *y* (*yérsey*). También lo son aquellas que terminan en *n*, *s* ante vocal o vocal y no llevan tilde (*colon*, *asas*, *caso*).

Podemos, pues, sintetizar estas reglas diciendo que la sílaba tónica de las palabras con acento ortográfico es aquella con la tilde. Las palabras sin tilde solo son llanas o agudas. Si acaban en *y*, en *n* o *s* ante vocal o en vocal, son llanas. El resto son agudas. A esto debemos añadir que los adverbios en *-mente* llevan un acento secundario además del principal. El principal es el del lexema y el secundario el de mente, de acuerdo este a la regla general para la palabra íntegra y ambos por separado (Quilis, 1984/2013, p. 26).

Por lo cual, atendiendo a que este género de sabandijas que llaman poetas son nuestros prójimos y cristianos —aunque malos—, viendo que todo el año idolatran mujeres y hacen otros pecados más enormes, mandamos que la Semana Santa recojan a los poetas públicos y cantoneros, como a malas mujeres y que los prediquen para convertirlos...

Francisco de Quevedo, *Premática del desengaño contra poetas güeros*

Métrica 5

5.1 La lengua de los poetas

El lenguaje poético del Siglo de Oro tuvo uno de sus máximos exponentes en el teatro. La representación dramática no tenía métrica propia, sino que recurría a las formas ordinarias de la poesía común (Navarro Tomás, 1991, p. 251). Por este motivo, resulta obligado abordarla de la misma manera que lo hacemos con la lírica.

La palabra *poesía* deriva de poiesis, *producción*, esto es el acto de crear. En sentido estricto, tanto la lírica como la narración y el teatro son composiciones poéticas, ya que todas ellas son actos de creación en el sentido al que se refiere la palabra *poesía*. Aristóteles apunta en su *Poética* (1488b20-24) que la poesía surge de nuestra predisposición para copiar y es nuestra familiaridad con el ritmo y la armonía lo que nos lleva a improvisar imitaciones. Difieren, sin embargo, en el modo en que se materializa esta creación, si se relata o se imita, si se cuenta o se enseña. No obstante, el sabio macedonio establece una relación genética entre las formas líricas y las dramáticas.

> Y, entre los antiguos, unos fueron poetas de versos heroicos y otros de yambos [...]. Una vez aparecidas la tragedia y la comedia, los que tendían a una y a otra poesía según su propia naturaleza, unos, en vez de yambos, pasaron a hacer comedias, y los otros, de poetas épicos se convirtieron en autores de tragedias, por ser estas formas de más fuste y más apreciadas que aquellas [...]. Habiendo, pues, nacido al principio como improvisación —tanto ella como la comedia; una gracias a los que entonaban el ditirambo, y la otra, a los que iniciaban los cantos fálicos, que todavía hoy

> permanecen vigentes en muchas ciudades—, fue tomando cuerpo, al desarrollar sus cultivadores todo lo que de ella iba apareciendo [...]. Por otra parte, la amplitud, partiendo de las fábulas pequeñas y de una dicción burlesca, por evolucionar desde lo satírico, se dignificó tarde, y el metro se convirtió de tetrámetro en yámbico. (Arist. *Poet.*, 1488b28-1449a-30)

No debemos, por tanto, caer en la trampa, de confundir la poesía en general con una de sus expresiones, la lírica, como se suele hacer comúnmente, y menos con las formas externas de esta. La versificación teatral, al contrario que la lírica, no está al nivel mimético, sino que los personajes se expresan como ellos mismos, están *hablando* —excepcionalmente podría estarlo si los actores declaman una poesía o cantan una canción, pero sería literatura dentro de la literatura; su equivalente dramático sería que los actores interpretaran a actores representando una función—. El verso se halla en la capa comunicativa entre el actor y el espectador, lo que ejemplifica de nuevo la doble enunciación del texto teatral, lo que el espectador reconoce como tal y acepta como parte de la naturaleza estética de la obra (Schaeffer, 1995, p. 618). En definitiva, la rima teatral no es mimética sino estética.

> En el teatro de la temprana Edad Moderna las personas usan un lenguaje rimado. No me parece una cosa secundaria, de la misma manera que el lenguaje en verso y la métrica no deberían ser cosas secundarias y es triste que hoy lo sean. El lenguaje rítmico consigue en el teatro lo mismo que consiguen otros modos [...]. Se produce como una tensión entre la construcción «natural» de la oración y la forma métrica [...]. ¿Cómo afecta esto a los sentimientos del espectador? Precisamos ahora esta pregunta: ¿Cómo afecta el lenguaje oscilante según ritmos fijos en los visitantes del teatro? ¿Qué aporta el ritmo cuando se trata de que olviden los sentimientos que han traído consigo y asuman los sentimientos fingidos de los intérpretes? (Aichinger, 2013, p. 133, traducción propia)

Así pues, si hemos de distinguir entre géneros, la concepción clásica resulta particularmente apropiada para el teatro áureo porque precisamente la forma externa que suele asociarse a la lírica es un elemento destacado del drama. Sin embargo, la configuración versificada va más allá del simple adorno. Esta no solo desempeña una función estética, sino que sirve como armazón estructural, como vimos en el capítulo anterior, «la métrica tiene en el teatro del Siglo de Oro una función estructural en la que es relevante el cambio de forma métrica» (Arellano, 2007, p. 68).

Sin embargo, en su composición sonora, los versos de la poesía lírica y la dramática no difieren en lo esencial. La estructura de un poema puede representarse mediante relaciones numéricas, esto es, el lenguaje poético unifica lengua y números (Aichinger, 2013, p. 45). En otras palabras, admite ser medido y cuantificado, no en vano nos referimos a ello con el término *metro*.

¿Pero qué es lo que medimos? Esta es la primera cuestión que hemos de resolver antes de embarcarnos en una mensura. Empezaremos para ello por el elemento mínimo, que es el verso. Caramuel indica que «los Poemas se componen de estrofas; las Estrofas de versos; los Versos de Palabras; las Palabras de Sílabas, y estas finalmente de Letras. Así, la mínima parte del Poema es la letra» (Caramuel Lobkowitz, 1665/2007, p. 35, mayúsculas en el original). No obstante, debemos precisar esto. La *letra* de Caramuel —nosotros la llamaríamos hoy *fonema*— no asume una función aislada, ya que es incapaz de crear un ritmo si no es en combinación con otros sonidos. La palabra sí podría hacerlo, aunque no necesariamente, en tanto que para ello se necesita un número mínimo de sílabas que no todas las palabras reúnen. De esta manera, la unidad mínima del poema es el verso, si bien este puede descomponerse en unidades menores pero insuficientes para constituir poemas. Más adelante, veremos cómo se agrupan. Una vez tenemos nuestro objeto de estudio, debemos determinar las unidades en función de las cuales asignaremos el valor a la medida de una magnitud. Aquí hemos de detenernos para considerar una particularidad de la lengua española. Debemos retrotraernos a la poética clásica, surgida en el ámbito de lenguas de ritmo acentual como el latín y el griego, basadas en la cantidad como ocurre hoy, por ejemplo, con el inglés (Quilis, 1984/2013, p. 21).

> Había sílabas *breves* o de rápida pronunciación, y otras *largas*, a manera de notas sostenidas, y por el tiempo que se gastaba en pronunciarlas, estaban las segundas con respecto a las primeras, que daban el tipo, en la proporción aproximada de 1 a 2.
>
> Seguramente los griegos y romanos al hablar, gozábanse en pronunciar con prolongado aliento las sílabas largas, y velozmente las breves, de un modo aproximado a lo que se practica en el canto gregoriano, y en los recitados de la ópera, aunque no extremando lo largo y breve, sino atemperando lo uno a lo otro, en la proporción de 1 a 2, y dentro de los límites naturales del aliento sonoro. (Bello, 1835/1981, p. 307)

Por este motivo, la unidad de las poéticas clásicas es el pie métrico. Lo que hace un hexámetro es estar compuesto de seis pies métricos. La lengua española, al contrario

que aquellas, es de ritmo silábico, por lo que su versificación ha de adaptarse a su naturaleza (Herrero Prado, 1996, p. 20; Torre, 1999, pp. 27-30; Quilis, 1984/2013, p. 21). Esto dota al verso español de unas características peculiares y diferenciada de los propios de otras lenguas de ritmo acentual (Cantero Serena, 2002, pp. 54-55), si bien no han faltado los intentos de ilustrísimas plumas por trasladarlo (Fernández Rodríguez, 2021). La consecuencia más importante de esto es que el metro castellano no lo medimos en pies sino en sílabas. En palabras de Navarro Tomás (1991, p. 34), «aplicar al estudio de una métrica extranjera el criterio de los hábitos adquiridos en la lengua propia es situarse en un equivocado punto de vista». El prurito clasicista de Díaz Rengifo (1592/2012, p. 174) le lleva a contemplar la cantidad, solo para, de inmediato, igualarla con el acento y señalar este como única distinción válida. No obstante, es usual que se intercalen pasajes de ritmo acentual en el teatro áureo porque incluye a menudo partes cantadas. En ocasiones, estos fragmentos dejan intuir su naturaleza métrica en ciertos indicios, como la irregularidad en el número de sílabas, que queda compensada por el ritmo acentual (Torrente, 2016, pp. 416-417). Desafortunadamente, incluso estando dispuestos a poner el pie en el terreno de los musicólogos y tratar cada estrofa cantada de manera individual, la tarea resultaría probablemente fútil debido a la escasez de las fuentes (pp. 323-324).

Por otra parte, la sílaba métrica tiende a corresponderse con su realización fonética —sea esta el fruto de una licencia poética o del normal uso del lenguaje— más que a una expresión fonológica idealizada de un segmento aislado (Domínguez Caparrós, 2014, p. 53). La poesía se aprovecha de esa característica para introducir accidentes suprasegmentales, como la sinalefa o la diéresis, ubicuos en el texto en verso como medio de ajuste silábico (Domínguez Caparrós, 2014, pp. 55-59). Así, para abordar el teatro aurisecular, hay que considerar el metro según su silabación fonética y tratar las irregularidades considerando los medios de compensación métrica. Por fortuna, las reglas de la división silábica del español son claras (Hualde, 1991), lo que simplifica la tarea de sistematizarlas en un programa informático. Ya discutimos estos fenómenos en el capítulo dedicado a presentar los fundamentos teóricos de la aproximación fonológica. Por lo tanto, daremos por supuestas también en la métrica esas nociones.

La sílaba del metro español se caracteriza por la presencia o ausencia de acento, y esta propiedad lo faculta para determinar la configuración rítmica del verso. Esto es, entendemos el verso como una serie de sílabas que adoptan un valor binario: tónica/átona. La distribución de estos valores puede responder a un patrón determinado, si bien este no resulta absolutamente necesario. En otras palabras, la única regla fija es que el último acento de un verso n-sílabo va sobre la sílaba $n - 1$, a partir de la cual no se consideran las sílabas inacentuadas, pero, dependiendo del tipo de

verso, los acentos suelen recaer en determinadas posiciones. Así, por ejemplo, una característica habitual de los versos endecasílabos es emplazar un acento en la sexta sílaba. Sin embargo, no ocurre igual en, por mencionar algunos, los sáficos puros o los de gaita gallega. Asimismo, existen otros acentos facultativos que caracterizan un tipo de verso. Por ejemplo, un verso heroico puro lleva acentos en la segunda, sexta y décima sílaba. Adicionalmente, el endecasílabo heroico admite otros acentos en la cuarta y la octava, de cuyas combinaciones dependerá que sea pleno, corto o largo.

Debemos advertir que en los textos teatrales existe una excepción a lo dicho. Muchas obras incluyen versos cantados que, como es lógico, no se atienen a la prosodia española sino a la musical. Por lo tanto, sería posible alargar o acortar sonidos, juntar o desdoblar sílabas. En estos casos, el texto tendría más que ver con los ritmos acentuales que con el silábico castellano, pero no podrían conocerse a ciencia cierta los pies que lo constituyen. Sin otra indicación que la letra, sin conocer la música, no cabe hacer más que conjeturas.

5.2 Elementos métricos

Como dijimos, el verso es la unidad poética mínima y, por lo tanto, qué más apropiado que comenzar por él. Navarro Tomás (1991, pp. 34-35) lo define como «una serie de palabras cuya disposición produce un determinado efecto rítmico». Este efecto, dice, se fundamenta en los acentos producidos al espirar las palabras. En la lengua española, el tono o la cantidad no tienen relevancia rítmica. Otros elementos, como la armonía vocálica o la aliteración, solo sirven, a lo sumo, de refuerzo, ya que no tienen capacidad de alterar el ritmo, mientras que el acento por sí mismo es suficiente para definirlo. Además de los acentos que moldean el verso, las sílabas y pausas marcan sus límites y divisiones internas. En cualquier caso, para considerar un verso como una unidad rítmica independiente, este ha de tener cuatro o más sílabas.

El verso español se atiene a la métrica románica y, como tal, sus elementos constitutivos elementales son el *acento de intensidad*, la *pausa métrica* y el *cómputo silábico* (Baehr, 1973/1997, p. 22). A partir de estos componentes básicos es como se construye el resto. Aquellos que no derivan directamente de ellos pueden ser complementarios pero no determinantes.

La poesía tiene raíces musicales. De esta manera, los acentos del verso se distribuyen formando un patrón rítmico. Según Navarro Tomás (1991, pp. 35-36), además del acento final, el verso poseería necesariamente otro apoyo rítmico al comienzo. El *periodo rítmico interior* sería así la porción del verso comprendida entre la sílaba que

recibe el primer apoyo rítmico hasta la que antecede al último acento. Las sílabas átonas que preceden al primer apoyo rítmico actuarían como anacrusa o antecompás. El *periodo de enlace* comenzaría, por su parte, en la última sílaba acentuada del verso, las sílabas que la sigan, la pausa al final del verso y, si las hubiera, las sílabas en anacrusa del verso siguiente (15). Estos dos periodos se repetirían periódicamente al modo de las composiciones musicales.

(15) $\overbrace{Je-ru-sa}^{Anacrusa}-\overbrace{lén\ las\ rui-nas\ del\ cal}^{Periodo\ rítmico}-\overbrace{de-o}^{Enlace}.$

(Moreto, *La cena del rey Baltasar*, v. 124)

Los periodos se organizan en *cláusulas* de dos o tres sílabas, cuyo equivalente en las lenguas de ritmo acentual es el pie métrico. No debe confundirse con este porque, como indicamos, la cláusula española es acentual y no cuantitativa. Todas las cláusulas de los versos españoles son disilábicas o trisilábicas. Las primeras son de ritmo *trocaico* si el acento recae en la primera o *yámbicas* si lo hace sobre la segunda. En cuanto a las cláusulas trisílabas, serán *dactílicas* si el acento recae en la primera sílaba, *anfibráquicas* si en la segunda y *anapésticas* si es en la tercera (Bello, 1835/1981, pp. 141-142). Navarro Tomás (1991, p. 37) nota que la interpretación del pie es subjetiva; lo ilustra con, entre otros, el primer verso del celebérrimo poema de Bécquer «De salón en el ángulo oscuro». Lo mismo sucede con el verso del ejemplo 16, que se lee tanto como anapéstico (16a) como dactílico con anacrusa (16b).

(16) *Caminad, caminad a la sombra*

a. `--+|--+|--+|-||`

b. `--|+--|+--|+-||`

(Calderón, *El viático cordero*, v. 1390)

Esto ofrece ya una primera pista de que el modelo de periodo rítmico de Navarro Tomás presenta algunos puntos problemáticos. Nos detendremos en ellos más adelante.

La *cantidad silábica*, si bien no se tiene en cuenta como medida del metro, tampoco es una magnitud inexistente en español, a pesar de no tener la relevancia de la que goza en otros sistemas métricos. Las sílabas tienen diferente duración, como se aprecia en los sonogramas. Esto suele compensarse en las cláusulas, cuya duración suele ser más uniforme, especialmente cuando se trata de dos del mismo tipo, de forma que estas se ajustan al compás (Navarro Tomás, 1991, p. 37). Este fenómeno carece de expresión a nivel fonológico y en la declamación se compensa alterando la realización de los sonidos para ajustar el periodo. Por este motivo, lo mencionamos aquí, pero no lo

tendremos en cuenta para la parte práctica del trabajo, que, después de todo, es un modelo fonológico que no aspira a representar exactamente el sonido físico.

A los elementos rítmicos fundamentales hay que añadir otros que, sin tener una función estructural, contribuyen a realzar los componentes básicos y aumentar la eufonía del verso. Podemos contar entre estos, por ejemplo, las aliteraciones, la simetría, la anáfora, el *lexaprén*, *macho e femea* y el *manzobre* (Domínguez Caparrós, 2014, pp. 126-129).

Los versos se clasifican atendiendo a varios criterios, en los que influye tanto el cómputo silábico como la posición de los acentos. Llamamos *métricos* a los versos que tienen un número fijo de sílabas y *amétricos* a los que no. Si los periodos del verso se distribuyen según una estructura prefijada, el verso será *monorrítmico* y será *polirrítmico* si esta disposición varía. Entre los versos amétricos, son *acentuales* aquellos compuestos por un número variable de cláusulas del mismo tipo, de modo que el ritmo se mantiene, mientras que serán *libres* los que no siguen un patrón. En el término medio entre ambos, se hallan los versos *fluctuantes* (Navarro Tomás, 1991, p. 39). Al final de cada verso hay una *pausa* y, en versos compuestos, hay además otra interna. Las pausas van precedidas de terminación oxítona, paroxítona o proparoxítona (Navarro Tomás, 1991, pp. 39-40).

La rima, aunque usada con muchísima cautela en la poesía clásica culta, abundaba en las canciones populares latinas. Sin embargo, su auge llegó en la Edad Media y, sobre todo, con la trova, que tomó la rima consonante como una de las máximas expresiones de su arte. La asonancia quedó relegada a las formas populares, pero cobró importancia en algunos géneros típicamente españoles (Navarro Tomás, 1991, pp. 40-41).

La agrupación de versos encuentra una unidad de orden superior en la estrofa. Además, la estructura del verso se apoya ocasionalmente en determinados recursos facultativos. El poeta tiene libertad para jugar con la distribución de las palabras y producir efectos como repeticiones o paralelismos. También disfruta la potestad de hacer corresponder las vocales en el interior del verso, particularmente aquellas en posición acentuada, de manera que se produzca una armonía vocálica o adaptar los sonidos consonánticos para evocar emociones que convengan al verso (Navarro Tomás, 1991, pp. 41-42).

5.3 Ritmo

El ritmo de un verso depende de sus acentos. El acento es un realce sonoro en determinadas sílabas para hacerlas destacar sobre el resto en los segmentos de lengua hablada. El primer tipo de acento que encontramos al escuchar hablar español es el *prosódico*. Sin embargo, debemos matizar aquí su carácter, en función del grado de complejidad de la verbalización. Así, si estamos tratando una palabra aislada, está llevará siempre un acento prosódico en mayor o menor medida, el acento *léxico*. Si, por el contrario, la palabra se halla en contexto, se distingue entre palabras tónicas y átonas. Este acento del habla normal también se mantiene en la pronunciación del verso, salvo en contadas ocasiones donde el poeta se concede una licencia para cuadrar el metro (Baehr, 1973/1997, p. 24).

> Acento es un sonido con que herimos y levantamos más una sílaba cuando la pronunciamos y nos detenemos más en aquella que en cualquiera de las otras de un mismo vocablo, como cuando decimos «agúdo», «poéta», herimos la «u» y la «e» y las levantamos sobre todas las otras sílabas. Y hablo aquí solamente del acento que predomina en cada dicción, no del grave o del circunflejo que ponen los latinos, los cuales nos conciernen poco para lo que al presente tratamos. Este acento predominante no puede ser más que uno en cada vocablo, ni se puede hallar sino en la sílaba última, como «perdí», «gané»; o en la penúltima, como «máno», «sagráda»; o en la antepenúltima, como «próspero», «alhóndiga». Llamo sílaba última la postrera de cada dicción; penúltima, la más cercana a la última; y antepenúltima, la que es tercera comenzando de la última. Supuesto este fundamento, aquella sílaba es larga que se pronuncia con el acento predominante, y todas las demás que estuvieren delante o se siguieren después della en un mismo vocablo serán breves, como en «caballéro» la «e» es larga y las demás breves, en «dignísimo» la «ni» es larga, las demás breves. Entre las que están delante contamos muchas dicciones de una sílaba, las cuales no tienen acento predominante, sino que son breves como las demás del vocablo a quien se arriman, como «la», «lo», «me», «te», «se», «sin», «con», «a», «de», «por», «en», etc. Así como «en vida con ampáro», la «en» y la «con» no tienen acento por sí, sino son breves como las demás a quien se ayuntan. Podría decir alguno que corre en «dignísimo» la «si» que no la «di», y lo mismo parecerá en todos los vocablos que tuvieren el acento en la antepenúltima. Pero no es así, antes todas son iguales; y la causa de parecer aquella más breve que las otras

> es el haber precedido inmediatamente la larga, en la cual, como se subió la voz, cuando baja a la breve que se sigue, parece que se despeña y que corre más por ella que por las otras, como en realidad en todas gaste un mismo tiempo. (Díaz Rengifo, 1592/2012, pp. 174-175)

Como ya apreció Díaz Rengifo, el acento prosódico depende del tipo de palabra y su origen histórico. Las observaciones del tratadista son certeras en tanto que, efectivamente, las preposiciones y los pronombres que funcionan como complemento verbal no preposicional son inacentuadas —de ahí que los últimos reciban el nombre de *pronombres átonos*—, mientras que, por el contrario, los sustantivos sí llevan acento (Quilis, 1984/2013, pp. 22-26). Este además recae en la última, penúltima o antepenúltima sílaba de la palabra. De esta manera, cada una lleva un solo acento prosódico, con la excepción de los adverbios terminados en *-mente* (Quilis, 1984/2013). No llama la atención que, ya a finales del siglo XVI, hubiera sido descrito el sistema con gran precisión, de forma que el mismo concepto se ha ido transmitiendo hasta las poéticas contemporáneas sin experimentar cambios cualitativos sustanciales.

Esto es una generalización aplicable a la mayoría de los casos. En realidad, las marcas acentuales presentan diferencias sutiles y podría hablarse de una gradación del acento con diferentes niveles (Navarro Tomás, 2004, sec. 173). Así, cuando concurren varios acentos contiguos, algunas de las sílabas tónicas pierden sonoridad, lo que Paraíso (Paraíso, 2000, p. 81) denomina «desacentuación rítmica secundaria». Esto justifica, por ejemplo, la taxonomía del octosílabo de Navarro Tomás (1973/2014, p. 64 y ss.), que veremos más adelante. El caso contrario también se cumple: una serie de sílabas inacentuadas tampoco tiene un nivel de intensidad uniforme.

Debemos advertir, no obstante, que los acentos contiguos también se emplean como *acento enfático*, un recurso expresivo (Paraíso, 2000, pp. 83-84). En el texto dramático esto se hace más evidente si cabe. Piénsese en una serie de réplicas y contrarréplicas monosilábicas como las del ejemplo 17, con un ritmo ++-++++.

(17) ERÍFILA Yo digo tal.
LEONATO Sí.
ERÍFILA ¿Yo?
LEONATO Sí.
(Lope de Vega, *Los locos de Valencia*, v. 233)

En este caso, no parece sensato reducir los acentos sobre las tablas, sino más bien todo lo contrario, que cada actor pronuncie su línea realzando la intensidad para

adecuarse a lo que demanda la acción, a despecho del ritmo ideal que pudiera tener el verso. Por ese motivo, solo consideraremos las reducciones *a posteriori*, como una posibilidad adicional de clasificación. Por el contrario, un único recitador se encontraría en dificultades para reproducir el diálogo a la manera de dos actores contando con solo un aparato fonador.

La cuestión necesita ser investigada en detalle, pues implica aspectos estilísticos, semánticos, fonológicos e incluso fonéticos (Sánchez Jiménez, 2017, pp. 57-58). Sánchez Jiménez se aviene a partir de los acentos gramaticales entretanto como solución de compromiso. Nosotros optaremos por lo mismo: se trata de definir un modelo que, incompleto, igual que todos, no abarca la totalidad del objeto de estudio; sin embargo, esa limitación facilita concentrarse en una de sus facetas.

Palabras acentuadas e inacentuadas

En la parte práctica necesitaremos determinar qué palabras del verso portan acento métrico.Nos guiaremos, pues, por ciertas reglas gramaticales para distinguirlas. Esta sección presenta tanto esas reglas y sus excepciones.

En español, las palabras solo tienen una sílaba acentuada (Quilis, 1984/2013, p. 22), con excepción de los adverbios terminados en *-mente* (18), que contienen dos sílabas tónicas (Quilis, 1984/2013, p. 26)[67]. El resto de los adverbios se acentúan una sola vez (19), salvo cuando cumplen otra función, pues ahí serán son átonos en algunas ocasiones. Un caso especial es *tanto* y su forma apocopada *tan*, ya que la primera es tónica y la segunda átona (Navarro Tomás, 2004, 169a). También es irregular *medio*, puesto que es inacentuada cuando actúa como adverbio (20), si bien es tónica (21) como adjetivo (Navarro Tomás, 2004, sec. 169c).

(18)	*so·la·men·te*	+-+-
(19)	*so·lo*	+-
(20)	*me·dio bo·rra·cho*	--|-+-
(21)	*me·dio li·tro*	+-|-+-

El verbo es la única clase verbal que va siempre acentuado en todas sus modalidades (Navarro Tomás, 2004, sec. 166). Además de cuando es predicativo, también es tónico en sus funciones semipredicativas o copulativas (22), incluso en su función auxiliar (23).

[67]Emplearemos la notación + para las sílabas acentuadas y - para las inacentuadas, en lugar de los símbolos de breve y larga, que no describen propiedades de la sílaba española.

(22) *es sue·ño* +|+-

(23) *ha di·cho* +|+-

Las formas nominales suelen ser también palabras acentuadas en español (24, 25) salvo en excepciones concretas (Navarro Tomás, 2004, sec. 167). Estas irregularidades ocurren en las fórmulas de tratamiento como *don* o *fray* antepuestas al nombre (26), la primera parte de una expresión vocativa compuesta (27), en caso de nombres compuestos, aquellos que anteceden al último (28), nombres prepositivos (29)

(24) *To·más* -+

(25) *el san·to* -|+-

(26) *San·to To·más* --|-+

(27) *¡San·to Dios!* --|+[68].

(28) *Jo·sé To·más* --|-+

(29) *santo del día* --|-|+-

Las preposiciones son átonas (30) excepto *según*, con independencia de su función (Navarro Tomás, 2004, sec. 170).

(30) *a ca·sa* -|+-

Los artículos determinados son inacentuados (31), al contrario que los indeterminados, que son tónicos (32). Constituyen una excepción estos últimos (Navarro Tomás, 2004, sec. 170) cuando sirven para dar una valoración aproximativa (33).

(31) *el hom·bre* -|+-

(32) *un hom·bre* +|+-

(33) *u·nas tres ho·ras* --|+|+-

Los adjetivos léxicos, esto es, los calificativos (34) y relacionales (35) llevan acento, no así todos los de las clases cerradas adjetivales. Los indefinidos son acentuados, pero el distributivo *cada* suele tener una relación átona (Navarro Tomás, 2004, 168.f)

(34) *san·to se·cre·to* +-|-+-

(35) *co·ro·na im·pe·rial* -+-|--+

[68]En este caso, en alternancia con el ritmo ordinario (+-|+) si se enfatiza la primera sílaba.

Otro tipo de palabras con uso adjetivo varían según el caso. De esta manera, los numerales, tanto ordinales como cardinales, son siempre tónicos por sí mismos (36). Sin embargo, al igual que ocurre con los sustantivos, el acento recae en el último elemento si estos constituyen una serie, siendo los precedentes átonos (37). Lo mismo sucede si desempeñan una función nominal (38).

(36) *pri·mer pre·mio* -+|+-

(37) *trein·ta mil* --|+

(38) *el trigésimo noveno* -|----|-+-

Los demostrativos son siempre tónicos, tanto en su función pronominal como adjetival.

(39) *es es·te* +|+-

(40) *es·te tex·to* +-|+-

El acento varía en los posesivos dependiendo de su función. Si actúan como determinante (41), son definidos (RAE, 2010a, 18.2.1a) y, como los personales, también átonos. Por el contrario, serán tónicos si desempeñan una función adjetival (42). Esto se deduce de la posición porque, en función determinativa, el posesivo ocupa una posición prenominal, mientras que es posnominal en función adjetival (RAE, 2010a, 18.2.2g).

(41) *nues·tro re·me·dio* --|-+-

(42) *re·me·dio nues·tro* -+-|+-

Las partículas exclamativas e interrogativas son tónicas (43), pero su contrapartida relativa es átona (44).

(43) *¿Cuán·do vie·nes?* +-|+-

(44) *Cuan·do ven·ga.* --|+-

Los pronombres personales son tónicos en el caso nominativo (45) y en el oblicuo (46) y átonos en dativo (47) y acusativo (48). Los pronombres reflexivos también son átonos (49). En los pronombres recíprocos con desambiguación, serán tónicos o no obedeciendo a las reglas de sus componentes, esto es, el componente reflexivo (*nos*, *os*, *se*) será átono mientras que tanto el nominativo como el oblicuo (incluyendo *sí* para la tercera persona) o adjetival (*mismo*) serán tónicos.

(45) *él vie·ne* +|+-

(46) *pa·ra mí* --|+

(47) *le da al·go* -|+|+-

(48) *lo co·ge* -|+-

(49) *se pei·na* -|+-

Son átonas, según Quilis (1984/2013, p. 24), las conjunciones copulativas (50), disyuntivas (51), la polivalente *que* (52), adversativas (53), causales (54), consecutivas (55), condicionales (56) y concesivas (57). No obstante, pueden ser tónicas en función adverbial (58). Constituyen la excepción las disyuntivas *ora*, *ya* y *bien* (59), la consecutiva *así* (60) y la temporal *apenas*, así como las compuestas adversativas (61), temporales (62), condicionales (63) y concesivas (64).

(50) *ni u·no ni o·tro* -|+-|+-

(51) *él o yo* +|-|+

(52) *da·le que te pe·go* +-|-|-|+-

(53) *sí, pe·ro no* +|--|+

(54) *pues·to que te gus·ta, qué·da·te·lo* --|-|-+-|+---

(55) *ya tie·nes lo tu·yo, con·que ve·te* +|+-|-|+-|--|+-

(56) *si pue·des, llá·ma·me* -|+-|+--

(57) *no lo creeré aunque lo vea* +|-|--+|--|-|+-

(58) *¿Y esto?* +|+-

(59) *ora unos, ora otros* +-|+-|+-|+-

(60) *no co·rre·rí·a a·sí lo fus·ti·ga·ran* +|--+-|-+|-|--+-

(61) *a·rre·gla·da no obs·tan·te in·for·mal* --+-|+|-+-|--+

(62) *se mar·chó no bien lle·gó* -|+-|+|+|-+

(63) *haz·lo con tal que fun·cio·ne* +-|-+-|-+-

(64) *a·cep·to mal que me pe·se* -+-|+|-|-|+-

Los adverbios empleados en función conjuntiva (65) o como parte de una locución conjuntiva (RAE, 2010a, 31.6.1b) se rigen por la regla general de la conjunción (66) y, por lo tanto, son átonos (Navarro Tomás, 2004).

(65) *pien·so, lue·go e·xis·to* +-|--|-+-

(66) *no ha·ce más que lo que de·be* +|+-|-|-|-|-|+-

Aquí identificamos un patrón considerando las conjunciones a partir de sus componentes individuales. Aquellas compuestas de palabras que individualmente son átonas serán asimismo átonas, mientras que otras que contengan una palabra que por sí misma es tónica conservarán normalmente su acento. De este modo, en las adversativas compuestas, *en tanto* conserva el acento en su segunda sílaba por ser tónica, pero *en cuanto* es átona en todas sus sílabas.

Acento métrico

Además del acento prosódico, para caracterizar el ritmo adecuadamente hay que tener en cuenta el *acento métrico* o *rítmico*. Se define este acento como los golpes tónicos intrínsecos al ritmo particular de un verso. De acuerdo con Baehr (1973/1997, pp. 8-9), en cada verso se hallan como mínimo dos acentos rítmicos, uno al final del verso en la rima y otro que varía su posición entre las cuatro primeras sílabas. Los acentos prosódicos no se traducen siempre en acentos métricos ni viceversa.

El verso español puede reproducir patrones rítmicos, aunque no necesariamente. Los versos endecasílabos (Cuadro 5.3), por ejemplo, se prestan a que el poeta aplique ciertos patrones para obtener un efecto estético de manera que, aparte del acento fijo en la décima sílaba, suele acentuar la sexta sílaba en la mayoría de los casos, pero un tercer acento facultativo en las cuatro primeras sílabas determina si el verso es enfático, heroico, melódico o sáfico, o incluso otro tipo de verso menos frecuente con otras combinaciones (Quilis, 1984/2013, p. 22). Los octosílabos áureos, por su parte, tienden a seguir un ritmo trocaico, dactílico o una de dos combinaciones de estos (Torre, 2000, pp. 93-94; Navarro Tomás, 1991, pp. 274-275). La realización del ritmo métrico, sin ser del todo coincidente con los acentos prosódicos, suele estar determinada por estos (Baehr, 1973/1997, p. 26), por lo que sería posible deducirla a partir de la prosodia.

De la misma manera que los versos endecasílabos se clasifican tradicionalmente en categorías según la posición de los acentos métricos, otros tipos de verso también se categorizan de forma análoga. En el caso del octosílabo, el más común en el teatro áureo, hay clasificaciones, como las de Navarro Tomás (1973/2014, 64 y ss.) o Varela Merino y col. (2005, pp. 149-158). Para el primero, los versos octosílabos podrían reducirse las sesenta y cuatro combinaciones rítmicas posibles de entre uno y siete acentos a tan solo cuatro clases métricas (Cuadro 5.1): trocaica (`--|+-|+-|+`)[69], dactílica (`+--|+--|+`), mixta a (`-|+-|+--|+`) y mixta b (`-|+--|+-|+`). La clasificación de Varela Merino y col. reconoce versos heroicos (`-+-+--+` y sus variantes `-+--+-+` y

[69]No marcamos las sílabas átonas que siguen al último acento, de forma que bajo la notación `-+-+--+` se hallan igualmente `-+-+--+-` y `-+-+--+--`.

-+----+), melódicos (--+---+ y sus variantes +-+-+-+, --+-+-+ y +-+---+), dactílicos (+--+--+), sáficos (---+--+), enfáticos (+---+-+ y +-----+) y vacíos (----+-+ o ------+).

Trocaico --|+-|+-|+	Dactílico +--|+--|+	Mixto a -|+-|+--|+	Mixto b -|+--|+-|+
7			
3-7	1-7	2-7	
5-7	4-7		
6-7			
1-3-7	1-4-7	1-2-7	1-2-7
1-5-7	1-6-7	2-4-7	2-5-7
2-3-7	4-5-7		2-6-7
3-4-7	4-6-7		
3-5-7			
3-6-7			
5-6-7			
1-2-3-7	1-2-4-7*	1-2-4-7	1-2-5-7
1-3-4-7	1-3-4-7*	2-3-4-7	1-2-6-7
1-3-5-7	1-4-5-7	2-4-5-7	
1-3-6-7	1-4-6-7	2-5-6-7	
1-5-6-7	4-5-6-7	2-4-6-7	
2-3-5-7			
2-3-6-7			
3-4-5-7			
3-4-6-7			
3-5-6-7			
1-2-3-4-7	1-3-4-6-7	1-2-4-5-7	1-2-5-6-7
1-2-3-5-7	1-2-4-6-7	1-2-4-6-7†	
1-2-3-6-7	1-4-5-6-7		
1-3-4-5-7	2-3-4-5-7‡	2-3-4-5-7	
1-3-5-6-7		2-3-4-6-7	
2-3-5-6-7		2-4-5-6-7	
3-4-5-6-7			
1-2-3-4-5-7‡	1-2-4-5-6-7	1-2-3-4-5-7	
1-2-3-5-6-7	1-2-3-4-6-7	1-2-3-4-6-7†	1-2-3-5-6-7†
1-3-4-5-6-7	1-3-4-5-6-7*		
2-3-4-5-6-7			
		1-2-3-4-5-6-7	1-2-3-4-5-6-7

* Si se da preferencia a la primera sílaba.
† Si se da preferencia a la segunda sílaba.
‡ Si se da preferencia a la tercera sílaba.

Cuadro 5.1: Clasificación del octosílabo de Navarro Tomás.

Para llegar a esta deducción a partir de los rasgos prosódicos, debemos considerar que los acentos rítmicos coinciden normalmente con sílabas que llevan acento prosódico por sí mismas. Sin embargo, estos no siempre son *activos*, sino que pueden ser *pasivos*

si no ejercen una función de apoyo rítmico. A la inversa, las sílabas débiles acarrean acentos rítmicos en ocasiones. Así, a la que le corresponde el último acento no es tónica por si misma, lo recibe igualmente. En el interior, la situación resulta un poco más confusa, como se infiere de las palabras de Navarro Tomás:

> Son variables los acentos prosódicos; son fijos los apoyos rítmicos en que se funda la estructura de cada tipo. Si el verso no posee más acento prosódico que el de la sílaba séptima o si, por el contrario, tienen acento prosódico todas las sílabas anteriores, los apoyos inexpresados se deducen del valor relativo de los vocablos y del ritmo de los versos precedentes. (1973/2014, p. 52).

Dicho de otra manera, en caso de ambigüedad, el verso por sí mismo resulta insuficiente para deducir una estructura. No es esta la única objeción, pues Baehr (1973/1997, pp. 29-30) plantea asimismo otras, cuyos motivos derivan en gran medida de la premisa —que nos comprometimos arriba a revisar— de la anacrusa en las sílabas que preceden al periodo interior. Empieza poniendo en duda que las sílabas anteriores al primer acento vayan necesariamente en anacrusa; arguye para ello que, de ser así, habría tendencia a evitarlas, cuando esto no ocurre. Considera Baehr aventurado atribuir a un ajuste métrico la pausa observada experimentalmente a final de verso, como hace Navarro Tomás, ya que esta responde igualmente a otros factores que el último no ha tenido en cuenta, como el sentido y función del sintagma. El mayor reparo de Baehr proviene de las implicaciones para la unidad del verso que provocaría el sistema de Navarro Tomás porque «el verso constituye unidad, que lo es de sentido, sintaxis y ritmo, señalada en su fin por el acento, la rima y la pausa» (Baehr, 1973/1997, p. 29).

Por todas estas razones, resulta poco fiable presentar una clasificación como datos extraídos del texto: a la evidencia, estaríamos añadiendo una buena parte de interpretación, lo que no corresponde a este trabajo, sino a quien tome sus resultados como herramienta. No obstante, al final del capítulo correspondiente de la parte práctica, presentaremos una posible solución para la clasificación.

5.4 Pausas

Un verso tiene, por lo general, una pausa larga obligada a su término. En versos de arte mayor, es común una pausa interna adicional que lo divide en dos partes. A esta pausa interna la llamamos *cesura* parencite[*s.v. cesura*]dominguez1985y cada una

de las partes se denomina *hemistiquio*. La diferencia entre una pausa y una cesura (Domínguez Caparrós, 1985, *s.v. hemistiquio*) es que «la primera permite el hiato y no la sinalefa, y la segunda, por el contrario, da lugar a la sinalefa y repugna el hiato» (Bello, 1835/1981, p. 151). Asimismo, al final de cada estrofa nos encontramos con otra pausa obligatoria de mayor duración que la versal.

La pausa versal se ve alterada si la estructura morfosintáctica no coincide con la del verso, de forma que aquella se ubica entre dos partes que no admiten separación en circunstancias normales. Nos encontramos, por tanto, ante la tesitura de dividir un elemento sintáctico indivisible, lo que sería violento, o saltarnos la pausa métrica, lo que provocaría un efecto no menos violento. Hablamos entonces de *encabalgamiento*, a cuyo elemento inicial denominamos *encabalgante* y al segundo *encabalgado*. Quilis (1984/2013, pp. 81-92) clasifica esta figura según su posición en el verso, a los elementos que escinde y a la longitud del verso encabalgado.

Respecto a la posición del encabalgamiento, este será *versal* si ocurre entre el final de un verso y el comienzo de otro o *medial* si se halla entre dos hemistiquios. Considerando los elementos que afecta, será un *encabalgamiento léxico* si divide en dos una palabra. Por el contrario, si lo que separa es un grupo de palabras que no admiten pausa entre sí, el encabalgamiento será *sirremático*. En español constituyen sirremas tanto la unión de un sustantivo con un adjetivo como aquel seguido de complemento determinativo. Tampoco la combinación de verbo y adverbio y los tiempos compuestos admiten pausa. Los pronombres átonos, conjunciones y artículos con el elemento que determinan rechazan asimismo la pausa. Tanto la estructura interna de los sintagmas preposicionales simples como estos y el elemento que modifican se pronuncian sin interrupción, al igual que las oraciones adjetivas especificativas. Un caso particular del encabalgamiento sirremático es el que se da entre una oración adjetiva especificativa y su antecedente; lo llamamos aquí *encabalgamiento oracional*. Finalmente, cuando el encabalgamiento termina antes de la quinta sílaba del verso encabalgado, este será *abrupto*[70], mientras que será *suave* si el encabalgamiento se prolonga más allá.

Buscar una regla general para dirimir cuál es la resolución más apropiada del encabalgamiento requeriría un trabajo propio, lo que se escapa de nuestro propósito.

[70]Es usual encontrar *braquistiquios* formados mediante un encabalgamiento abrupto respetando la pausa rítmica.

5.5 Cómputo silábico

Una vez que tenemos las nociones elementales del acento y los ritmos que producen, las aplicamos al cómputo de las sílabas del verso. Para ello debemos tener en cuenta la forma de los accidentes comunes que caracterizan al verso. No entraremos en esta sección en licencias poéticas como la supresión o adición arbitraria de sílabas por el poeta. Si estas se encuentran señaladas explícitamente, no se hace necesario más que dejarse llevar por las indicaciones ortográficas. Si, por el contrario, no hay marca visible de una irregularidad arbitraria, tampoco habrá manera posible de inferirla a ciencia cierta a partir de la grafía, más allá de lo que es posible hacerlo las irregularidades de carácter accidental.

Atendiendo a su número de sílabas, aquellos versos de hasta ocho sílabas se denominan de *versos de arte menor* (Domínguez Caparrós, 1985, *s.v. verso de arte menor*), mientras que los versos mayores de esta medida se conocen como *versos de arte mayor* (*s.v. verso de arte menor*). En la notación más popular, se indican los primeros con una letra minúscula y a los segundos con una mayúscula.

(67) Lo primero, con garbo y con denuedo,
es entrar por la puerta de Toledo,
irse al balcón del Rey con gallardía,
[...] hacerle una profunda cortesía,
luego a las damas otras muy *perfetas*.
(Esas son cortesías con corvetas).
Terciar la capa con gentil decoro,
empuñar el rejón, salir el toro,
aguardarle cubierto,
darle en la nuca y ¡zas! dejarle muerto.
Que aquesto hecho con modo y sin recelos
Parecerá, don Cosme, de los cielos.

(Calderón, entremés *El toreador*, vv. 131-142)

Excepcionalmente, los versos se *quiebran*, de manera que, si se divide el verso regular en dos partes al modo de los hemistiquios de uno de arte mayor, el patrón del quebrado corresponde con la primera de las dos partes. Así, el heptasílabo es el quebrado del endecasílabo y el tetrasílabo del octosílabo (Díaz Rengifo, 1592/2012, pp. 178-180; Domínguez Caparrós, 1985, *s.v. quebrado*). En el ámbito dramático este es un recurso que, empleado con maestría, puede aportar matices importantes a la interpretación.

Así, por ejemplo, Hiergeist (2018, p. 186) llama la atención sobre cómo se vale Calderón del quebrado en un entremés para provocar una pausa dramática antes de alcanzar punto culminante del discurso (67). Saludar, prepararse, aguardar... ¡y zas!

Accidentes métricos por último acento versal

El fenómeno más importante está condicionado por la última palabra del verso, que normalmente llevará acento métrico con independencia de que tenga o no prosódico. La sílaba tónica de esta palabra es la *cumbre tonal* (Balbín Lucas, 1968, pp. 17-18) del verso y determina la longitud del metro, ya que la posición postónica subsiguiente cuenta como una sílaba adicional con independencia del número de sílabas que contenga, incluso hallándose vacía. De esta manera, en un verso acabado en palabra esdrújula, también denominada proparoxítona[71], se resta una sílaba al recuento, puesto que las dos átonas cuentan como una sola; si es llana o paroxítona, no altera el recuento porque la posición postónica contiene una sola sílaba átona y, finalmente, si el verso acaba en palabra aguda u oxítona, se añade una sílaba al recuento, al estar vacía la posición posacentual (Domínguez Caparrós, 2014, pp. 60-65; Quilis, 1984/2013, pp. 43-44).

Debemos tener una consideración adicional para los versos de arte mayor, ya que la cesura podría conferirles particularidades de las que carecen los de su contrapartida de arte menor, contra la opinión de Navarro Tomás (1991, p. 40). En lo que a nuestro estudio respecta, esto concierne en concreto a los endecasílabos, que se dividen por regla general *a minore*, con un primer hemistiquio de cinco sílabas seguido de otro de seis tras la cesura, si el acento rítmico central va en la cuarta sílaba, o *a maiore*, con siete y cuatro si va en la sexta (Baehr, 1973/1997, p. 136). Sin embargo, sería posible tener dos hemistiquios *a minore* de seis sílabas ortográficas siempre y cuando la última palabra del primero fuera esdrújula o, incluso, un endecasílabo de nueve sílabas si ambos hemistiquios terminan con sendas palabras esdrújulas. La situación opuesta no se da. Esto es, no se añade una sílaba al primer hemistiquio si acaba en palabra aguda (Martínez Zuviría y Herrera, 1938, pp. 389-393). La razón debemos buscarla en el origen musical del verso, en el que cada pie corresponde a una nota, y una sílaba larga equivale a dos breves. Veámoslo con un ejemplo: tomemos el verso de 68a y dividámoslo en sus sílabas, considerando sus acentos, por una parte, y su cantidad como si fuera una canción. Ahora tomemos *alma* y convirtámoslo en *ánima* como en el ejemplo 68b. Aparentemente, tenemos una sílaba supernumeraria en el primer hemistiquio que produciría un verso dodecasílabo. Pero, ahora, imaginemos que es una verso de una canción y establezcamos una equivalencia entre largas y acentuadas.

[71] Por claridad, usaremos una nomenclatura para las palabras y otra para las rimas.

Las sílabas *-nima* equivaldrían a dos breves, que podrían resolverse como una sola larga sin alteración del ritmo.

(68) a. que en vos el álma ‖ más que en mí vivía ‖
(Calderón, *Bien vengas, mal si vienes solo*, v. 915)
b. que en vos el ánima ‖ más que en mí vivía ‖

El caso inverso, por el contrario, no resulta aceptable al oído, a pesar de que Quilis (1984/2013, p. 80) sea de la opinión contraria: sumar una sílaba al primer hemistiquio si este es agudo no satisface los requisitos necesarios. La razón debemos buscarla también en los tiempos rítmicos. Tomemos, por ejemplo (69a): la cuarta sílaba del primer hemistiquio podría descomponerse en dos sílabas breves (69b). Sin embargo, en la métrica española la sílaba acentuada equivale a una larga, por lo que no podría realizarse (69c), ya que esto obligaría a introducir una larga, para la que faltaría medio tiempo (Martínez Zuviría y Herrera, 1938, pp. 393-396).

(69) a. que en vos el mal ‖ más que en mí vivía ‖
b. que en vos el mal ‖ más que en mí vivía ‖
c. *que en vos el mal ‖ más que en mí vivía ‖

Fusión de vocales

El fenómeno más importante que afecta a las vocales en contacto es la sinalefa. Esta, como vimos en el capítulo que dedicamos a los aspectos fonéticos-fonológicos del español, se produce cuando dos palabras adyacentes tienen sendas vocales en contacto la una con la otra. En ese caso, una de ellas pierde su cualidad vocálica y se une a la otra sílaba. Nebrija (1492/1981, p. 47) nota que, si una palabra acaba en vocal y a esta la sigue otra que comienza también con vocal, la primera *se echa fuera*. Comparte esta opinión Díaz Rengifo (1592/2012, p. 186), aunque este no echa del todo la primera vocal, sino que se limita a no contarla, y proporciona dos pautas para aplicar la sinalefa y excepciones en su uso. La primera es que la segunda sílaba es la que adopta la función de núcleo vocálico. La segunda indicación que da es que esta figura solo se da en el interior del verso y no entre versos. Finalmente, advierte de que la sinalefa puede romperse —hablaríamos aquí de *dialefa*—, especialmente si la primera vocal pertenece a un monosílabo y va acentuada. Caramuel (1665/2007, p. 51) matiza diciendo que la primera sílaba no desaparece, pero se debilita[72].

[72]Las distintas opiniones se prestan a especulación, pues ninguno de los naturales de la Extremadura Castellana es tan radical como el andaluz. Decía Juan Valdés de Nebrija que, «como aquel hombre no

Sin embargo, esto resulta problemático si prestamos oídos a Navarro Tomás (2004, sec. 135), quien sentencia lo siguiente:

> El oído de un buen poeta es siempre un excelente guía en lo que se refiere, dentro de su idioma, al acento y al cómputo silábico de las palabras. Por otra parte, aunque en el lenguaje poético haya palabras, giros y modos de expresión que no se usen de ordinario en la lengua corriente, sabido es que en lo que a la articulación y a la dicción se refiere, no existe en español una pronunciación poética distinta de la que se usa en el discurso, en la escena o en las conversaciones de las personas ilustradas.

Recordemos que habíamos dicho en el capítulo anterior que la sinalefa respondía al mismo criterio de perceptibilidad que regía para la diptongación. Por lo tanto, nos encontramos ante dos propuestas diferentes, que podrían llegar a ser contradictorias si ambas partes de la unión son tónicas, pero la primera es más perceptible. Así, por ejemplo, ⟨va este⟩, sería para los tratadistas históricos /ˈba̯este, mientras que para la lingüística moderna sería /ˈbã̯este/, por primarse la más perceptible entre acentuadas. Aunque las diferentes lecturas no repercuten en el patrón rítmico, sí lo hacen en la armonía vocálica, al variar la calidad del núcleo de la sílaba. Menos riesgo hay de que llegue a afectar a la rima, ya que su empleo en la última sílaba tónica es un recurso poco común. En palabras de Bello (1835/1981, p. 118), «no hay causa que legitime más el hiato que la circunstancia de hallarse la dicción acentuada al fin de la frase o del verso. El concurso de ambas circunstancias haría particularmente inaceptable la sinalefa».

¿Cómo resolvemos esta dicotomía? La fonética experimental acude en nuestro auxilio (Martínez, 2004). Lo cierto es que los grupos fónicos /j+e/ y /ẽ̯+a/ son los más numerosos, pero, incluso entre inacentuadas, aparece como tercero /o+ẽ̯/ y /a+ũ̯/ está también entre los más habituales. Por lo tanto, debemos entender la observación de los tratadistas clásicos no como una norma inmutable sino como una descripción del comportamiento más frecuente[73]. En cualquier caso, en la realización de la sinalefa intervienen otros factores, que van desde el dialecto hasta la intención concreta del hablante en el momento concreto de la elocución.

Otra particularidad hoy desaparecida de la poética áurea es la productividad fonológica de la *h* a comienzo de palabra con raíz etimológica en la *f* latina. Este sonido

era castellano, sino andaluz, *hablaba* y escribía como en Andalucía, y no como en Castilla» (Valdés, 1737/2015, p. 190, énfasis añadido, ortografía modernizada).

[73]Podríamos plantearnos como hipótesis si este fenómeno, irónicamente, no se daría incluso de una forma más habitual entonces, dado que diptongos decrecientes como /uĩ̯/ eran aún productivos hasta el punto de determinar la rima de un verso, como vimos.

era históricamente aspirado, aunque, entrado el siglo XVII, esta aspiración había quedado reducida al ámbito dialectal. Sin embargo, el carácter típicamente arcaizante del texto poético hace que la aspiración sea no solo lícita, sino de uso común como dispositivo de compensación métrica, ya que obliga a romper la sinalefa que, de otro modo, habría de producirse (Quilis, 1984/2013, p. 50).

La sinéresis, por su parte, consiste en la unión de dos vocales en el interior de la palabra. Distingue Díaz Rengifo (1592/2012, pp. 187-188) entre diptongos decrecientes y crecientes, aceptando los primeros como tales, pero adscribiendo los últimos a la sinéresis, no obstante, admitiendo que son naturales a la lengua. Hay circunstancias que facilitan la sinéresis, entre ellas que la vocal inacentuada sea /e/, por ser «la menos llena de las llenas» (Bello, 1835/1981, p. 88).

La sinéresis puede, a su vez, desencadenar otros fenómenos, como el desplazamiento acentual o, por el contrario, puede estar provocada por este. Lo ilustra Sánchez Jiménez (2020, pp. 303-304) con las sinéresis posibles en *tenía* o *María*. En esos casos, la vocal alta no puede ser nuclear, por lo que el rasgo tónico se desplaza necesariamente, sea sistólicamente a la sílaba anterior, por lo que queda *tenia*, o diastólicamente, que resulta en *teniá*. El primer caso no solo alteraría el ritmo, sino también el recuento silábico del verso.

Cabe aquí añadir que, en el teatro, las uniones vocálicas han de resolverse en ocasiones mediante estrategias diferentes a la reducción vocálica. En un verso compartido, donde sendos locutores tienen un sonido vocálico en los márgenes colindantes de sus respectivos parlamentos, una resolución normal obligaría a comenzar la línea con una vocal no silábica aislada, algo que resulta ciertamente difícil de concebir para el oído castellano. De esta manera, algunos versos contravienen la métrica al introducir una sílaba supernumeraria, como en (Figura 5.1).

En este ejemplo, la sinalefa entre el segmento medio y último no solamente no se hace, sino que la actriz fuerza que la división sea más destacada pronunciando como tónica una sílaba que, por sí misma, no debería serlo. La palabra *hacienda* es declamada *ha cienda*, lo que deja tres acentos sucesivos. Tal vez una pronunciación neutra hubiera bastado para hacer percibir la unión como una sinalefa. Si bien las series de acentuadas en réplicas y contrarréplicas no son infrecuentes, el poeta suele perseguir con estas un efecto de velocidad en el discurso —con recurso cómico, no pocas veces—, pero, en cualquier cosa, sin contravenir la métrica. En este caso, la responsabilidad no es suya sino de la actriz o del director de escena[74].

[74]Durante un receso del Hispanistentag celebrado en Graz, Ramón Valdés aludió a la tensión dialéctica entre el escritor-poeta y el escritor-dramaturgo que impregna el proceso creativo de la obra

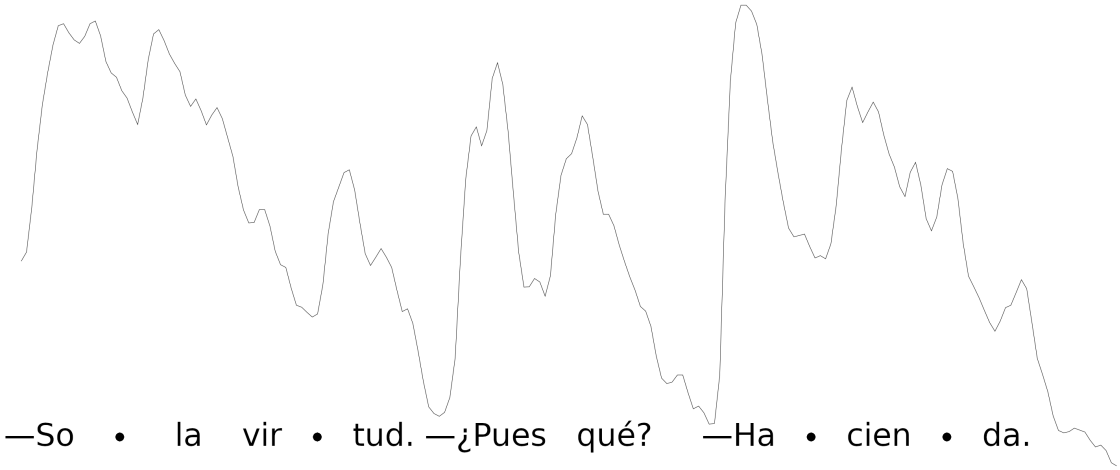


Figura 5.1: Verso 42 en una representación (Vega Carpio, 1982) de *La discreta enamorada*.

Separación de vocales

Se denomina *diéresis* a la ruptura de una unión natural entre vocales, de forma que el diptongo da lugar a dos sílabas independientes. En lo concerniente a nuestro trabajo, hay que tener en consideración que, a pesar de que estas vienen señaladas explícitamente en las ediciones más minuciosas (por ejemplo, ⟨rüin⟩ /ruˈin/ frente a ⟨ruin⟩ /ˈrwin/), no tiene por qué ser así en todos los casos y, de hecho, no suele serlo. Sin embargo, hay palabras cuyos diptongos se rompen de manera más frecuente en los versos áureos, como *juez*, *suave*, *cruel*, *fiel*, *ruin* y sus derivados, por lo que, si se presenta la necesidad de producir una sílaba adicional para cuadrar el verso, este tipo de palabras se prestan sobremanera a la diéresis. Incluso vates tan poco dados a extravagancias como Quevedo recurren al hiato aprovechándolas, si bien con mucho más comedimiento que sus contemporáneos culteranos (Llamas Martínez, 2020, pp. 55-64).

Otro fenómeno divisorio es el *hiato* o la ruptura de una sinalefa. Quilis (1984/2013, pp. 50-52) atribuye el fenómeno a diversas causas, como la acentuación de una de las dos vocales implicadas, causa de sinalefa violenta. En la poesía del Siglo de Oro hay que contemplar asimismo la posible aspiración de la hache inicial en la segunda palabra que, como dijimos, también podría impedir la sinalefa.

---

teatral clásica. En este caso, la *vis* dramatúrgica de Lope parece que sucumbió ante los embates de la poética, quizás incluso a sabiendas de las complicaciones escénicas que entrañaría tal decisión.

5.6 Rima

El medio más evidente de alcanzar eufonía es la rima. Este recurso lírico consiste en la repetición de sonidos desde la última vocal acentuada o cumbre tonal del verso. Puede ser de diferentes clases dependiendo de los tipos de sonido que se repitan (Domínguez Caparrós, 2014, pp. 103-115). No es la única vía para crear un efecto melódico, sino que es frecuente encontrar también otras figuras de repetición tales como aliteración, simetría o anáfora (Domínguez Caparrós, 2014, pp. 126-129). Podemos definir la rima, pues, como «la total o parcial semejanza acústica, entre dos o más versos, de los fonemas situados a partir de la última vocal acentuada» (Quilis, 1984/2013, p. 37).

De esta manera, hablaremos de rima oxítona o *masculina* si el verso acaba en palabra aguda, paroxítona o *femenina* si acaba en palabra llana y proparoxítona o dactílica si acaba en palabra esdrújula.

Caramuel (1665/2007, p. 57) distingue entre rimas *asonantes*, también llamadas parciales, cuyas vocales coinciden, pero no así sus sonidos consonánticos (70), y rimas *consonantes*, también llamadas rimas totales, cuyas vocales y consonantes coinciden desde el último acento (71).

(70)

Yo, acudiendo a mis es-	t/*ú*-	di/*o*/s
en ellos y en todo	m/*í*-	r/*o*
que Segismundo se-	r/*í*-	*a*
el hombre más atre-	v/*í*-	d/*o*,

(Calderón, *La vida es sueño*, vv. 708-711)

(71)

Hipogrifo vio-	l/*énto*,
que corriste parejas con el	vi/*énto*

(vv. 1-2)

No todos los sonidos son parte de la rima. En el caso de la rima proparoxítona, como ya indicamos, la constituyen los elementos de la sílaba tónica y la última sílaba (72), de manera que *fábula* rima con *águila*. En las rimas asonantes, solo los núcleos vocálicos son relevantes para la rima, de forma que *inercia* rima con *lenta* (Herrero Prado, 1996, p. 89). Aquí es donde se hacen más vez patentes los beneficios de la decisión inicial de partir de una transcripción fonológica silabeada en lugar de la representación ortográfica (73). Por una parte, en tanto que tenemos el verso ya separado en sus sílabas, trabajaremos directamente con estas, corrigiendo según los fenómenos intervocálicos si es menester. Por otra parte, al distinguir entre los alófonos vocálicos silábicos y no silábicos, la resolución de la rima asonante es trivial, pues

basta con tener en cuenta exclusivamente los fonemas vocálicos silábicos del *axis* —que discutiremos abajo— de cada verso.

(72)

Al Uni-	gé-	ni-	to,
al Padre	m/*á*-	xi-	m/*o*
y al Santo Es-	pí-	ri-	tu,
de ambos Pa-	r/*á*-	cli-	t/*o*,
pidamos	hú-	mi-	les,
que en estos	*á*/s-	pe-	r/*o*-/s
valles de	lá-	gri-	mas
desiertos y	*á*-	ri-	d/*o*/s,
su Amor a-	yú-	de-	nos,
su Gracia	s/*á*/l-	ve-	n/*o*/s.

(Calderón, *El año sancto de Roma*, vv. 855-863)

(73)

Que estas señoras da-	rán,	
de irlas sirviendo li-	c/*é*/n-	ci/*a*.
Y más cuando fuera	cul-	pa,
que los crïados que	d/*é*-	j/*a*/n
a sus dueños en vi-	si-	ta,
por ellos, Félix, no	vu/*é*l-	v*a*/n.

(Calderón, *¿Cuál es mayor perfección...?*, vv. 938-943)

Los versos que acaban con palabras homógrafas suponen un caso particular. Diremos que son *equisonantes* si estas difieren léxicamente o en su función gramatical, es decir, si se trata de una diáfora. Lo vemos en el ejemplo 74, donde *pasa* toma el valor de 'entrar' en el segundo de los versos, mientras que vale 'acontecer' en el tercero. Si las palabras, además de ser homógrafas, también comparten significado y función, hablamos de rimas *unisonantes* (75), que resulta inaceptable para la prescriptiva.

(74) [...]
pero tu hermano viene. Aquí escondida
le he de escuchar. Pues ya a tu cuarto *pasa*.
Y así saber espero lo que *pasa*.
(Moreto, *No puede ser el guardar una mujer*, vv. [1411-1413)

(75) Colgad ese saco ahí
para que diga, ay de *mí*:
«En tal puesto me colgó
Paulo, que no mereció

la gloria que encierro en *mí*».
(Tirso de Molina, *El condenado por desconfiado*, vv. 1885-1889)

En cualquier caso, estas restricciones solo afectan a la rima, por lo que, por ejemplo, el primer y segundo verso del ejemplo 76 no estarían sujetos a estas restricciones y, si bien podría achacársele a Rojas Zorrilla un uso repetitivo del léxico en este caso, esto no tendría repercusión en su observancia de las reglas de la métrica.

(76) En tu atención. El amor,
¿quién le colige en lo a*tento*?
La atención supone amor,
disgusto el diverti*miento*;
bien quiere aquel que escuchando
se transforma en los con*cetos*;
(Rojas Zorrilla, *Obligados y ofendidos*, vv. 31-36)

Sea como fuere, en el corpus que manejamos, el recurso a la equisonancia se utiliza de manera muy esporádica. La unisonancia es prácticamente inexistente, puesto que esta suele ser fruto de la falta de habilidad del poeta y, si las ediciones digitales de literatos áureos consagrados son escasas, más aún lo son las de los diletantes. La adscripción a uno u otro caso de homofonía de las ocurrencias que hemos encontrado suele ser susceptible de discusión. En cualquier caso, la máquina contaría con los medios para identificar estas rimas, ya que la determinación del ritmo obliga a identificar las funciones morfosintácticas de las palabras del verso. Por lo tanto, requeriría apenas verificar que ambos homógrafos de la rima tienen etiquetas diferentes.

Además de la rima externa, aparece aunque no resulta frecuente —Lope se vale de este recurso en, por ejemplo, *El amigo hasta la muerte* y *El mayordomo de la duquesa de Amalfi*— una rima interna italianizante, a imitación del *endecasillabo incatenato*. Esta rima interna la conforman las del axis rítmico y otro interno que pasa por la sexta sílaba, en el siguiente verso (Sánchez Jiménez, 2014).

5.7 Agrupaciones de versos

El verso por sí mismo no constituye un elemento lírico. Para que este sea digno de su nombre, debe aparecer en combinación con otros versos (Quilis, 1984/2013, p. 95), de manera que formen una estructura superior que produzca el contexto sonoro de la poesía. Desde el punto de vista semántico, el verso es autosuficiente, pero no así desde el rítmico y sonoro, ya que ha de mostrar un patrón. Como es obvio, para que este

pueda existir, se requiere, al menos, una serie de dos elementos. A esta agrupación estructurada de versos la denominamos estrofa. A su vez, cabe agrupar varias estrofas en un poema estrófico, pero también pueden unirse versos directamente en poemas no estróficos.

La estrofa

De acuerdo con Quilis (1984/2013, pp. 95-99), la estrofa ha de cumplir cuatro requisitos para ser tal en propiedad: debe tener un *axis rítmico*[75], consistir de un número determinado de rimas, poseer una estructura sintáctica determinada y estar dispuesta según un modelo versal estructurado.

El axis rítmico, según como lo define Balbín (1968, pp. 38-60), está constituido por la inflexión distensiva de cada uno de los grupos melódicos de los que se compone la estrofa en torno a la que se concentran y organizan los factores rítmicos. Dicho de otra manera, el axis rítmico de la estrofa se establece a partir de la última sílaba acentuada de sus versos (77). Esta es la que carga con el acento versal que, a su vez, altera la cantidad del verso. Aquí se produce asimismo la inflexión del tono; en torno a este punto se distribuyen las pausas para lograr la eufonía y se define el tipo de encabalgamiento. El centro del eje provee el elemento más importante de la rima. En consecuencia, resulta crucial para nuestro modelo identificar esta parte del verso porque varios de los análisis que han de hacerse se sustentan en su correcta caracterización. Esto significa, además del silabeo, localizar la sílaba que carga con el acento léxico de la última palabra del verso.

(77)

¿cuál de vosotras, cuál desde su a-	*si/é/n*	-to
es la que influye en mí desdichas	*t/á*	-les
¿Cuál de vosotros, astros desi-	*gu/á*	-les
a su cargo tomó mi sufri-	*mi/é/n*	-to

(Calderón, *Sueños hay que verdad son*, vv. 653-656)

El número de rimas resulta asimismo fundamental, pues la función de estas no se limita a la ejercida sobre el propio verso, sino que se combinan unas con otras de forma que, de esta unión, resulta una estrofa, cuya configuración es producto del número de rimas y de su distribución. Como veremos un poco más adelante, clasificamos las estrofas en primer lugar por el número de versos que las constituyen y, además,

[75]Emplearemos este cultismo —que Quilis adopta de Balbín Lucas (1968) —, pues *eje* podría llevar a confusión, ya que también equivale a la sexta sílaba del endecasílabo si esta va acentuada (Domínguez Caparrós, 1985, *s.v. eje*).

distinguimos entre estrofas con el mismo número de versos por la disposición de las distintas rimas de estos.

Por otra parte, la estrofa encuentra una correspondencia en la estructura sintáctica completa, de manera que la pausa estrófica coincide con el final de la unidad sintáctica. Aunque así se ha hecho tradicionalmente, algunos poetas con querencia por los modelos grecolatinos contravienen la prescriptiva de la métrica castellana para disociar una unidad sintáctica en dos estrofas (Quilis, 1984/2013, p. 97). De la importancia de esta costumbre para reproducir el metro clásico da cuenta el elogio de Menéndez Pelayo a Francisco Medrano, al que el erudito se refiere como

> de todos los imitadores de Horacio el más latino, el más sobrio, el más rápido, el que mejor ha remedado la marcha de los períodos rítmicos de Horacio, *el que más se le parece en la manera de encabalgar las estrofas*, y el que más le anda a los alcances en el arte de economizar las palabras. (Menéndez Pelayo, 1908/1951, p. 14, énfasis añadido).

En lo que respecta al sistema de estructurar las rimas, cada estrofa debe componerse de un número determinado de versos de tipos concretos dispuestos según un patrón dado. De esta manera, encontramos *estrofas isométricas*, en las que todos sus versos tienen la misma cantidad, como en el cuarteto del ejemplo 78, y *estrofas heterométricas*, en las que se mezclan versos con distinto número de sílabas (79).

(78) Cuando la fama en lenguas dilatada
vuestra rara hermosura encarecía,
por fe os amaba yo, por fe os tenía,
Leonor, dentro del alma idolatrada.
(Calderón, *A secreto agravio, secreta venganza*, vv. 746-749)

(79) Ilustre es, si no rico,
el condado Anspurg en Alemania;
a su quietud me aplico,
pues los jardines de la gran Campania
no igualan a este monte
que el Ártico nos muestra en su horizonte.
(Mira de Amescua, *La casa de Austria*, vv. 89-94)

Para representar la estructura de las estrofas, emplearemos, además de la notación tradicional categórica de letras mayúsculas y minúsculas, la de subíndices si la caracterización de la estrofa requiere una descripción cuantitativa de sus versos. De esta

manera, representaremos la estrofa del ejemplo 79 como *aBaBcC* o $a_7b_{11}a_7b_{11}c_7c_{11}$, según las características que convenga observar en cada momento. La segunda forma resulta muy conveniente en una obra dramática, en la que, al contrario de lo que suele ocurrir en los poemas, se alternan diferentes metros, sobre todo de arte menor. De esta manera, resulta más sencillo explorar los versos en su contexto para conocer el uso de una determinada rima y condicionarlo al número de sílabas o viceversa.

Según la disposición de la rima en la estrofa, se distingue entre rima *continua*, con una estructura *aaaa* (80); *rima gemela*, según el patrón *aa* : *bb* (81); *rima encadenada*, con el patrón *abab* (82) y rima abrazada, *abba* (83).

(80) Ay de opinión tan ciega,
que los principios a la Fe le niega,
donde a mover la Caridad no llega,
que huye a Misericordia que le ruega!

(Calderón, *El nuevo hospicio de pobres*, vv. 779-982)

(81) Sí, mas de ambiguas razones
en sus ojos mis pasiones

han visto lo que me estima.
Vana esperanza te anima

cuando penetra mi amor
el que me tiene interior.

Cuando tu soberbia abajes
y amor se obligue a mis gajes,

tu engaño conocerás.
Yo sé que me envidiarás.

(Tirso de Molina, *Santo y sastre*, vv. 908-917))

(82) Con el cuidado que el Amor, Diana,
pone en un pecho, que aquel fin desea,
que la mayor dificultad allana,
Él mismo quiere que te adore y vea.

(Lope de Vega, *El perro del hortelano*, vv. 689-692

(83) Tienes, Laurencia, razón;
que en dejando de querer,
más ingratos suelen ser
que al villano el gorrión.

(*Fuente Ovejuna*, vv. 249-242)

Tipología estrófica

Como indicamos al comienzo de este capítulo, el metro dramático no se diferencia del lírico. Por lo tanto, si hemos de encontrar patrones en las agrupaciones de versos, debemos recurrir a la poética tradicional. Añadiremos que el teatro no solo toma prestada la métrica de la poesía, sino que, además, lo hace extensivamente, aprovechando todos los recursos que aquella pone a su disposición. Así, Fernández Guillermo (2021, pp. 213-242) encuentra en apenas veinte comedias de Lope de Vega —un único autor— las siguientes formas: redondillas, romances con y sin estribillo con hexasílabos y octosílabos, quintillas, pareados, tercetos sueltos y encadenados, endecasílabos y otros versos sueltos, octavas, octavas reales, sonetos, canciones, décimas, sextinas, letrillas, coplas, coplillas, seguidillas, seguidillescas, glosas de tipos diversos, silvas, romancillos de ocho y siete sílabas, liras, sextetos-lira, cuartetas asonantadas, canciones petrarquistas, ensaladas, coplas de pie quebrado, villancicos y zéjel, así como partes cantadas con otras medidas y estrofas. A cuenta de esto, resulta necesario caracterizar las agrupaciones versales que encontraremos porque habremos de identificarlas en la segunda parte de este trabajo.

Nos aproximaremos a la clasificación estrófica típica del Siglo de Oro considerando formas recogidas por Morley y Bruerton (1968) en su seminal trabajo sobre la datación de la obra lopesca. Optamos por este y no por la obra similar calderoniana de Hilborn (1938) por encontrarse esta última más simplificada (Antonucci, 2023, pp. 3-4). Nos centraremos en las formas españolas (Morley y Bruerton, 1968, pp. 38-39) e italianas 1968, pp. 39-41, como también los metros que constan en las tablas de Fernández Guillermo (2021, pp. 213-242). Partiremos del número de versos que componen cada estrofa, tomando las descripciones de los tratados de métrica (Herrero Prado, 1996, pp. 102-153; Quilis, 1984/2013, pp. 100-119), e iremos ilustrando sus particularidades con ejemplos del corpus teatral. Esta clasificación será la que tomemos como modelo inicial para reconocer patrones en la segunda parte.

La estrofa más elemental es el *pareado*, ya que está compuesta de tan solo dos versos de igual o distinto metro y es autosuficiente. Al ser solo dos versos independientes del contexto, solo admiten un patrón de rima en la forma AA (84). Si, además, los versos son octosílabos con rima consonante a_8a_8, la estrofa se denomina *aleluya*. El pareado no ha de ser necesariamente isométrico, sino que también se da entre versos de distinta medida. Un ejemplo paradigmático lo ofrece la combinación de heptasílabos y endecasílabos de los pares de versos que componen la *silva de consonantes* del monólogo con el que Rosaura da comienzo a *La vida es sueño* (85).

(84) Éste dice: «Señor, yo soy Estacio,
que estoy en los jardines de palacio,

y enseñado a plantar hierbas y flores,
planté seis hijos, a los dos mayores

suplico que les deis...».
Basta, ya entiendo,
con más cuidado ya premiar pretendo.
(Lope de Vega, *El castigo sin venganza*, vv. 2467-2472)

(85) Hipogrifo violento,
que corriste parejas con el viento,
¿dónde, rayo sin llama,
pájaro sin matiz, pez sin escama,
y bruto sin instinto
natural, al confuso laberinto
de esas desnudas peñas
te desbocas, te arrastras y despeñas?
(Calderón, *La vida es sueño*, vv. 1-8)

De cualquier manera, al haber una única rima posible, el nexo entre estrofas del mismo tipo ha de ser débil, en tanto que la siguiente o bien tiene la misma rima, en cuyo caso no hay un cambio, o una diferente, por lo que el salto será abrupto. No es factible una transición entre pareados que además sea suave. Para que esto sea posible, debemos contemplar la estrofa respecto a otras, lo que es el caso del *perqué* o *aquelindo*, que alterna rimas que encadena con la siguiente estrofa según la estructura *ab/bc/cd/de...*

(86) Escucha, la que veniste
de la jerezana tierra

a hacer a Sevilla guerra
en cueros, como valiente;

la que llama su pariente
al gran Miramamolín;

la que se precia de ruïn,
como otras de generosas;

la que tiene cuatro cosas,
y aun cuatro mil, que son malas;

la que pasea sin alas
los aires en noche escura;

la que tiene a gran ventura
ser amiga de un lacayo;

la que tiene un papagayo
que siempre la llama puta;

la que en vieja y en astuta
da quinao a Celestina;

la que, como golondrina,
muda tierras y sazones;

la que a pares, y aun a nones,
ha ganado lo que tiene;

la que no se desaviene
por poco que se le dé;

la que su palabra y fe
que diese, jamás guardó;

la que en darse a sí excedió
a las godeñas más francas;

la que echa por cinco blancas
las habas y el cedacillo.

(Cervantes, *El rufián dichoso*, pp. 556-596)

Del mismo modo, las transiciones entre estrofas se multiplican en el momento en que introducimos un nuevo verso. Las estrofas de tres versos se denominan *tercetos* en su versión de arte mayor ABA (87) y tercerillas en las de arte menor *aba* (88). De esta manera, se dan estrofas AAA, que diferirían de los pareados en la cantidad de versos, aunque no en la cualidad de la rima, pero también ABA o ABB, que introducirían una alternancia en la rima. Esto, a su vez, expande las posibilidades de combinar estrofas porque ahora podemos encadenarlas en la forma $ABABCBCDC...$, de modo que la nueva rima sirve de solución de continuidad para la transición de un verso, pero, al contrario que en la estrofa de dos sílabas, la estrofa puede conservar una rima característica distinta de la siguiente. En la poesía medieval se cultivaba una variante del terceto conocida como *trístico monorrimo*, que tiene la misma rima en sus tres versos según el modelo AAA (89).

(87) Si yo pensara, Conde, que te diera
tanta tristeza el casamiento mío,
antes de imaginarlo me muriera.

(Lope de Vega, *El castigo sin venganza*, vv. 114-116)

(88) Si aborrecidas adoran,
si adoradas aborrecen,
¡lo que son mujeres!

(Rojas Zorrilla, *Lo que son mujeres*, vv. 2849-2851)

(89) Pues los músicos digan a coros:
No están todos
en la casa de los locos.

(vv. 2701-2703)

Las estrofas de cuatro versos ofrecen más variedad. Si son consonantes de arte mayor, se denominan *cuartetos*, y tienen rima cruzada $ABBA$ o encadenada $ABAB$ (90), en

cuyo caso reciben el nombre de *serventesio*. En cuanto a las de arte menor, tenemos la *cuarteta* según el esquema *abab* (91). La combinación de arte menor se conoce como *redondilla*[76] si la rima es abrazada en la forma *abba* (92). Una variante de esta estrofa es aquella que solo rima en sus versos pares según el modelo $-a-a$. Se da también la modalidad *asonantada* o *tirana* (93). La redondilla en secuencia con el romance es la estrofa más común en el teatro de Lope (Fernández Guillermo, 2007, p. 169), cuya obra representa un porcentaje notable de la producción dramática del Siglo de Oro que ha llegado hasta nosotros.

(90) Amor de ser amado satisfecho
cuando agraviado imaginó vengarse,
Templado el fuego y el furor desecho,
Adonde pudo arderse pudo helarse.

(Lope de Vega, *La Dorotea*, p. 403)

(91) Razón, fortuna, amor, celos
son pasiones que se mudan:
la razón falta a su tiempo
y se cansa la fortuna;

(Calderón, *La banda y la flor*, vv. 2068-2071)

(92) Yo os juro que, si os agrado,
que de vos lo voy también,
y que, procediendo bien,
os doy amor por cuidado.

(Lope de Vega, *El villano en su rincón*, vv. 65-68)

(93) Inventó el amor un juego
donde en gustosos descuidos,
pagando en prendas sus yerros,
se vino a quedar desnudo.

(Moreto, *El hijo pródigo*, vv. 1309-1312)

Las estrofas de cuatro versos se prestan bien a diversas combinaciones heterométricas. Así, la *seguidilla* combina versos impares heptasílabos y pares pentasílabos en su variedad habitual (94)[77]. Navarro Tomás(1991, pp. 292-293) recoge variaciones con

[76]No confundir con la *redondilla* según la terminología de Díaz Rengifo (1592/2012, pp. 193-207), quien emplea el término para referirse a estrofas de octosílabos de cinco versos y sus variaciones de entre cuatro y ocho.

[77]No confundir con la llamada *seguidilla gitana*, compuesta de versos hexasílabos a excepción del tercero que es decasílabo o endecasílabo, que sería una variante de la endecha (Hanssen, 1909/1957, pp. 185-186).

hexasílabos en lugar de pentasílabos, así como de tres versos, que se reparten según las disposiciones $5-7-5$ y $7-5-7$, y Morley y Bruerton (1968) con siete versos $XaYa : bZb$. Asimismo, Sor Juana Inés, entre otras modalidades, usó una conocida como *seguidilla real*, según el patrón $x10a_6x_{10}a_6$. También heterométrica es la *estrofa sáfica* (95), que se compone de tres endecasílabos sáficos (`---+x+xxx+-`) y un pentasílabo adónico (`+xx+x`) o, en su adaptación a la rítmica castellana, un hexasílabo según el patrón `+x+x+-` (Luque Moreno, 2010, p. 119). En la poética castellana, gozaba de gran tradición el *tetrástrofo monorrimo*, más conocido por *cuaderna vía*, que se compone de cuatro versos alejandrinos, a su vez formados por dos hemistiquios heptasílabos cada uno, y una sola rima consonante, pero su presencia es escasa en el Siglo de Oro (Navarro Tomás, 1991, p. 277). Entre las formas heterométricas destacan las que constituyen la composición conocida como *endecha real* (96). Si la *endecha* es un romancillo de versos de menos de ocho sílabas de tema triste (Domínguez Caparrós, 1985, *s.v. endecha*), la real presenta la particularidad de que sus tres primeros versos son heptasílabos y el cuarto endecasílabo.

(94) Alegría, zagales,
que a casa vuelve
hoy el hijo perdido.
Todos se alegren.

(Moreto, *El hijo pródigo*, vv. 2843-2856)

(95) A- | mor pode- | roso ‖ en cielo y en tierra, ‖
dul- | císima | guerra ‖ de nuestros sentidos, ‖
¡oh, | cuántos per- | didos ‖ con vida inquïeta ‖
tu im- | perio su- | jeta! ‖

(Lope de Vega, *La Dorotea*, p. 158)

(96) No con más diligencia
la diosa de las mieses
buscó a su hija amada
hasta los escondrijos del infierno,

(Cervantes, *La entretenida*, vv. 515-518)

Hay que hacer notar una característica del ejemplo 95 que será conveniente considerar para diseñar el sistema de escansión. Si atendemos a la prosodia, la estrofa está compuesta de tres versos de doce sílabas y uno de seis. Sin embargo, dijimos que este tipo de estrofa se compone de endecasílabos y un pentasílabo. Vemos en la correspondencia con los pies del modelo latino del pentasílabo y los primeros hemistiquios

de los endecasílabos que tenemos una sílaba suelta antes del dáctilo y el troqueo. En efecto, está sílaba está en anacrusa, por lo que no se considera en el cómputo silábico ni en el patrón acentual —de cantidad en el modelo latino—. De esta manera, los acentos de los endecasílabos recaen en la primera y cuarta sílaba tras la anacrúsica y en la primera y tercera en el pentasílabo. Dada la escasa frecuencia con que aparecen metros inusuales, cabe plantearse considerar este fenómeno ante versos de metro inesperado con sílabas átonas a comienzo del verso. Por otra parte, el último endecasílabo presenta otra particularidad, ya que requiere contemplar la interjección como inacentuada.

Entre las estrofas isométricas de cinco versos, el *quinteto*, de arte mayor, no es habitual en la poesía aurisecular, al contrario que su equivalente de arte menor, la *quintilla* (97). Ambas formas presentan restricciones, pues han de tener rima consonante, no contienen más de dos versos seguidos con la misma rima, los dos últimos versos no riman entre sí y no hay versos sueltos, de manera que solo se dan las combinaciones *ABABA*, *ABAAB*, *ABBAB*, *AABAB* y *AABBA*. Entre las estrofas heterométricas cabe destacar la *lira*, que combina dos endecasílabos y tres heptasílabos según la estructura rimante *aBabB*. Aunque apenas encuentra uso dramático tras Rey de Artieda y Juan de la Cueva (Navarro Tomás, 1991, p. 257), da pie a otros metros más extendidos como el sexteto-lira.

(97) Y así, en lirio transformado,
siendo el morado color
jeroglífico del prado,
se vio entre el lirio y la flor
el amor enamorado.

(Calderón, *Hado y divisa...*, vv. 1604-1608)

(98) ¿Es, ingrata Sigura,
el casamiento próspero que aguardo?
Callar será cordura,
pues de ira y desdén ardo,
y de envidia me hielo y acobardo.

(Rey de Artieda, *Los amantes*, vv. 533-537)

Las estrofas isométricas de seis versos más destacadas son la *sextilla* (99), con restricciones semejantes a las de la quintilla en cuanto a la disposición de las rimas, así como su equivalente de arte mayor, el *sexteto*. La *sextina real* o *sexta rima* es una configuración italianizante de versos endecasílabos según el patrón *ABABCC*, «como una octava real a la que hubiesen recortado los dos primeros versos» (Baehr,

1973/1997, p. 276). Presenta a veces algunas variaciones, como el empleo de versos oxítonos en los versos tercero y sexto; no obstante, gozó de escasa estima entre los dramaturgos áureos (Navarro Tomás, 1991, p. 256).

Versos	Estrofa	Rima	Distribución
2	Pareado	indiferente	$a_x a_x$
	Aleluya	indiferente	$a_8 a_8$
	Perqué	indiferente	$a_8 b_8 / b_8 c_8 / ...$
	Silva de consonantes.	consonante	$a_7 a_{11}$
3	Tercerilla	consonante	aba
	Terceto	consonante	$a_{11} b_{11} a_{11}$
	Trísticos	indiferente	$a_x a_x a_x$
4	Cuarteta	consonante	$abab$
	Redondilla	consonante	$abba$
	Serventesio	consonante	$ABAB$
	Cuarteto	consonante	$ABBA$
	Copla	asonante	$xaxa$
	Seguidilla	indiferente	$x_7 a_5 x_7 a_5$
	Cuaderna vía	consonante	$x_{14} x_{14} x_{14} x_{14}$ *
	Estrofa sáfica	indiferente	$x_{11} x_{11} x_{11} x_5$
5	Quintilla	indiferente	$abbab$†
	Quinteto	consonante	$ABBAB$†
	Lira	indiferente	$a_7 b_{11} a_7 b_7 b_{11}$
6	Sextilla	consonante	$abaaba$†
	Sexteto	consonante	$ABAABA$†
	Sexteto correlativo	consonante	$a_7 b_7 c_{11} / a_7 b_7 c_{11}$
	sexteto-lira	consonante	$a_7 b_{11} a_7 b_{11} c_7 c_{11}$
	Sexta rima	consonante	$a_{11} b_{11} a_{11} b_{11} c_{11} c_{11} / a_{11} c_{11} c'_{11} b_{11} b_{11} c'_{11}$
	Estrofa manriqueña	consonante	$a_8 b_8 c_4 a_8 b_8 c_4$
7	Séptima	indiferente	$abbab$
	Seguidilla compuesta	indiferente	$x_7 a_5 x_7 a_5\ b_7 a_5 x_7 b_5$
8	Octavilla	consonante	$a_8 b_8 b_8 c'_8\ d_8 e_8 e_8 c'_8$
	Octava real	consonante	$a_{11} b_{11} a_{11} b_{11} a_{11} b_{11} c_{11} c_{11}$
	Copla castellana	consonante	$a_8 b_8 b_8 a_8\ c_8 d_8 d_8 c_8$
	Copla de arte mayor	consonante	$a_{12} b_{12} b_{12} a_{12}\ a_{12} c_{12} c_{12} a_{12}$
	Octava italiana	consonante	$ABBE'\ CDDE'$
10	Décima	consonante	$abba\ ac\ cddc$
	Copla real	consonante	$abbab\ cdcdc$
	Ovillejo	consonante	$a_8 a_4 b_8 b_4 c_8 c_4\ c_8 d_8 d_8 c_8$

* También de dieciséis sílabas.

† Las rimas alternas de acuerdo a las restricciones descritas.

Cuadro 5.2: Estrofas.

Encontramos asimismo estrofas heterométricas, como el sexteto-lira, compuesto de

heptasílabos y endecasílabos en versos alternos, tal como en $aBaBcC$ o $abCabC$ (100). Existen también combinaciones heterométricas de arte menor, como la sextilla *de pie quebrado*[78]. Un caso particular de esta estrofa con gran tradición en las letras castellanas es la llamada *estrofa manriqueña*, por haberla usado Jorge Manrique en el siglo XV para componer las *Coplas a la muerte de su padre*. En esta estrofa, los tetrasílabos se disponen en el tercer y sexto verso, tal que resulta $a_8b_8c_4a_8b_8c_4$. También de seis versos son las *estrofas aliradas*, que alternan endecasílabos y heptasílabos según el esquema $aBaBcC$ (102), pero que admiten además otras combinaciones.

(99) Si lo que la Infanta yerra
peregrino huésped, curas,
haciendo al infierno guerra,
dirán todas las criaturas:
«¡Gloria a Dios en las alturas,
y paz al hombre en la tierra!».
(Calderón, *El veneno y la triaca*, vv. 1328-1334)

(100) Aneguen mis enojos
estos campos con llanto de mis ojos.
Y este monte, que ha sido
áspero monumento,
aumente el sentimiento,
aun sin tener sentido,
y, enternecido, el suelo
muestre en su llanto eterno desconsuelo,
(Calderón, *Judas Macabeo*, vv. 492-497)

(101) ¡Qué ventura
que solo el tiempo os destroce,
cuando el sol solo os conoce;
y en esta selva segura,
lo que vuestra vida dura,
libres siempre, nadie os goce!
(Calderón, *A secreto agravio, secreta venganza*, vv. 299-304)

(102) Un sabio que escribía
en su cama, una vez incorporado,
la mano que movía,

[78]También aparece como *copla de pie quebrado* (101). Sin embargo, dado que la copla tiene un tamaño variable, optaremos por denominar este metro como sextilla en lugar de utilizar el *hiperónimo* copla.

por ser entonces el invierno helado,
de suerte se le helaba
que apenas letra ni razón formaba.
(Lope de Vega, *Poder vencido y amor premiado*, vv. 3033-3038)

Las estrofas de siete versos no resultan tan productivas en las letras castellanas como las otras que hemos visto. Mencionaremos la *seguidilla compuesta* entre las estrofas de este número de sílabas, que se compone de una seguidilla simple y un bordón o estribillo de tres versos adicionales, distribuidos según la forma $x_7a_5x_7a_5/b_5x_7b_5$. No obstante, la estrofa varía en sus versos según el bordón. Asimismo, encontramos una extensión a siete versos de la estrofa alirada repitiendo el metro en los dos últimos (103).

(103) Amor, que lo imposible facilitas
con poderosa fuerza blandamente,
allanando las cumbres:
¿por qué las nubes de mi sol no quitas?
¿Por qué no muestras por algún Oriente
las dos hermosas cumbres
que dan rayos al sol, luz a tus ojos,
por quien te rinde el mundo sus despojos?
(Cervantes, *La entretenida*, vv. 577-584)

Las estrofas de ocho versos gozan de mayor estima en la literatura española. La agrupación isométrica de arte mayor más típica es la *octava*, pero existen diferentes variantes de esta. Así, se conocen estrofas como la *copla de arte mayor* —también llamada de Juan de Mena, por haberla cultivado este poeta con gran acierto—, que adopta el patrón $a_{12}b_{12}b_{12}a_{12} : a_{12}c_{12}c_{12}a_{12}$. Sin embargo, es muy difícil de encontrar en el teatro del Siglo de Oro. La configuración típica de la copla de arte menor es *abba* : *acca* (Navarro Tomás, 1991, p. 266). La *octava real* surge como una modificación de la octava real siciliana, en la que se altera la última rima para obtener un pareado en acorde a *ABABABCC* (104), y se emplea en el teatro para parlamentos graves (Díaz Rengifo, 1592/2012, p. 252; Navarro Tomás, 1991, p. 255).

En cuanto a las estrofas de arte menor, desde el Renacimiento se encuentran *octavillas*, pero estas no son más que la unión de dos redondillas en las que una de sus rimas es coincidente. Para el periodo que nos ocupa, entendemos la estrofa como una composición de otras menores, ya que no adquiere entidad propia hasta entrado el siglo XVIII, cuando se introduce la *octavilla italiana*, que aporta el rasgo distintivo

Enfático	puro		1-6-10
Enfático	pleno		1-6-8-10
Heroico	puro		2-6-10
Heroico	pleno		2-4-6-8-10
Heroico	corto		2-4-6-10
Heroico	largo		2-6-8-10
Heroico	difuso		2-4-10
Melódico	puro		3-6-10
Melódico	pleno		1-3-6-8-10
Melódico	largo		3-6-8-10
Melódico	corto		1-3-6-10
Sáfico	puro		4-8-10
Sáfico	puro	pleno	1-4-8-10
Sáfico	pleno		1-4-6-8-10
Sáfico	corto		4-6-10
Sáfico	corto	pleno	1-4-6-10
Sáfico	largo		4-6-8-10
Sáfico	largo	pleno	2-4-8-10
Sáfico	difuso		4-10
Sáfico	difuso	pleno	1-4-10
Sáfico	inverso		1-6-7-10
Vacío	puro		6-10
Vacío	largo		6-8-10
Dactílico	puro		4-7-10
Dactílico	pleno		1-4-7-10
Dactílico	corto		2-4-7-10
Galaico	antiguo		5-10
Italiano	puro		7-10

Cuadro 5.3: Clasificación del endecasílabo.

del último verso agudo de cada semiestrofa. La *copla castellana* no difiere en esto, aunque sí en la rima de la segunda semiestrofa, de manera que presenta un patrón de según el modelo *abba* : *cddc* (105). Esta forma tuvo cierto predicamento hasta la época de Juan de la Cueva, pero las siguientes generaciones optaron por la redondilla emancipada (Navarro Tomás, 1991, p. 267). Pueden ser también de *de pie quebrado* si alternan versos de ocho y cuatro sílabas.

(104) «Señor, mirad por vuestra casa atento;
que el Conde y la Duquesa en vuestra ausencia...»
No me ha sido traidor el pensamiento,
habrán regido mal, tendré paciencia.
«ofenden con infame atrevimiento
vuestra cama y honor», ¿Qué resistencia
haran a tal desdicha mis enojos?

«Si sois discretom os lo diran los ojos».
(Lope de Vega, *El castigo sin venganza*, vv. 2484-2491)

Las últimas estrofas que repasaremos en este capítulo son las compuestas de diez versos. No iremos más allá, puesto que estamos de lleno en el terreno en el que las agrupaciones se consideran estructuras complejas a partir de la combinación de estrofas simples. Sirva para ilutrarlo la copla castellana del ejemplo 105, que podría dividirse en dos estrofas simples, redondillas en este caso. De esta manera, una vez hayamos establecido en la parte práctica el modo de agrupar las estrofas simples, podremos extrapolar el método a cualquier otra forma compleja para clasificarla según patrones estróficos, análogamente a lo hecho con los versos.

(105) A Dios pongo por testigo,
si a tal quisiera venir,
mas puédeseme decir:
«mensajero sois, amigo».
Que bien saneado estó
que dirán de mi llegada:
«aunque traéis la embajada,
no merecéis culpa, no»[79].
(Juan de la Cueva, *La muerte del rey don Sancho...*, vv. 172-178)

Así, entre las estrofas de diez versos citaremos la *copla real*, que podría analizarse en realidad como la unión de dos quintillas; la *décima* o *espinela* —ya que se atribuye su invención al ingenio de Vicente Espinel—, que, a su vez, está compuesta de dos redondillas unidas por dos versos de enlace en los que se repite la primera rima de cada una de las redondillas (106) y, por último, el *ovillejo* o *séptima real*, una forma compuesta de tres pareados octosilábicos y tetrasilábicos y una redondilla octosilábica (107).

Existen no obstante construcciones estróficas de verso variable, como la *estancia*, que combina entre nueve y veinte versos heptasílabos y endecasílabos con rima consonante. En su modalidad más regular, consta de una *fronte* o *frente* y una *sirma* o *coda*. La frente la componen dos *piedi* o *pies* de tres versos e igual esquema rítmico, denominado el primero *vuelta* o *tornata* y el segundo *revuelta*. La coda comienza con un *eslabón*

[79]Nótese que se emplea este metro tradicional para tratar un tema clásico del romancero. No resulta probable que esto sea producto de la mera casualidad, sino que, más bien, sugiere una percepción de la copla castellana como una estrofa arcaizante, lo que bien podría explicar el desuso en el que cayó con la llegada de la Edad Moderna.

(también *chiave*) heptasílabo que rima con el último verso de la fronte, al que siguen otros versos hasta concluir con una coda de pareados.

(106)

Dorotea: litigantes
sobre tu amor, Lelio y yo,
la esperanza nos citó
a tus estrados amantes.

Amigos éramos antes;
mas pleitos de tu beldad

mudan nuestra voluntad
en competencia enemiga,
que si es cuerdo, no hay quien diga
que en pleitos hay amistad.

(Tirso de Molina, *Santo y sastre*, vv. 847-856)

(107)

Céfiro, ¿en quién dicha espera?
En una fiera.

¿Y quién a Ifis da desmayo?
Un bello rayo.

¿En quién Pigmaleón no medra?
En una piedra.

Ninguno llegue a ser yedra
del laurel que ama, porque hoy
lloren todos, que yo soy
la fiera, el rayo y la piedra.

(Calderón, *La fiera, el rayo y la piedra*, vv. 2060-2069)

Poemas estróficos

Denominamos poemas estróficos a aquellos que, como su propio nombre sugiere, están compuestos por estrofas. Esto es, los versos se agrupan en unidades intermedias dentro del poema. En la literatura española se han usado con más o menos profusión diferentes formas, pero los siguientes ocupan un lugar destacado (Quilis, 1984/2013, pp. 126-150).

La *canción* es un tipo de poema que proporciona gran libertad compositiva al poeta, ya que, además de no exigirle fijar un número de estrofas concreto, estas también varían en su número de versos. En general, las estrofas oscilan entre seis y doce versos en la canción provenzal o nueve y veinte en la petrarquista, así como quince en el modelo de Boscán o trece el de Garcilaso. Tampoco hay una regla fija que dicte la naturaleza de la rima y su distribución. A pesar de todo, hay una estructura global fija, ya que la disposición de la primera estrofa se repite en las siguientes. Cada una de estas estrofas tiene dos partes, unos versos iniciales de frente, que se divide a su

vez, por lo general, en dos pies, y una parte final o coda, compuesta por uno más versos o uno o más *volte*[80] a imagen de los pies de la frente. La canción concluye con una estrofa más corta, la *tornata* o *envío*. Puede darse asimismo un eslabón de unión entre frente y coda. Su presencia en el teatro es escasa, aunque no del todo desconocida. En el ejemplo 108, frente y coda tienen dos partes de cuatro versos, el envío también de cuatro y sin verso de enlace.

(108) Llorente pidió a su prima
Constanza le dé a beber,
y ella quísolo hacer
y echóle el cántaro encima.

Sintiéndose fatigado
de sed, de amor y calor,
le demandó por favor
agua, estando ya abrasado.

No se esquiva aunque se estima,
y en empezando a beber
ella le dejó caer
el cántaro todo encima.

Rió, desque así lo vido,
y él comenzó a sacudirse,
y acometió para irse,
colorado de corrido.

Ella dijo: — ¿Esto os lastima?
Torna si queréis beber,
y dejaros he caer
el cántaro y agua encima

(Juan de la Cueva, *Los siete infantes de Lara*, vv. 503-522)

La *glosa* carece de una extensión fija. Se caracteriza por componerse de una poesía breve llamada *texto* y la glosa propiamente dicha, que consiste en un comentario al anterior distribuido en tantas estrofas (décimas por lo general) como versos componen el texto. Deberemos, pues, verificar si hay repeticiones de versos y si la estructura métrica sugiere una poesía de este tipo. En el ejemplo 109, tenemos el texto en la primera estrofa en cursiva, seguido de una quintilla que anuncia la glosa de décimas que sigue, cuyo último verso —correspondiente al texto, o sea, a la primera quintilla— hemos enfatizado también mediante letra cursiva.

(109) *En fin, señora, me veo*
sin mí, sin vos, y sin Dios.
Sin Dios, por lo que os deseo;
sin mí, porque estoy sin vos;
sin vos, porque no os poseo.

Y por si no lo entendéis,
haré sobre estas razones
un discurso, en que podréis
conocer de mis pasiones
la culpa que vos tenéis.

[80]Tomaremos prestado el término de la nomenclatura italiana porque la forma *verso* podría inducir a confusión con uso más común en la métrica.

Aunque dicen que el no ser
es, señora, el mayor mal,
tal por vos me vengo a ver,
que para no verme tal,
quisiera dejar de ser.
En tantos males me empleo,
después que mi ser perdí,
que aunque no verme deseo,
para ver si soy quien fui,
en fin, señora, me veo.

A decir que soy quien soy,
tal estoy, que no me atrevo,
y por tales pasos voy,
que aun no me acuerdo que debo
a Dios la vida que os doy.
Culpa tenemos los dos,
del no ser que soy agora,
pues olvidado por vos
de mí mismo, estoy, señora,
sin mí, sin vos y sin Dios.

Sin mí no es mucho, pues ya
no hay vida sin vos, que pida
al mismo que me la da;
pero sin Dios, con ser vida,
¿quién si no mi amor está?
Si en desearos me empleo,
y él manda no desear
la hermosura que en vos veo,
claro está que vengo a estar
sin Dios, por lo que os deseo.

¡Oh, qué loco barbarismo
es presumir conservar
la vida en tan ciego abismo
hombre que no puede estar
ni en vos, ni en Dios, ni en sí mismo.
¿Qué habemos de hacer los dos,
pues a Dios por vos perdí,
después que os tengo por dios,
sin Dios, porque estáis en mí,
sin mí, porque estoy sin vos?

Por haceros sólo bien,
mil males vengo a sufrir;
yo tengo amor, vos desdén,
tanto, que puedo decir:
¡mirad con quién y sin quién!
Sin vos y sin mí peleo
con tanta desconfianza.
Sin mí porque en vos ya veo
imposible mi esperanza;
sin vos, porque no os poseo.

(Lope de Vega, *El castigo sin venganza*, vv. 1916-1975)

La *sextina* está formada por seis estrofas de seis versos de arte mayor no rimados acabados en seis sustantivos diferentes de dos sílabas pero en un orden distinto en cada estrofa, y una *contera*, que es una estrofa de tres versos que contiene los seis sustantivos en la forma *ABCDEF FAEBDC CFDABE ECBFAD DEACFB BDFECA A-B D-E C-F* (110). Sin embargo, los dramaturgos de la época se prodigaron poco en ella, aunque aún se encuentran en unas pocas obras de comienzos del siglo XVII.

(110) Hermosas, claras, cristalinas fuentes,
jardines frescos, celebrados árboles,
que aquí me vistes de Jarifa hermano,

ya no soy el hermano de Jarifa;
ya puedo ser su amante y ser su esposo:
dad todos parabién a Abindarráez.

Ya no soy aquel triste Abindarráez
que os daba tanto llanto, puras fuentes;
ya no escribiré hermano, sino esposo,
por las cortezas de los verdes árboles.
Pero, si no me quiere mi Jarifa,
¡cuanto mejor me fuera ser su hermano!

Mas aunque no me quiera, el ser su hermano
ya quita la esperanza a Abindarráez
de la gloria que el alma ve en Jarifa.
Dirán que esto es verdad las sordas fuentes,
y sus hojas harán lenguas los árboles:
tanto es el bien de poder ser su esposo.

Si solo el ser posible ser su esposo
estorbaba del todo el ser su hermano,
jardines, hiedras, flores, plantas, árboles,
aquí donde lloraba Abindarráez,
hechos sus ojos caudalosas fuentes,
aquí se llama esposo de Jarifa.

¡Cielos! ¿Que gozar puedo de Jarifa?
¿Que ya es posible que yo sea su esposo?
Riendo lo murmuran estas fuentes,
que me llamaron tristemente hermano.
Decid que soy su esposo Abindarráez,
que el viento os dará voz, amigos árboles.

¡Que de veces al pie de aquestos árboles
miré los bellos ojos de Jarifa,
y ella me dijo: «¡Hermano Abindarráez!»
Pues ya su esposo soy, no soy su hermano;
o, a lo menos ya puedo ser su esposo:
decidselo, si vuelve, claras fuentes.

Fuentes, ya cesa el llanto; verdes árboles,
ya parto a ser esposo de Jarifa,
que ya no soy su hermano Abindarráez.

(Lope de Vega, *El remedio en la desdicha*, vv. 309-437)

El popular *soneto* (111) gozó del mayor reconocimiento entre los poemas de endecasílabos en el Siglo de Oro (Navarro Tomás, 1991, p. 252). Se caracteriza por sus catorce versos de once sílabas en su modalidad más extendida, divididos en dos cuartetos y dos tercetos, cuyo esquema clásico es $ABBA\ ABBA\ CDC\ DCD$, aunque admite otras combinaciones. En los tercetos, siguen al modelo clásico en popularidad $CDE : CDE$, $CDE : DCE$, $CDC : EDE$, $CDE : DEC$, $CDC : CDC$ y $CDE : EDC$, sin que un poeta se incline necesariamente siempre por una u otra variante.

(111) Apenas el hibierno helado y cano
este monte con nieves encanece,
cuando la primavera le florece
y el que helado se vio se mira ufano.

Pasa la primavera y el verano
los rigores del sol sufre y padece;
llega el fértil otoño y enriquece
el monte de verdor, de fruta el llano.

Todo vive sujeto a la mudanza:
de un día y otro día los engaños
cumplen un año, y este al otro alcanza.

Con esperanza sufre desengaños
un monte que, a faltarle la esperanza,
ya se rindiera al peso de los años.

(Calderón, *Eco y Narciso*, vv. 1385-1398)

El *villancico* es una composición versos octosílabos u hexasílabos que se divide en dos partes: un *estribillo* de entre dos y cuatro versos xa y un *pie de estrofa* de seis o siete versos $xxxxaa$, los últimos de los cuales riman con el estribillo (112).

(112) Entra mayo y sale abril,
¡cuán garridico le vi venir!

Entra mayo coronado
de rosas y de claveles,
dando alfombras y doseles
en que duerma amor, al prado.
De trébol viene adornado,

de retama y toronjil.

Entra mayo y sale abril,
¡cuán garridico le vi venir!

(Tirso de Molina, *La peña de Francia*, vv. 2045-1006)

El zéjel es una composición usualmente de versos octosílabos que está formada por un estribillo de uno o dos versos (a), una segunda estrofa llamada *mudanza* de tres versos monorrimos (b) y un último verso o *vuelta* (c) que rima con el estribillo, quedando en la distribución *aa* : *bbba* (113).

(113) ¡Ay, Fortuna,
cógeme esta aceituna!

Aceituna lisonjera,
verde y tierna por defuera,
y por de dentro madera,
fruta dura e importuna.

¡Ay, Fortuna,
cógeme esta aceituna!

Fruta en madurar tan larga
que sin aderezo amarga;
y aunque se coja una carga,
se ha de comer sola una.

¡Ay, Fortuna,
cógeme esta aceituna!

(Lope de Vega, *El villano en su rincón*, vv. 2043-2056)

Poemas no estróficos

Además de los poemas estróficos, los versos se agrupan también por sí mismos en otras estructuras métricas de extensión variable sin necesidad de estar integrados en uniones intermedias como la estrofa. Existen varios tipos de poemas no estróficos de extensión variable que aparecen de manera recurrente en la literatura castellana (Quilis, 1984/2013, pp. 150-169; Navarro Tomás, 1991, p. 254). El más relevante de todos ellos para este trabajo es, tal vez, el *romance* (114), tanto por su tradición como por la profusión con la que los poetas lo han usado, pero, sobre todo, por ser la agrupación variable más común del teatro áureo. Se trata de una serie ilimitada de octosílabos con rima asonante en los pares y suelta en los impares. En su vertiente culta, estos se agrupan en cuartetas y suele intercalarse un estribillo popular con heptasílabo o endecasílabo, un pareado de octosílabos con rima diferente al romance, o un dístico endecasílabo con la misma rima. No es extraño encontrar combinaciones de metros diferentes, como octosílabos y hexasílabos.

(114)

Un gallardo caballero
hermosamente vestido
a nuestra quinta ha llegado.
¡Ay, Lauro, yo soy perdido,
sin duda, es aqueste el rey!
¿Quién es?
Es un hombre erguido,
tan resuelto y tan bizarro
que solo de haberle visto
vengo temblando de miedo.
El rey es.
Él no ha pedido
licencia, que ya se ha entrado.
¿Qué hay, Elena?
Señor mío,

(Enríquez Gómez, *Engañar para reinar*, vv. 777-788)

Aparte del habitual romance octosilábico, en la métrica castellana se da también una variedad menor de este, compuesta de versos de menos de ocho sílabas, que se denomina *romancillo* (115). También relacionado con el romance, así como con el villancico, encontramos el poema llamado *letrilla*, que se compone de un número indeterminado de redondillas o quintillas dobles y un estribillo o dos alternados. Suele constar de versos octosílabos o hexasílabos y aparece indistintamente con rimas asonantes y consonantes.

(115) ¡Ay, riguroso estado,
ausencia mi enemiga,
que dividiendo el alma
puedes dejar la vida!
¡Cuán bien por tus efetos
te llaman muerte viva,
pues das vida al deseo
y matas a la vista!
¡Oh, cuán piadosa fueras,
si al partir de Medina
la vida me quitaras
como el alma me quitas!
En ti, Medina, vive
aquella Inés divina,
que es honra de la corte
y gloria de la villa.
Sus alabanzas cantan
las aguas fugitivas,
las aves, que la escuchan
las flores, que la imitan.
Es tan bella que tiene
envidia de sí misma,
pudiendo estar segura
que el mismo sol la envidia;
pues no la ve más bella,
por su dorada cinta,
ni cuando viene a España
ni cuando va a las Indias.
Yo merecí quererla.
¡Dichosa mi osadía,
que es merecer sus penas
calificar mis dichas!
Cuando pudiera verla,
adorarla y servirla,
la fuerza del secreto
de tanto bien me priva.

Cuando mi amor no fuera
de fe tan pura y limpia,
las perlas de sus ojos
mi muerte solicitan.
Llorando por mi ausencia
Inés quedó aquel día,
que sus lágrimas fueron
de sus palabras firma.
Bien sabe aquella noche
que pudiera ser mía.
Cobarde amor, ¿qué aguardas,
cuando respetos miras?
¡Ay, Dios, qué gran desdicha,
partir el alma y dividir la vida!

(Lope de Vega, *El caballero de Olmedo*, vv. 1610-1659)

Un poco menos frecuente que el romance, pero también de gran popularidad en el teatro del Siglo de Oro, es la *silva* (116). Esta combina normalmente versos heptasílabos y endecasílabos con rima total, aunque admite versos sueltos intercalados y construcciones con endecasílabos exclusivamente. Junto a la silva hallamos el *madrigal*, que comparte con aquella la carencia de una cantidad fija de estrofas o versos por estrofa y también hace combinaciones variables de heptasílabos y endecasílabos. Lo que diferencia una y otra composición es que la segunda expresa un pensamiento amoroso de extensión breve. En términos computacionales, esto implica que hay que plantear una hipótesis en consonancia con el número de versos porque, para identificar el tema automáticamente, habría que emplear técnicas de *topic modelling*, lo que incrementaría la complejidad del trabajo[81].

(116) Cuestión fue, no apurada hasta este día,
¿cuál hace más, aquel que desafía
a otro a un sitio aplazado,
o el que al sitio salió desafiado?
Y bien ahora pudiera
la cuestión resolver el que me viera
batallando conmigo,
porque no hay tan cruel fiero enemigo,
como es el pensamiento del que aguarda.
Mucho don Félix tarda.
Sin duda que ha escogido,
de don Diego celoso y ofendido,
verse con él primero.
Mas yo no cumpliré si no le espero.
¿Quién en el mundo, cielos,

[81] No obstante, este es un campo que hemos empezado a explorar con vistas a futuras investigaciones.

> se vio, sin dama, sin amor, sin celos,
> en tal lance empeñado?
> ¡Que el prestar a un amigo mi criado
> de suerte lo disponga,
> que mi opinión en tal empeño ponga!
> Digo que aquestos días
> toda mi vida es caballerías,
> pues no hallo en ella cosa
> que parecer no pueda fabulosa.
> Una dama tapada me ha dejado,
> sin decirme quién es, enamorado;
> un criado me ha puesto,
> porque así su ignorancia lo ha dispuesto,
> en trance de perderme; y un amigo,
> sin quererlo, me ha dado un enemigo.
> Mas ¿qué me admiro? Si hallo a cada paso
> que estos son los empeños de un acaso.
>
> (Calderón, *Los empeños de un acaso*, vv. 2046-2076)

Asimismo, encontramos *poemas de versos sueltos*, que responden a diversas necesidades, como el afán de imitar los modelos latinos o adaptarse a partes cantadas. Todavía más complicada llega a ser la *ensalada*, que no solo mezcla metros a discreción sino también métricas, ya que aúna varios idiomas (Díaz Rengifo, 1592/2012, p. 314; Domínguez Caparrós, 1985, *s.v. ensalada*). En estos casos, habremos de dejar la clasificación a la mano humana porque, además de reconocer palabras, ritmos y rimas, obligaría a identificar lenguas y métricas distintas a la silábica castellana, lo que sobrepasa las posibilidades de esta investigación y requeriría un trabajo propio.

Usaremos en cualquier caso esta selección de agrupaciones como base clasificatoria de la parte empírica del trabajo, a la que podremos añadir otras más específicas a la vista de los versos sueltos que resultaren de procesar el corpus de pruebas.

Parte II

Aplicación empírica

Es muy misteriosa, llena de figuras y cifras, oscura y no patente para todos. Tienen sus vocablos y maneras de hablar muy diferente significación de la que saben los vulgares trilingües. Por donde el que construyera la letra y tomare el sentido que resulta de la construcción gramatical caerá en muchos errores.

Juan Huarte de San Juan, *Examen de ingenios para las ciencias*

De la teoría a la práctica 6

6.1 Computador y papel

Bien pertrechados de fundamentos teóricos sobre los que sostener el análisis de los textos, podemos por fin abordarlos empíricamente con garantías. En esta segunda parte formalizaremos lo visto hasta ahora para instruir al computador sobre el modo de aplicarlo a los textos y extraer de ellos la información requerida. ¿De qué manera lo haremos? Como adelantamos en la introducción, la idea es ir tomando lo expuesto en la parte que acabamos de concluir y formalizarlo en esta parte práctica empleando una notación algorítmica de *pseudocódigo*[82]. En cierto modo, lo que ofrecen estas páginas no va a dejar de seguir siendo un desarrollo teórico, pues no usaremos código fuente *real* de un lenguaje de programación específico que pueda ser copiado y ejecutado directamente en un computador, sino que encontraremos la idealización de formalizaciones generales. Estas idealizaciones se representan mediante una tecnología que ha probado su valía durante milenios: la escritura. Si bien intentaremos descomponer cada proceso en bloques relativamente sencillos para que puedan ser seguidos sin dificultad, un papel y un bolígrafo podrían ser de ayuda para orientarse en los pasos menos evidentes.

Son tres las razones que nos animan a plantear la labor de esta manera y no comentando los listados de código del lenguaje de programación en el que hemos hecho las pruebas. En primer lugar, conviene proponer una idea de la forma más clara y concisa posible, por lo que haremos bien en abstraernos de las particularidades a las que están sujetos los lenguajes de programación. Estos obligan a emplear una sintaxis estricta y funciones muy específicas propias de cada uno de ellos, lo que exige

[82] Llamamos *pseudocódigo* o lenguaje de descripción algorítmica a una representación informal compacta del paradigma de un algoritmo. Su función no es ser interpretado por una máquina, sino mostrar el principio operativo del algoritmo.

descomponer las ideas abstractas en procesos menores concretos, más por satisfacer necesidades técnicas que conceptuales. Esto es, deberíamos atenernos a la literalidad de la computación de según unas reglas particulares y renunciar a los atajos contextuales que confieren al lenguaje natural su agilidad. Incluso cuando hay una traducción estructural biunívoca, existen diferencias en cómo expresar el desarrollo que requeriría aclaraciones. No solo serían así mucho más largos los listados —lo que sería un inconveniente menor—, sino también más farragosos y, por lo tanto, difíciles de seguir, lo que ya sí es una preocupación de primer orden.

En segundo lugar, cada lenguaje posee de un repertorio propio de expresiones internas con las que ejecutar operaciones sintéticas. Sin embargo, obligaría a hacerlo analíticamente allá donde no. Aunque todos los lenguajes suelen ser capaces de realizar operaciones elementales, la disponibilidad de recursos para abordar otras más complejas varía de un lenguaje a otro. Estas últimas —como, por ejemplo, aquellas relacionadas con el PLN— requieren *bibliotecas*[83] externas en el mejor de los casos; en el peor, obligan a elaborar los procedimientos desde cero. Sea como fuere, aunque estas bibliotecas actúan como una caja negra cuyo mecanismo interno es ajeno al programa que las invoca, los datos de entrada, que hay que facilitarles, y los de salida, que devuelven, pueden variar significativamente en su formato, incluso cuando son conceptualmente equivalentes. Esta diferencia en el repertorio de los diferentes lenguajes hace que, como decimos, recurrir a uno en concreto ilustraría el uso particular en este, pero complicaría la traducción a otros. De ahí que el pseudocódigo sea una buena alternativa tanto por una cuestión didáctica como por su generalidad.

En tercer lugar, en relación con lo anterior, la explicación en un lenguaje de programación concreto ha de presuponer cierta familiaridad con algunos elementos sintácticos fundamentales que, de lo contrario, requerirían ser explicados. Dado que este no es el sitio apropiado para ello, una sistematización sin otras particularidades que las propias de una notación general facilitan la comprensión, si bien a cambio de llenar los huecos que deja la falta de un lenguaje de programación real con el sentido común del lector.

Sin embargo, no debemos olvidar que el objeto de este trabajo es práctico, queremos dar una solución a un problema real y, no solo eso, la solución debe ser empírica. Por esa razón, no dejaremos lo aquí expuesto en el mundo inteligible, sino que produciremos su realización sensible. Dicho de otro modo, no ha de preocuparse el lector, pues le ahorraremos el trabajo de traducir los algoritmos aquí presentados en programas

[83] Un *biblioteca* (*library* en inglés, por lo que, con más frecuencia de la deseable, hay que leer traducido con el barbarismo *librería*) es una colección externa de funciones que un programa importa para usarlas como propias.

informáticos: para poner a prueba lo discutido programamos una implementación completa en Python 3, la misma que, una vez depurada, se ha estado usando en el proyecto *Sound and Meaning*, cuya última versión disponible a fecha de la entrega de este trabajo se encuentra en los apéndices o, en un formato más conveniente y en su última versión disponible, en los repositorios públicos que se han habilitado en Internet para ello.

Como ya habíamos adelantado, la ubicación de este capítulo antecediendo a la formalización de los aspectos teóricos no responde a la casualidad. Al contrario, se trata de una elección que —creemos— ayuda a comprender el resto del trabajo. En ese sentido, este capítulo es un puente entre los textos y programas reales y la idealización de estos que usaremos en los capítulos siguientes. Aquí se explica la elección de bibliotecas externas y otras diseñadas *ad hoc* en las que delegarlas, así como sus interrelaciones. Esto no es obvio y solo cobra sentido si se conoce de antemano la estructura de dependencia de los módulos en la implementación del modelo. Asimismo, tras el presente capítulo, asumiremos textos en condiciones ideales y no está de más conocer que llegar a ellos no es una cuestión trivial. Excúsese, pues, este interludio técnico.

6.2 Modelo de pruebas

Como hemos dicho, en esta segunda parte no encontraremos el prototipo final. En su lugar, trazaremos un plano en los siguientes capítulos para explicar qué ha de hacerse, aunque sin entrar más de lo estrictamente necesario en cómo ha de llevarse a cabo. Hallaremos, por ejemplo, *listas*, pero no una forma concreta de añadirles nuevos elementos o recorrerlas, ya que esto entraría en las particularidades de cada lenguaje de programación: damos por hecho que el lector es capaz de componer mentalmente una lista y recorrerla de la manera que considere oportuno. Permanecemos, pues, en el ámbito de las ideas. Este capítulo contiene asimismo el conjunto de especificaciones técnicas del prototipo que pretendemos construir a partir de las instrucciones del «plano». El producto final, el resultado de aplicar todo esto —más bien su fotografía, como captura inerte del modelo real— es lo que encontramos en los apéndices.

Lenguaje de programación

Para realizar la implementación del modelo, podríamos haber empleado cualquiera de los muchos lenguajes de programación disponibles en el mercado, pues no faltan

aquellos con capacidad para representar con sencillez las estructuras de datos y algoritmos que requerimos. No obstante, nos decidimos por la versión 3 de Python por varias razones que pasamos a detallar.

Nos hacemos una idea de las cualidades de este lenguaje de programación a partir de su definición como «*general-purpose programming language that blends procedural, functional, and object-oriented paradigms*» (Lutz y Ascher, 2013, p. 6, énfasis en el original). Esto es, tenemos un lenguaje que permite abordar todo tipo de tareas y aproximarse a ellas según diferentes paradigmas. Concretamente, la orientación a objetos de Python contempla la encapsulación y reutilización sin complicaciones (Bird y col., 2009, p. xiii). En la última década, Python se ha convertido en uno de los lenguajes de programación más importantes para tratamiento de datos, *machine learning* y desarrollo de *software* tanto en la academia como en la industria (McKinney, 2018, p. 2).

Python es, además, un *lenguaje de programación interpretado*. Esto es, el código fuente no se traduce a valores binarios de máquina[84] para producir objetos autónomos[85], sino que es ejecutado por un intérprete para correr el programa. Esto tiene aspectos positivos y negativos. Por una parte, el proceso de interpretación es mucho más lento que la ejecución de código nativo. Además de la traducción a instrucciones de máquina, el código ejecutable suele estar más optimizado para la arquitectura física de la máquina. Sin embargo, compilar un programa requiere un tiempo considerable, por lo que tiene sentido en el caso de un programa pensado para ser distribuido como una versión definitiva, pero resulta inconveniente si se va a modificar el programa constantemente, más aún si este no va a ejecutarse intensivamente más que un par de veces frente a muchas de prueba. Esta es precisamente nuestra situación, por lo que el pequeño sacrificio en el tiempo de ejecución lo compensa con creces la agilidad que demuestra su despliegue, lo que en inglés se conoce como *rapid turnaround*: favorece un desarrollo interactivo y ágil, probando en el momento, sin necesidad de recompilar el código fuente cada vez que se introduce una mínima modificación. No solo eso, sino que Python es además *dinámico*, de manera que se asignan atributos y definen variables al vuelo (Bird y col., 2009, p. xiii), al contrario que en los programas compilados de lenguajes clásicos, donde estos han de ser definidos por anticipado.

Otra de las ventajas de Python es su portabilidad. El mismo código se ejecuta en distintas máquinas con distintos sistemas operativos. El único requisito es que dis-

[84] En la jerga técnica se conoce este proceso como *compilar*.

[85] Una vez *enlazados*, estos objetos podrían ser programas ejecutables independientes. Este es el caso, por ejemplo, del lenguaje de programación C, cuya compilación produce código binario para la máquina objetivo, o Java, en cuyo caso, el código binario está destinado a una máquina virtual.

pongan de un intérprete de Python. Este, por su parte, se puede descargar libre y gratuitamente. En nuestro caso, debía correr en todos los computadores de los miembros del proyecto *Sound and Meaning* (Linux, MacOS y Windows), por lo que un lenguaje compilado hubiera planteado dificultades técnicas de cierta envergadura para portar los programas a máquinas con *hardware* y un sistema operativo distinto al PC de tipo Unix con el que se ha llevado a cabo el desarrollo (Lutz y Ascher, 2013, pp. 16-17).

A pesar de esta flexibilidad, Python es un lenguaje de programación poderoso, dotado de recursos para llevar a cabo todo tipo de trabajos, incluso bajo la demanda computacional más exigente. Facilita la construcción modular de grandes programas y provee estructuras de datos predefinidas para ejecutar directamente operaciones relativamente complejas, pues estas están dotadas de herramientas para operar sobre ellas. Si las estructuras de datos básicas no satisfacen la demanda, las bibliotecas que incluye de serie expanden las facultades del lenguaje de programación mucho más allá de su funcionalidad básica. Y, como hemos dicho, a esto hay que sumar las bibliotecas de libre distribución de terceros (Lutz y Ascher, 2013, pp. 17-18).

A pesar de toda esta finura técnica, Python es fácil tanto de usar como de aprender, gracias a la transparencia de su semántica y sintaxis (Bird y col., 2009, p. xiii). Incluso comparado con lenguajes de programación de alto nivel diseñados específicamente para fines didácticos, como Modula-2 , Python resulta ganador en cuanto a sencillez de uso y legibilidad. Esto es así hasta el punto de que incluso ha llegado a denominarse *pseudocódigo* ejecutable (Lutz y Ascher, 2013, p. 20). El lenguaje tiene fama asimismo de poder aprenderse rápidamente, aun sin experiencia previa en la programación, mucho más si ya se posee alguna con otros lenguajes.

Python es además libre y de código abierto. Esto no solo significa que su uso es gratuito, sino que se encuentra bajo una licencia que obliga a proporcionar el código y permite su redistribución sin restricciones, con independencia del grupo destinatario o los fines para los que se vaya a emplear, incluso para la creación de obras derivadas, que pueden ser redistribuidas en los mismos términos. La licencia compromete a preservar la integridad de la obra original en la distribución, pero puede ir acompañada de *parches* para ser aplicados *a posteriori*. Cualquier usuario al que le llegue el programa está sujeto a estas condiciones, que no cambian si el programa se distribuye como parte de otro y, recíprocamente, tampoco afectan a otros programas que se distribuyan junto a él (Open Source Initiative, 2007).

No nos proponemos alterar el código de Python; ya hace lo que necesitamos muy bien. Por lo tanto, solo aprovecharíamos directamente de su carácter libre la posibilidad

de usarlo sin costes adicionales. Sin embargo, esto presenta varias ventajas indirectas que nos afectan sobremanera. El movimiento del *código abierto* surgió en el seno de la comunidad del *software libre*, este último gestado en un ambiente académico como una adaptación de los usos de ese entorno.

> In a math department, anyone would be free to tinker with a proof that someone offered. If you thought you had a better way to prove a theorem, you can take what someone else did and change it. In a classics department, if you believed a colleague's translation of a recently discovered text was flawed, you were free to improve it. Thus, to Stallman, it seemed obvious that you should be free to tinker with and improve the code that ran a machine. (Lessig, 2004, p. 279)

En palabras del ideólogo histórico del software libre Richard M. Stallman (1998/ 2002, pp. 57-62), el código abierto es una concesión pragmática que pasa por la renuncia al compromiso ético del movimiento original, para el que el software cerrado no es solo un inconveniente práctico, sino un problema social. No obstante, el uso de programas de código abierto supone un apoyo al software libre en la medida que ofrece una alternativa moralmente válida al código cerrado. En otras palabras, usando programas de código abierto, promovemos el libre intercambio de ideas y el acceso universal al conocimiento.

Sin renunciar a ese compromiso, lo cierto es que las ventajas pragmáticas del código abierto —si bien sus causas están íntimamente entretejidas con la naturaleza libre del software— no son desdeñables. En el caso de Python, esto se traduce en una enorme comunidad de usuarios y desarrolladores entusiastas. De esta manera, hay disponible una vasta cantidad de documentación que cubre la mayor parte de los problemas técnicos que uno se encuentra. Por otro lado, los miles de bibliotecas disponibles, tanto para llevar a cabo tareas generales como especializadas, alivian el desarrollo de los programas, pues evitan tener que reinventar la rueda una y otra vez en cada ocasión que a alguien quiere echarse a rodar[86].

[86]Existen bibliotecas de PLN como Stanza (Qi y col., 2020) o SpaCy (ExplosionAI, 2023) con modelos avanzados para la lengua española. Hay, además, repositorios públicos de paquetes abiertos a contribuciones de los usuarios, por lo que no solo resulta instalar los desarrollos ajenos, sino compartir los propios. Esta flexibilidad para el usuario novel está muy por delante de la del resto de lenguajes de programación. Como nota adicional, si bien no de índole técnica, pero también con cierto peso en la elección, cabe mencionar el hecho poco conocido de que el nombre Python es un guiño al grupo de humoristas británicos Monty Python (Lutz y Ascher, 2013, pp. 20-21).

Excurso II: Equipo

Podríamos valernos del lenguaje de programación como una formalización abstracta de los algoritmos que queremos implementar, de manera que tendríamos un pseudocódigo algo inflexible, pero bien estructurado, completo y sin ambigüedades. No obstante, nuestro propósito es aplicarlo a la práctica, por lo que necesitamos un computador que corra el código. Por lo tanto, lo aquí expuesto debe ser realizable y, en consecuencia, los planteamientos teóricos deben permitir una realización física factible. Dado que esto constriñe en cierta medida lo que haremos, es menester mencionar estas cortapisas.

Con un presupuesto limitado, se hace necesario encontrar un compromiso de mínimos. Resulta mucho más práctico disponer no solo de una máquina capaz de correr programas en Python, sino también de llevar a cabo otras tareas, que ofrezca un entorno de desarrollo potente y versátil. De las soluciones que encontramos, la que más se ajustaba a las necesidades de este trabajo era una *distribución* de GNU/Linux, por varias razones que pasamos a enumerar. En primer lugar, ofrece de serie una plataforma global con convenientes herramientas para la manipulación de texto como *sed*, *awk* o *grep*. Provee asimismo editores de texto avanzados. Estos, además, admiten extensiones de terceros para incrementar sus capacidades, lo que permite configurar un entorno de programación completo dentro del editor (Figura 6.1).

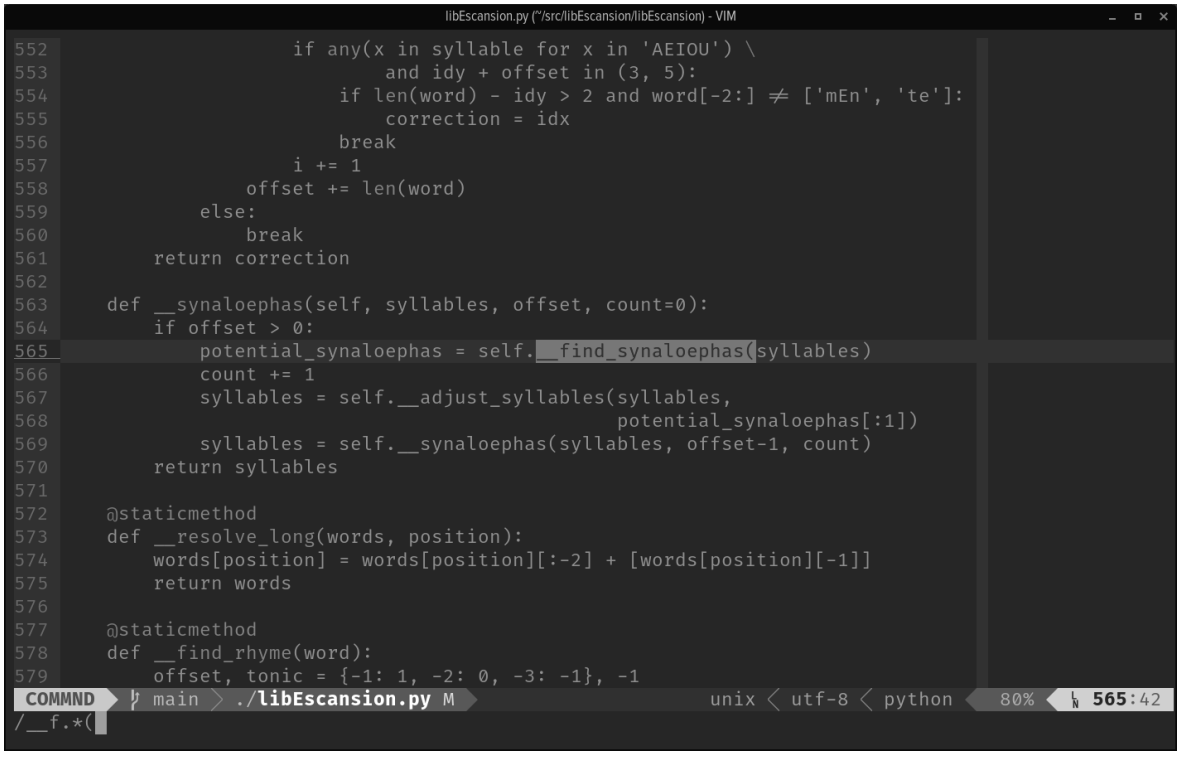


Figura 6.1: Código Python en el editor de texto Vim.

Otra cuestión a tener en cuenta es la filosofía GNU y el software libre. Los paquetes empleados en el desarrollo de este trabajo son de código abierto y se encuentran bajo diferentes licencias libres. Esto garantiza el acceso a los programas y la libertad para usarlos sin restricciones ni coste alguno. Contra lo que podría suponerse, que los programas sean libres no compromete su calidad. Al contrario, se trata de estándares *de facto* tanto en el entorno industrial como en el académico. Considérese, por ejemplo, el paquete de cálculo estadístico R (R Core Team, 2019) o el propio Python en el ámbito del *machine learning*: ambos son herramientas de uso mayoritario en su terreno. El sistema operativo Linux, por su parte, ha demostrado su valía en aplicaciones científicas y comerciales hasta el punto de que los primeros quinientos supercomputadores más potentes del planeta operaban sobre alguna distribución de este sistema operativo en noviembre de 2022 (TOP500.org, 2022) y domina el mercado de los servidores de internet (W^{3}Techs, 2023). Creemos que nuestro trabajo, con unas exigencias mucho más modestas, bien puede aprovecharse también de estos recursos. No se trata, sin embargo, de una cuestión meramente práctica, sino que una porción sustancial del peso responde al compromiso ético con la libre difusión del conocimiento que encarna el software libre.

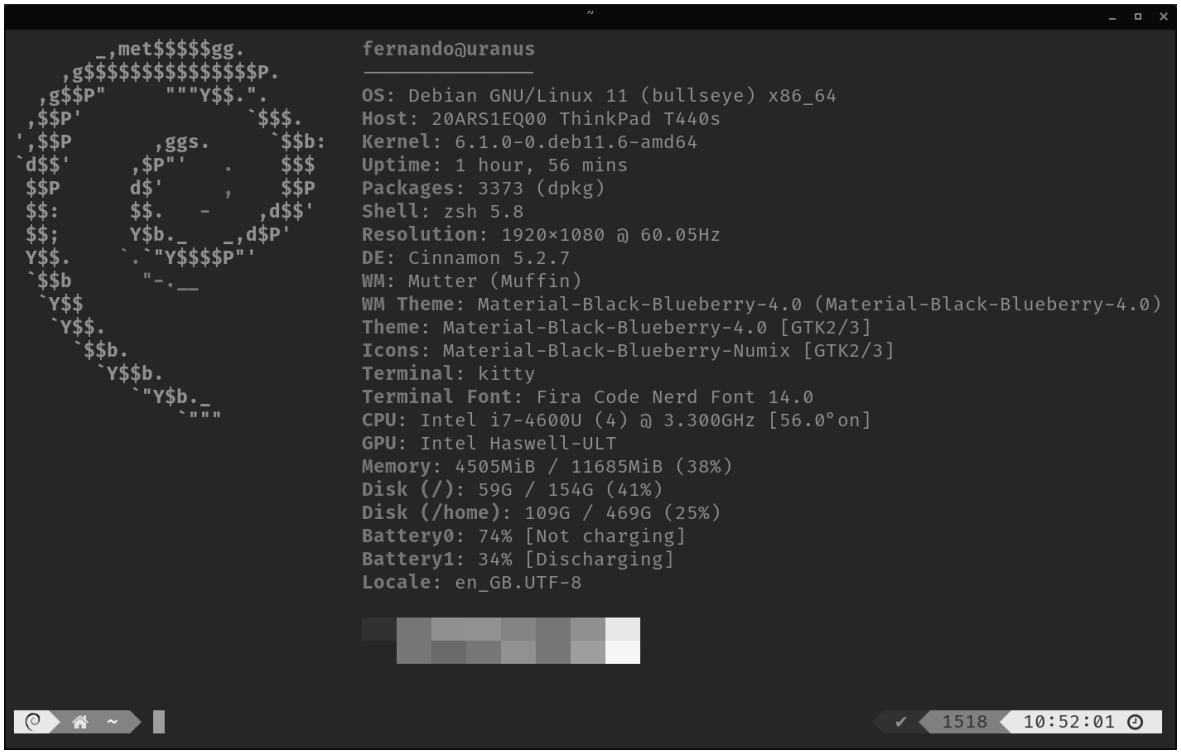


Figura 6.2: Equipo de desarrollo.

En concreto, optamos por la distribución Debian GNU/Linux (2022) en su versión 11 (*Bullseye*), por aunar estabilidad con un completo catálogo de aplicaciones, a la vez que sigue una estricta política de licencias para seleccionar el software incluido.

En cuanto al equipo físico, la mayor parte del trabajo se llevó a cabo en un Lenovo ThinkPad T440s con 12 GB de memoria RAM y un procesador de cuatro núcleos Intel CORE i7-4600U con una frecuencia de reloj de 3,3 GHz[87] (Figura 6.2).

6.3 Estructura del modelo

Para plantear cómo construiremos el modelo, debemos conocer antes lo que va a hacer. En Figura 6.3 observamos una visión panorámica de los procesos que llevaremos a cabo, surgidos de depurar el concepto inicial (Sanz-Lázaro, 2023c). A grandes rasgos, tomamos los textos tal como los encontremos y los preparamos según su estructura dramática para obtener datos semiestructurados con los que el computador empiece a trabajar. No obstante, como analizaremos verso por verso, resulta más conveniente un formato que permita moverse en vertical de esta manera, por lo que tabulamos los datos y obtenemos uno estructurado. Estos datos son los que someteremos al análisis métrico, tal que la salida es una tabla con nuevas columnas para la información métrica de cada verso.

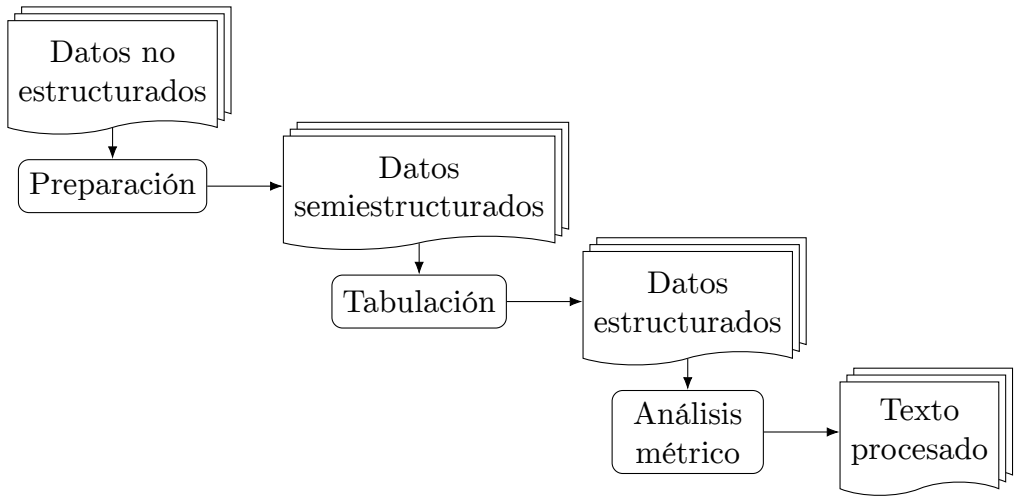


Figura 6.3: Estructura general.

En cuanto a la preparación, consta de una parte de limpieza semimanual variable, que consiste en convertir las fuentes en un formato legible para el programa y organizar el texto de acuerdo con su estructura dramática; esto lo veremos en detalle en el capítulo siguiente. En la primera parte habremos de recurrir en muchos casos a

[87]Posteriormente, para procesar textos de manera masiva, encontramos que correr fragmentos del código en una GPU optimizada en lugar de la CPU del computador aceleraba sustancialmente los cálculos. Así, para la aplicación de los resultados del trabajo, empleamos los controladores comerciales de la empresa NVIDIA para poder ejecutar los programas en una tarjeta NVIDIA GeForce GTX 1060. De la misma manera, los programas completos se corrieron en diversas versiones del sistema operativo MacOS. En nuestro descargo alegaremos que, a pesar de que la portabilidad del programa deja correrlo en entornos más o menos cerrados, en su desarrollo nos ceñimos de manera estricta a software libre.

herramientas externas, mientras que la estructuración dramática se alcanza en gran medida mediante sustituciones con *expresiones regulares*[88] una vez que el texto tiene un formato aceptable. Nos saltaremos este paso en ocasiones, siempre y cuando el texto a tratar esté ya semiestructurado, como es el caso de las ediciones en XML-TEI. De cualquier forma, si los datos se hallan en un formato adecuado, se convierten a datos estructurados de manera rápida sencilla mediante una herramienta para la manipulación automática de tablas.

Los datos tabulados resultantes son los que someteremos al análisis métrico. Este se lleva cabo recorriendo la tabla por filas para escandir los versos uno a uno; la fila correspondiente se actualiza según los resultados con nuevas columnas. El producto es una tabla de versos con anotaciones sobre las particularidades de cada uno en consonancia con su estructura métrica, función y jerarquía en la obra dramática.

6.4 Módulos del programa

Ya sabemos que Python cuenta con una inmensa cantidad de bibliotecas disponibles para llevar a cabo funciones comunes (y no tanto). Hemos de preguntarnos qué tareas podríamos delegar en ellas, si hay implementaciones capaces de encargarse de ello accesibles y, si es así, si estas satisfacen nuestras demandas. De lo contrario, no queda sino diseñar el procedimiento para resolver esas tareas desde cero. Como es natural, aceptaremos sin entrar en disquisiciones las bibliotecas que proveen fórmulas para llevar a cabo operaciones primitivas, esto es, aquellas que sirven para manipular tipos de datos básicos o interactuar con el sistema operativo. Nos centraremos, por el contrario, en las funciones abstractas tales como las representadas en la figura anterior. En nuestro modelo de pseudocódigo, al igual que hacemos en la implementación con el lenguaje de programación real, daremos por supuestas las funciones de tales bibliotecas y las emplearemos sintéticamente sin preocuparnos de su mecánica interna.

Por otra parte, no creamos programas para la limpieza y preparación de los datos, sino que trabajamos con herramientas preexistentes o sencillos *scripts* para organizarlas, por lo que la configuración modular de los pasos más avanzados no afecta a esta etapa inicial. Los sistemas operativos de tipo Unix están equipados de fábrica con potentes programas ya listos para hacer sin esfuerzo las operaciones que requerimos llevar a cabo. Los clásicos `sed` y `awk` para la manipulación de texto, por ejemplo, simplifican un tratamiento minucioso de los más pequeños detalles y admiten regex para

[88] Las *expresiones regulares* (abreviado *regex*, del inglés *regular expression*) son una secuencia de caracteres con unas reglas sintácticas dadas que representan un patrón de búsqueda en un texto.

acotar las modificaciones. Otras herramientas facilitan los trabajos menos delicados, como convertir archivos de texto plano entre diferentes codificaciones. Incluso hay disponibles programas, como los editores de texto Vim y Emacs, capaces de hacer ambas cosas. Por lo tanto, este paso aconseja herramientas orientadas a un uso particular, más que recurrir a lenguajes de programación de uso general, sean reales, como Python, o ficticios, como nuestro pseudocódigo. Resulta mucho más sencillo resolver directamente las contingencias que surjan mediante pequeños mandatos de la línea de órdenes provistos por el sistema operativo.

Una vez disponemos de los datos semiestructurados, empezaremos a automatizar realmente las tareas, ahora ya solventada la necesidad de supervisión humana. Para esto, por una parte, necesitaremos que el computador entienda el texto de entrada y, por otra, que conozca cómo estructurarlo en el formato de salida. Como veremos más adelante, tenemos previstos dos formatos semiestructurados: XML-TEI y VED, este último una notación simplificada propia para representar piezas dramáticas. Abordaremos cada uno de ellos de manera distinta. Para el primero, necesitaremos poder interpretar árboles XML y, para el segundo, aplicar expresiones regulares a las búsquedas. Sin embargo, entender la entrada, como decimos, es solo la mitad del trabajo. Hemos de ser también capaces de traducirla adecuadamente, por lo que necesitaremos algún tipo de herramienta que dé la posibilidad de trabajar con tablas. Las tablas, por su parte, son apenas el modo que tenemos de ordenar los versos. Una vez que hemos recorrido las líneas y llegado a la que deseamos, hay que extraer el verso de ella, escandirlo y modificar la línea mediante la adición de los valores correspondientes en nuevas columnas creadas al efecto.

De esta manera, si deseamos leer el texto semiestructurado, requeriremos herramientas para analizar jerarquías XML y realizar búsquedas con expresiones regulares. Tendremos que usar también algo que facilite la manipulación de tablas. Necesitamos, asimismo, algún instrumento para escandir versos. Para todas estas funciones existen diferentes bibliotecas disponibles listas para usar. La cuestión es: ¿debemos hacerlo? Empezaremos a responder la pregunta considerando las partes menos peliagudas.

El tratamiento de archivos XML, la creación de un sistema de sustitución mediante expresiones regulares y la manipulación de tablas son tareas que queremos completar, pero el modo de hacerlo es irrelevante para el propósito de este trabajo. Nos interesan los aspectos lingüísticos y literarios del análisis, no el mecanismo interno de las herramientas empleadas. Por otra parte, la escansión entra de lleno en las competencias de nuestra disciplina, así que nos ocuparemos de ella en detalle. En las pruebas de concepto, nos limitaremos a utilizar las bibliotecas estándar de Python para expresiones regulares y XML y, para las tablas, emplearemos Pandas, por ser esta tal vez la

biblioteca más popular para este tipo de tareas y, por lo tanto, de la que resulta más sencillo encontrar ayuda y documentación en caso de que se requiera. En el trabajo, obviaremos esas particularidades y asumiremos simplemente que podemos trabajar con tablas y expresiones regulares, sin entrar en cómo hacerlo.

De las bibliotecas de escansión que vimos en la revisión bibliográfica, apenas media docena se encuentran libremente disponibles. Asimismo, estas están orientadas a poemas, que no solo son textos monométricos, sino que además no tienen en consideración algunas irregularidades de los textos dramáticos. Esto hace necesario diseñar una herramienta de escansión específica para nuestro tipo de texto. En Figura 6.4 se esquematiza cómo se ha de llevar a cabo la escansión. Vemos que, además del conjunto de versos de entrada, tiene como salida los mismos versos ya procesados; una forma y otra se encuentran separadas por el programa encargado de análisis métrico. Dentro de este, se va recorriendo la tabla por filas para analizar los versos uno a uno; la fila correspondiente se actualiza en función de los resultados del análisis con nuevas columnas. De esta manera, escandimos cada uno de los versos y, para ello, etiquetamos gramaticalmente sus palabras para distinguir entre tónicas y átonas y hacemos la transcripción fonológica de cada palabra para trabajar con idealizaciones de sonidos y no grafemas.

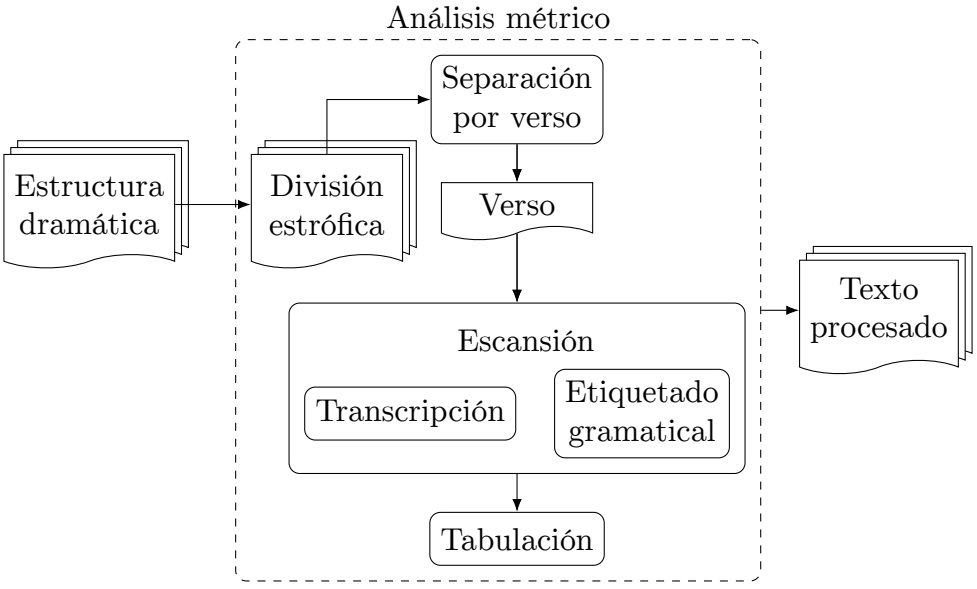


Figura 6.4: Módulos.

Cabría cuestionar la conveniencia de delegar la transcripción y el etiquetado gramatical en bibliotecas externas. En cuanto a lo primero, se trata de una tarea compleja que requeriría, como mínimo, invertir tanto tiempo en ella como el que hemos dedicado a esta tesis. Por otra parte, existen diferentes herramientas disponibles capaces de llevar a cabo el proceso con una gran precisión. Por lo tanto, parece sensato dar un

paso atrás aquí y tomar lo que se nos ofrece. En concreto, hemos optado por Stanza NLP. Se trata de una *caja de herramientas*[89] para Python de código abierto capaz de procesar más de sesenta lenguas, que incluye, entre otras funciones, la tokenización, lematización, expansión de tokens de varias palabras, etiquetado gramatical y análisis sintáctico. Dicho de otro modo, todo cuanto necesitamos. Esto lo ofrecen otros analizadores, como spaCy, pero Stanza rinde los mejores resultados para el español (Qi y col., 2020). Asimismo, se trata de un proyecto de investigación académica muy activo, por lo que el repositorio está abierto a contribuciones mediante el reporte de errores, que se resuelven en cuestión de horas. En el pseudocódigo consideramos tan solo una función genérica de PLN, lo que, recalcamos, se puede traducir de muy diversas maneras en un lenguaje de programación, de modo que Stanza NLP es solo una posible forma de entenderlo.

El caso de la transcripción difiere del anterior. Las distintas bibliotecas para la fonología del español que hemos evaluado presentan dos problemas que desaconsejan usarlas. Por una parte, hay una dificultad de naturaleza técnica. Las bibliotecas están preparadas para transcribir exclusivamente texto con ortografía normativa, esto es, únicamente admiten los diacríticos comunes del español. Por lo tanto, no completan su labor si encuentran caracteres ajenos a esta lengua o, incluso, signos diacríticos típicos del verso, como la crema sobre vocales distintas de *u*. El segundo inconveniente es conceptual: la división silábica no considera las excepciones a los diptongos ortográficos. Mientras que el primer obstáculo sería relativamente sencillo de solucionar mediante la edición del código fuente —que si es abierto, lo contempla— o la sustitución durante la preparación de los textos, el segundo requeriría una reescritura a fondo de los programas. Considerando el tiempo que habría que invertir en poner parches, estimamos que salía más a cuenta escribirlo desde cero, tomando nuestros requerimientos particulares como punto de partida y no como extras añadidos *a posteriori*. Por el mismo motivo, el módulo de división silábica lo construimos también desde cero para ajustarlo a las demandas específicas de este trabajo. Esto permite, entre otras cosas, trabajar directamente con símbolos del AFI en las palabras a tratar.

En la implementación práctica nos valimos de otras bibliotecas adicionales, si bien estas resultan aún más irrelevantes para las disquisiciones de esta tesis. Entre ellas, se incluyen las dedicadas a tareas auxiliares como dibujar barras de progreso o cronometrar el tiempo de ejecución de los programas. Al ser estos elementos de control y no tener una función en el desarrollo del programa, como decimos, obviaremos tanto su uso como los motivos que llevaron a su elección.

[89] *Toolkit* en el original.

6.5 Formatos de las fuentes bibliográficas

Para hacer las pruebas con el modelo necesitamos, huelga decirlo, textos: en particular, textos digitalizados. Hemos de dejar a un lado las condiciones ideales y enfrentarnos a las penurias del mundo real, despegarnos del *texto* como idea y aproximarnos a sus realizaciones visibles. En definitiva, describiremos aquí los tipos de texto y procedencia de aquellos de los que nos hemos valido para llevar la abstracción a lo concreto.

Hay que empezar por decir que no existe un repositorio unificado de textos ni un formato estandarizado para codificarlos. Al contrario, los textos están muy repartidos y cada fuente parece tener una estructura propia, aunque superficialmente podrían antojarse similares. Los fondos semiestructurados suponen una minoría, si bien su número aumenta constantemente. Ha de tenerse en cuenta que, por lo general, el interés de la crítica se concentra en el reducidísimo conjunto de autores del parnaso canónico y, de su producción, apenas se presta atención a unas pocas obras selectas. Esto no es menos cierto en el ámbito de las humanidades digitales, por lo que la probabilidad de dar con una buena digitalización de una obra menor es ciertamente escasa.

Desde el punto de vista técnico —por lo menos en lo que concierne al texto, sin entrar en cuestiones de composición y tipografía—, la labor se reduce a media decena de tipos de archivos, varios de ellos tratables con las mismas herramientas. Tenemos datos semiestructurados en XML-TEI y otros no estructurados de texto, como HTML y texto plano, así como también otras codificaciones más complejas, como PDF. Asimismo, encontramos formatos para procesador de texto, como DOC[90].

El propósito de este trabajo es, además, equipar al proyecto Sound and Meaning con una herramienta capaz de extraer datos de obras teatrales áureas e incorporar los textos junto a sus metadatos a un corpus. Este está destinado a servir como fuente para la investigación, por lo que es prioritario que los resultados de estos sean válidos. Esto obliga a restringir las fuentes según una serie de criterios que lo garanticen. Así, para hacer pruebas solo hemos tenido en cuenta las ediciones con las que trabajaba el proyecto. No debe entenderse esto como una limitación técnica, pues, en teoría, el modelo podría analizar textos que no se atuvieran a estos criterios, a pesar de que únicamente aseguraría resultados adecuados cuando aquellos se observan.

Las posibles obras candidatas para el análisis se reducen si tomamos la precaución de considerar solamente aquellas que cuentan con ortografía normalizada y modernizada.

[90] Lamentablemente, los formatos de archivo comerciales predominan sobre los abiertos, lo que, más allá de nuestro trabajo, pone trabas al intercambio de información y conocimiento cualesquiera que sean los fines.

Aunque hemos intentado que los modelos sean capaces de procesar la gran mayoría de las grafías que se encuentran en las digitalizaciones de obras teatrales auriseculares, con independencia de la edición, no debemos olvidar que asumimos criterios ortográficos y fonológicos modernos. De esta manera, los resultados producidos por un texto normalizado y otro sin normalizar podrían no ser comparables debido a la imprevisibilidad del último. Es más, es posible que este tenga inconsistencias internas, lo que podría invalidar incluso un examen individual. Por lo tanto, quien haya de servirse de nuestro trabajo debería tener esto en cuenta a la hora de seleccionar las fuentes que va a usar, pues la consistencia de los análisis con ortografías no normalizadas ofrece garantías insuficientes, máxime si las convenciones de partida divergen.

La escasez de textos es aún más cierta si lo que buscamos son buenas ediciones. Lamentablemente, no todas las fuentes que se encuentran digitalizadas responden a unos criterios mínimos de calidad ecdótica. Por lo general, no presentan problemas generalizados, pero unas pocas ediciones de origen oscuro muestran deficiencias graves, alguna de ellas capaces de alterar el valor de las distribuciones de los ritmos. Por este motivo, el proyecto Sound and Meaning solo ha trabajado con textos de prestigio, pues lo contrario habría obligado a llevar a cabo un minucioso examen filológico de cada edición de procedencia no fiable para verificar su calidad; ahora multiplíquese ese tiempo por los cientos de posibles candidatos. De esta manera, aunque es probable que se hayan excluido muchos textos perfectamente válidos por carecer del aval de un editor, lo cierto es que simplifica la composición del corpus, al permitir integrar nuevos textos de inmediato. Por consiguiente, si ha de incluirse un texto de procedencia incierta, conviene tomar la precaución de examinarlo bien antes de encomendarse a los resultados de su análisis para extraer conclusiones.

Sea como fuere, con estos mimbres hay que hacer el cesto, por lo que más vale acostumbrarse a ello. En una situación óptima, dijimos, encontramos datos semiestructurados, como en el caso del corpus calderoniano (Lehmann y col., 2022), por ejemplo. Asimismo, las obras de Artelope (Oleza Simó y col., 2022) y T$^{\underline{c}}$/12 (TC/12, 2014), aunque no se encuentran directamente accesibles en XML-TEI, aparecen como datos estructurados incrustados en el código HTML de las páginas donde se presentan las obras. Por los notables paralelismos entre las estructuras, podría sospecharse incluso que la página web se ha generado automáticamente a partir de una edición en XML-TEI. Sin embargo, si bien la situación ha mejorado notablemente en la última década, siguen estando tan vigentes como entonces los *desiderata* de Valdés Gázquez (2104a).

La mayoría de las fuentes de donde partimos presentan sus textos en formato PDF. Las principales Prolope (Prolope, 2023) para comedias de Lope de Vega y GRISO (2020), que dispone de una completa colección de autos sacramentales de Cal-

derón. Esta disponibilidad, junto a los textos de CalDraCor, hace que, a día de hoy, la obra calderoniana sea posiblemente la más apta de entre sus contemporáneas para llevar a cabo estudios digitales. No podemos dejar de mencionar la Biblioteca Virtual Miguel de Cervantes (Fundación Biblioteca Virtual Miguel de Cervantes, 2021), aunque, en este caso, dada la diversidad de las procedencias editoriales de sus fondos, conviene seleccionar las obras individualmente.

Finalmente, tenemos obras recopiladas individualmente gracias a la generosidad del propio editor. Estas suelen encontrarse en los formatos propios de los procesadores de texto que incluyen los paquetes ofimáticos comunes. Dado que, en este caso, los textos se obtienen normalmente de forma individual e, incluso un mismo editor se vale de diversas herramientas a lo largo del tiempo —téngase en consideración que el rango temporal de estas ediciones va desde principios de la década de 1990 a textos que acaban de terminarse y aún no están publicados[91]—, estas ediciones son tremendamente heterogéneas. Dada esta variedad de procedencias, nos enfrentamos a diferentes obstáculos para preparar los textos de manera que podamos emplearlos directamente en el computador.

6.6 Diferencias entre textos, textos y textos

La heterogeneidad de los textos requiere aproximarse individualmente a los problemas que presenta cada uno de ellos. No solo nos hallamos ante una plétora de formatos incompatibles entre sí, sino que, incluso, encontramos diferencias en la manera en que se codifican dos textos en el mismo formato. La diferencia principal estriba en que algunos de esos formatos proveen facilidades para la manipulación y análisis textual; otros, por el contrario, suponen un obstáculo que, en ocasiones, resulta insalvable.

El primero de los retos que uno se encuentra para tratar un texto digitalizado —y no por obvio deja de ser el más importante— es obtener el susodicho texto. Aquí se distinguen varias tipologías: por una parte, los textos accesibles públicamente y, por otro, aquellos que no lo están y han de obtenerse a través de diversas vías, como el préstamo o la compra. Estos textos, con independencia de cómo se hayan conseguido, pueden ser libres o tener restricciones legales o técnicas. Esto es, hay textos que prohíben la manipulación y copia en cumplimiento de los derechos de autor, lo que se fuerza en ocasiones mediante la aplicación de un sistema de protección digital anticopia. No obstante, se encuentran textos de dominio público con la misma protección, lo que también impediría tratar un texto, aunque fuera perfectamente legítimo.

[91] Prolope, Instituto de Teatro Clásico de Almagro, comedias.org y CANON60, entre otras.

Asumiremos que trabajaremos legalmente con los textos de los que disponemos y que estos no están restringidos por protección digital alguna. Los formatos más comunes son PDF, TXT, XML y cualquiera de los tipos de archivos privados e incompatibles entre sí —incluso entre diferentes versiones de supuestamente el mismo programa— de los diversos procesadores de texto[92]. Una vez se dispone de acceso al archivo, el objetivo es convertirlo a datos semiestructurados. En el caso de XML, se obvia ese paso porque ya lo está.

6.7 Peor escenario: PDF

Los archivos PDF albergan el texto completo, pero este no es directamente accesible. No se trata de un texto plano ni de un lenguaje de marcado como XML, sino de una descripción de las páginas. Para poder verlo, requerimos de la mediación de un programa capaz de interpretar el código binario del archivo y representarlo de manera visual en la pantalla. Se presenta el problema de que, de esta forma, no hay posibilidad de trasladar la composición de la página directamente y, por lo tanto, tampoco la estructura de la obra. Se necesita convertir ese documento a texto plano y, a la vez, preservar el mayor número de elementos visuales posible, de manera que se minimice el trabajo manual. La forma de hacerlo dependerá de como se haya compuesto el archivo, ya que esto determina el modo en que se codifica el texto.

Cada una de las páginas contiene bloques de instrucciones que definen segmentos de contenidos y sus características. En el caso de los bloques de texto, estos suelen estar repartidos de acuerdo no a un criterio textual, sino a las necesidades de la maquetación. Por ejemplo, para representar *Ejemplo*, ubicaría la E en la posición (x,y) y el segmento *jemplo* en $(x + 1 - n, y)$, de modo que desplazaría el segundo segmento n puntos hacia la izquierda para hacer un acoplamiento parecido a *Ej*, en lugar de conservar el espaciado como en *Ej* (Glyph & Cog, 2022). Esto hace que los programas para la extracción automática del texto aborden la tarea colocándolo en orden de lectura sin más. La complejidad de la maquetación y su proximidad a la jerarquía textual son determinantes para decidir la manera de abordar el texto.

Si la maquetación del documento se aviene hasta cierto punto a la estructura textual, recurriremos a herramientas de conversión de formato como `pdftotext`, que dan la

[92] En el curso de los trabajos llevados a cabo para el proyecto Sound and Meaning, pudo comprobarse que, lamentablemente, los archivos creados con un procesador de textos suelen estar en alguno de los diferentes formatos cerrados de Microsoft Office; excepcionalmente, aparece algún archivo en Open Document Format (ODF). Como anécdota, algunas de las obras del corpus del proyecto estaban compuestas con WordPerfect 5.2, una versión del programa salida al mercado en 1992; aunque los procesadores de texto comerciales modernos fueron incapaces de reconocer ese formato, el paquete ofimático libre LibreOffice.org no tuvo problema para abrirlo.

opción de conservar la distribución espacial de los elementos en la página, aunque no consigan un resultado perfecto. Después, el trabajo consistiría en afinar esta conversión mediante expresiones regulares. De esta forma, se eliminan cabeceras de página, numeración de los versos múltiplo de cinco y aparato crítico, así como las voladas de llamada a nota si las hubiera. Esta parte no presenta muchas dificultades, ya que los saltos de página vienen marcados con un metacarácter.

Sabemos que la primera línea tras el salto de página es la cabecera, por lo que se elimina. Los números de verso aparecen a la derecha del texto, precediendo de manera inmediata al final de línea y después de uno o más espacios o tabuladores que los separan del texto. Sabiendo esto, suprimimos los guarismos que cumplan esas condiciones, ya que no encontraremos un equivalente textual. No suele haber llamadas a nota porque la numeración se corresponde con el número de verso. Si las hubiera, no obstante, esto tampoco presentaría excesivo problema, ya que las llamadas van anejas a una palabra o a un signo de puntuación no precedido por un guarismo y, en este último caso, seguidas de espacio. Aunque, hipotéticamente, podrían darse casos como *123*[1], los guarismos intratextuales son prácticamente inexistentes, por lo que asumimos simplemente que varios de ellos a final de línea son cifras del mismo número.

Las notas a pie de página, sin embargo, admiten el tratamiento directo con dificultad. Sabemos que entre las notas y el cuerpo textual hay varias líneas de espaciado, a semejanza de la composición del texto de origen. Desafortunadamente, esto no se da siempre ni todas las veces que lo hace indica la separación entre ambas entidades. Con la numeración de las notas sucede lo mismo: da una pista, pero no es ni necesaria ni suficiente. La segunda línea y subsiguientes de una nota de más de un renglón no van precedidas de un guarismo, como tampoco lo van la continuación en la siguiente página de una nota extensa. Por otra parte, cuando alternan varios personajes genéricos del mismo tipo, estos se nombran a veces simplemente mediante sus ordinales, como *1.º*, *2.º*, etc., por lo que, aparte de la numeración a principio de línea, se requieren otros criterios. Un tercer indicio es la longitud de la línea, si bien aquí también encontramos tanto parlamentos en prosa de líneas largas como notas tan cortas como un verso de arte menor. La combinación de estos criterios no es una garantía lógica de los aciertos, si bien, en la práctica, es suficiente. Así, buscaremos líneas largas antes del final de página que estén separadas por varios espacios del cuerpo principal y líneas largas que comiencen por un número y las siguientes hasta el final de la página.

El cuerpo del texto no es uniforme. Por una parte, las líneas y las didascalias de personaje suelen presentar pocas dificultades porque el sangrado revela la función y

este se conserva al convertir entre formatos. Hay, empero, dos elementos textuales que requieren una inspección visual detenida: las acotaciones[93] y los versos compartidos. Las acotaciones, salvo que vayan centradas en la página, no se distinguen de otros componentes textuales porque el texto plano no tiene la capacidad de aplicar formato a los caracteres. Por lo tanto, los fragmentos en cursiva del archivo PDF de origen se traducirían en redonda en el de texto plano. Esto hace inevitable el marcado manual. Las continuaciones de versos compartidos, por su parte, se ven o no dependiendo de si el espaciado original es suficiente como para que la herramienta de conversión lo interprete como tal. Un espaciado insuficiente implica que en el archivo de texto no habrá distinción en el sangrado de un verso normal y la continuación de un verso compartido. Si bien suelen encontrarse fallos en la segunda parte del verso, a partir de la tercera, al llevar dos niveles de sangrado, resulta más infrecuente. En cualquier caso, conviene hacer un examen manual para marcar estos versos. El proceso consiste en un primer cotejo visual para buscar últimos versos de un parlamento y primeros del siguiente más cortos que los del bloque les rodea. Una vez localizados, se verifica que ambas líneas componen un solo verso y se ajusta el sangrado si procede.

Puede darse la situación de que un archivo PDF con una estructura interna compleja produzca unos resultados tan pobres s al convertirlo tratando de conservar la maquetación, que requiera tal cantidad de correcciones manuales que se dispare el tiempo necesario para preparar el texto. En algunos de estos casos, es posible usar una alternativa geométrica. Ahora emplearíamos otra herramienta para convertir a lenguaje de marcado, tal como `pdftohtml`, para traducir el archivo PDF a XML. El resultado no se trataría de un XML como en la jerarquía definida por TEI, sino que los elementos se caracterizarán por su posición espacial. Estos poseen atributos que indican mediante pares de coordenadas su ubicación física en la página. Con esto en mente, estimamos los rangos en los que comienzan los elementos horizontalmente en cada página. Es importante porque, además de variar la maquetación en general entre páginas pares e impares, hay elementos que no aparecen en todas las páginas. Así, los fragmentos centrados incrementarían el rango, cuyo punto de inicio sería la mitad de cuerpo de texto para un solo carácter. Lo mismo sucede con las continuaciones de versos compartidos, cuya posición inicial depende de la longitud de la parte precedente del verso en la línea o líneas anteriores. Por el desplazamiento vertical sabemos a qué línea corresponden los segmentos. Sea como fuere, ya es factible aproximar con cierta exactitud la distribución de los elementos y pasar el XML a texto. A pesar de ser un método más engorroso que el anterior, presenta al menos una ventaja

[93] Trataremos la didascalia de personaje como un elemento distinto al resto de acotaciones, pues estas tendrán entidad propia, mientras que aquella proporcionará un atributo.

sobre aquel, en tanto que se conservan los atributos de los caracteres, con lo que se encuentran las acotaciones con más facilidad. No resulta esta tampoco una solución infalible porque, por ejemplo, una expresión latina intercalada a comienzo de verso sería indistinguible de una acotación interna como *Canta* si esta última no va entre paréntesis.

La carga de temporal de este tipo de conversiones es variable, pero el coste por obra (C) es inversamente proporcional al número total de obras (n), en tanto que se requiere invertir una cantidad notable de tiempo (T_0) para establecer reglas que evalúen los patrones de la primera y aplicarlas a cada una de las siguientes (i) de manera mecánica, lo que toma una pequeña cantidad de tiempo (t) en comparación con el necesario para encontrarlas.

$$C = \frac{T_0 + \sum_{i=0}^{n} t_i}{n}, t_i \ll T \tag{6.1}$$

En términos llanos, esto es que, si disponemos de varios archivos con la misma estructura interna, solo tenemos que encontrar las reglas en uno de ellos y, una vez con ellas, podemos aplicarlas en bloque al resto de textos. Por lo tanto, cuantos más fuentes de la misma procedencia obren en nuestro poder, más eficiente en tiempo será convertir cada uno de ellas. Por el contrario, el peor escenario en términos de efectividad sería tratar una obra suelta. En el caso del corpus de *Sound and Meaning*, casi un centenar de archivos PDF procedían de las digitalizaciones del GRISO (2020) de autos calderonianos. Todos ellos respondían al mismo modelo y pudieron ser tratados mediante la conversión directa a texto plano. Esto hizo posible crear un corpus de varios cientos de obras a pesar de partir de PDF.

6.8 Escenario regular: archivos ofimáticos

Con archivos ofimáticos nos referimos a aquellos formatos producidos por el programa de edición de textos de un paquete para trabajos de oficina, como Write de LibreOffice.org o Word de Microsoft. Nos encontramos con estos archivos sobre todo en casos en los que el propio editor de la obra se la ha facilitado al proyecto *Sound and Meaning*. La manera de codificar el texto con formato varía sustancialmente; oscila entre estándares abiertos como ODF, que internamente usan archivos XML comprimidos, y cerrados diseñados específicamente para mantener un mercado cautivo, como es el caso del formato DOC de Microsoft y su plétora de versiones y revisiones apenas parcialmente compatibles, incluso entre sí. En cualquier caso, una vez conseguimos

abrirlos mediante el programa correspondiente, trabajaremos con ellos sin complicaciones, ya que no requerimos recurrir a las extensiones propias más esotéricas, sino simplemente algunas funciones elementales para manipular texto con formato. En concreto, tienen importancia la alineación del párrafo, las sangrías y algunas propiedades de los caracteres (cursiva y versalitas, sobre todo). El objetivo es convertir a texto plano marcando de alguna manera esas propiedades para poder conservarlas.

Los elementos que afectan al párrafo y la línea se marcan atendiendo a su espaciado. No es así con las propiedades de los caracteres, ya que estos pueden ir intercalados entre otros sin formato adicional. De este modo, marcaremos estos con una etiqueta reconocible, esto es, una serie de caracteres que no se vayan a dar en el texto. Si bien esto añade artefactos al formato, reconocerlos y tratarlos como tales es trivial, por lo que las ventajas superan con creces los inconvenientes, que son de orden estético en cualquier caso.

Para este trabajo, nos hemos servido del paquete LibreOffice.org porque da la posibilidad de buscar y reemplazar mediante la combinación de expresiones regulares estandarizadas y elementos de formato, por lo que los pasos que lo requieren pueden darse fácilmente. La extensión *Alternative Find & Replace for Writer* hace posible otras sustituciones. Las alternativas existentes, como Microsoft Office, ofrecen herramientas con una funcionalidad restringida en comparación. Así pues, optamos por la alternativa libre.

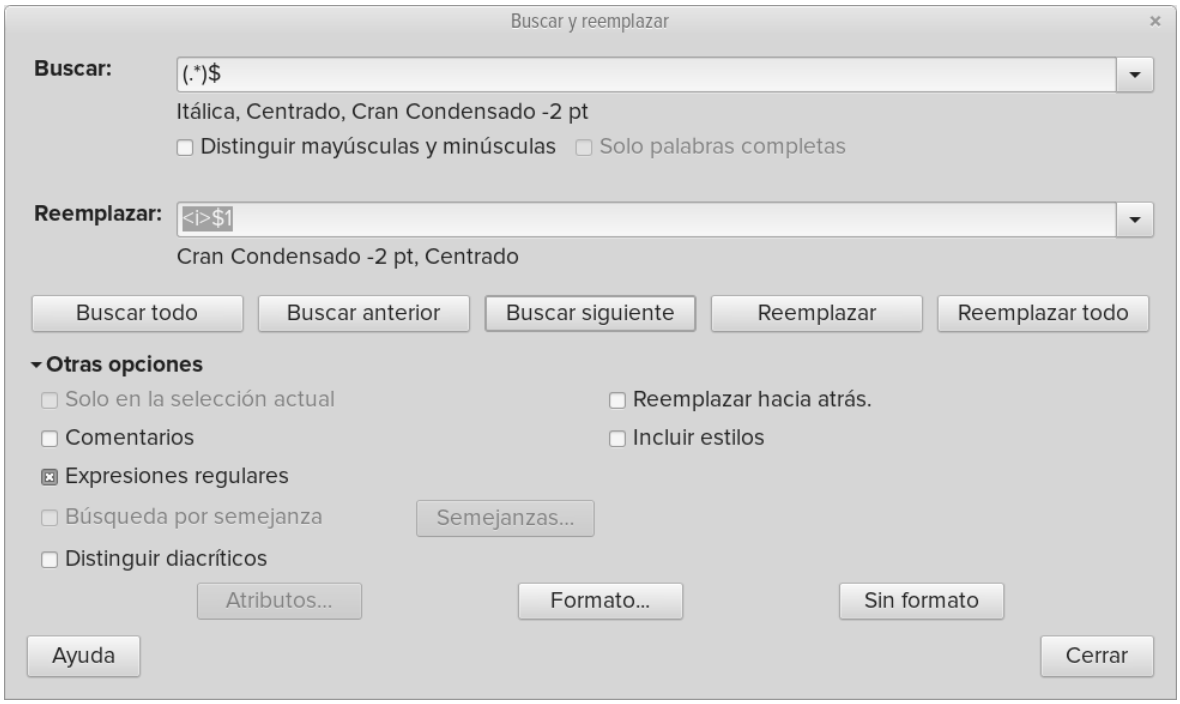


Figura 6.5: Diálogo de búsqueda de LibreOffice.org.

Esta opción haría hipotéticamente posible salvar el texto en formato ODT y escribir un pequeño programa que llevara a cabo la preparación procesando el código XML después de descomprimirlo. Al ser pocos los archivos en este formato, resulta más práctico tratarlos de forma manual. El inconveniente es que esto obliga a abordarlos de uno en uno, por lo que, de haber una cantidad grande de estos, convendría considerar procesar automáticamente el código XML. De cualquier manera, conviene restringir las sustituciones desde el procesador de textos ofimático a aquellas que no puedan resolverse sencillamente en el archivo de texto con la línea de órdenes del sistema operativo, ya que el tratamiento individual de los archivos acarrea un alto coste temporal.

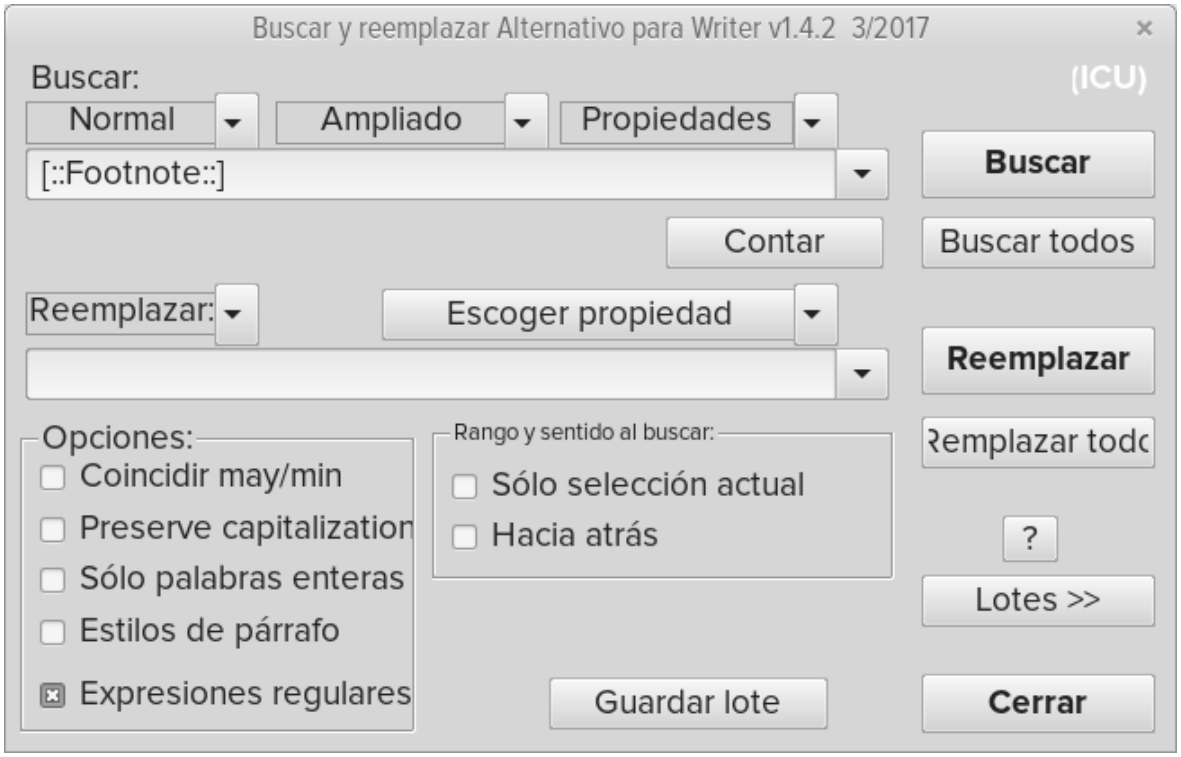


Figura 6.6: Extensión de búsqueda *Alternative Find & Replace for Writer*.

Para las sustituciones en el cuerpo del texto, se selecciona `Buscar y reemplazar`, que se encuentra en el menú `Editar`. Esto abre un diálogo con diferentes opciones para nuestro propósito (Figura 6.5). Aquí debe marcarse la casilla correspondiente para habilitar el uso de expresiones regulares, y empezar a hacer las sustituciones requeridas. La expresión de la ilustración, por ejemplo, buscaría y capturaría (paréntesis) cualquier cadena de texto (`.*`) hasta el final de la línea ($), siempre que la línea esté centrada y el texto en cursiva. Sustituiría el texto por el patrón capturado (todo hasta el final de la línea), pero, por ejemplo, con la cadena `<i>` antepuesta[94], de modo que

[94]La elección no es casual, sino que se aviene a las etiquetas del formato VED, que veremos más en detalle algunas páginas más adelante.

se identificaría esta línea una vez perdida la alineación y las propiedades del tipo de letra. Al pulsar `Reemplazar todo`, se aplicaría al texto completo. Sabemos también que los versos están alineados a la izquierda, mientras que el texto en prosa suele ir justificado a ambos lados de la página. Esto se aplica para buscar esta característica y marcar los pasajes necesarios.

Por su parte, para eliminar bloques paratextuales, también en el menú `Editar` seleccionaremos el elemento `Alt. buscar y reemplazer`[95] [*sic*]. Esto abre otro diálogo con opciones parecidas a la búsqueda estándar, aunque algo diferente (Figura 6.6). Aquí también marcaremos la casilla que activa el uso de expresiones regulares. En este caso, no buscaremos cadenas de texto, sino bloques (notas a pie de página, al final, etc.). Dejaremos vacío el campo con la sustitución, pues así indicaremos que el bloque buscado ha de reemplazarse por un elemento vacío. De esta manera, la expresión del ejemplo, eliminaría todo el cuerpo de notas a pie de página.

Los diferentes niveles de sangrado se indican mediante tabuladores, tantos como niveles haya. El espaciado interior también es de utilidad, ya que nos valdremos de él para diferenciar entre unidades estructurales. Una separación de un solo espacio divide componentes de una misma unidad, mientras que un número mayor de espacios vacíos indica una separación de otra índole. De esta manera, distinguiremos entre didascalia de personaje y parlamento con independencia de las propiedades del texto. El ejemplo 117 capturaría en un primer grupo la didascalia de personaje y en otro el parlamento, de manera que podríamos alterarlos de la manera más conveniente.

(117) `^\s*(\S+(?:\s\S+)*)(\s{2,})(\S+)$`

Podríamos modificar la expresión regular sustituyendo `^\s*`, que denota la presencia de cero o más (`*`) espacios (`\s`), por un rango de entre i y j espacios, por ejemplo con `^\s{`i`,`j`}`, de manera que encontraríamos la presencia de versos compartidos. Así, adaptaríamos estas expresiones a los casos particulares y, de igual forma, abordaríamos el archivo de texto resultante.

6.9 Escenario óptimo: XML-TEI

El mejor caso posible es la conversión desde un archivo XML-TEI. Este presenta todos los elementos necesarios para producir una forma estructurada y sin ambigüedades. El esquema de Figura 6.7 muestra una descripción de la estructura básica de una obra de teatro, como las que se encuentran, por ejemplo, en DraCor. De la raíz `<TEI>`

[95] La errata está en el menú de la barra de tareas.

cuelgan dos ramas. La primera es `<TeiHeader>` es una cabecera de metadatos. La segunda es `<text>`, con el texto propiamente dicho, como se observa en el esquema de la jerarquía XML-TEI (Figura 6.7). El estándar TEI define un detallado conjunto de etiquetas, cuyas directrices se extienden a lo largo de casi dos mil páginas en su última versión (TEI Consortium, 2022), por lo que aquí solo nos ocuparemos sucintamente de aquello que afecta directamente al análisis y, aun así, de manera muy superficial.

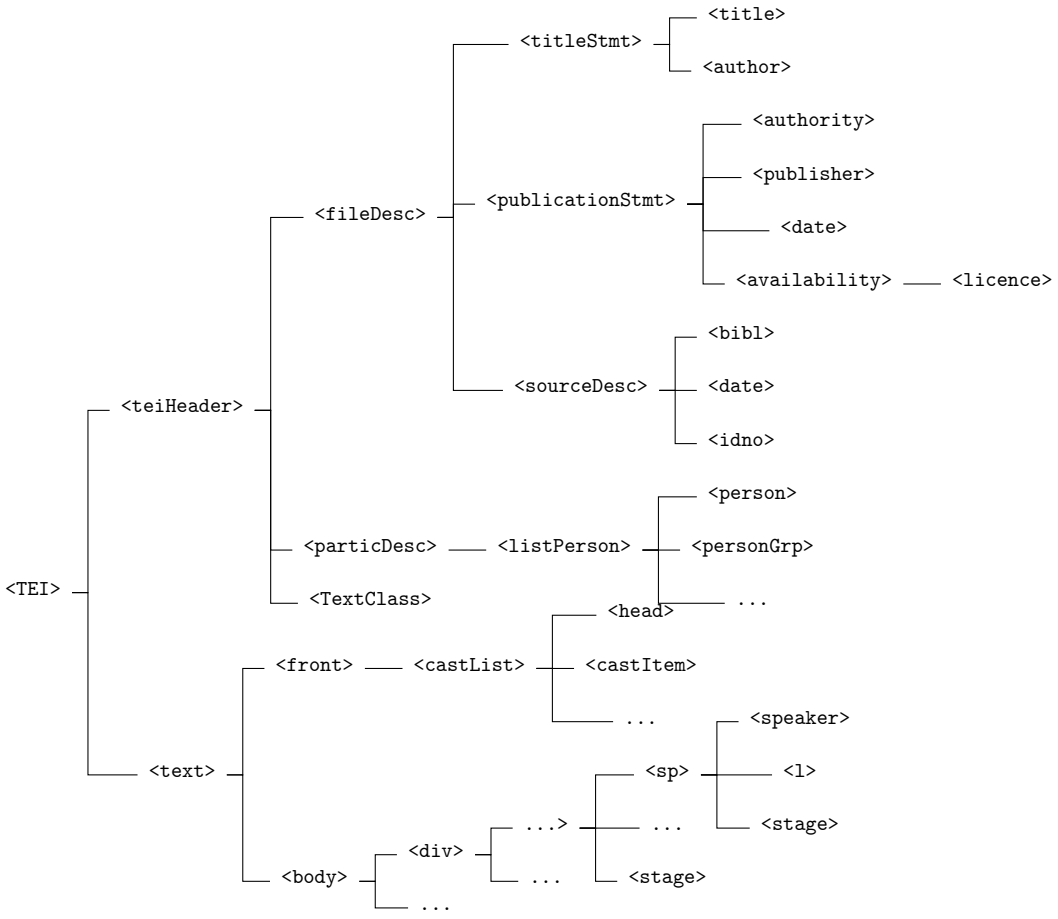


Figura 6.7: Estructura de XML-TEI.

De la cabecera cuelgan otras subestructuras. En `<fileDesc>` hay información sobre la edición. Tal vez la más relevante de estas ramas sea `<titleStmt>`, pues incluye el título y el autor. Aquí se usan varias etiquetas, añadiendo argumentos si es necesario, como, por ejemplo, `<title type"main">` y `<title type"subtitle">`, para reflejar el título principal y el subtítulo, o crear una subsestructura en `<author>` especificando detalles como nombre o apellidos. Los datos de la publicación cuelgan de `<publicationStmt>`, como la organización responsable de la edición, en `<authority>`, la editorial en `<publisher>`, la fecha en `<date>` y la disponibilidad `<availibility>`, donde caben etiquetas como `<licence>` para expresar los términos legales de uso

del texto (118). Los datos de la obra original están bajo `<sourceDesc>`, con diversos identificadores, así como la fecha.

(118)
```
<publicationStmt>
    <authority>University of Vienna,Institute of Romance Languages and
     Literatures</authority>
    <publisher xml:id=""/>
    <date when="2022-01-12">
        <time when="14:08:57"/>
    </date>
    <idno type="quadramaX">La siembra del Señor (Los obreros del
     señor)</idno>
    <availability>
        <licence>
            <ab>CC-BY 4.0</ab>
            <ref target="https://creativecommons.org/licenses/by/4.0/">
                Licence
            </ref>
        </licence>
    </availability>
</publicationStmt>
```

Resulta de la mayor importancia la rama `<particDesc>`, ya que en ella se configura la lista de personajes que intervienen en la obra. Esta lista consta de una entrada única de tipo `<person>` para cada personaje o, en el caso de personajes colectivos, de tipo `<personGrp>`, como ilustra el ejemplo 119. Asimismo, cada personaje se caracteriza más en detalle mediante atributos de la etiqueta, entre los cuales consideraremos dos. Tenemos, por ejemplo, `sex`, que se emplea para denotar si el personaje es masculino, femenino o lo que convenga. Otros atributos similares se emplean de manera análoga. Tanto o posiblemente más relevante para el tratamiento digital que los atributos de clase, es el identificador, pues asigna a cada uno de ellos un código individual que se mantiene a lo largo de la obra. Esto simplifica seguir a cada personaje, con independencia de que se referencia a un mismo locutor con diferentes didascalias, como abreviaturas, títulos de cortesía o, incluso, que un personaje cambie de nombre a mitad de la obra. Para ello, además de la didascalia bajo la etiqueta `<speaker>`, disponemos de un indicador paratextual `xml:id` único para cada personaje, siempre el mismo con independencia de la didascalia con que se nombre. De esta forma, una vez que hemos definido `xml:id="scipion"`, cada uno de los parlamentos del personaje llevará ese identificador, escríbase la didascalia Scipión o Scip.. De esta manera, los versos bajo ambas se reconocen de forma sencilla como pertenecientes a un mismo personaje *Scipión*.

(119)
```
<person sex="MALE" xml:id="scipion">
    <persName>SCIPIÓN</persName>
</person>
<personGrp sex="UNKNOWN" xml:id="musicos">
    <persName>MÚSICOS</persName>
</personGrp>
```

Pueden existir más ramas de este árbol, algunas de las cuales resultan extremadamente útiles para facilitar la clasificación de la obra, como, por ejemplo, `<textClass>` (120).

(120)
```
<textClass>
        <keywords>
                </person>
                <term type="genreTitle">Comedia</term>
                <term type="subgenreTitle">De capa y espada</term>
        </keywords>
</textClass>
```

Respecto a los datos de la cabecera, cuantos más se encuentren a nuestra disposición, tantas más serán las posibilidades en lo concerniente al análisis del texto y en lo que atañe a cuestiones de archivo. Resulta evidente que la inclusión de la fecha inicial de la publicación, datos de los personajes o del género de la obra permite cruzar esa información para buscar patrones. Pero no menos importantes son otras cuestiones, como la licencia de uso, que condicionará la manera de emplear los datos, o los identificadores de la obra y el autor, que facilitan la búsqueda y clasificación de esta. Las ediciones usadas proporcionan toda la información requerida, pero la cuestión podría cobrar importancia con fuentes menos minuciosas.

La rama `<text>` contiene el texto propiamente dicho, con dos partes denominadas `<front>` y `<body>`. Esta rama tiene la particularidad de que los contenidos de las etiquetas se muestran en el texto, por lo que los metadatos han de señalarse como atributos de la etiqueta. Bajo `<front>`, del ingleś *front matter*, tendríamos, en principio, los preliminares. Sin embargo, en nuestro caso, aquí se incluye también el elenco bajo una estructura `<castList>` que contiene elementos del tipo `<castItem>` y `<head>`, este último para una cabecera del tipo «Personajes que hablan» o «*Dramatis personae*».

La rama `<body>` se divide en partes bajo la etiqueta `<div>`. Aquí es habitual especificar el tipo y la numeración, cuando es necesario, para poder encontrar ramas bajo la

etiqueta, como en `<div type="act" n="1">`. Bajo estas, puede haber asimismo subdivisiones, en la que, en lugar de `type="act"`, aparece, por ejemplo, `type="scene"` y, por lo general, especificando también la numeración. Finalmente, en la rama `<div>` más profunda, se encuentra el texto de la obra.

Este texto[96] tiene dos tipos principales de etiqueta, `<stage>` y `<sp>` (121). La primera señala las acotaciones y la segunda los parlamentos. En estos últimos, además de la parte correspondiente al texto, debe tenerse muy en cuenta el atributo `who` de la etiqueta, que es el identificador único que se había definido en la cabecera y que permite, por ejemplo, encontrar redes de relaciones entre personajes. Es preferible usar este identificador en lugar de la etiqueta `<SPEAKER>`, ya que esta se refiere a la didascalia y no al locutor, por lo que, como indicamos, pude variar, lo que dificultaría su seguimiento. La última etiqueta que nos interesa es `<l>`, que marca cada línea.

Además del atributo `n` que, como los elementos anteriores, denota la numeración, puede aparecer el atributo `part`. Este sirve para indicar el comienzo de un verso (`"I"`), su continuación (`"M"`) y su término (`"F"`), para identificar versos distribuidos a lo largo de dos o más líneas. Podría darse el caso de que, en lugar de la etiqueta `<l>`, halláramos otra del tipo `<p>`. Esta denotaría que no encierra una línea de parlamento, sino un párrafo y, por lo tanto, prosa. Se emplea, por ejemplo, cuando un personaje lee una carta. Tampoco es raro que, inmediatamente bajo `<sp>`, no se abra un bloque mediante una etiquetas de tipo `<l>`, sino que sea una estructura intermedia `<lg>` y, por debajo de esta, se encontrarían las líneas. Por ejemplo, `<lg type="stanza">`. Las acotaciones internas se representan mediante etiquetas `<stage>` dentro de una estructura `<sp>`, en lugar de a su mismo nivel como las externas.

(121)
```
<sp who="#morlaco">
    <speaker>MORLACO</speaker>
    <l met= "+-++-++--+-" rhyme="ina">¿Yo beldad? Loco me es sobre
     gallina</l>
    <l part="I">este príncipe.</l>
</sp>
<sp who="#siroes">
    <speaker>SIROES</speaker>
    <l part="M">¿Dónde...</l>
</sp>
<sp who="#morlaco">
    <speaker>MORLACO</speaker>
    <l part="F" met="+-+--+-+++-" rhyme="ada">No sé nada</l>
```

[96] Se trata de una codificación propia a partir de una edición todavía inédita de Ignacio Arellano del auto calderoniano *La exaltación de la Cruz*.

```
        <l n="1665" met="+--+++-+-+-" rhyme="ada">y hágase allá, que es
         burla muy pesada.</l>
    </sp>
```

Cabe la posibilidad de incluir atributos `met` para el patrón rítmico y `rhyme` para la rima. No resulta común encontrarlo en textos españoles y, de hecho, no hemos hallado ninguna edición que lo incluya, aunque sea bastante habitual en lenguas clásicas, el griego sobre todo. Así que, si de bien poco sirve tener la posibilidad en el análisis careciendo de textos que hagan uso de ella, la recordaremos para aprovecharla más adelante y reflejar los resultados de nuestros análisis.

Conociendo la estructura, la extracción de datos resulta trivial: recorremos recursivamente las subestructuras de XML-TEI hasta llegar al nivel más profundo de cada rama, leemos los datos que en ella se encuentran y los tratamos de la manera que resulte más conveniente. Por ejemplo, para tratar los parlamentos procederemos como en Algoritmo 1.

Algoritmo 1: Conversión de XML-TEI a texto.

```
para cada acto ∈ actos hacer
    para cada escena ∈ acto hacer
        para cada estructura ∈ escena hacer
            si estructura es parlamento entonces
                para cada línea ∈ estructura hacer
                    si (¬ part ∨ part = ⟨F⟩) entonces
                        GuardaVerso(anteriores + línea)
                        anteriores ←∅
                    si no
                        anteriores ←anteriores + línea
            si no
                GuardaAcotación(estructura)
```

El algoritmo consiste en recorrer los actos, y en cada uno de ellos —supondremos que no hay escenas ni otros niveles intermedios—, recorremos a su vez las estructuras inferiores, como los parlamentos y acotaciones. Aquí se bifurca el flujo de datos según el tipo de estructura. Si es un parlamento, lo recorremos línea a línea. Primero probamos si la línea se corresponde a un verso, comprobando que la línea no lleva atributo de parte o es la sección terminal de un verso. En este caso, procesaremos las partes anteriores del verso —que será un elemento nulo de no ser un verso compartido—, seguidas de esta línea y reiniciamos la variable intermedia donde guardamos los elementos anteriores. Si la línea no es terminal, vamos añadiendo las nuevas partes del verso a la variable intermedia. Si, por el contrario, la estructura no es un parlamento, la procesaremos como acotación. De manera análoga, haremos lo propio con todos

los elementos de la cabecera, aunque, en ese caso, los terminales serán normalmente de un solo tipo —con excepciones como la descripción de los personajes— y no hará falta recorrer en un bucle, sino buscar determinados elementos únicos.

6.10 Precauciones con la codificación

Hasta aquí hemos visto las estructuras de datos como meras abstracciones. Sin embargo, a la hora de trabajar en la práctica con ellas, debemos observar ciertas precauciones de índole más técnica que conceptual, ya que estamos sujetos a la representación interna de los caracteres del texto empleada por el computador. Incluso cuando el procedimiento empleado por todos los archivos de texto es muy similar, hay algunas diferencias en la codificación que son susceptibles de causar dificultades y errores. En nuestro trabajo, esto redunda tanto en la organización de los textos como en la manera de tratar las fuentes. Para entender el problema nos remontaremos a los primeros estándares de codificación informática de texto.

En las computadoras personales típicas, el máximo común divisor es *American Standard Code for Information Interchange* básico, más conocido por sus siglas ASCII (Wells, 1997). Este estándar surgió a principios de la década de 1960 a expensas de la American Standard Association. Se diseñó con el objetivo de representar textos escritos en inglés y, más específicamente, de la variedad americana de esta lengua. Por este motivo, no era capaz de mostrar caracteres ajenos a ese alfabeto, como los que emplean muchas lenguas europeas, el español entre ellas. La cantidad de caracteres no alfanuméricos también estaba limitada y, por ejemplo, no preveía el uso de símbolos monetarios distintos del dólar estadounidense. Como es de suponer, otros sistemas de escrituras no basados en el alfabeto latino quedaban totalmente excluidos. Este ASCII se representa mediante 7 bits, por lo que hay un máximo de 2^7 combinaciones, que se reparten entre 95 *caracteres* imprimibles y 33 de control[97] en su revisión más moderna (MacKenzie, 1980, pp. 423-433).

Esta codificación es precisamente la que usa el *Speech Assessment Methods Phonetic Alphabet* (SAMPA). Se trata de un alfabeto que se desarrolló en el marco del proyecto European Strategic Programme on Research in Information Technology (ESPRIT) de la Comunidad Económica Europea para representar en soporte informático la fonología de lenguas europeas occidentales. Una de las premisas de su diseño era que

[97]Un carácter no equivale necesariamente a un grafema, sino a un patrón de bits al que le corresponde un significado. Este admite diferentes formas: puede ser gráfico —una letra, un signo, un símbolo o un espacio en blanco—, en cuyo caso hablaremos de caracteres imprimibles, pero también puede representar una acción no visible —borrar el carácter anterior o marcar el final de una línea—, y hablaremos de caracteres de control (Haralambous y Dürst, 2019, p. 139).

debía garantizar la compatibilidad con las computadoras de la época. De acuerdo con esto, podríamos valernos de ella para representar las transcripciones fonológicas. Sin embargo, esto presentaría problemas para los fragmentos grafémicos.

Debido a las limitaciones de ASCII, diversos fabricantes y organizaciones concibieron sus propios métodos de codificar los caracteres extendidos, entre los que se encuentran las letras con diacrítico y muchos de los signos de puntuación que necesitamos para escribir en español. Así, encontramos que la ISO propuso su estándar ISO-8859-1 y Microsoft desarrolló Windows-1252 para las lenguas europeas. Ambos son de 1 byte, lo que son suficientes para representar 2^8 combinaciones o, lo que es lo mismo, 256 caracteres.

Hoy en día, hemos superado aquellas constricciones impuestas por la técnica, podemos representar en texto plano no solo todas las grafías que demanda una correcta ortotipografía en cualquier lengua europea, sino muchos más caracteres. El desarrollo de las computadoras personales hizo irrelevantes las limitaciones que prevenían la aparición de nuevos sistemas de codificación. *Unicode* se presentó en 1991 con la idea de ofrecer una solución a las constricciones de las codificaciones existentes y, especialmente, para dar soporte a todas las lenguas vivas y muertas, así como otros sistemas de símbolos cuya inclusión estuviera justificada. Cada carácter Unicode tiene el mismo número de bits, por lo que todos ellos se codifican como un único carácter para cada grafema, tanto del español como del AFI. Por defecto, Unicode emplea dos bytes para representar un solo carácter, esto es, dieciséis bits. Esto permite usar un total de 2^{16} o, lo que es lo mismo, 65.536 caracteres (Bettels y Bishop, 1993). Los sistemas operativos modernos de tipo Unix más extendidos —Linux y MacOS— soportan Unicode de forma nativa en prácticamente todas sus versiones posteriores al año 2000. De esta manera, no resulta ya necesario emplear SAMPA, pues pueden representarse con la misma codificación tanto los textos grafémicos españoles como su transcripción al AFI.

Esto no solo simplifica la comprensión del texto, ya que la notación de los programas se corresponde con la impresa, sino que tiene implicaciones en el análisis informático de los textos transcritos. En concreto, la limitación del número de caracteres disponibles obliga a que SAMPA represente dígrafos y ligaturas mediante dos caracteres, mientras que Unicode los representa con uno. Por ejemplo, SAMPA obligaría a preprocesar el fonema /t͡ʃ/ al representarlo con los caracteres ASCII *t* y *S*, como *tS*. Unicode, por su parte, aúna los símbolos del AFI *t* y *ʃ* mediante un único carácter digrafémico *ʧ*. De esta manera, se vuelve innecesario crear una excepción para tratar este sonido africado, al poder representarse como una única unidad.

Aunque el formato final de nuestros datos está codificado según Unicode, esto no implica necesariamente que los archivos de origen también opten por el mismo sistema. Por lo tanto, hay que convertir aquellos archivos de texto que partan de codificaciones distintas. En el caso de ASCII, no sería necesario porque es un subconjunto de Unicode; desafortunadamente, como ya dijimos, es un subconjunto que no incluye todos los diacríticos y signos ortográficos necesarios para escribir en español de acuerdo con su ortografía, por lo que ningún texto aceptado va a estar en ASCII puro. Las otras aproximaciones de un byte a la representación de caracteres extendidos provocan serios problemas, pues producen artefactos que alteran el texto (Figura 6.8). Si bien los signos ortográficos no afectarían a nuestro propósito, los diacríticos supondrían un problema grave. Por lo tanto, se hace necesario unificar formatos en algún momento y, por razones prácticas, lo haremos cuanto antes.

```
¡Oíd, mortales, oíd,                    Â¡OÃ d, mortales, oÃ d,
y al pregón de la Fama                  y al pregÃ³n de la Fama
todos acudid!                           todos acudid!
```

Figura 6.8: Transliteración de la variante ASCII ISO-8859-15 a Unicode.

La solución pasa por identificar de antemano el tipo de codificación y convertir el archivo a Unicode. Esto no presenta mayor problema en un entorno Unix, cuyas herramientas proveen soluciones simples. Mediante el mandato `file` identificamos la codificación del archivo y, utilizando un programa al efecto[98], solventamos el inconveniente.

A pesar de que Unicode es la opción que mejor se adapta a nuestro propósito, tampoco está exenta por completo de algunas fuentes potenciales de complicaciones. Una de ellas, que debemos señalar para poder evitarla, deriva del propio diseño de la codificación. Aunque esta se adecúa a las ortografías de las lenguas más conocidas y es capaz de representar un gran número de caracteres, soluciona las combinaciones de dos maneras diferentes, indistinguibles para el ojo, pero no precisamente triviales de tratar digitalmente de manera elegante. Para representar un carácter compuesto, pongamos por ejemplo la letra eñe minúscula, se usa la codificación Unicode simple LATIN SMALL LETTER N WITH TILDE. Esta es el carácter U+00F1 para el computador y, para nosotros, el grafema ⟨ñ⟩. También existe la posibilidad de representarlo mediante la combinación de los caracteres LATIN SMALL LETTER N ⟨n⟩ y COMBINING TILDE ⟨◌̃⟩. En este último caso, aunque nosotros veríamos igualmente el grafema ⟨ñ⟩ en la pantalla, el computador leería la secuencia U+006E U+0303: esto es, dos

[98] Por poner dos ejemplos de entre los empleados, la biblioteca GNU C de Linux provee `iconv` y MacOS dispone de `textutil` para la tarea.

caracteres. Como dijimos, hasta aquí no sería problema para atenerse a la ortografía española. Desafortunadamente, no puede decirse lo mismo de otros sistemas de representación.

Concretamente, las vocales no silábicas, marcadas con un símbolo diacrítico de breve invertida inferior ⟨◌̯⟩, solo admiten representación en Unicode mediante una combinación, de forma que hemos de tratarlo como dos caracteres (Moran y Cysouw, 2018, p. 17). Por esta razón, ante la dicotomía de introducir excepciones en el código por una cuestión técnica visual o mantener la fidelidad a los principios rectores, optamos por lo segundo, dando primacía a la claridad conceptual del proceso sobre la estética del resultado[99]. De este modo, representaremos las semivocales /i̯/ y /u̯/ con el mismo símbolo que las semiconsonantes /j/ y /w/, respectivamente, y usaremos el carácter individual con diacrítico de breve integrado (⟨ă⟩, ⟨ĕ⟩ y ⟨ŏ⟩) para el resto de vocoides.

La codificación de los símbolos que señalan acentos primarios o secundarios en las sílabas (ˈ, ˌ) no se ven afectados gracias a que, al contrario que los modificadores diacríticos, estos anteceden a la sílaba que modifican. Por consiguiente, cuando aparece uno, se tiene en cuenta como modificador, pero se ignora para lo demás *a priori*.

Otro problema que se plantea son las distintas formas en las que los diferentes sistemas representan el final de línea (LF) y el retorno de carro (CR). Las máquinas con un sistema operativo de tipo Unix —GNU/Linux y MacOS, por ejemplo—, emplean únicamente LF. Por su parte, los computadores Windows usan CR LF, mientras que sistemas Apple antiguos emplean CR. Esto supone para nosotros que, cuando abrimos un texto con saltos de línea para Windows, veremos al final de cada fila un artefacto correspondiente al carácter CR, sin uso en nuestro sistema. En un texto creado en una máquina Apple antigua, lo que aparecerá será tan solo una larga línea con artefactos dispuestos a intervalos, ya que no tenemos el carácter LF que indica el final de la línea, pero sí CR, para el que no encontramos uso. Por esta razón, es necesario convertir aquellos textos que así lo requieran. De nuevo, usaremos la herramienta estándar de Unix `file` para identificar el tipo de final de línea que tenemos y programas específicos para convertir las marcas de un sistema a otro[100].

Una vez podemos manipular las fuentes para tratar el texto con la asistencia del computador, nos hallamos en condiciones de empezar a identificar y clasificar las entidades que requerimos para caracterizar la obra teatral. En realidad, como veremos en el capítulo siguiente, la cuestión no es tanto encontrar estos elementos, pues ya hemos visto que la codificación XML-TEI ha resuelto este reto y proporciona un buen

[99] Dejamos abierto para futuras versiones de la biblioteca modificar esto.

[100] Bajo Linux, `mac2unix` o `dos2unix` o `textutil` en MacOS. Se puede hacer usando tan solo las herramientas estándar de Unix, como `sed` o `awk`.

modelo, sino seleccionar aquellos componentes que necesitemos, así como encontrar una disposición que facilite los análisis cuantitativos.

Quédese vusté con Dios,
ya no salgo a la comedia
y ya me voy a mi casa,
porque no quiere el poeta
que le haga estorbo el gracioso
cuando hay un paso de veras.

Rojas Zorrilla, *Peligrar en los remedios*

Estructura digital del texto dramático 7

7.1 Texto dramático, datos dramétricos

En este capítulo retomamos lo visto sobre los datos y las diferentes formas de usarlos para ser aplicados a las entidades dramáticas. El objetivo ahora es definir estas como estructuras de datos, sea como elementos singulares o mediante las relaciones entre ellos. Damos por supuesto que, tras seguir lo expuesto en el capítulo anterior, disponemos ya de textos *limpios* y preparados para trabajar digitalmente de modo automático.

Si bien el formato XML-TEI se presta perfectamente a almacenar información y da facilidades para su manipulación, su estructura jerárquica requiere un paso intermedio para el tratamiento estadístico. Los paquetes de mayor difusión en la academia y la industria trabajan con marcos de datos[101], que no son otra cosa que una vía de abstracción para representar la sempiterna tabla. Esto es, una serie de elementos distribuidos a lo largo de filas, cuyas características se definen en columnas. De esta manera, si queremos conocer la característica j del elemento i, nos desplazaremos verticalmente hasta la fila i-ésima y, en ella, miraremos el contenido de la celda correspondiente a la columna j-ésima.

Esto naturalmente requiere conocer en primer lugar qué características cuantificaremos para definir el formato de la tabla. También hemos de saber cómo se representarán esas características para asignarles un valor numérico o categórico, de lo que depende el formato del valor de la celda. Asimismo, necesitamos conocer qué

[101] *Data frame* en inglés.

elementos poseedores de tales características observaremos. De esta manera, nos interesa definir las entidades que constituyen la forma externa del texto dramático para poder estructurarlas después en forma tabular.

Como dijimos, estas entidades y sus relaciones estaban indicadas explícitamente en los archivos en XML-TEI. Lamentablemente, no todas las fuentes se encuentran disponibles en este formato ni todas lo que lo están son fiables (Valdés Gázquez, 2104a). Podríamos codificarlos nosotros mismos —de hecho lo haremos más adelante—, pero estaríamos en la misma tesitura porque, para hacerlo, debemos identificar primero los elementos textuales y sus relaciones.

La solución por las que nos inclinamos pasa por un sistema de tratamiento de textos semimanual. Esto es, los textos de origen se limpian primero mediante los procesos semiautomatizados presentados en el capítulo anterior, si bien, en lugar de dejarlos tal cual, aprovechamos para que el proceso los organice según ciertas convenciones que nos resulten más convenientes. El formato nos ha de permitir identificar las entidades textuales sin ambigüedades para luego automatizar su tratamiento. Proponemos un sistema de codificación de textos dramáticos que hemos denominado Visually-Encoded Drama (VED) para alcanzar este propósito. Hemos desarrollado este formato específicamente para el proyecto *Sound and Meaning* como una aportación extra al limpiado de los textos. Esto es, surgió como una evolución de la preparación directa de los archivos, aquella consistente en eliminar información metatextual como encabezados, paginación o aparato crítico. Con unos cambios mínimos, que apenas requieren operaciones adicionales más que adaptar las ya usadas al nuevo objetivo, el texto resultante puede proporcionar información adicional sobre su estructura, así como metadatos relevantes.

7.2 De texto plano a VED

La manipulación de las fuentes plantea varias necesidades. Por una parte, hay que disponer de un formato de representación de las ontologías propias del drama. Por otro lado, crearlo a partir del formato de partida debe implicar la mínima intervención manual. Además, todo ello debería poderse llevar a cabo sin conocimientos técnicos. De ahí el carácter visual del formato que ideamos. Este reproduce la disposición de los elementos estructurales que se hallan en la página, de manera que, al menos los concernientes al texto, pueden codificarse sin necesidad de aprendizaje (7.1).

Listado 7.1: Ejemplo mínimo de codificación VED.

```
UNO
```

```
        Es el primer verso de Uno,
        y este es uno...
OTRO
                ...compartido.
        y este lo dice solo Otro.
```

Resulta evidente la similitud con el formato que encontramos en cualquier edición moderna, si bien la didascalia de personaje toma una línea para ella sola. Naturalmente, esta información es insuficiente, por lo que hemos de añadir otros datos de alguna manera. Nos valdremos para ello de etiquetas simples, pero siempre respetando la estructura visual del texto. Dada la flexibilidad que ofrece esta codificación, resulta sencillo construir herramientas para traducir este formato a otros, como XML-TEI, por lo que también las hemos construido para llevar a cabo esta tarea. Por ahora, empezaremos por describir el formato VED y, a continuación, explicaremos cómo llegamos hasta él.

El formato VED está concebido con la idea de que sea intuitivo. Por lo tanto, nada mejor para explicarlo que con un ejemplo. Vemos elementos que reproducen la forma arriba descrita, así como otros menos obvios, por estar precedidos de marcas de control. Aquí tenemos dos tipos: por un lado, etiquetas, que se asemejan a las usadas en HTML o XML, aunque, al contrario que esas codificaciones, no van por pares con una etiqueta para abrir y otra para cerrar, ya que no cumplen esa función. Al establecer una relación biunívoca entre entidad y línea, los límites de esta son los de aquella. La etiqueta, pues, solo se utiliza a efectos clasificatorios.

Listado 7.2: Ejemplo de archivo VED.

```
<x>Comentario
<a>Autor
<t>Título
<tt>Subtítulo
<g>Género
<gg>Subgénero
<o>Obtenido de la edición tal*https://foo.bar/fuente
<f>1600-1699
<el>PERSONAS QUE HABLAN EN ELLA|Uno*Otro, otro personaje*UN TERCERO
<j>Primera Jornada
<i>Acotación
UNO #uno
        El primer verso de Uno,
        segundo, de Uno, el verso.
        Tercero es medio.
OTRO #otro
                que
```

```
UNO #uno
                          sigue
        en este otro parlamento.
        <i>Acotación interna.
        Quinto verso, también fin.
UNO Y OTRO #uno #otro
        <e>¡Fin!
<i>Vanse. Acotación externa. Entra un tercero.
TERCERO #tercero
        <i>Lee.
        <p>Esto no es un verso sino un texto en prosa.
```

Como su nombre indica, el formato VED reproduce la estructura espacial de la pieza teatral. Se trata de un simple archivo en texto plano —por lo tanto, puede ser abierto y editado directamente en cualquier sistema sin necesidad de instalar programas adicionales—. Cada elemento textual se representa en una línea; estas líneas se configuran de manera interna de diferentes formas para marcar su función en el texto. Los parlamentos se distinguen de las didascalias de personaje mediante el sangrado, mientras que otras características del texto, así como el resto de acotaciones, elementos textuales dramáticos inusuales y metadatos se indican mediante un reducido conjunto de etiquetas. Esto resulta en una composición legible directamente, pero semiestructurada, que se procesa automáticamente con un programa específico. Por otra parte, al remedar la disposición sobre el papel, este sistema es muy intuitivo, por lo que apenas requiere unos minutos para poder empezar a preparar textos.

Al comienzo, vemos una serie de líneas sin sangrar y con una etiqueta que recuerda a las que se emplean en HTML. Su función es, sin embargo, un poco diferente, ya que no indica atributos sino categorías. La etiqueta `<x>` la usamos para añadir comentarios que no han de ser tenidos en cuenta en el análisis. Resulta útil cuando hay que introducir advertencias o aportar información adicional necesaria para el desarrollo del trabajo. El nombre del autor lo señalamos con la etiqueta `<a>`. Si estamos codificando la obra para procesarla e incorporarla a un corpus, sería importante convenir una forma estandarizada, de manera que varias obras del mismo autor no aparezcan con distintas variantes del nombre; el computador es incapaz de notar algo aparentemente obvio como que tanto `Calderón` como `Calderón de la Barca` y `Pedro Calderón de la Barca`, o incluso `Calderón de la Barca, Pedro`, son formas de nombrar a la misma persona. En la práctica, hemos optado por emplear versiones abreviadas del nombre en la codificación y recurrir a una tabla de conversión externa para sustituirla por el nombre completo. Aquí, confiaremos en el sentido común del lector.

Las etiquetas `<t>` y `<tt>` corresponden al título de la obra y a su subtítulo, respectivamente. La fecha la introducimos tras la etiqueta `<f>`. En el caso normal, indicamos la fecha mediante un año representado por cuatro guarismos arábigos. Si no conocemos la fecha concreta, optaremos entre usar un rango mediante dos años separados por un guion o una fecha insegura, que indicamos con un año y un signo de cierre de interrogación. Cualquier otro formato sería válido, pero, dado que no es posible contemplar todas las variantes posibles, al convertirlo a TEI produciría un formato de fecha no estandarizado. Asimismo, para dejar constancia de la procedencia editorial de la fuente, empleamos la etiqueta `<o>`. Aquí incluiremos cuantos datos estimemos convenientes y, de manera opcional, una dirección de una página web asociada introducida por un asterisco. Este campo no se encuentra estandarizado, ya que no está concebido por ahora para ser estudiado, sino con fines de archivo y clasificación. Volvemos aquí a la cuestión ontológica: ¿son los datos editoriales de la edición moderna entidades susceptibles de ser estudiadas mediante la lectura distante? Dada la falta de publicaciones que así lo consideren, lo dejaremos por ahora como un dato estrictamente metatextual no cuantificable; tal vez, habría que revisar este punto en el futuro y normalizar el formato para poder tenerlo en consideración como una variable más.

El género y el subgénero se indican mediante `<g>` y `<gg>`. Aquí debemos considerar la misma precaución que observamos con el nombre del autor, porque el computador sería incapaz de reconocer, por ejemplo, `capa y espada` como equivalente a `de capa y espada` sin complicar el algoritmo de evaluación de los términos a comparar. Una solución eficaz consiste en asignar códigos mnemotécnicos sencillos para cada género y escribir luego de manera automática el nombre completo usando una tabla de conversión.

Tenemos además la etiqueta `<el>` con la que introducimos el elenco. Aquí pondremos opcionalmente la cabecera que utiliza la obra para introducirlo (por ejemplo, «Personajes que intervienen») seguida de una barra vertical y, a continuación, los personajes. Estos van separados por asteriscos e, internamente, puede constar, además del nombre, de un epíteto opcional si lo hubiera, introducido por una coma. Conviene cuidar el detalle aquí, porque esta línea la emplearemos en la conversión a XML-TEI no solo para los metadatos, sino para presentar las partes impresas en el cuerpo del texto. En efecto, la información necesaria para producir el bloque del cuerpo textual bajo la rama `<front>` la obtenemos a partir de la información proporcionada en estos metadatos. Gracias a ella disponemos del autor, los títulos y las *dramatis personae*.

La etiqueta `<j>` señala un nuevo acto o jornada. La aplicaremos incluso en obras sin

macrodivisiones, pues su primera ocurrencia es la marca que señala el comienzo de la parte estructural de la obra. No existe una etiqueta específica para señalar las escenas porque no es habitual que este tipo de segmentación se señale explícitamente en las obras. A partir de aquí, comienza la representación en sí y señalamos la jerarquía estructural del texto mediante el espaciado.

Dejamos las didascalias de personaje en una línea sin sangrar y sin etiqueta que introduzca el texto. Podemos, además, señalar facultativamente el identificador `<xml:id>` de cada personaje para que se tenga en consideración al convertirlo a TEI. Esto puede resultar útil cuando aparecen parlamentos colectivos, para indicar con ello quiénes toman parte. En cualquier caso, poniendo el nombre de un personaje, hemos señalado el comienzo de un nuevo parlamento. Ahora sangraremos el texto para todo el bloque de orden jerárquico inferior supeditado a la intervención de este personaje, que incluye tanto texto como acotaciones internas. Podemos encontrar versos completos o comienzos de verso, que no necesitan más información, ecos, que indicaremos con la etiqueta `<e>` o fragmentos en prosa, que denotamos mediante la etiqueta `<p>`. De la misma manera, si queremos señalar una acotación interna, lo indicaremos sangrando la línea antes de etiquetarla con `<i>`; para una externa, evitaríamos el sangrado. En el caso de versos compartidos, la continuación o continuaciones de un verso inconcluso se marcan con unidades de sangrado adicional para que reproduzca visualmente la edición impresa. En la práctica, dos unidades de sangrado son suficientes para el computador, pero es recomendable emplear niveles sucesivos, ya que estos ayudan a la identificación visual. Su equivalente en XML-TEI lo vemos en Listado 7.3.

Listado 7.3: Equivalente XML-TEI.

```
<sp who="#uno">
    <speaker>UNO</speaker>
    <lg>
        <l>El primer verso de Uno,</l>
        <l>segundo, de Uno, el verso.</l>
        <l part="I" >Tercero es medio.</l>
    </lg>
</sp>
<sp who="#otro">
    <speaker>OTRO</speaker>
    <lg><l part="M">que</l></lg>
</sp>
<sp who="#uno">
    <speaker>UNO</speaker>
    <lg>
        <l part="F">sigue</l>
        <l n="5">es este otro parlamento.</l>
```

```
            <stage>Acotación interna.</stage>
            <l>Quinto verso, también fin.</l>
        </lg>
    </sp>
    <sp who="#uno #otro">
        <speaker>UNO y OTRO</speaker>
        <lg><l>¡Fin!</l></lg>
    </sp>
    <stage>Vanse. Acotación externa. Entra un Tercero.</stage>
    <sp who="#tercero">
        <speaker>TERCERO</speaker>
        <stage>Lee.</stage>
        <p>Esto no es un verso sino un texto en prosa</p>
    </sp>
```

De esta manera, ya tendríamos nuestro texto como datos semiestructurados, lo que nos permitiría tratarlo con el computador para todo tipo de estudios cuantitativos sobre las entidades definidas. Asimismo, el formato facilitaría otras aproximaciones. Pensemos, por ejemplo, en una prueba estilométrica clásica, en la que la entidad medida es la frecuencia de las palabras. En este caso, no hemos definido esa entidad para su medición, aunque hacerlo llevaría muy pocos pasos y, además, sencillos. Considerando que los programas de estilometría toman el texto en sí como entrada, deberíamos eliminar tanto los elementos paratextuales (título de la jornada) como aquellos otros no dramatizados (acotaciones...). Bastaría para esto aplicar una serie de expresiones regulares para eliminar secuencialmente los elementos deseados. Empezaríamos, por ejemplo, por todas las líneas sin sangrar, luego las que empiezan con una etiqueta de acotación y, finalmente, a discreción de quien lleve a cabo la prueba, tal vez las líneas introducidas por etiquetas de prosa o eco.

Listado 7.4: Preparación de archivo VED para análisis estilométrico.

```
$ sed '/^\S/d' pieza.ved | sed '/<i>.*/d | sed 's/<.*>//'
```

Podríamos llevar una cuenta de personajes en escena según un procedimiento análogo al de determinar personajes colectivos. La aproximación más simple es incluirlos en una lista a medida que intervienen. La distribución de esta cambiaría con cada intervención, de manera que el último personaje pasaría a ocupar siempre la posición final. De esta forma, tras una acotación *vase*, se sabría exactamente a quién alude. Más complicado sería *Vanse*, porque atañe al menos a dos personajes, pero desconocemos a cuántos en concreto. Aproximaríamos aquí, dejando el resto a la revisión manual. Si se especifican nombres en la acotación, esta afectaría a los aludidos sin importar

la posición que ocupen. Lo vemos en Algoritmo 2, donde evaluamos una lista *fuera* indica los personajes en escena por orden de salida y la acotación.

Algoritmo 2: Personajes que se van.

Método `EnEscena`(*fuera, acotación*)**:**
  si *'Va[n]se'* $\in$ *acotación* **entonces**
    si $\exists personaje_i \in acotación, \forall personaje \in fuera$ **entonces**
      fuera $\leftarrow fuera - personaje_i$
  si no, si *'Vase'* $\in$ *acotación* **entonces**
    fuera $\leftarrow fuera_{1..n-1}$
  si no, si *'Vanse'* $\in$ *acotación* **entonces**
    fuera $\leftarrow fuera_{1..n-2}$
  devolver *fuera*

En nuestro caso, hemos usado el procedimiento para determinar los personajes colectivos (Apartado B.5), de manera que si aparece, por ejemplo, LAS DOS, serán los dos últimos personajes femeninos que han precedido la última intervención, o, si es todos, —de modo algo impreciso—, la lista entera salvo el personaje que acaba de hablar. Sin embargo, bastaría con utilizar ese valor para otros propósitos, como los descritos en Capítulo D si así se desea.

7.3 Excurso III: De VED a XML-TEI

Como hemos visto, el formato VED define una ontología que constituye un subconjunto de XML-TEI y permite representar las principales entidades textuales definidas por este último para el drama. Esto facilita emplear los archivos VED para codificar una obra y convertir la edición automáticamente a XML-TEI de manera rápida, incluso sin estar familiarizado con ninguno de ambos formatos. Para llevar a cabo este proceso de transformar datos semiestructurados de forma visual en otro jerárquico no estrictamente secuencial, emplearemos el programa `text2tei` (Sanz-Lázaro, 2022b), cuyo código fuente se encuentra en Apartado B.5 en su última versión a la fecha de entrega de este trabajo. Su mecanismo, a grandes rasgos, consiste en ir recorriendo las líneas del archivo VED e ir asignando los valores en el lugar correspondiente de la estructura XML-TEI.

Para realizar esto, definimos primero las estructuras de datos que emplearemos. Como datos de entrada, tomamos un archivo que guardamos en una variable, que separamos como una lista de sus líneas. Seguidamente, creamos otra variable, en la que establecemos el esqueleto de la estructura XML-TEI donde colocaremos la información, con la cabecera de metadatos, los preliminares y el cuerpo del texto.

Algoritmo 3: Conversión de VED a XML-TEI.

```
vedtext ← texto.ved
líneas ← líneas + {línea}∀línea ∈ vedtext
metadatos ← {(a,∅), (t, ∅), (tt, ∅), (g, ∅), (o, ∅), (f, ∅), (el, ∅)}
árbol ←CreaXML()
árbol.AñadeElemento({cabecera, preliminares, cuerpo})
variable_α ← línea − α, ∀línea ∈ vedtex ∧ línea empieza con α
metadatos_⟨f⟩ ← EvalúaFecha(variable_⟨f⟩)
metadatos_⟨a⟩ ← EvalúaAutor(metadatos_⟨a⟩, tabla_autores.xml)
metadatos_⟨el⟩ ← EvalúaElenco(metadatos_⟨el⟩)
árbol.cabecera ← CompletaÁrbol(metadatos)
árbol.preliminares ← CompletaÁrbol(metadatos)
para línea ∈ líneas hacer
    si línea empieza con ⟨j⟩ entonces
        árbol.cuerpo.NuevoActo()
    si no, si ⟨i⟩ ∈ línea entonces
        si línea empieza con espacio entonces
            árbol.cuerpo.acto.IntroduceAcotación(línea − ⟨i⟩)
        si no
            árbol.cuerpo.acto.parlamento.IntroduceAcotación(línea − ⟨i⟩)
    si no, si línea empieza con letras entonces
        árbol.cuerpo.NuevoParlamento(línea)
    si no, si línea empieza con espacio entonces
        árbol.cuerpo.acto.parlamento.EvalúaLínea(línea)
```

El siguiente paso será guardar los elementos de la cabecera en las variables recién definidas para manipularlas de la forma más conveniente. Así, buscamos su valor en las líneas con la etiqueta que corresponda. Como dijimos, la fecha presenta variaciones, por lo que la volvemos a evaluar para asignarle el formato preciso según se define en las directrices de TEI. Hacemos lo mismo respecto al autor y completamos el nombre consultando un archivo externo, donde tenemos la correspondiente denominación única completa de acuerdo con las convenciones de notación onomástica TEI (Listado 7.5). Con todos los datos de la cabecera en un formato adecuado, llenamos las ramas correspondientes en el árbol. Como ya disponemos de los componentes necesarios para completar los preliminares, lo hacemos. Con esto, tenemos listos los elementos estructurales fijos y comenzamos a tratar los variables.

Para disponer de los elementos del cuerpo del texto, iteramos por todas las líneas, ignorando, claro está, aquellas que sean comentarios. Este bucle se diferencia del anterior en que ha de introducir un número indeterminado de elementos que, a su vez, pueden contener una cantidad variable de entidades subordinadas. De esta manera, cada vez que aparezca una etiqueta señalando un nuevo acto, producimos un nuevo

bloque en el cuerpo del texto, dentro del cual iremos construyendo las estructuras que contiene.

Listado 7.5: Formato de autores.

```
<author cert="high">
<persName>
<forename>Pedro</forename>
<surname sort="1">Calderón</surname>
<nameLink>de la</nameLink>
<surname>Barca</surname>
</persName>
<idno type="wikidata">Q170800</idno>
<idno type="pnd">118518399</idno>
</author>
```

Así pues, si aparece una indicación de acotación, la introduciremos como elemento hijo del acto con el que estamos trabajando si la etiqueta es el primer elemento de la línea o como elemento hijo de un parlamento si va precedida de algún espacio. Si la línea no empieza por espacio ni por etiqueta[102], será una didascalia de personaje indicando el locutor; por lo tanto, creamos un nuevo parlamento en el acto dentro del que nos encontremos. Si la línea empieza con espacio, se trata de texto —de haber llevado la etiqueta de acotación, nos habría obligado a entrar en la segunda bifurcación condicional y no habríamos entrado en esta—. Para agregar la nueva línea, evaluaríamos en primer lugar si lleva etiquetas de eco o prosa, que nos obligaría a considerar de manera distinta la cuenta de versos. De ir sin etiquetar, se comprobaría el sangrado para, de la misma manera, mantener invariable la numeración versal si estamos ante la continuación de un verso compartido y, además, introducir el atributo descriptivo de la parte del verso que representa tanto la línea del elemento actual como la línea precedente.

7.4 Estructurando los datos

Al tener definidas estas entidades, produciremos archivos mediante la traducción entre diferentes formatos, a condición de que compartan una misma ontología. Así, crearemos archivos VED tomando como fuente los diferentes formatos iniciales para, a partir de ahí, por ejemplo, codificar la obra según los principios de XML-TEI. De este modo, nos encontramos en condiciones de producir material que puede ser ya

[102] Aunque no necesariamente por un carácter alfabético, ya que puede suceder que, habiendo varios personajes del mismo tipo (por ejemplo, PASTOR 1.º, PASTOR 2.º y PASTOR 3.º), la didascalia de personaje se abrevie mediante el ordinal (1.º), el numeral (1) o que el editor marque un personaje implícito con corchetes ([PASTOR] 3.º).

empleado para diversos propósitos finalistas, como representar redes de personajes, analizar la dominancia escénica y, en general, cualquiera de los análisis a mano que ilustra Capítulo D. Dicho de otra manera, estos resultados resultan ya productivos por sí mismos; buena muestra de ello son las comedias de Calderón aportadas a CalDracor (Lehmann y col., 2022), todas ellas preparadas de los originales como archivos VED, traducidos posteriormente de manera automática a XML-TEI según lo descrito en este capítulo.

A pesar del valor intrínseco que pudiera tener ese resultado, es apenas un paso intermedio en este estudio. Nuestra intención es ir más allá. Téngase en cuenta que los textos en XML-TEI no dejan de ser una fuente que, aunque con una estructura definida, requiere programas capaces de clasificarla y cuantificarla y de llevar a cabo cálculos partiendo de esas cuantificaciones. Si bien esto puede hacerse hoy de manera sencilla —tanto que podría darse por supuesto, ya que muchos programas orientados a ese tipo de tareas se encuentran libremente accesibles—, no deja de ser un paso intermedio que, además, está limitado a entidades y relaciones predefinidas, aparte de que su estructura jerárquica obliga a emplear *trucos* cuando dos elementos estructurales concurrentes se superponen (TEI Consortium, 2022, p. 692). Nuestra intención es simplificar más aún ese análisis para poder usar los datos sin modificar en exámenes de lectura distante; esto pasa por presentar las entidades textuales cuantificadas para que los cálculos puedan hacerse de forma directa, así como extraer otra información relevante —precisamente una de especial interés para el proyecto Sound and Meaning: el ritmo—. Se trata de organizar los datos de tal manera que puedan ser leídos sin mediación por los programas más habituales de cálculo estadístico y que, además, contengan una ontología que represente los entes y relaciones que nos interesan; para ello, necesitamos no solo identificar entidades textuales, sino ordenarlas.

¿Pero cómo se traslada el formato dramático a una tabla? El modelo de los cuadros ideado por Marcus (1973) para representar la intervención de los personajes en las obras teatrales ofrece una buena idea de cómo definir el eje temporal. Observemos el ejemplo Figura 7.1, que sigue los principios de ese modelo: las unidades que señalan el curso de la acción son los parlamentos. A cada uno de ellos le corresponde una fila, caracterizada mediante un valor ternario por su relación con los personajes de la obra: *presente hablando*, *presente en silencio* y *ausente*. Si bien es una aproximación aceptable, disponemos de medios para afinarla.

En efecto, en la obra teatral áurea, el texto se dispone en líneas, por lo que el transcurso del tiempo se divide en unidades discretas. Está claro que no es una medida exacta, en tanto que el teatro es polimétrico y, por ejemplo, un verso hexasílabo será métricamente más breve que un endecasílabo y las líneas acusaran normalmente la

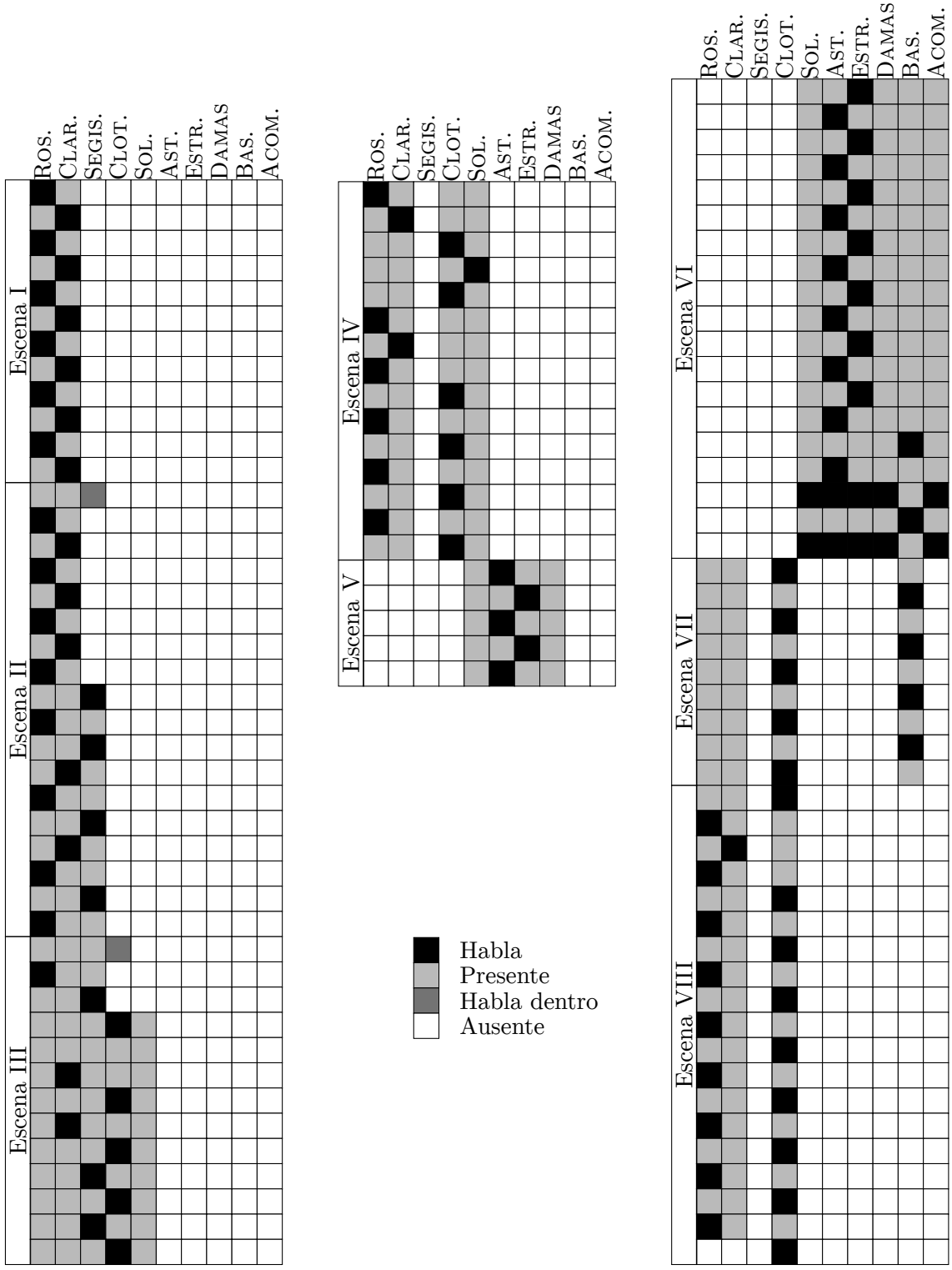


Figura 7.1: Estructura de la configuración dramática.

desproporción de igual manera. Asimismo, la acción dramática determina la cadencia de la declamación, por lo que dos líneas idénticas en el papel pueden realizarse más o menos aprisa sobre las tablas. Por lo tanto, trabajamos con una idealización en

la que suponemos la línea como una medida abstracta del paso del tiempo escénico. Definimos, en cualquier caso, una ontología en la que existe esta clase línea (L), con tantos individuos l_i ordenados sucesivamente como líneas tenga la obra, tal que $l_i \prec l_{i+1}$. De esta manera, encontraríamos en el índice una relación fundamental entre las entidades que hemos definido, pues indicaría la posición única en el tiempo de cada línea.

¿Cómo caracterizar la línea? Podríamos hacerlo a la manera de Marcus, pero nos topamos con el inconveniente de que cada pieza requeriría un formato de tabla propio, lo que complicaría trabajar con un corpus de obras teatrales. Debemos, por lo tanto, unificar las columnas para que sean las mismas para todas las obras. Por otra parte, como ya indicamos, no disponemos de información detallada sobre la presencia escénica, por lo que resultaría harto complicado determinarla automáticamente, y tremendamente costoso en recursos llevarlo a cabo de manera manual. La alternativa que proponemos es considerar una sola columna categórica, cuyos valores son los nombres de los personajes de las obras, y que acepte la composición de categorías. Esto es, adoptaría valores como Rosaura o Clarín, pero también tomaría el valor Rosaura $\oplus$ Clarín, si la didascalia de personaje indica que hablan los dos.

Por otra parte, el propio texto de la línea podría ser un valioso atributo que nos ayudaría a establecer relaciones léxicas y semánticas. Asimismo, al tener personajes, deberíamos añadir columnas con información adicional. Si bien hay nombres de personaje que son prototípicos —raro es que un *Chichón* no sea un gracioso—, hay otros más comunes que pueden aparecer representando tipologías distintas en diferentes obras. Por otro lado, es posible que el sexo sea ambiguo. Por lo tanto, no asumiremos una identificación preestablecida, sino que conviene indicarla en función de la obra. Para ello, necesitaremos columnas al efecto. Si bien el sexo de los personajes resulta sencillo de predecir de forma automática en muchos casos, cuando se indica mediante un nombre de uso habitual, va antecedido por un título de cortesía o un artículo —al menos en singular, ya que resulta complicado no pocas veces, incluso leyendo, distinguir si el plural es masculino o neutro—, la tipología del personaje lo es menos si no se indica en la presentación del elenco.

Debemos, además, tener en cuenta que no todos los elementos textuales son líneas propiamente dichas y hay algunos fuera de la diégesis. En concreto, las acotaciones no solo no corresponden a personajes, sino que tampoco ocupan un lapso de tiempo, por lo que van paralelos a la acción. Necesitamos incluirlos de tal manera que esto se tenga en cuenta. A despecho de forzar la terminología —en pro de la sencillez, añadiremos en nuestro descargo—, incluiremos estos elementos en la ahora difusa clase de las *líneas*. Para ello, podríamos definir un nuevo atributo con una numeración

alternativa al índice. Así podría llevarse la cuenta si se van marcando las acotaciones con números negativos decrementales para ordenarlas entre sí o ubicarlas de acuerdo con los versos. No obstante, esto resulta redundante porque disponemos de suficiente información con los atributos que representan el índice y el personaje. Si a este último, por ejemplo, lo denominamos ACOTACIÓN, sabríamos tanto su ubicación exacta como que la línea es extradiegética.

De la misma manera, dadas las características del drama áureo, podríamos incluir una segunda numeración adicional que nos indique el número de verso. Así, en uno compartido, el valor de los atributos para el índice y línea avanza con cada individuo, mientras que el que indica el verso permanece invariable hasta superar la última parte del verso actual. En el caso excepcional de que sea una línea extradiegética o un parlamento en prosa, el atributo estaría vacío; formalmente, $v_i \preceq v_{i+1} \vee v_i = \emptyset$.

Al contrario que en XML-TEI, donde los metadatos se indican una sola vez, el formato tabular demandaría que se repitieran en cada fila de la obra, en tanto que estos datos afectan a todos los versos. El autor y el título de la obra, información de la edición, fechas, categorización genérica o el subtítulo, son elementos que se contarían entre los invariables dentro de la obra. Debemos tener cuidado aquí de que la distinción sea efectiva. Por consiguiente, no está de más contar con una columna de subgénero y subtítulo, de manera que si el título es *La vida es sueño*, sepamos si nos encontramos ante la tragedia o el auto sacramental homónimo y, en este último caso, poder identificar la versión.

7.5 Implementación del proceso

Veamos, pues, cómo plasmamos esto en una representación formal. Recordemos que partimos de un texto en formato VED. Nuestro objetivo es obtener a partir de él una tabla de *líneas*[103], cuyos atributos nos permitan caracterizarlas de tal manera que sea posible realizar con ellas análisis cuantitativos directamente. Empezaremos definiendo los atributos de la línea y asignándoles un valor por defecto. Antes de empezar a recorrerlas, no tenemos definido un acto, parlamento o verso, sino que cobran entidad al leer la primera *línea*. Por lo tanto, estarán a cero hasta entonces. Definimos asimismo variables auxiliares para llevar la cuenta de los parlamentos y las acotaciones por separado. Los atributos de la clase, al no tener ninguna instancia de esta aún, estarán también a cero, salvo el locutor, que pondremos como *acotación*. Esto es, como habíamos decidido, la acotación es una instancia especial de la clase

[103] Usaremos letra redonda para la línea dramática en propiedad y cursiva para la entidad abstracta que hemos definido, que comprende acotaciones además de estas.

línea, que se diferencia de las partes habladas por su locutor y por tener número de línea negativo.

Algoritmo 4: Cabecera para la tabulación.

```
acto, parlamento, verso, parlamento_n, acotación_n ←0
pers ←'∅'
autor, Título, Género, sexo, s, edición, fecha ←∅
personaje ←ACOTACIÓN
tabla ←Tabla({Autor, Título, Subt., Género, Subg., Edición, Fecha,
 Jornada, Parlamento, Verso, Locutor, Tipo, Sexo, Texto})
nombres ←{∅}
```

Después de esto, pasaremos a predecir el sexo de los personajes. El método comete errores y necesita una revisión manual en cualquier caso. Sin embargo, dada la relativa predictibilidad de la onomástica hispana, el porcentaje de fallos se reduce hasta el punto de aliviar sustancialmente la anotación artesanal. Utilizaremos una función a la que le pasamos el locutor.

Buscaremos en las letras voladas de los ordinales, como en SOLDADO 1.º o DAMA 2.ª, de haberlos, ya que indican con precisión el sexo de los personajes individuales de un colectivo. De no existir el ordinal, optaremos por una aproximación basada en las flexiones nominales del español. Esto es, asumiremos que la terminación *-o* es masculina y *-os* es su plural y, al contrario, la terminación *-a* es femenina y el plural se construye de manera análoga. Esto no se cumple en todos los casos, por lo que afinaremos indicando, por ejemplo, que la terminación *-ías*, como en *Ananías* debe considerarse masculina. Si no responde a ninguno de estos patrones, se marca como indeterminado para un ulterior ajuste manual. Los nombres bíblicos y mitológicos son especialmente proclives al error.

Algoritmo 5: Determinación del sexo.

```
Método AsignaSexo(locutor):
    sexo ←X
    si nombre es femenino entonces
        sexo ←F
    si no, si nombre es masculino entonces
        sexo ←M
    si no
        sexo ←X
    nombres_nombre ←sexo
    devolver nombres_nombre
```

Con esto, recorreremos las líneas del texto. Supondremos que se encuentra en formato VED, pero los principios rectores deberían ser aplicables a otros de manera análoga

a la conversión que hicimos desde XML-TEI. Se trata de ubicar los elementos textuales y metatextuales y trasladarlos a una nueva estructura respetando sus relaciones. Distinguiremos dos tipos de línea, las que indiquen parlamento y personaje, y otras.

Algoritmo 6: Recorrido de las líneas de la obra (etiqueta).

```
para línea ∈ líneas hacer
    guardar_línea ←Falso
    si etiqueta ∈ línea entonces
        etiqueta, texto ←línea_{etiqueta,texto}
        si etiqueta es <jornada> entonces
            jornada += 1
        si no, si etiqueta es <título> entonces
            Título ←texto
        si no, si etiqueta es <autor> entonces
            autor ←texto
        si no, si etiqueta es <género> entonces
            género ←texto
        si no, si etiqueta es <edición> entonces
            edición ←texto
        si no, si etiqueta es <fecha> entonces
            fecha ←texto
        si no, si etiqueta es <acotación> entonces
            guardar_línea ←Verdadero
            verso ←−|verso|
            acotación_n ←acotacion_n − 1
            si sangrado entonces
                personaje ←ACOTACIÓN INT.
            si no
                personaje ←ACOTACIÓN
                parlamento ←acotación_n
            sexo ←∅
        si no, si etiqueta ∈ {<prosa>, <eco>} entonces
            guardar_línea ←Verdadero
            verso ←−|verso|
            personaje ←p
            sexo ←s
            texto ←línea
        si no
            verso ←|verso|
    si no
        /* Algoritmo 7 */
```

Antes de nada, definimos una variable para señalar si la línea corriente contiene texto que ha de ser guardado o solo modifica los valores de los atributos de las líneas a guardar. Inicializamos esta variable como $falsa$, si bien invertiremos su valor en caso

de que el tipo de estructura textual así lo demande. Aquí, como indicamos, distinguimos los diferentes tipos de entidad mediante etiquetas y el uso del sangrado. Si encontramos una línea etiquetada, separaremos el texto y examinaremos la etiqueta para modificar los valores en consecuencia. Si la etiqueta es de jornada, lo que denotaría el comienzo de un nuevo acto, incrementaríamos el contador correspondiente y no haríamos nada más. Lo mismo ocurriría si aparecen otras etiquetas que señalen contenido metatextual, como las alusivas al título y al subtítulo, al autor, género y subgénero de la obra, edición o fecha.

Si, por el contrario, la etiqueta indica contenido textual especial, como una acotación o un pasaje en prosa, activaremos la variable que habíamos definido para considerar la línea como parte del texto. De este modo, si la etiqueta señala que se trata de una acotación, guardamos su posición en el texto. Aquí hemos optado por el valor negativo del verso precedente, de manera que podamos ubicarla en el conjunto respecto a este y, a la vez, distinguirla de las partes representadas. Por comodidad, nos quedamos también con su posición en la lista de acotaciones para que, además de ahorrar algunos cálculos al tener valores consecutivos, diferenciemos entre dos acotaciones seguidas. Distinguimos entre acotaciones internas y externas según el sangrado. En nuestro modelo, la acotación interna va sangrada al nivel de los versos, mientras que la externa no. En caso de acotación interna, el número de parlamento permanece invariable, mientras que si es externa lo señalaremos cambiándolo por el número de acotación —negativo, recordemos—, para facilitar el uso de nuestros resultados. Esto es, a la hora de trabajar con la tabla que produciremos, basta con ignorar las filas cuya celda destinada al número de línea contiene un valor negativo.

En el caso de que tengamos una línea declamada, pero que no se considere para la cuenta de versos, sea por hallarse en prosa o por ser el eco de un verso precedente, activamos la señal si queremos que guarde el verso, pero cambiamos los valores correspondientes a los atributos que lo describen, de manera que, después, sea posible identificarlo y tratarlo por separado si así se desea. Para ello, le asignamos un número de verso negativo, pero con el locutor apropiado. De esta manera distinguimos entre acotaciones, donde el nombre del locutor señala que es una acotación y los números de línea y verso son negativos y partes no versadas del parlamento, también con cuenta negativa en el verso, aunque con el correspondiente personaje como locutor y número de línea positivo. Al contrario que con las acotaciones, ahora sí guardamos el sexo.

En cualquier otro caso, tomaremos simplemente la línea como un verso a ignorar. Sabemos qué hacer con las líneas etiquetadas; nos faltan, sin embargo, aquellas carentes de etiqueta (Algoritmo 6). Estas pueden ser de tres tipos: sin sangrado, con un nivel

Algoritmo 7: Recorrido de las líneas de la obra (no encuentra etiqueta).

```
si no
    /* Viene de Algoritmo 6 */
    verso ←|verso|
    si linea = ∅ entonces
        nada
    si no, si ¬sangrado entonces
        pers ←línea
        parlamento_n ←parlamento_n + 1
        s ←AsignaSexo(p)
    si no, si |sangrado| = 1 entonces
        guardar_línea ←Verdadero
        verso ←verso + 1
        personaje ←p
        sexo ←s
    si no
        guardar_línea ←Verdadero
        personaje ←p
        sexo ←s
    parlamento ←parlamento_n
línea ←{Autor: autor, Título: título, Género: género, Edición: edición,
 Fecha: fecha, Jornada: jornada, Parlamento: parlamento, Verso: verso,
 Locutor: locutor, Sexo: sexo, Texto: texto}
si guardar_línea es Verdadero entonces
    AñadirLínea(línea, tabla)
```

de sangrado o con más de uno. Las líneas correspondientes al primer caso indican el nombre del locutor de las siguientes líneas, las del tipo segundo denotan inicio de un verso y las últimas del último señalan su continuación.

Lo primero que nos encontramos es una línea vacía. En un escenario ideal, estas no existen. En un texto digitalizado real, como los descritos en el capítulo anterior, sí. Así que, si bien es superfluo para nuestra descripción conceptual, llamamos la atención sobre ello como aviso a navegantes que quieran llevar a la práctica las ideas aquí expuestas. Ahora evaluamos el sangrado. Si es inexistente, se trata de una didascalia de personaje y, por lo tanto, del comienzo de un nuevo parlamento con un otro locutor, por lo que incrementamos el valor de la cuenta de estos y asignamos el contenido de la línea al locutor. Además, aplicamos en este momento la función que habíamos diseñado para adivinar *grosso modo* el sexo del personaje.

Si detectamos un nivel de sangrado, estamos ante una línea de texto porque, de tratarse de una acotación interna, se habría cumplido la condición de Algoritmo 6 y no habría entrado en esta. Por lo tanto, activamos la variable de guardar la línea e incrementamos la cuenta de versos, a la vez que asignamos los valores correspondientes

al locutor y su sexo.

Título	Acto	Parl.	Verso	Pers.	Sexo	Texto
La vida es sueño	I	-1	-1	*Acotación*		Sale en lo alto de un monte ROSAURA en hábito de hombre,de camino, y en representando los primeros versos va bajando.
La vida es sueño	I	1	1	ROSAURA	♀	Hipogrifo violento,
La vida es sueño	I	1	2	ROSAURA	♀	que corriste parejas con el viento,
La vida es sueño	I	1	3	ROSAURA	♀	¿dónde rayo sin llama,
La vida es sueño	I	1	4	ROSAURA	♀	pájaro sin matiz, pez sin escama
La vida es sueño	I	1	5	ROSAURA	♀	y bruto sin instinto
La vida es sueño	I	1	6	ROSAURA	♀	natural, al confuso laberinto
La vida es sueño	I	1	7	ROSAURA	♀	de esas desnudas peñas
La vida es sueño	I	1	8	ROSAURA	♀	te desbocas, te arrastras y despeñas?
La vida es sueño	I	1	9	ROSAURA	♀	Quédate en este monte,
La vida es sueño	I	1	10	ROSAURA	♀	donde tengan los brutos su Faetonte;
La vida es sueño	I	1	11	ROSAURA	♀	que yo, sin más camino
La vida es sueño	I	1	12	ROSAURA	♀	que el que me dan las leyes del destino,
La vida es sueño	I	1	13	ROSAURA	♀	ciega y desesperada,
La vida es sueño	I	1	14	ROSAURA	♀	bajaré la cabeza enmarañada
La vida es sueño	I	1	15	ROSAURA	♀	deste monte eminente
La vida es sueño	I	1	16	ROSAURA	♀	que arruga el sol el ceño de la frente.
La vida es sueño	I	1	17	ROSAURA	♀	Mal, Polonia, recibes
La vida es sueño	I	1	18	ROSAURA	♀	a un extranjero, pues con sangre escribes
La vida es sueño	I	1	19	ROSAURA	♀	su entrada en tus arenas;
La vida es sueño	I	1	20	ROSAURA	♀	y apenas llega, cuando llega a penas.
La vida es sueño	I	1	21	ROSAURA	♀	Bien mi suerte lo dice;
La vida es sueño	I	1	22	ROSAURA	♀	mas ¿dónde halló piedad un infelice?
La vida es sueño	I	-2	-22	*Acotación*		(Sale CLARÍN, gracioso.)
La vida es sueño	I	2	23	CLARÍN	♂	Di dos, y no me dejes
La vida es sueño	I	2	24	CLARÍN	♂	en la posada a mí cuando te quejes;
La vida es sueño	I	2	25	CLARÍN	♂	que si dos hemos sido
La vida es sueño	I	2	26	CLARÍN	♂	los que de nuestra patria hemos salido
La vida es sueño	I	2	27	CLARÍN	♂	a probar aventuras,
La vida es sueño	I	2	28	CLARÍN	♂	dos los que entre desdichas y locuras
La vida es sueño	I	2	29	CLARÍN	♂	aquí habemos llegado,
La vida es sueño	I	2	30	CLARÍN	♂	y dos los que del monte hemos rodado,
...						

Cuadro 7.1: Tabla de atributos dramáticos.

En caso contrario, esto es, si no se trata ni de una línea sin sangrar ni con un solo nivel de sangrado, estamos ante la continuación de un verso, ya que la única opción que incumple esas dos condiciones es que se dé más de un nivel de sangrado. En consecuencia, repetiremos lo hecho con un solo nivel, salvo el incremento del número de verso porque, al tratarse de una continuación, le corresponde el mismo valor que al inicial. Con esto, creamos una estructura para definir la línea con sus atributos, a los que asignamos los valores que hayan tomado las variables que definimos en función de las condiciones a las que se ajusta la línea. Si, además, se cumple la condición de guardar, añadimos la línea a la tabla.

Hemos obtenido así una tabla de la estructura dramática del texto parecida a Cuadro 7.1. Además de las columnas indicadas de la tabla de ejemplo, naturalmente,

se incluiría datos adicionales[104], tanto para indicar metainformación —verbigracia, el autor o la procedencia de la edición—, como para complementar las líneas con atributos dinámicos, tales como el tipo de personaje, dama y gracioso en este caso.

```
In [1]: import pandas as pd
        vida = pd.read_csv('calderon_la-vida-es-sueno.csv')
        vida = vida.loc[(vida['Act'] == 1) & (vida['Character'] != 'STAGE DIR.')]
        vida.groupby('Character')['Speech'].nunique().plot(kind='bar')
Out[1]: <AxesSubplot: xlabel='Character'>
```

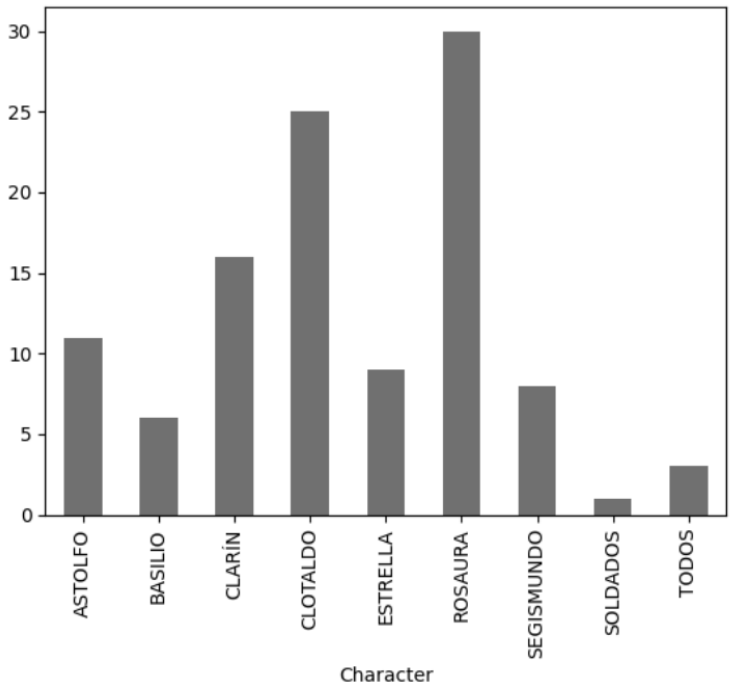


Figura 7.2: Ejemplo de aplicación.

La tabla permite un uso flexible y obtener información de manera rápida y sencilla. Por ejemplo, pueden compararse cuántas líneas dice cada personaje o, ya que disponemos del texto en sí, contar palabras por personaje (Figura 7.2); podría asimismo diferenciarse entre personajes femeninos o masculinos, o seguir el grado de participación de cada uno a lo largo de las escenas en función de sus parlamentos. Conociendo esto de cada una de las obras de un corpus, podrían compararse entre sí, en masa incluso, para encontrar piezas que se salgan de la norma. Si se agrupan, podría observarse las diferencias entre épocas, autores, géneros, o en función de varias variables, como la evolución de un autor en la producción de un género concreto a lo largo del tiempo.

[104]Las dimensiones del papel lo desaconsejan aquí. No obstante, se han representado todos los datos dinámicos.

A las pronunciadas voces
de blandas músicas junten
sus no pronunciadas solfas
las aves, siendo a su numen.

Calderón, *La nave del mercader*

Transcripción automática 8

8.1 Sobre la transcripción automática

Para poder analizar digitalmente los diálogos acercándonos a su declamación, su representación grafémica convencional se antoja insuficiente. Al contrario que el lector de carne y hueso, el computador es incapaz de inferir la prosodia a partir del texto si no se le indica explícitamente. La máquina lee lo que se le ordene leer y lo interpreta como se le haya instruido hacerlo. Lo ideal sería olvidarnos de los textos y recurrir a grabaciones de calidad tomadas durante puestas en escena de las obras que deseemos examinar, tanto mejor si son recitadas por actores bregados en la interpretación de textos áureos. A partir de esas grabaciones podríamos extraer y estudiar la representación de las ondas sonoras. Sin embargo, la grabación proporcionaría la mejor descripción posible del sonido de *cada* función, por lo que sus resultados solo serían totalmente válidos para sus respectivos casos (Sánchez Jiménez, 2017, p. 12); otro elenco de actores, un nuevo director de escena o, incluso, las variaciones en la forma de entender el personaje tras representarlo un par de veces, conducirían a cambios en la entonación. Podríamos entonces tomar varias muestras diferentes, pero esto dispararía los recursos requeridos hasta hacer inasumible un estudio extenso.

Otro inconveniente que se plantearía a la hora de componer un corpus sería la necesidad impuesta por el material disponible, ya que este se limita a aquellas funciones de las que existe una grabación y esta es accesible. Como dijimos, la idea de la lectura distante es, sin embargo, el examen del bosque y no de los árboles más altos. Y, si únicamente una mínima parte de piezas están editadas adecuadamente, un reducidísimo porcentaje de estas se lleva a las tablas. De esas representaciones, las que han

sido grabadas son una minoría y, de esa minoría, solamente un porcentaje ínfimo se encuentra al alcance del público. Afortunadamente, hay un máximo común divisor entre interpretaciones, en el que todas las lecturas fieles concuerdan, por lo que no es necesario llegar a este de forma inductiva. Al contrario, el acuerdo de mínimos es el propio texto fijado. Nos adaptaremos, pues, a las condiciones a las que nos enfrentamos y nos conformaremos con el texto escrito, un modelo del objeto real. Como en todo modelo, hay elementos que se pierden, pero esto permite también concentrarse en los otros, los que hemos seleccionado. En nuestro caso, los fonológicos, a despecho de los sonidos auténticos de la representación.

En una función hay actores que median entre el texto escrito y lo que escucha el público. Esto es, los actores *interpretan* el texto. Leen la grafía asumiendo que responde a unas reglas dadas y la pronuncian aplicando a lo que ven un conjunto de operaciones que han aprendido de antemano. Estas operaciones las desconoce el computador *a priori*. Por suerte, el texto escrito según la ortografía normativa contiene casi toda la información necesaria para pronunciarlo, siempre y cuando se conozcan sus convenciones y las reglas que dictan como llevar a cabo la traducción. La ecuación $algoritmos + estructuras de datos = programas$ que da título a una obra clásica de la ciencia informática (Wirth, 1976) sintetiza esta situación: la ortografía y la sintaxis española definen la forma en que se estructuran los datos de entrada, pero debemos definir los algoritmos que empleamos para tratarlos.

Estas operaciones se corresponden a las reglas de la fonología, que el usuario competente de la lengua maneja orgánicamente y sin reparar en ello. Nuestra labor consiste en formalizarlas en una biblioteca informática que funcione de manera autónoma. Esto es, el programa principal importará un módulo de esta biblioteca en el que delegará la transcripción fonológica, de forma que, una vez introducida la cadena de texto a transcribir, recibe de vuelta su transcripción, aun sin ver lo que sucede entre bambalinas. Eso permitiría teóricamente sustituir este módulo por otro sin necesidad de modificar el programa principal, de modo que, ante la eventualidad de una solución mejor, sería trivial reemplazar la nuestra.

Al comenzar a trasladar los elementos teóricos a código, este módulo formaba parte integral de las funciones propias de la escansión. Sin embargo, poco tiempo después, se plantearon otros usos ajenos al originario, por lo que hubo que decidir entre reutilizar fragmentos del código en un programa nuevo independiente o externalizar los algoritmos de transcripción fonológica. Dado que la modularidad es una enorme ventaja a la hora de encontrar fallos, hacer modificaciones o añadir mejoras, optamos por lo segundo. Este desarrollo independiente también ha propiciado la introducción de nuevas funcionalidades que no se requieren en la investigación principal, pero que

resultan de gran ayuda para otros fines, como la transcripción fonética y la transliteración a otras codificaciones. No obstante, además incluye funcionalidades orientadas exclusivamente a la lírica áurea, como la marcación de posibles haches aspiradas. La implementación en Python del módulo que detallamos en este capítulo está disponible de forma independiente como en la biblioteca `fonemas` (Sanz-Lázaro, 2023a) y `silabeador` (Sanz-Lázaro, 2023d), cuyas versiones más recientes a fecha de publicación de estas páginas se encuentran en Apartado B.2.

8.2 Módulo transcriptor

Descripción general

A grandes rasgos, el mecanismo consiste en tomar una cadena de caracteres y separarla en una lista, cada uno de cuyos elementos es una consonante, una vocal o una unión de vocales en caso de diptongo o triptongo. Después se recorre la lista para distribuir las consonantes en torno a los núcleos silábicos vocálicos. Finalmente, se establece la posición del acento prosódico de conformidad a las reglas ordinarias del español.

Gracias a este diseño modular, desde la perspectiva del procedimiento de escansión, la función transcriptora se reduce a un operador que recibe un verso y devuelve la transcripción de sus sílabas. Sin embargo, para llegar a ese resultado se requiere llevar a cabo una serie de procesos bajo la mesa para manipular el texto hasta dejarlo en su forma final. Simplificando los detalles del proceso, este consiste en aplicar reglas de transformación de primitivas a la cadena grafémica y, una vez transformada esta, invocar un módulo de división silábica para que separe las sílabas. En las siguientes líneas examinaremos cómo hemos construido el módulo de transcripción fonética (Algoritmo 8), recorriendo paso a paso los procesos de transcripción y silabación.

El programa requiere algunas funciones de bajo nivel para operar. Parte de ellas las provee el propio lenguaje de programación, mientras que otras específicas tenemos que definirlas nosotros primero para poder usarlas después. Por lo tanto, antes de todo, hemos de preparar el entorno para que disponga de cuanto vayamos a necesitar. De ahí que las primeras instrucciones carezcan de una finalidad operativa. Estas sirven para componer las estructuras de datos que requeriremos para llevar a cabo la transformación. Atendiendo a esto, importaremos funciones de los diversos módulos que vayamos a requerir: en la implementación real, serían procedimientos específicos; aquí, el buen entendimiento que aporta el lector al asumir pasos implícitos.

Algoritmo 8: Definiciones previas de la biblioteca.

```
Importar(Bibliotecas)
Clase Valores():
    palabra ← {}
    slbs ← {}
```

La *clase*[105] de datos la requerimos para definir los valores que tomarán los *atributos* del *objeto* que representa el texto transcrito. Denominaremos a esta clase `Valores`, y la dotaremos de dos atributos, válidos para los distintos tipos de transcripción que llevamos a cabo. Cada uno de estos atributos contiene dos valores, que definimos como sendas listas vacías *palabras* y *sílabas*, cuyos elementos son palabras y sílabas transcritas, respectivamente. Una vez disponemos de los módulos que requeriremos y definida la clase auxiliar que sirve de plantilla para los valores, tenemos lo necesario para empezar con la clase principal.

Definición de la clase de transcripción

Como se observa en el Algoritmo 9, esta clase acepta un argumento [106]. La clase `Transcripción` tiene asimismo cuatro atributos y cuatro métodos públicos que pueden emplearse sin restricciones en programas externos. El argumento obligatorio es el propio texto que deseamos transcribir. A este lo denominaremos *texto* en la clase, y espera recibir una cadena de caracteres Unicode, sea en español o su transcripción en el AFI.

Algoritmo 9: Cabecera del módulo de transcripción.

```
Clase Transcripción(texto):
    texto ←Limpia(texto)
    si reordenar entonces texto ← Reordena(texto)
    fonología ← TranscripciónFonológica(texto)
    fonética ← TranscripciónFonética(fonologia)
    sampa ← TransliteraciónSAMPA(fonetica)
```

Respecto a los atributos de la clase `Transcripción`, todos ellos deben ser accesibles desde el exterior. El primer atributo es *texto*, que almacena la cadena de caracteres resultante de preprocesar la entrada con los métodos descritos en Apartado 8.2. El

[105] En programación orientada a objetos, una *clase* es una plantilla para crear instancias de una estructuras de datos (*objetos*) a partir de unos valores iniciales. La clase describe variables (*atributos*) y procedimientos (*métodos*) para operar sobre estas.

[106] La implementación en Python admite además cuatro argumentos opcionales mpara alterar el comportamiento de la transcripción cuando las circunstancias así lo aconsejan, de forma que disponemos así de una herramienta flexible para usar en casos particulares.

atributo *fonología* es el resultado de aplicar el método `TranscripciónFonológica` al texto de entrada. Este contiene una instancia de la clase `Valores`, que ya habíamos definido, y su contenido serán sendas listas de palabras y sílabas transcritas según el AFI. El atributo *fonética* se obtiene mediante la aplicación del método `TranscripciónFonética` al valor del atributo *fonología*. Como este, es una instancia de la clase `Valores`, por lo que también contendrá subatributos *palabras* y *sílabas*, pero, en este caso, lo que almacena las transcripciones fonéticas del texto. Finalmente, el atributo *sampa* es la salida del método `TransliteraciónSAMPA` cuando se le pasa el contenido de *fonética*. Al igual que los dos atributos precedentes, contiene una instancia de `Valores` con los subatributos *palabras* y *sílabas*, aunque con su transcripción al Speech Assessment Methods Phonetic Alphabet (SAMPA).

Excurso IV: Argumentos de la implementación en Python

En la implementación en Python, incluimos los argumentos opcionales para indicar si las palabras monosilábicas han de marcarse con símbolo de sílaba tónica (/ˈ/ en AFI) o no. Esto es, *eres tú* se resuelve por defecto como /ˈeres tu/, aunque puede forzarse a /ˈeres ˈtu/ si así se lo indicamos. Lo hacemos mediante una variable de tipo booleano que está inicializada por defecto como *falso*. Si creamos una instancia de la clase de transcripción indicando que el valor de ese argumento es *verdadero*, forzamos el comportamiento contrario al predefinido. En cualquier caso, aquí consideramos tan solo la sílaba acentuada de cada palabra hasta que, más adelante, el análisis morfosintáctico permita identificar los acentos prosódicos. En la discusión que sigue lo obviaremos por ser irrelevante en el aspecto teórico.

El programa de Python ofrece la posibilidad de señalar explícitamente las haches a comienzo de palabra. Si bien la aspiración ya se había perdido en Castilla entrado el siglo XVII (Lapesa, 1981/2008, sec. 91.3), esta sobrevivía en hablas dialectales (Salvador Plans, 1982) —como lo eran las de muchos dramaturgos—, pero también en la poética como herramienta de compensación métrica. Por este motivo, conviene marcar la posición de las haches de manera que, en caso de varios hiatos posibles, la posible aspiración tenga valor determinante para inclinarse por esa sílaba. Se ha diseñado como una característica opcional para facilitar la transcripción de textos modernos, que no requieren de esta funcionalidad. De nuevo, ignoramos la capacidad de elegir de la que hemos dotado a la prueba práctica y marcaremos por defecto las haches iniciales para tenerlas en cuenta para valorar sinalefas.

Otra opción es la capacidad de alterar la posición de los fonemas consonánticos en coda a final de palabra si la siguiente sílaba es vocálica, de manera que se evitan

sílabas trabadas donde sea posible. De esta forma, si lo ponemos a *verdadero*, *dos uno* se interpretará como /doˈsu.no/ en lugar de /dosˈu.no/. Esto responde a una cuestión fonética más que fonológica y no afecta a la calidad de los fonemas ni a su posición fonémica absoluta, sino únicamente a su posición relativa respecto a la serie silábica. En ciertas circunstancias puede resultar útil, pero no lo consideraremos aquí.

En el diseño para el uso en un escenario real, hemos tenido que introducir asimismo modificaciones más prácticas. Una de ellas es la opción de elegir el carácter que representa el acento primario en la transliteración SAMPA. Esto responde a la necesidad de guardar en ocasiones los resultados en archivos en un archivo CSV, acrónimo de *Comma Separated Values*. Este formato permite organizar información al modo tabular de una hoja de cálculo en texto plano, de manera que cada línea corresponde a una fila, cuyas columnas se separa una coma. En caso de que los valores de un campo lleven comas, estos se entrecomillan para indicar que todo cuanto se encuentre entre las comillas ha de interpretarse literalmente. SAMPA emplea comillas dobles para señalar la posición de las sílabas tónicas, que, en un archivo CSV, se interpretan como una la apertura de una cadena de caracteres literal. Por esta razón, resulta necesario introducir un símbolo alternativo cuando se desea guardar información en un archivo de este tipo.

La implementación de Python también ofrece la posibilidad de decidir el comportamiento del programa ante una *s* líquida. Esta función responde a la frecuencia con que se dan eses líquidas en los textos, sobre todo en fragmentos latinos intercalados, lo que hace necesario, por lo menos, tener en consideración su existencia. Estos casos se resuelven reinterpretando la consonante como coda de una vocal /e/ protética.

La *s* líquida

Aunque en la implementación de Python damos la posibilidad de desactivar la funcionalidad de solucionar esto observando la fonotáctica española, para la explicación daremos por sentada esta estrategia. La *s* líquida es aquella en que /s/ aparece en ataque de la sílaba inicial seguida inmediatamente de consonante. Si bien era habitual en latín, desapareció en su evolución a algunas de las lenguas romances, el español entre ellas. Los préstamos que la emplean, se resuelven mediante un /e/ protésica ante la *s*. Este segmento gana carácter silábico, por lo que la sílaba inicial se desdobla (Clavería Nadal, 2018, p. 165).

¿De qué forma afecta este elemento al análisis de dramas del Siglo de Oro? Como decíamos, existen numerosas instancias de eses líquidas en los textos. Dado que los editores críticos de cada caso han optado por hacer una excepción en la modernización

ortográfica y conservar la ortografía original, cabe pensar que esta decisión podría responder a un criterio fonológico. Esto es, el verso *spíritu ha de lograrla* del Ejemplo 6 se resolvería conforme a la edición como /ˈspi.ɾi.tu ˈa de lo.ˈgɾaɾ.la/, que, tras encontrar los acentos prosódicos, quedaría como {*í*,*i*,*u*,*á*,*e*,*o*,*á*,*a*}. Por el contrario, si introducimos una *e* epentética, la transcripción será /esˈpi.ɾi ˈtwa de lo.ˈgɾaɾ.la/, sus núcleos vocálicos se organizarían de la manera {*e*,*í*,*i*,*á*,*e*,*o*,*á*,*a*} una vez hechas las sinalefas pertinentes. Estas dos soluciones producirían patrones rítmicos diferentes, `+--+--+-` y `-+-+--+-`.

La cuestión se torna más apremiante al considerar que, en muchas obras, se intercalan fragmentos latinos, a los que la ese líquida no es ajena, al contrario que sucede con su presencia en el español. ¿Debemos asumir que se conserva una ese líquida culta en las obras teatrales a imitación del modelo latino? Veamos los ejemplos que proporciona nuestro corpus de autos sacramentales, donde encontramos las siguientes instancias de ese líquida antecediendo a /p/ y /t/ o /k/:

(1) a. sin que una alba splendor de otra alba sea (Calderón, *A María el corazón*, v. 17)
b. del Spíritu) toda (v. 179)
c. el spíritu encomiendo. (v. 1296)
d. los spíritus (¡qué ahogo!) (v. 1319)
e. Ave, maris stella (v. 1481)
f. Spiritui Santo (v. 1517)

(2) a. Spíritu de Dios era (*El sacro Pernaso*, v. 246)
b. el dulce spiritual (v. 551)
c. del spiritual manjar (v. 883)
d. Spíritu de consuelo. (v. 1207)
e. In te, Domine, speravi, (v. 1237)
f. carear estos dos testos de Scritura, (v. 1510)
g. Yo un nuevo spíritu en este (v. 1984)

(3) In te, Domine, speravi, (*Santo rey don Fernando (II)*, vv. 2200, 2202)

(4) su splendor se presume (*La cura y la enfermedad*, v. 302)

(5) a la imaginada scena (*La redención de cautivos*, v. 354)

(6) spíritu ha de lograrla, (*Las espigas de Ruth*, v. 1738)

(7) te da el splendor divino (*Los alimentos del hombre*, v. 796)

Aunque la mayoría de los casos mostrados arriba permiten una resolución aceptable mediante poco más que jugar con las sinalefas y las diéresis, en (1b) necesitamos la epéntesis para alcanzar el heptasílabo, que es como los versos colindantes aconsejan escandirlo. Por ese motivo, resulta apropiado disponer de la opción de intercalar una vocal antes de la ese líquida.

Más dificultad plantea el caso de (2b). Aquí debemos contar ocho sílabas, por lo que, si hacemos la epéntesis, se plantea la dicotomía de no hacer sinalefa en *dulce espiritual*, forzando un hiato entre dos vocales iguales, ambas átonas, o recurrir a la diéresis en *espiritual*, lo que además sería su pronunciación regular. Esta última posibilidad se antoja más plausible, por lo que el verso no ofrece pistas sobre la epéntesis, pues no influiría en la resolución. En (2c) encontramos el caso contrario: si hacemos epéntesis, debemos diptongar *espiritual* para que el verso siga siendo octosílabo. El ritmo aporta poca ayuda, ya que ambas soluciones serían `----+--`. La solución no epentética '/del spi/ implicaría el préstamo fonológico latino. La otra ordenación posible, /dels pi/ también es ajena a la fonología española, donde /ls/ en coda se da exclusivamente en préstamos y nunca en interior de palabra.

En cuanto a la pronunciación latina, tenemos (1e), (1f) y (3). En los dos primeros casos nos hallamos ante hexasílabos cantados y ante un octosílabo en el último. El octosílabo aporta poco a la discusión, pues la sinalefa provocaría que el verso se resolviera exactamente igual con o sin epéntesis, por lo que discutiremos los otros dos. El ejemplo (1e) es un verso cantado por los músicos, y debe leerse en consonancia con la pronunciación latina para concordar con el verso posterior 8a, así como con los siguientes parlamento de los músicos (8b y 8c).

(8) a. Dei mater alma (Calderón, *A María el corazón*, v. 1482)

b. Atque semper virgo,
felix celli porta. (vv. 1485-1486)

c. Summens illud Ave
Gabrielis ore. (vv. 1489-1490)

El verso 1f es ambiguo, por prestarse a la resolución tanto mediante epéntesis como diéresis. En este último caso, no obstante, obligaría a acentuar la vocal cerrada, provocando un acento antirrítmico. Parece más probable que de nuevo se alcance el metro valiéndose de la epéntesis. De esta manera, la estrategia protésica se antoja la solución más cabal a este tipo de palabras, trátense de voces españolas o no.

No hay que descartar, sin embargo, que puedan darse casos en los que se prefiere la pronunciación latina. Véase por ejemplo el siguiente ejemplo en 9, que, aun sin proceder de una obra teatral, obliga a tomar precauciones:

(9) En Scilas ni en Caribdis no repara
(Cervantes, *Los trabajos de Persiles y Segismunda*, Lib. 1, cap. 9)

En situaciones como esta, posponemos la resolución hasta el ajuste métrico, dejando abierta la posibilidad de eliminar la epéntesis y emplear la pronunciación latina si lo demanda el verso para obtener el recuento silábico apropiado. No obstante, en el caso del teatro, en el que la declamación se aproxima más al habla corriente que en la lírica —sea siquiera porque la mayor cantidad de versos dificultan esmerarse tanto en cada uno como se haría, por ejemplo, con los catorce versos de un soneto—, cabría plantearse hasta qué punto el actor tendería a resolver un verso semejante forzando la pronunciación latina en lugar de añadir una sílaba supernumeraria atendiendo a la fonología española.

Preprocesado

El preprocesado tiene como meta eliminar, enmendar y expandir caracteres. Por una parte, elimina del texto todos aquellos caracteres no alfabéticos que se encuentren en él. Por otro lado, deletrea los nombres de las consonantes en caso de encontrar una como un grafema aislado, de manera que transcriba *letra b* no como /ˈletɾab/ sino como /ˈletɾa ˈbe/. Finalmente, translitera diacríticos ajenos al repertorio del español. Para llevar a cabo la tarea definimos la función que describimos en Algoritmo 10. Esta recibe como entrada el texto que deseamos preparar y lo devuelve preprocesado.

En primer lugar, definimos las equivalencias entre los grafemas consonánticos y su deletreo, lo que en Algoritmo 10 hemos llamado *letras*. La finalidad de esto es instruir al computador para tratar consonantes aisladas que encuentre en el texto, como en 10, pues la máquina ignora la pronunciación; si no le indicamos que haga una excepción, tomará la consonante aislada como un grafema y no como el nombre de una letra, por lo que la sustituirá por el fonema correspondiente. Salvo en el caso de las vocales, esto produciría palabras monofonemáticas consonánticas, ajenas por completo a la fonología española. Dicho de otro modo, si encuentra ⟨b⟩ sin contexto ha de tratarlo como ⟨be⟩ y si da con ⟨h⟩, como ⟨hache⟩. Para evitarlo, deletreamos el nombre de cada letra no vocálica cuando aparece una consonante aislada y hacemos una sustitución si es menester.

(10) El abecé del marido.
La A, quiere decir hacer
La B significa bobo
La C, callar; atended:
«hacerte bobo y callar» ,
¿qué os parece el abecé?

(Mira de Amescua, *Los pastores de Belén*, vv. 177-182)

Excluiremos no obstante ⟨y⟩, dado que en su valor de conjunción tiene pronunciación vocálica monofonemática. En el corpus no hemos encontrado ningún ejemplo de su uso literal como letra, valga la redundancia, mientras que su uso conjuntivo es extensivo, por lo que optamos por considerar únicamente la transcripción vocálica del grafema. Si ocurriera un caso que requiriese la transcripción /ˈi̯ɡɾje.ga/, la transcripción /i/ que empleamos obligaría a resolver el verso con entre una y dos sílabas de menos respecto a las reales, dependiendo de la distribución de las sinalefas, por lo que se evidenciaría la irregularidad del metro, lo que facilitaría sobremanera su localización y corrección manual.

Algoritmo 10: Módulo de preprocesado.

```
Clase Limpia(texto):
    letras ← {a, b, c, d, ...}
    deletreos ← f : {(grafema_1, deletreo_1), ..., (grafema_n, deletreo_n)}
    símbolos ← f : {símbolo_1, ..., símbolo_n}
    diacríticos ← f : {(diacrítico_1, transliteración_1), ..., (diacr._n, transl._n)}
    texto ← Minúscula(texto)
    para cada carícter ∈ deletreos hacer
        si letras_i ≯ carácter ≯ letras_j entonces
            carácter ← deletreos_carácter
    devolver texto
```

Algunos caracteres pueden transcribirse de forma directa a su equivalente fonológico. En concreto, esto es posible si grafema y fonema son homógrafos o si el fonema no tiene equivalente grafémico. Por eso transcribiremos directamente algunos grafemas a su fonema, mientras que otros han de conservar momentáneamente la representación grafémica para no alterar transformaciones posteriores. De esta manera, escribimos ⟨haʧe⟩, pues el grafema ⟨a⟩ existe como fonema y se corresponde biunívocamente con este. Por el contrario, el grafema ⟨r⟩ puede representar /ɾ/ o /r/ si está duplicado. Esto desaconseja alterarlo previamente. El grafema ⟨ch⟩ tiene también correspondencia biunívoca con el fonema /ʧ/. Al no existir el grafema ⟨ʧ⟩ en el repertorio del español, no hay reglas posteriores que le afecten y lo transcribiremos ⟨haʧe⟩, sin que afecte a

operaciones ulteriores. La *h* inicial, no obstante, hemos de conservarla para decidir más adelante si la marcamos como posible aspiración.

A continuación, creamos una lista de signos ortográficos de puntuación que llamaremos *símbolos*. Al carecer de equivalente fonemático, no se necesita definir una transformación para cada uno, pues todos ellos serán reemplazados por un elemento vacío. En previsión de diferentes estilos ortotipográficos —tanto aceptados como no—, debemos incluir todas las variaciones posibles. Por ejemplo, comillas españolas e inglesas, tanto rectas como rizadas, simples y dobles. También, grafías irregulares de la raya ortográfica (—) y el guion (-). Hemos comprobado que la fidelidad a las prescripciones académicas (RAE, 2010b, p. 373) varía notablemente de un texto a otro, por lo que, además de los susodichos símbolos, hemos incluido la raya intermedia[107] (–).

Seguidamente, tratamos grafemas ajenos a la ortografía española o aquellos que, formando parte del repertorio castellano, se hallan modificados por un símbolo diacrítico extraño a este. Para ello definimos de nuevo las equivalencias entre los caracteres con diacrítico foráneo y su transliteración según las normas de la ortografía española. No es raro hallar ediciones que emplean diacríticos y ligaturas de diversas lenguas, como el alemán (11a), el catalán (11b), el italiano en Lope (11e) o el latín (11c).

(11) a. El húngaro a Nördlingen
(Calderón, *El primer blasón de Austria*, v. 113)

b. ¿Què voleu? Les paradetas (*El pintor de su deshonra*, v. 1868)

c. mutans Evæ nomen (*A María el corazón*, v. 1494)

d. porque el gran Luis de Camões
(*A secreto agravio, secreta venganza*, v. 106)

e. Un seculo e più, segnora, (Lope de Vega, *El anzuelo de Fenisa*, v. 3054)

Estos símbolos diacríticos poco comunes pueden darse incluso en palabras patrimoniales españolas, sea por una ortografía arcaizante o por una cuestión editorial. De esta manera, dependiendo de la edición, se encuentran acentos circunflejos en sustitución de un dígrafo vocálico con reducción fonética (12a), el anticuado acento grave (12b) o la no menos antigua cedilla (12c), al lado de la habitual diéresis editorial mediante crema para señalar un hiato (12d).

[107] Aunque no es propia del español, se usa extensivamente en inglés (*en dash*) y en alemán (*Halbgeviertstrich*), por lo que no es extraño encontrar ocasionalmente este símbolo reemplazando la raya española en ediciones poco cuidadas.

(12) a. hoy vengo buscándôs: basta(Calderón, *Amar después de la muerte*, v. 61)
decirle que crêr en Dios. (v. 549)

b. Conmigo à un tiempo y con ella, (*El pintor de su deshonra*, v. 1868)

c. A ti te pareçe eso, (*El diablo mudo*, v. 1005)

d. te parece que Dïana, (*El verdadero Dios Pan*, v. 484)

La solución pasa por la transliteración, cuyo caso general consiste simplemente en eliminar el diacrítico. No obstante, debemos considerar algunas excepciones. El acento grave suele indicar una sílaba tónica tanto en otras lenguas como en grafías anticuadas del español, salvo en el caso de palabras monografémicas. Así pues, primero los grafemas ⟨à⟩ y ⟨ò⟩ por ⟨a⟩ y ⟨o⟩ cuando van orlados por espacios en blanco, y todos los demás casos por su equivalente con acento agudo.

Las ligaturas pueden o bien desdoblarse en dos grafemas, por lo que representarían dos sílabas ortográficas, o bien transliterarlas al grafema del repertorio español que represente el sonido más cercano. Tradicionalmente, ⟨æ⟩ ha dado el diptongo /je/ o /e/ (considérese *cielo* y *celestial*, ambas voces procedentes del latín *cælus*), como vimos en el ejemplo 8b.

En caso de encontrar diéresis, diferenciaremos entre las editoriales y las ortográficas. Para las primeras, sustituimos por el grafema sin símbolo diacrítico y añadimos un guion bajo precediendo a la vocal para señalar más tarde al módulo de división silábica que ahí ha de romper el diptongo. Finalmente, reemplazamos el grafema ⟨ç⟩ por /θ/, pues su pronunciación anterior /t͡s/ debía de haber completado la desafricación ya en la segunda mitad del siglo XVI (Lapesa, 1981/2008, sec. 92.4). En caso de que la diéresis proceda de un extranjerismo, se trata indistintamente, asumiendo que se translitera como una de las cinco vocales españolas y no como un diptongo o un fonema ajeno a la fonología española.

Ahora ya disponemos de las descripciones de los elementos que emplearemos para hacer las sustituciones. Por una cuestión de eficiencia, convertimos el texto a minúsculas antes de empezar a sustituir. De otra manera, estaríamos obligados a evaluar cada vez si estamos ante una variación mayúscula o minúscula de una misma letra. Esto duplicaría el tiempo de la operación, lo que podría resultar perceptible cuando se procesa una gran cantidad de texto, como es frecuente en nuestro caso. Con el texto en minúscula, cada letra está representada por un único grafema y no se da este problema.

La primera de las series de sustituciones consiste en recorrer las claves de *letras* y, si encuentra una ocurrencia de una de ellas que no esté orlada a izquierda y derecha por

otras letras, estamos ante una consonante aislada, por lo que procedemos a sustituir las ocurrencias de esa letra solitaria en el texto por el valor de la clave correspondiente.

$$x \longrightarrow f(x) \ / \ \emptyset_ \ \emptyset$$

A continuación recorremos la lista de símbolos y sustituimos las ocurrencias de sus elementos en el texto por un espacio en blanco. De esta manera tenemos únicamente palabras.

$$x \longrightarrow \emptyset$$

Para finalizar introduciremos los casos de *e* epentética. Sustituimos ⟨s⟩ en posición inicial de palabra siempre que no vaya seguido por una vocal.

$$\mathrm{s} \longrightarrow \mathrm{e,s} \ / \ \emptyset_\langle \neg VOC \rangle$$

Tras realizar estas operaciones, el método devuelve la cadena de caracteres transformada.

Excurso V: Reordenación

El segundo de los métodos de preprocesado lo contemplamos como idea para la traducción computacional de los algoritmos vistos aquí. Se emplea si el argumento para reordenar del programa de Python se ha puesto deliberadamente a *verdadero*. Para la escansión no lo utilizaremos, salvo que necesitemos afinar con una agrupación silábica fonológica, ya que los núcleos vocálicos permanecen constantes. El objeto es, pues, proporcionar unas herramientas tan flexibles como podamos producir y, para ello, como en este caso, intentamos adelantarnos a las necesidades de los hipotéticos beneficiarios de la biblioteca.

Algoritmo 11: Módulo de reordenación.

```
Función Reordena(texto):
    vocales ←⟨vocales⟩
    para cada posición_i ∈ texto hacer
        cuenta ←|posición_i|
        si i ≠ 0 ∧ |posición_{n−1}| > 1 entonces
            si posición_i empieza con vocales entonces
                si posición_{i−1} ¬acaba con vocales entonces
                    posición_i ← posición_{i−1,n} + posición_i
                    posición_{i−1} ← posición_{i−1} − posición_{i−1,n}
    devolver texto
```

En Algoritmo 11 vemos cómo se lleva a cabo. Tomamos el texto y empezamos a recorrerlo por sus sílabas. Queremos comprobar si la sílaba correspondiente a cada

iteración no es la primera del texto, esto es, si existe una sílaba precedente plurifonemática cuya coda es redistribuible. De cumplirse todas las condiciones, nos aseguramos de que la sílaba que estamos examinando empiece por vocal y la anterior acabe por consonante, en cuyo caso procederemos a reubicar la última consonante de la coda de la sílaba precedente en el ataque de la sílaba actual.

Transcripción fonológica

Con el texto de entrada dispuesto, se aborda la transcripción fonológica. Esta se produce mediante las tablas y reglas de sustitución del método `TranscripciónFonológica`, cuyas primeras líneas vemos en el Algoritmo 12. Toma como argumento una variable *texto*, además de heredar una instancia de la clase `TranscripciónFonológica`. El argumento *texto* es una copia del texto a transcribir.

Algoritmo 12: Método de transcripción fonológica.

Método `TranscripciónFonológica(`*texto*`):`
- diacríticos $\leftarrow f : \{diacrítico_1 : transliteración_1), ..., diac._n : transl._n\}$
- consonantes $\leftarrow f : \{grafema_1 : fonema_1, ..., grafema_n : fonema_n\}$
- texto $\leftarrow$ `SustituyeR(`*texto*`)`
- **si** *aspiración* **entonces**
  - texto$(\langle h \rangle) \leftarrow marca, \forall \emptyset \succ h \in texto$
- texto$(consonantes_c) \leftarrow consonantes_v, \forall consonantes_c \in texto$

Antes de empezar a modificar el texto, definimos una tabla de sustitución de grafemas con diacrítico y su equivalente sin este. Para ello, nos valemos de un *diccionario*, como especificamos el tipo de datos consistente en un conjunto de pares *clave*, *valor*. La clave nombra al elemento y el valor, como su propio nombre indica, es el contenido que asignamos al elemento nombrado por la clave. De esta manera, creamos claves con todos los grafemas vocálicos con símbolo diacrítico y les damos como valor su equivalente sin diacrítico. Por ejemplo, $\{\acute{a} : a, \grave{a} : a, ..., \ddot{u} : u\}$. Se incluye de manera adicional la raya baja, pues la emplearemos para señalar hiatos en caso de diéresis. Al no ser esta más que una marca diacrítica, no le corresponde fonema alguno.

Definimos un segundo diccionario para los sonidos consonánticos. Este también tiene un grafema o una combinación grafémica como claves, pero sus valores son sus respectivos fonemas equivalentes en la forma $\{v : b, z : \theta, qu : k, ...\}$. Al contrario que el diccionario anterior, este contiene combinaciones fonológicas para tener en cuenta el contexto, de modo que crearemos dos claves distintas *ce* y *ca*, a las que asignaremos además valores diferentes θe y *ka*. Las claves deben seguir un orden específico que refleje cómo deben ser aplicadas las transformaciones. Por ejemplo, en el caso

de ⟨j⟩, x debe preceder a j en el diccionario. De lo contrario (13), estaríamos ante una transformación errónea, pues si transforma primero j, al llegar a la siguiente, se encontraría con caracteres x que corresponderían tanto al grafema ⟨x⟩ como al fonema /x/.

(13)

entrada:	a	j	o
j ⟶ x		↑	
	a	x	o
x ⟶ ks		↑	
salida:	a	ks	o

Las sustituciones comienzan por el grafema ⟨r⟩ cuando le corresponde el fonema /r/. Dado que el mismo símbolo se emplea para el grafema correspondiente a /ɾ/, sustituimos los grafemas de /r/ por el carácter `R`, esto es, detrás de ⟨n⟩, ⟨n⟩ o ⟨s⟩ o a comienzo de palabra, o si ⟨r⟩ aparece duplicado.

$$\mathrm{r} \longrightarrow \mathrm{R} \;/\; \left\{ \begin{matrix} \mathrm{n} \\ \mathrm{l} \\ \mathrm{s} \\ \emptyset \end{matrix} \right\} _$$

$$\mathrm{r}, \mathrm{r} \longrightarrow \mathrm{R}$$

A continuación comprobamos si existen instancias de del grafema ⟨h⟩ a comienzo de palabra. De ser así, las sustituimos por una marca en lugar de hacerlas desaparecer como las interiores. Esto tiene por objetivo advertir de la presencia de esta hache, de manera que pueda tenerse en cuenta después a la hora de predecir una potencial aspiración. Seguidamente, recorremos las claves del diccionario *consonantes* y, cuando alguna de ellas aparece en el texto, sustituimos todas sus instancias por el valor de la clave correspondiente. Esto, sin embargo, no es suficiente para tratar las transformaciones en su totalidad porque ⟨y⟩ y ⟨g⟩ presentan particularidades de tal cariz que hacen necesario un tratamiento específico (Algoritmo 13).

Algoritmo 13: Caso de ⟨y⟩.

texto(⟨*y*⟩) ←/*i*/, ∀∅ ≯ ⟨*y*⟩ ≯ ∅ ∈ *texto*
texto(⟨*y*⟩) ←/*j*/, ∀⟨*y*⟩ ≯ ∅ ∈ *texto*
texto(⟨*y*⟩) ←/ʝ/, ∀⟨*y*⟩ ∈ *texto*
texto(*diacríticos*$_v$ + /ʝ/) ←*diacríticos*$_c$/*i*/,
    ∀*diacríticos*$_v$ + /ʝ/ ≯ ∅, *diacríticos*$_v$ ∈ {⟨*á*⟩, ⟨*é*⟩, ⟨*í*⟩, ⟨*ó*⟩, ⟨*ú*⟩}
consonantes ← *f* : *consonantes*
texto(/ʝ/) ←/*i*/, ∀/ʝ/ + *consonantes*$_n$ ∈ *texto*

En el caso de que detectemos que el texto contiene el grafema ⟨y⟩, se sustituyen todas las ocurrencias aisladas de esta, correspondientes a la conjunción, por /i/. Si aparece a final de palabra, se considera como semiconsonante parte de un diptongo decreciente, por lo que se reemplaza por /j/. En los demás casos, se cambia a /ʝ/.

$$\mathrm{y} \longrightarrow \mathrm{i} \;/\; \emptyset_\emptyset$$

$$\mathrm{y} \longrightarrow \mathrm{j} \;/\; _\emptyset$$

$$\mathrm{y} \longrightarrow \mathrm{ʝ}$$

El paso siguiente consiste en tomar las claves con acento agudo del diccionario *diacríticos* para sustituir su valor —esto es, la vocal sin acento— en el texto por la vocal acentuada si la vocal precede a a /ʝ/ a final de palabra.

$$\left\{ \begin{matrix} \mathrm{a} \\ \mathrm{e} \\ \mathrm{i} \\ \mathrm{o} \\ \mathrm{u} \end{matrix} \right\}, \mathrm{ʝ} \longrightarrow \left\{ \begin{matrix} \mathrm{á} \\ \mathrm{é} \\ \mathrm{í} \\ \mathrm{ó} \\ \mathrm{ú} \end{matrix} \right\}, \mathrm{i} \;/\; _\emptyset$$

Para finalizar el tratamiento excepcional de ⟨y⟩, se comprueba que no se ha introducido accidentalmente /ʝ/ seguido de consonante y, de ser así, se reemplaza por ⟨i⟩.

$$\mathrm{ʝ} \longrightarrow \mathrm{i} \;/\; _\langle +CONS \rangle$$

El segundo grafema que tratamos de manera excepcional es ⟨g⟩, ya que este puede representar los fonemas /g/ y /x/ dependiendo del contexto. Para ello, confirmamos en primer lugar si este aparece en *texto* para, de darse tal circunstancia, emprender los pasos oportunos para su tratamiento (Algoritmo 14).

Algoritmo 14: Caso de ⟨g⟩.

débiles $\leftarrow f : \{e, i\}$
fuertes $\leftarrow f : \{a, o, u\}$
si $\langle g \rangle \in texto$ **entonces**
  texto$(\langle g \rangle + d) \leftarrow \langle x \rangle + d, \forall \langle g \rangle + d \in texto, \forall d \in débiles$
  texto$(\langle gu \rangle + d) \leftarrow \langle g \rangle + d, \forall \langle g \rangle + d \in texto, \forall d \in débiles$
  texto$(\langle gü \rangle + d) \leftarrow \langle gw \rangle + d, \forall \langle gü \rangle + d \in texto, \forall d \in débiles$
  texto$(\langle gu \rangle + f) \leftarrow \langle gw \rangle + f, \forall \langle gu \rangle + f \in texto, \forall f \in fuertes$

En este caso, se procede primero a asignar el fonema /x/ a los grupos ⟨ge⟩ y ⟨gi⟩ y /g/ a ⟨gue⟩ y ⟨gui⟩. Consideramos más allá de las formas canónicas de ⟨e⟩ e ⟨i⟩ e incluiremos sus variaciones diacríticas.

$$\text{g} \longrightarrow \text{x} \;/\; _\left\{\begin{matrix}\text{e}\\ \text{i}\end{matrix}\right\}$$

$$\text{g,u} \longrightarrow \text{g} \;/\; _\left\{\begin{matrix}\text{e}\\ \text{i}\end{matrix}\right\}$$

Para completar ⟨g⟩, abordamos los casos en los que va seguido de ⟨u⟩ con o sin diéresis.

$$\text{g, ü} \longrightarrow \text{g, w} \;/\; _\left\{\begin{matrix}\text{e}\\ \text{i}\end{matrix}\right\}$$

$$\text{g,u} \longrightarrow \text{g, w} \;/\; _\left\{\begin{matrix}\text{a}\\ \text{o}\end{matrix}\right\}$$

Con esto habríamos terminado prácticamente de transcribir los caracteres grafémicos en fonológicos. No obstante, quedan algunos diacríticos vocálicos que necesitamos para dividir el texto en palabras, que es precisamente de lo que se ocupan las líneas finales del método (Algoritmo 15). Esta división pasa por invocar al método `DivideVariables` (Algoritmo 16), pasándole como argumentos *texto*. Este crea una instancia de la clase `Valores`, que asignamos a la variable *transcripción* —enseguida entraremos en detalle en su modo de operar—. Para finalizar, como ya no son necesarios los símbolos diacríticos una vez termina la división silábica, los sustituimos por las formas canónicas en *transcripción* antes de devolver este objeto como salida.

Algoritmo 15: Caso de ⟨y⟩.

```
transcripción ←DivideVariables(texto)
transcripción(diacríticos_c) ←diacrticos_v
devolver transcripción
```

8.3 Métodos auxiliares de la transcripción

División entre palabras y sílabas

Como hemos dicho, para transcribir fonológicamente empleamos el método `DivideVariables`. Este, a su vez, delega algunas de sus tareas en otros. Veamos primero el mecanismo general del método y, cuando lo tengamos, entremos en sus componentes, de una manera parecida a lo que acabamos de hacer con `TranscripciónFonológica`. Primero de todo, definimos las variables *palabras* y $sílabas_{texto}$, que son las listas donde iremos almacenando todas las palabras y las sílabas del texto —debemos diferenciarla de la variable $sílabas_{palabra}$, en la que almacenaremos las sílabas de una

palabra cada vez—. Hecho esto, empezamos a iterar por los componentes que resultan de dividir el texto en una lista con las palabras de las que se compone.

Comprobamos si la palabra que estamos evaluando es un adverbio en *-mente*. Para ello, verificamos en primer lugar que esa sea efectivamente su terminación y que tenga más de cinco letras. Si este es el caso, guardamos en las variables *sílabas* y *tónica* los resultados del método `Silabación` si le pasamos la raíz de la palabra, —esto es, excluyendo *-mente*—. Trataremos `Silabación` por ahora como una caja negra, en la que introducimos la palabra por un extremo y sacamos del otro sus sílabas, sin preocuparnos de lo que sucede internamente para que esto ocurra; dejaremos para más adelante la descripción analítica de su mecanismo (Apartado 8.4). Retomando el hilo, si la palabra es *claramente*, aplicamos la silabación a *clara*. Con las sílabas almacenadas en *sílabas*, evaluamos el tamaño de esta variable.

Algoritmo 16: Creación de variables para sílabas y palabras.

Función `DivideVariables`(*texto*)**:**
- palabras ←{}
- $\texttt{sílabas}_{texto}$ ←{}
- **para cada** *palabra* ∈ *texto* **hacer**
  - **si** *palabra* **acaba con** ⟨*mente*⟩ ∧ |*palabra*| > 5 **entonces**
    - sílabas, tónica ←`Silabación`(*raíz*)
    - **si** $|sílabas_{palabra}| > 1$ **entonces**
      - $\texttt{sílabas}_{palabra} \leftarrow sílabas_{palabra} + \{'men,te\}$
      - tónica ← *tónica* − 2
  - **si no**
    - slbs, tónica ←`Silabación`(*palabra*)
  - $\texttt{sílabas}_{palabra}$ ←`Diptonga`$(palabra, sílabas_{palabra})$
  - $\texttt{sílabas}_{palabra^{+\text{TON}}} \leftarrow \langle' \rangle + sílabas_{palabra^{+\text{TON}}}$
  - palabra ←∅
  - $\texttt{sílabas}_{texto} \leftarrow ílabas_{texto} + sílabas_{palabra}$
  - palabras ←$palabras + palabra$
- **devolver** `Valores`$(palabras, sílabas_{texto})$

Si el resultado tiene más de una sílaba, dejamos el acento secundario y les añadiremos las sílabas de *mente* también con su correspondiente acento primario. La posición del acento secundario la hallaremos restando las dos sílabas de *mente* a *tónica*, lo que dará la posición contando desde la última sílaba hacia la primera. Por el contrario, si la raíz es monosilábica, asumimos que la palabra es llana y la resolvemos concatenándole $\{'men,te\}$ con un único acento, el primario, e indicando que este se ubica en la penúltima sílaba. Si, por el contrario, no nos encontramos ante una palabra acabada en *-mente*, hallamos su división silábica de manera directa y almacenamos las sílabas sin modificar en $sílabas_{palabra}$ y la posición de la sílaba tónica en *tónica*.

Definimos ahora la identificación de los diptongos. Para abordarlos, se llama al método `Diptongos` —que veremos en detalle después (Algoritmo 17)—, con *palabra* y $sílabas_{palabra}$ y como argumentos. Este devuelve una lista de sílabas reevaluadas en función de las reglas de diptongación, que se asignan a *sílabas*, reemplazando su valor previo. La variable *tónica*, obtenida anteriormente, se emplea para marcar la sílaba donde recae el acento en la posición correspondiente de la variable *sílabas*. Como el contenido de *palabra* ya no es necesario en esta iteración, se reinicializa para volver a usar la variable de nuevo en la próxima palabra del texto[108].

El bucle que recorría las palabras termina añadiendo al final de $sílabas_{texto}$ los elementos de $sílabas_{palabra}$ y *palabra* tras el último elemento de *palabras*. Hecho esto, la función devuelve un objeto de la clase `Valores`, cuyos atributos son *palabras* y $sílabas_{texto}$.

Diptongos

El último de los métodos auxiliares es `Diptonga` (Algoritmo 17), como su propio nombre indica, es el encargado de abordar las uniones vocálicas intrasilábicas. Este recibe el argumento *sílabas*. Nótese que su función no es determinar la división silábica, de lo que se encarga el módulo externo de silabación, sino de sustituir los fonemas correspondientes a un vocoide por los símbolos adecuados, según les corresponda, para eliminar ambigüedades. Este pequeño y sencillo paso desempeña un papel crucial, pues la corrección del resultado determinará más adelante el análisis de ciertos fenómenos prosódicos.

Algoritmo 17: Manejo de diptongos.

Función `Diptonga(`*sílabas*`):`
  para cada $sílaba_n \in sílabas$ **hacer**
    si $vocales_n \succ débil \wedge débil \succ vocales_n$ **entonces**
      $\mathsf{sílaba}_{n,\text{débil}} \leftarrow \mathsf{sílaba}^{-\text{SIL}}_{n,\text{débil}}$
  devolver *sílabas*

El procedimiento es sencillo: recorremos todas las sílabas y, si encontramos dos vocales seguidas, intentamos sustituir la vocal alta por su equivalente semivocálico o semiconsonántico. Primero probamos por la izquierda, en consideración de la tendencia del español a los diptongos crecientes (Alarcos Llorach, 1964; Navarro Tomás,

[108] En la demostración en Python, se quitarían las marcas acentuales de los monosílabos en función de lo que se le haya indicado explícitamente al programa.

2004; Quilis, 1993/2019), y después por la derecha. Hechas las sustituciones en la variable *sílaba*, esta reemplaza el valor inicial.

$$\left\{ \begin{matrix} \text{i} \\ \text{u} \end{matrix} \right\} \longrightarrow \left\{ \begin{matrix} \text{j} \\ \text{w} \end{matrix} \right\} \; / \; _ \langle +VOC \rangle$$

$$\left\{ \begin{matrix} \text{i} \\ \text{u} \end{matrix} \right\} \longrightarrow \left\{ \begin{matrix} \text{j} \\ \text{w} \end{matrix} \right\} \; / \; \langle +VOC \rangle _$$

Transcripción fonética

Aunque la transcripción fonética no es completamente imprescindible para llevar a cabo los pasos posteriores en el análisis métrico, hemos incluido un modelo de esta por completitud. No nos detendremos, por tanto, a examinar con detenimiento cada detalle, pero sí pretendemos mostrar la idea general subyacente. Como se ve, `TranscripciónFonética` es menos complejo que su equivalente para la transcripción fonológica. Esto se debe, en gran medida, a que el transcriptor fonético se basa en este, pues el único parámetro que admite es el objeto de transcripción fonológica que devuelve (Algoritmo 18). En realidad, apenas se ocupa de hacer de intermediario entre un objeto de la clase `Valores` y la función de sustitución más detallada `SustituciónFonética` que veremos de inmediato (Algoritmo 19). En concreto, `TranscripciónFonética` separa las dos listas que conforman un objeto de la clase *fonología* y las convierte a cadenas de caracteres para, seguidamente, crear una nueva instancia de la clase `Valores` con los resultados de tratar cada una de esas cadenas de caracteres con el método `SustituciónFonética`.

Algoritmo 18: Transcripción fonética.

```
Método TranscripciónFonética(fonología):
    palabras ←SustituciónFonética(Une(fonología.palabras))
    sílabas ←SustituciónFonética(Une(fonología.sílabas))
    devolver Valores(palabras, sílabas)
```

Las sustituciones comienzan de forma secuencial en *palabras*. Reemplazamos todos los fonemas oclusivos sonoros por sus respectivos alófonos aproximantes —esto es, [ß], [ð] [ɣ]—, por ser esta su realización más habitual. Seguidamente, aquellos casos en los que vayan precedidos de una consonante nasal o pausa, en el caso de /d/ también lateral, se asigna el alófono oclusivo. Después de esto, se sonorizan las fricativas si van precedidas de consonante sonora. Para finalizar, se define el diccionario *alófonos* de entornos fonémicos y su realización fónica, con pares como $\{(nb,mb), (nf,ɱf), (nk,ŋk), ...\}$. Se recorre el diccionario y, cada vez que se encuentra una ocurrencia de alguna clave del diccionario, se sustituye esta por su valor. Si

encontramos nb, lo sustituiremos por mb, si hallamos nf, lo cambiaremos por $ɱf$, y así de manera sucesiva hasta completar todas las sustituciones posibles.

Algoritmo 19: Sustituciones fonológico-fonéticas.

Función `SustituciónFonética(`*transcripción*`):`
  transliteración(i) $\leftarrow alófonos_i, \forall i \in transcripción$
  alófonos $\leftarrow f : \{(entorno_1, alófono_1), (entorno_2, alófono_2),...\}$
  transliteración(j) $\leftarrow alófonos_j, \forall j \in transcripción$
devolver *transcripción*

Con esto tenemos la transcripción fonética codificada en una cadena de caracteres con las sílabas separadas por guiones, por lo que ya solo resta crear una lista con sus sílabas.

Excurso VI: Transliteración AFI-SAMPA

Para permitir la utilización del transcriptor en otros entornos, hemos incluido en la realización para Python[109] una función para traducir de AFI a SAMPA (Algoritmo 20). Aunque no empleamos esta última notación para desarrollar los algoritmos, la transformación ilustra como se traduce entre diferentes formatos, por lo que hemos considerado que tiene sentido incluirlo aquí.

Algoritmo 20: Transliteración IPA-SAMPA.

Función `TransliteraciónSAMPA(`*transcripción*`):`
  transliteración $\leftarrow f : \{(ipa_1, sampa_1), (ipa_1, sampa_1), ...\}$
  transcripción(i) $\leftarrow alófono_i, \forall i \in transcripción$
  alófonos $\leftarrow f : \{(entorno_1, alófono_1), (entorno_2, alófono_2),...\}$
  transcripción(j) $\leftarrow entorno_j, \forall j \in transcripción$
  devolver *transcripción*

La piedra angular de este método es el diccionario *transliteración*. Sin seguir un orden concreto, no es más que una serie de fonemas cuya representación en el AFI difiere de SAMPA. Usamos los primeros como clave y les asignamos como valor su equivalente SAMPA. Para acabar, se recorre el diccionario haciendo las sustituciones en el texto en instancias que se correspondan con claves del diccionario.

[109] En ocasiones, los algoritmos aquí mostrados requieren una adaptación adicional al computador por motivos más técnicos que lógicos. Por ejemplo, la traducción a Python para la demostración requirió asignar a la variable *ipa* copias explícitas de las variables. Los atributos de la clase reaccionaban no como los valores de las variables sino punteros a dichas variables. Por lo tanto, si no se hacía esta asignación de copias, el método terminaba modificando los valores del atributo de la clase que le pasamos como argumento, con el resultado de que perdíamos la transcripción fonética en el AFI.

8.4 Módulo silabeador

Sobre la división silábica

La división silábica se lleva a cabo mediante un módulo externo específico para la tarea. Como vimos cuando hablábamos de modularidad, esto permitiría reemplazarlo por otro que realizara la misma función de diferente forma. Al contrario que la biblioteca de transcripción fonológica, que era parte integral del módulo de escansión y se separó posteriormente de este, la biblioteca de división silábica se concibió desde el origen como una pieza aparte. El motivo es que, en principio, consideramos delegar la tarea en una biblioteca externa. Sin embargo, durante las primeras pruebas, se observa que las herramientas disponibles no tienen en consideración las excepciones vistas en Apartado 4.7, lo que redunda en la precisión general. Por ese motivo, decidimos crear nuestro propio módulo de silabación que tuviera en cuenta dichas irregularidades. Demostró ser una decisión afortunada, a la luz de la evaluación de la precisión (Apartado 8.4).

Dado que las aplicaciones del módulo no se limitan a este trabajo en concreto, tiene sentido distribuirlo como una biblioteca aparte sin obligar a usarla junto al transcriptor. Por este motivo, en la implementación de Python introdujimos la opción de modificar el comportamiento de serie, de manera que la silabación pueda adaptarse a distintas necesidades. Esta versión para usos diversos (Apartado B.2) se encuentra disponible en línea tanto el paquete listo para su instalación directa como su código fuente (Sanz-Lázaro, 2023d). La instalación, al igual que la de Fonemas, se hace de manera sencilla mediante el gestor de paquetes de Python.

Funcionamiento práctico del transcriptor

Podemos ver el resultado más en detalle en el código, empezando por Algoritmo 21. La biblioteca es autosuficiente hasta cierto punto, pues, incluso en su versión de Python, se limita a importar la biblioteca estándar de expresiones regulares. La clase de silabación propiamente dicha toma una palabra y genera dos atributos, *sílabas* y *tónica* mediante la aplicación secuencial de diversos métodos a la palabra que hemos introducido. Aquí debemos definir también dos categorías, *vocales*, que contiene fonemas[110] vocálicos con todas sus variedades diacríticas posibles, y *cerradas*, con una lista de las vocales altas, así como sus respectivos alófonos no silábicos[111].

[110]La implementación en Python da la opción de aplicar la división silábica a textos grafémicos, por lo que, además, incluiría mayúsculas y sus variantes puntuadas. El computador las considera caracteres diferentes y, de no indicarle de manera explícita que las incluya, no las tendría en cuenta.

[111]Por claridad, en los bloques algorítmicos no haremos distinción en los vocoides entre semivocales y semiconsonantes, por lo que usaremos indistintamente /w/ y /j/ para representar ambos alófonos.

Tomamos *palabra* y la modificamos para marcar en el texto excepciones de forma explícita. En la implementación en Python, como con las opciones de la transcripción, también puede desactivarse esta función de manera que la división silábica ignore las excepciones y las trate atendiendo a su ortografía.

Primero, `HacerExcepciones` señala las excepciones descritas en Apartado 4.7 para que se tengan en cuenta en la silabación. El resultado volvemos a modificarlos mediante `Latín`. Este método se encarga de buscar términos latinos con morfemas incompatibles con las reglas de acentuación del español.

Tenemos el texto preparado, de manera que está marcado explícitamente dónde ha de hacerse una excepción al dividir las sílabas y se ha añadido acentuación explícita a algunos latinismos. Aplicaremos el método de división silábica `Silabea` y almacenaremos el resultado en el atributo *sílabas*. Procesando este atributo mediante el método `EncuentraTónica`, hallamos la sílaba acentuada, cuya posición asignamos al atributo *tónica*.

Algoritmo 21: Cabecera del módulo silabeador.

```
Clase Silabación(palabra):
    vocales ←f : {a,e,i,o,u, á, é, ...}
    cerradas ←f : {i,u, j, w}
    palabra ← HacerExcepciones(palabra)
    palabra ← Latín(palabra)
    slbs ← Silabea (palabra)
    tónica ← EncuentraTónica (sílabas)
```

Veamos como el módulo lleva a cabo el tratamiento de las excepciones. El método `HacerExcepciones` toma primero las palabras ambiguas (Domínguez Caparrós, 2012) y las divide. Nótese que en la implementación física hay que indicarle de forma explícita al módulo que haga esto, pues tiene sentido solamente si, como en nuestro caso, estamos trabajando con realizaciones fonológicas y no ortográficas y, después tenemos dispuesto un mecanismo para hacer una sinéresis cuando no se requiera el hiato.

Tras esto, el método se vale de una lista externa de términos que requieren tratamiento individualizado y la manera en que este ha de hacerse. En la implementación de Python, lo hacemos mediante expresiones regulares (ver Algoritmo 22). Emplear una lista como archivo separado tiene la ventaja de que permite añadir o eliminar excepciones rápidamente sin modificar el programa, tan solo editando el fichero que contiene sus descripciones. Veámoslo con un ejemplo real tomado del archivo que

empleamos en la versión de Python, en concreto, la parte que se ocupa de los verbos en *-uir*.

```
# Infinitivo
([^fqg]|\b)ui([rr])\b \1u_i\2
#Participio
([^fqg]|\b)uid([oa]s{,1})\b \1u_id\2
# Presente
([^fqg]|\b)ui(mos|s)\b \1u_i\2
# Indefinido
([^fqg]|\b)u(i|iste|isteis)\b \1u_\2
# Futuro !
#([^qg]|\b)ui([rr])(é|ás|á|emos|éis|an)\b \1u_i\2\3
# Condicional !
#([^qg]|\b)ui([rr])í(a|as|amos|ais|an)\b u_\1i\2í\3
```

En este fragmento vemos que hay líneas que comienzan con una almohadilla (#). Estas líneas son ignoradas, por lo que las usaremos para escribir comentarios, como en los que indicamos los tiempos verbales tratados, o para ignorar reglas, como hacemos en el ejemplo con el condicional y el futuro (Quilis, 1993/2019, pp. 184-186, ver Apartado 4.7). En las líneas sin comentar, tenemos dos campos separados por un espacio. En el de la izquierda indicamos la cadena que buscamos mediante expresiones regulares y, en el de la derecha, la sustitución. Los grupos encerrados entre paréntesis se *capturan*, y podemos referenciarlos después durante el reemplazo. Por ejemplo, en el caso del infinitivo que aparece en la primera fila, tenemos la expresión regular `([^fqg]|\b)ui([rr])\b`, cuyas coincidencias queremos permutar por `\1u_i\2`. Esto es, disponemos de la secuencia de o bien un fonema distinto de /f/, /q/ o /g/ (`[^fqg]` —queremos romper el diptongo en *huimos*, pero no *fuimos* y, en el caso de un texto grafémico, no tendría sentido hacerlo en *seguimos* o *delinquimos*, pues ⟨u⟩ no tiene ahí contrapartida fonológica— o bien el límite de palabra (`\b`), seguido de /ui/ y este de o bien de /ɾ/ o bien ⟨r⟩[112] antes del límite de palabra. En la sustitución, indicamos que ha de poner el primer grupo capturado, seguido de `u_i` y del segundo grupo. Formalmente, lo expresamos de la siguiente manera:

$$\mathrm{u, i, } \longrightarrow \mathrm{u, _, i} \; / \; \mathrm{_} \left\{ \begin{smallmatrix} \mathrm{r} \\ \mathrm{ɾ} \end{smallmatrix} \right\}, \emptyset$$

En la lista de excepciones incluimos las formas de los verbos en *-iar*, *-uir* y *-uar*, excepto los terminados en *-cuar* y *-guar*. No hacemos la salvedad de Quilis (1993/2019,

[112] Dado que la implementación digital del módulo silabeador se ha concebido para poder ser usada también con textos grafémicos, las expresiones regulares de la lista incluyen caracteres correspondientes a fonemas y grafemas indistintamente cuando es posible.

p. 185) para el condicional y el futuro, pues las pruebas empíricas con espectrogramas que hemos realizado no son concluyentes. También incluimos los adjetivos en *-uoso* y sus derivados, así como las voces individuales y sus flexiones indicadas en Apartado 4.7.

Algoritmo 22: Procesamiento de las excepciones.

Método `HacerExcepciones(`*palabra*`):`
  `CargarLista(`*lista*`)`
  palabra(i) $\leftarrow lista_i, \forall i \in palabra$ **devolver** *palabra*

En la versión de Python, la carga de la lista es un poco más elaborada de lo que mostramos aquí, pues, después de guardar el archivo en una variable, tenemos una larga cadena de texto. Sin embargo, como hemos visto, hay una sola regla por línea, por lo que conviene dividirlo y crear una lista de casos. Por eso, lo siguiente que hacemos es reemplazar cada elemento de la lista por sus dos componentes, excepto si dicha línea está vacía o comienza con un símbolo de almohadilla, en cuyo caso no la incluiríamos. De esta manera, ahora disponemos de una lista de duplas cuyos elementos son ambos miembros de expresión de sustitución. Restaría recorrer la lista e ir aplicando las sustituciones una detrás de otra.

Latinismos homógrafos

Hemos hallado una circunstancia en el corpus de prueba que no encuentra respuesta en las poéticas y requiere también un tratamiento excepcional. O, mejor dicho, las poéticas *españolas* no dan la respuesta. Se trata de los latinismos intercalados en su ortografía original. Aunque muchas veces pueden analizarse con normalidad para obtener un resultado aceptable, este no es el caso de determinadas declinaciones adjetivo-nominales, algunas de cuyas flexiones, como sabemos, son en -um, -ōrum y, en el de la tercera, también en -em. A esto hay que añadir, en el sistema verbal, las voces pasivas en -ur. Según las reglas de acentuación del español, estas palabras serían agudas, pues no acaban en *n*, *s* o vocal y tampoco llevan tilde, pero vemos que este no es realmente el caso, como demuestra el Ejemplo 14.

(14) a. *et omne factum est ita* (Calderón, *El divino Orfeo*, v. 80)
b. *Tantum ergo Sacramentum.* (*La vacante general*, v. 1278)
c. *Te Deum laudamus* (*El sacro Pernaso*, vv. 1170, 1192, 1208, 1239)
d. *Te, Dominum, confitemur...* (vv. 1171, 1193, 1209, 1240)
e. *non confundar in eternum.* (v. 1238)
f. *Adsum*, y pues es decente (v. 1629)

Lo que planteamos en Apartado 4.8 es también válido para esta situación. Podemos usar diccionarios o reglas. En cuanto a los diccionarios, es posible incluir en `exceptions.lst` tantos casos como estimemos, incluyendo la división silábica además de la acentuación. En ese sentido incluimos por defecto las siguientes líneas.

```
([aeiouáéíóú])([uea]m|[aie]t|[aeui]nt|ur)\b \1_\2
([Dd])o(m).num \1ó\2inum
```

Con la primera sustitución le indicamos al programa que separe silábicamente el lexema del morfema si aquel termina en vocal y este empieza por otra en palabra potencialmente latina. La segunda, indica que ponga tilde en la antepenúltima sílaba de *Dominum*. Esto presenta una dificultad, y es que se necesitaría tener en cuenta todos los términos posibles. Por este motivo lo haremos de una forma alternativa.

La solución que planteamos es capturar estas palabras al vuelo, reescribirlas con acentos ortográficos y adaptarlas a la ortografía española conforme a reglas generales que aligeren la imprescindible corrección manual. De acuerdo con las normas de acentuación latina, la última sílaba es siempre átona sea cual sea su cantidad, por lo que las palabras bisilábicas son llanas. Las dificultades surgen en las de tres o más sílabas, pues la posición de la sílaba tónica la determina la cantidad de la penúltima, que no siempre conocemos de antemano. Si esta es larga, el acento recae sobre ella. Si, por el contrario, es breve y va precedida de otra sílaba, es la antepenúltima la acentuada, independientemente de su cantidad.

Para hallar si la sílaba decisiva es larga o breve, debemos recurrir a los patrones que sigue la lengua latina. De este modo, la cantidad de una sílaba puede deberse a la naturaleza de la vocal nuclear o a su posición. Esto es, la sílaba será larga si se forma en torno a un núcleo largo, que estaría constituido por una vocal larga o por un diptongo; la sílaba puede ser larga también si es cerrada, independientemente de la cantidad de su núcleo (Stowasser y col., 1998, p. XI).

Aunque podemos detectar una sílaba larga posicional, no resulta igual de sencillo averiguar la cantidad de una vocal interior tan solo con reglas estrictamente grafémicas si esta no va indicada mediante símbolos diacríticos. No obstante, es posible identificar sílabas largas posicionales y los diptongos, siendo los siguientes los más habituales:*ae*, *oe*, *au*, *ceu*, *heu*, *seu*, *neuter*, *cui* e *huic* (Ceccarelli, 1999, p. 28). Generalizaremos esto y los dejaremos tentativamente en ⟨æ⟩, ⟨œ⟩, ⟨au⟩, ⟨eu⟩ y ⟨ui⟩. Así tenemos, resueltos todos los casos salvo aquellos en los que la penúltima vocal es abierta, ya que desconocemos *a priori* su cantidad. Lo que haremos es asumir que la penúltima es larga,

salvo que la antepenúltima sea cerrada o su núcleo sea un diptongo, en cuyo caso asumiremos que es breve. Como es de esperar, esta simplificación no produce resultados perfectos, pero reduce el número de excepciones individuales que necesitamos añadir a la lista hasta producir una salida aceptable.

Un último problema que se plantea, cuya resolución es más incierta, es el caso de términos de morfología homográfica. El ejemplo más claro lo ofrecen las primeras personas de la voz pasiva latina, como *amor*, *amābor*, *amābar*, *amer*, *amārer*. En términos puramente grafémicos, estas formas son indistinguibles de palabras agudas españolas como *amor*, *labor*, *acabar* o *temer*, todas ellas agudas, por lo que no recibirán un tratamiento excepcional sin producir cambios en palabras españolas legítimas. En el ejemplo 14d tenemos un caso muy claro de la situación. Podríamos resolver mediante diccionario aquellos casos que no tuvieran homógrafo exacto en español, incluyendo también el lexema, como en el ejemplo de *dominum*, pues no podríamos basarnos completamente en el morfema. Aun así, hay algunos casos difíciles de abarcar sin provocar errores en palabras españolas, como, por ejemplo, con la palabra *amor*, que está en el repertorio de ambas lenguas pero acentuada en posiciones inversas. Dado que los latinismos son casos excepcionales y, entre estos, aquellas palabras con acentuación contraria a las reglas españolas son una minoría, creemos que es factible el tratamiento manual de aquellos casos no cubiertos si los hay en el corpus. Otros morfemas en vocal o *s* no los automatizamos, pues el ahorro que supondría en la corrección manual sería mínimo. Sin embargo, al igual que los otros casos, también están abiertos a ser tratados individualmente empleando el archivo de excepciones.

Latinismos no transliterados

Lo que acabamos de exponer lo formalizamos con el método `Latín`. Como vemos en Algoritmo 23, este método toma un único argumento. En él definimos la lista `flexiones`, en la que guardamos una lista de morfemas latinos que darían lugar al problema que estamos tratando. Hacemos lo propio con los diccionarios *acentos* y *diptongos*. En el primero tenemos una tabla de transformación para las vocales. En ella, la clave de cada elemento es una vocal y su valor, la misma vocal acentuada. En el diccionario `diptongos` tenemos como clave aquellos diptongos latinos no excepcionales que no lo son en castellano, a los que adjudicamos como valor el mismo diptongo en forma monografemática. En el programa de Python, por ejemplo, lo hacemos cambiando lo dos grafemas por el dígrafo con ligatura. De esta manera, tenemos la misma información en una sola posición.

Antes de emprender acción alguna, verificamos que la palabra se trata de un caso

especial, tal como los que hemos descrito. Si, efectivamente, resulta ser así, seguimos adelante con el proceso; si no, devolveremos la palabra tal cual sin modificaciones. Para esto, comprobamos si la palabra en minúscula termina con alguno de los elementos de la lista *flexiones* y no tiene marca acentual explícita. Si se cumplen ambas condiciones, comenzamos los cambios[113].

Algoritmo 23: Procesamiento de latinismos.

Método `Latín(`*palabra*`)`**:**
  flexiones $\leftarrow f : \{-um, -em, -at, -ant, ...\}$
  acentos $\leftarrow \{(a, á), (e, é), (i, í), (o, ó), (u, ú)\}$
  diptongos $\leftarrow \{(ae, æ), (oe, æ)\}$
  si *palabra* **acaba con***flexiones* **entonces**
    si $(á \vee é \vee í \vee ó \vee ú) \notin$ *palabra* **entonces**
      para *flexión* $\in$ *flexiones* **hacer**
        si *palabra* **acaba con** *flexión* **entonces**
          palabra(*flexión*) $\leftarrow \langle _ \rangle +$ *flexión*, $\forall$ *flexión* $\in$ *palabra*
      palabra($diptongo_c$) $\leftarrow diptongo_v$
      slbs $\leftarrow$`Silabea(`*palabra*`)`
      cuenta $\leftarrow |palabra|$
      si $n = 1$ **entonces** {Nada}
      si no, si $n = 2$ **entonces**
        palabra $_{n-1}acentos_c \leftarrow acentos_v$
      si no, si $n > 2$ **entonces**
        si $diptongo_i \in sílabas_{n-2}$ **entonces**
          slbs $_{n-1}acentos_c \leftarrow acentos_v$
      si no, si $diptongos_i \in sílabas_{n-2} \vee$
       `CuentaVocales(`$sílabas_{n-3}$`)` $\geq 2 \vee$
       $\neg sílabas_{n-3}$**acaba con** $\{vocales\}$ **entonces**
        slbs $_{n-2}(acentos_c) \leftarrow acentos_v$
      si no
        slbs $_{n-1}(acentos_c) \leftarrow acentos_v$
  devolver *sílabas*

Recorremos uno detrás de otro los dos diccionarios que habíamos definido. Primero lo hacemos con *flexiones* en busca del morfema de la palabra y lo separamos del lexema. En la prueba con el computador, hemos optado por un guion bajo, pues es un elemento poco común en los textos y que, además, refleja ópticamente la idea de lo que pretendemos hacer: deseamos separar con esta marca lexema y morfema, de manera que, si uno de los lados limita con una vocal alta —lo que implicaría

[113]En la implementación en Python, como ya dijimos, hay distinción entre mayúsculas y minúsculas, por lo que en este punto pasaríamos la palabra entera a minúsculas para evitarnos hacerlo posteriormente cada vez que necesitáramos hacer una comparación entre dos caracteres en mayúscula y minúscula para la misma letra, aunque aquí resulte irrelevante.

un diptongo propio de la fonología española—, podamos indicar la existencia de un hiato facultativo. En el siguiente bucle reemplazamos grupos vocálicos que forman diptongo en latín pero no en español por un carácter equivalente que represente un dígrafo ligado. Así, por ejemplo, al grupo vocálico ⟨ae⟩, que es bisilábico en español, le correspondería ⟨æ⟩, de manera que lo interpretaríamos después con un núcleo vocálico singular. Una vez hecho esto, dividimos las sílabas normalmente, valiéndonos del método `Silabear`. De esta manera guardamos en la variable *sílabas* una lista de las sílabas de la palabra.

A continuación comprobamos si la palabra es monosilábica, lo que daría una solución trivial. Si este no es el caso, seguimos probando; examinamos si tiene exactamente dos sílabas y, de ser así, aplicamos el acento de forma directa en la segunda, como prescriben las reglas de la prosodia latina. Si tampoco es el caso, verificamos que la palabra reúna las condiciones que la hagan susceptible de las mismas reglas que rigen para una palabra bisilábica. Esto se averigua cotejando la penúltima sílaba por si contiene un diptongo o es una sílaba trabada. Sabemos si el núcleo es un diptongo examinando si alguno de los valores del diccionario *diptongos* —con las ligaturas— está en la penúltima sílaba o esta contiene más de dos vocales. Si la sílaba es cerrada se dilucida *ad contrarium*, verificando que su última letra no esté entre las claves de *acentos*, que, recordemos, contiene las vocales. Si se cumplen estas premisas, sabemos que la penúltima sílaba es larga y, por lo tanto, va acentuada. En ese caso, se procede a introducir un acento gráfico en la sílaba. Se recorren los elementos de *acentos* por clave y valor, cerciorándose por cada elemento de si su clave se encuentre en la penúltima sílaba. Si es así, se reemplaza en la penúltima sílaba dicha clave por su valor y se interrumpe el bucle tan pronto se lleva a cabo la sustitución.

Si la palabra tiene tres sílabas o más y no se cumplen las condiciones descritas para la segunda sílaba, solo queda intentar adivinar si la antepenúltima sílaba es tónica, pues no conocemos la cantidad de la penúltima. En ese caso, se introducen sustituciones específicas en el archivo de excepciones, como las siguientes, que cubren las primeras personas plurales de los futuros de indicativo de la primera y segunda declinación, así como la tercera persona singular del futuro de la voz activa:

```
abi([tm])u(rs) ábi\1u\2
abi([tm])u(rs) ábi\1u\2
ebi([tm])u(rs) ébi\1u\2
ebi([tm])u(rs) ébi\1u\2
```

La única respuesta viable pasa entonces por un diccionario extensivo. En tanto que escapa del propósito de este trabajo, aproximaremos simplemente una solución acep-

table, si bien imperfecta[114], a partir de reglas ortográficas generales. Refinaremos aún más el resultado si consideramos que es más probable la alternancia entre sílabas breves y largas que encontrar dos largas seguidas, por lo que será más plausible asumir que la penúltima es breve si la antepenúltima sílaba es larga y después corregir a mano los fallos.

Volvemos a repetir las comprobaciones que habíamos hecho con la penúltima sílaba, pero examinamos la antepenúltima en esta ocasión. Primero verificamos si alguna de las ligaturas de *diptongos* se encuentra en esa sílaba o si contiene dos vocales, lo que hallamos recorriendo las claves de *acentos*. Sabremos que es cerrada observando si su última letra no es una clave del diccionario. Acabamos recorriendo *acentos* por sus elementos y tratando de sustituir cada clave por su valor, para abandonar el bucle en la primera ocasión que lo consigamos.

Si ninguna de las condiciones anteriores se ha cumplido, asignamos el acento a la penúltima sílaba, marcando la primera vocal que encontramos. Ahora sí, devolvemos las sílabas, bien en su estado originario si la palabra lleva acento gráfico o no termina en uno de los morfemas de las flexiones, o una lista de sílabas con las transformaciones indicadas si la palabra cumplía las condiciones para entrar en la bifurcación condicional inicial.

Silabación

Aquí ya se hallan todas las palabras en igualdad de condiciones, una vez ajustadas aquellas cuya ortografía no se correspondía con la prosodia. Por lo tanto, invocamos el método `Silabea`, al que le pasamos como parámetro el atributo *palabra*[115].

Algoritmo 24: Método silabeador.

```
Método Silabea(palabra):
    sílabas ← {}
    si palabra es cadena entonces
        extraños ←{(à, a), (õ, o), (ffi, ffi), ...}
        palabra(ajenos_c) ←ajenos_v, ∀extraños_c ∈ palabra
        sílabas ← IdentificaNucleos(palabra)
        sílabas ← DistribuyeConsonantes(sílabas)
    devolver sílabas
```

[114] Aquí vemos las ventajas de la modularidad de nuevo, pues podríamos sustituir en el futuro este método por una biblioteca específica para textos latinos sin tener que alterar el resto del programa.

[115] En la implementación en Python, necesitaríamos, además, pasarle un parámetro indicando que estamos trabajando con el AFI y no con una palabra representada con caracteres puramente grafémicos. De esta manera, puede usarse directamente para otros fines aparte de este particular que nos ocupa, bastaría con advertirlo al invocar el método.

Para comenzar, definimos una lista vacía a la que llamaremos *sílabas*, pues el propósito de este método es organizar las sílabas de la palabra como elementos de una lista. Por esta razón, comprobaremos primero que *palabra* sea una cadena de caracteres, de modo que, si es ya una lista de sílabas[116], devuelve el atributo que recibe el método tal cual sin alterarlo. Si, en efecto, se trata de una cadena de caracteres y no de una lista de sílabas, emprende ahora sí la división silábica.

Creamos un diccionario del que nos valdremos para limpiar grafemas poco habituales o impropios del español al que denominaremos *extraños*. Las claves de este diccionario son tanto diacríticos impropios de la lengua castellana (*à* u *õ*, por ejemplo) como ligaturas de imprenta que el computador representa como un único carácter y cuyas letras constitutivas necesitamos tratar por separado (por ejemplo, *ﬁ* o *ﬂ*). Aunque las ligaturas son hoy en día escasas, y más en ediciones electrónicas, estas pueden aparecer por diferentes motivos. Esto van desde una elección tipográfica consciente en la edición electrónica, a una digitalización precisa de un texto impreso que las usaba o a una mala digitalización de un texto impreso que carecía de ellas. En cualquier caso, es necesario tenerlas en cuenta para que estos caracteres sean considerados por los grafemas que representan y no como un símbolo distinto. Los valores que adoptan son la transliteración castellana, en el caso de los diacríticos o una secuencia de grafemas en el de las ligaturas.

Empezaremos a trabajar con la palabra. Recorremos los caracteres que componen la variable *palabra*, que la almacena, para ir sustituyendo las claves del diccionario *extraños* por sus valores, si es que los hallamos. Esta cadena de caracteres la transformamos en una lista cuyos elementos corresponden a caracteres de *palabra* y la guardamos en la variable *sílabas*. Con esto, abordamos la silabación en sí.

Añadimos los elementos a la lista en dos pasos: primero separamos los elementos de *sílabas* si nada lo impide por medio el método `IdentificaNucleos`. Esto es, dividimos todos los elementos salvo aquellos que puedan constituir un diptongo, en cuyo caso son asignados a un único elemento. Luego unimos si es posible los elementos de la lista resultante valiéndonos del método `DistribuyeConsonantes`, que se encarga de repartir las consonantes en ataque y coda de los núcleos vocálicos. Devolvemos la lista así obtenida, que contiene las sílabas. Queda explicar en detalle los procesos de separación y unión que hemos empleado.

[116]Este sería precisamente el caso de una palabra procesada por `Latín`. Dicha subrutina ya habría realizado el trabajo por encontrarse ya en ese método una llamada a `Silabea`, de manera que ya estaría silabeado.

Separación de caracteres

Empezaremos por la separación de los núcleos silábicos de la cadena en elementos de una lista mediante el método `IdentificaNucleos`. Como vemos en Algoritmo 25, se trata de una función recursiva que recibe la cadena de caracteres `palabra` y la lista `sílabas`, esta última en principio vacía. La cadena corresponde a la palabra que queremos procesar y la lista se va rellenando con sílabas tentativas, con las vocales unidas en caso de diptongo, que finalmente devolveremos.

Algoritmo 25: Separación de núcleos silábicos.

```
Método IdentificaNucleos(palabra, sílabas):
    dipt ← EncuentraDiptongo(palabra)
    dígrafos ← {ll, ch, rr}
    si diptongo entonces
        slbs ←
          IdentificaNucleos(palabra − diptongo_i, {diptongo_i} + sílabas)
    si no, si palabra acaba con dgrafos entonces
        slbs ← IdentificaNucleos(palabra_{1:n−2}, {palabra_{n−1}, palabra_n} +
          sílabas)
    si no, si palabra entonces
        slbs ← IdentificaNucleos(palabra_{1:n−1}, {palabra_n} + sílabas)
    devolver sílabas
```

Primero buscamos si la cadena de caracteres acaba con un diptongo o un triptongo según las reglas de formación vistas en Apartado 4.6[117]. El resultado de la búsqueda lo almacenamos en la variable *diptongo*. Por conveniencia, creamos una lista de dígrafos ortográficos del español. El último no es relevante en nuestro caso porque estaríamos ya trabajando con los equivalentes de su transcripción fonológica. Sin embargo, permitiría usar el módulo de forma independiente en otros contextos que requirieran trabajar con textos tal cual sin transcribir.

Después entramos en una división condicional. Primero comprobamos si hemos encontrado una agrupación vocálica a final de palabra verificando que la variable *diptongo* no es nula. De contener un valor, invocamos recursivamente la propia función `IdentificaNucleos`, pero, ahora, la llamaremos pasándole como argumentos la cadena de caracteres sin la unión vocálica final y la lista *sílabas* con dicha fusión insertada en la primera posición. Guardamos lo devuelto en *sílabas*.

Si la palabra acaba en un dígrafo, guardamos en *sílabas* la salida de una nueva llamada a la propia función con la cadena de caracteres sin el dígrafo y este como un nuevo elemento al final de la lista. En cualquier otro caso, guardamos en *sílabas* el

[117] En la implementación en Python, hacemos esto valiéndonos de expresiones regulares.

resultado de la llamada recursiva a `IdentificaNucleos` con la cadena sin su último carácter y la lista *sílabas* con este en primera posición.

En el siguiente ejemplo vemos el funcionamiento de una llamada concreta al método `IdentificaNucleos('Sigüenza')`[118]. Primero tomaría la *a* y guardaría en *sílabas* el resultado de `IdentificaNucleos('Sigüenz', {'a'})`. Este, a su vez, sería el resultado de `IdentificaNucleos('Sigüen', {'z', 'a'})`, y así sucesivamente hasta que la llamada más profunda de la recursión se encontrara con una cadena de caracteres nula, por lo que procedería a devolver la lista, que iría pasando de un nivel de la recursión al precedente hasta alcanzar la llamada inicial.

```
0: word: []                                    letters: Sigüenza
1: word: ['a']                                 letters: Sigüenz
2: word: ['z', 'a']                            letters: Sigüen
3: word: ['n', 'z', 'a']                       letters: Sigüe
4: word: ['üe', 'n', 'z', 'a']                 letters: Sig
5: word: ['g', 'üe', 'n', 'z', 'a']            letters: Si
6: word: ['i', 'g', 'üe', 'n', 'z', 'a']       letters: S
8: word: ['S', 'i', 'g', 'üe', 'n', 'z', 'a']  letters:
```

Unión de caracteres

El método `DistribuyeConsonantes` (Algoritmo 26) recibe una lista de agrupaciones fonémicas por medio del argumento *palabra*. Los núcleos silábicos se encuentran ya ordenados en monofonemáticos, diptongos y triptongos. La subrutina empieza por definir dos listas. La primera es *ataqueIndivisible*, que está compuesto por agrupaciones consonánticas en ataque silábico del tipo *mutae cum liquidis*, según lo visto en Apartado 4.7, dígrafos y el grupo /gw/. La segunda lista es *codaIndivisible* que, como su propio nombre indica, se compone de agrupaciones consonánticas indivisibles en coda si preceden a otra consonante.

Se define la variable *sílabas*, que es el valor a devolver. En esta variable se irán guardando los núcleos silábicos y las agrupaciones consonánticas que les corresponden; se inicializa como una lista vacía. También se crea una variable auxiliar *resto*, donde se van poniendo temporalmente las consonantes que aún queden por asignar a un núcleo silábico; esta se inicializa como una cadena de caracteres vacía[119]. Con esto, ya está todo lo necesario para empezar a iterar por los elementos de *palabra*.

[118] Ciudad mitrada castellana de monumental legado arquitectónico, histórico y cultural; ha presentado su candidatura a Patrimonio Mundial.

[119] En la implementación en Python, además, para completar los pasos previos hay que verificar si el texto es una transcripción fonológica o es grafémico, en cuyo caso el programa considerará los

Algoritmo 26: Distribución silábica de consonantes.

```
Método DistribuyeConsonantes(palabra, silabas):
    ataqueAdmisible ← f : {ataques admisibles}
    ataqueIndivisible ← {pl, fl, ...}
    codaIndivisible ← {ns, bs, ...}
    sílabas ← {}
    para posición_i ∈ palabra hacer
        /* Bloque de comprobaciones (Algoritmo 27)          */
    /* Asignación final del resto (Algoritmo 28)            */
    devolver sílabas
```

Ya dentro del bucle que hemos iniciado, establecemos una serie de bifurcaciones condicionales (Algoritmo 27). Como primera comprobación, miramos si el carácter examinado es el escogido para señalar los hiatos facultativos. Si este es el caso, solo actuaremos si la lista de sílabas contiene al menos un elemento; de lo contrario, se trata del comienzo de la palabra, por lo que no haremos nada y, simplemente, pasaremos a la siguiente iteración. Supongamos que no es este el caso; entonces, nos encontramos ante una división silábica forzada manualmente, por lo que añadiremos el contenido de *resto* en coda a la última sílaba de la cadena y reiniciaremos la variable.

Si no encontramos un hiato introducido de forma manual, pasamos a la siguiente prueba. Entramos en el bloque de instrucciones si el elemento de *grupo* que examinamos en esta iteración contiene exclusivamente consonantes. Si es así, vemos si acaba con *y* —lo que, de nuevo, se emplearía únicamente en contextos en los que la cadena de entrada es grafémica y no fonológica—, lo que llevaría a un tratamiento especial. De ser cierto, comprobamos si, además, la variable *resto* está constituida solo por *y*, esto es, si no va precedida de consonante. Si se cumple esa condición, asignamos la *y* a la sílaba anterior si acaba en vocal, pues estaremos ante una cadena V-*y*-C. Si no, asignamos la *y* a una sílaba nueva de *sílabas*, ya que estaríamos ante C-*y*-C. Finalmente sobrescribiríamos *resto* con el valor del grupo que estamos observando.

Si la consonante final no es *y*, se la añadimos a *resto*. Guardamos en la variable *media* la posición donde queremos dividir *resto* en caso de que haya que repartirlo entre dos sílabas. Asimismo, introducimos aquí la excepción para evitar que la hache intercalada vaya en ataque después de consonante.

Si ninguna de las dos condiciones anteriores se cumple, probamos a ver si lo hace la siguiente. Examinamos *resto* para ver si contiene como máximo una consonante o

caracteres ⟨j⟩ y ⟨w⟩ como grafemas consonánticos. En esta descripción en pseudocódigo, no lo haremos, sino que los tomaremos como alófonos de /i/ y /u/.

Algoritmo 27: Bloque de comprobaciones.

```
n ← |sílabas|
si posición_i = ⟨_⟩ entonces
    si n > 0 entonces
        slbs_n ← resto
        resto ← ∅
si no, si ∄x ∈ vocales, ∀x ∈ posición_i entonces
    si resto acaba con ⟨y⟩ entonces
        si resto = ⟨i⟩ ∧ posición_{i−1} ∈ {a, e, o} entonces
            slbs_n ← silabas_n + resto
        si no
            slbs ← sílabas + {resto}
        resto ← posición_i
    si no
        resto ← resto + posición_i
        si n = 0 entonces med ← |resto| mod 2
        si restoacaba con ⟨h⟩ entonces med ← 0
si no, si |resto| ≤ 1 ∨ n = 0 entonces
    slbs ←sílabas + {resto + posición_i}
si no, si resto acaba con ataqueIndivisible entonces
    si n > 0 entonces
        slbs_n← sílabas_n + resto_{n−1:n}
        slbs ← sílabas + {resto_{n−1:n} + posición_i}
    si no
        slbs ← posición_i + {resto + posición_i}
    resto ← ∅
si no, si resto empieza con codaIndivisible ∧ dim resto > 2 entonces
    slbs_n← sílabas + resto_{1:n−2}
    slbs ← sílabas + {resto_{3:n} + posición_i}
    resto ← ∅
si no, si resto_{n−1} + resto_n ∉ ataquesAdmisibles entonces
    slbs_n ←sílabas_n + resto_{1:n−1}
    slbs ← sílabas + {resto_n + posición_i}
    resto ← ∅
si no
    slbs_n ← sílabas_n + {resto_{1:med−1}}
    slbs ← sílabas + {resto_{med:n} + posición_i}
    resto ← ∅
```

si la variable *sílabas* está vacía. Esto es, nos encontramos ante un núcleo vocálico, ya que no se ha cumplido que sea la marca de hiato o consonante. De la misma manera, resulta que, o bien tenemos una o ninguna consonante sin asignar, o bien estamos ante la primera agrupación vocálica de la palabra. En tal caso, añadimos un nuevo elemento a *sílabas*, puesto que tenemos un núcleo vocálico. Este estaría

formado, además, por *resto* en ataque. Una vez asignado esta variable, la vaciamos de su contenido antes de la siguiente iteración.

Si tampoco esta se cumple, estamos ante un núcleo vocálico que no es el de la primera sílaba. Así pues, cotejamos las últimas consonantes de *resto* con la lista de ataques indivisibles. De ser uno de ellos y ya tenemos un núcleo vocálico en *sílabas*, dividimos *resto* en dos: el grupo consonántico indivisible del final y las demás consonantes. El comienzo se lo añadimos en coda al último elemento de *sílabas* y el grupo indivisible lo pondríamos en un nuevo elemento como coda del vocálico que estamos examinando. Nótese que si *resto* lo componen dos consonantes, por ejemplo, $resto = \{p, ɾ\}$ la división se seguiría haciendo, pero, en este caso los grupos serían $resto_1 = \{\}$, que iría en coda de la sílaba anterior y $resto_2 = \{p, ɾ\}$, en ataque de la nueva sílaba. Finalmente, vaciamos la variable auxiliar una vez distribuidos sus componentes.

Si el caso anterior tampoco se cumple, verificamos si *resto* comienza por alguna de las consonantes que componen una coda indivisible. De ser cierto, hacemos un proceso parecido al precedente; dividimos la variable auxiliar en dos partes: por un lado, las dos primeras letras y, por otro, las demás si las hay. El primer grupo se lo asignamos al último núcleo vocálico en coda, mientras que el segundo irá en una nueva sílaba en ataque de la vocal o agrupación vocálica que estamos examinando. Como en las ocasiones anteriores, una vez distribuidas las consonantes, reinicializamos la variable auxiliar.

Si los casos anteriores no se cumplen, examinamos las dos últimas letras de *resto* para ver si forman un par inaceptable en ataque silábico y, por lo tanto, susceptible de división. Comprobamos si tenemos una consonante de /d/, /f/, /k/ o /t/ (*ánfora*); /θtk/ seguido de /g/ (*pazguato*), ; dos veces /l/ o /m/ (*colmena*); o /k/ ante /θ/ (*acción*). Nótese que el dígrafo ⟨ll⟩, por ejemplo, aunque satisface los criterios de esta bifurcación condicional, no podría entrar en ella nunca porque lo habría hecho antes en la prueba en que se comprueba si *resto* acaba en un ataque indivisible. En caso de que el algoritmo entre en esta condición, añade todas las letras de *resto*, salvo la última, al elemento final de *sílabas* y después introduce una nueva sílaba compuesta por la última letra de *resto* y el elemento vocálico que estamos examinando para, finalmente —como ya sabemos—, borrar el contenido de la variable una vez repartidos sus componentes.

En cualquier otro caso, concatenamos a la última sílaba de *sílabas* las consonantes de *resto* hasta la posición *media* e introducimos una nueva sílaba compuesta por las letras de *resto* desde la posición indicada por *media* hasta el final en ataque de la vocal que estamos examinando. También en este caso, borramos el contenido de *resto*

una vez usado.

Algoritmo 28: Asignación final del resto.

si *resto* **entonces**
  si *resto* **acaba con** $\langle y \rangle \wedge |resto| = 1$ **entonces**
    $\mathsf{slbs}_n \leftarrow resto$
  si no, si *resto* **acaba con** $\langle y \rangle$ **entonces**
    $\mathsf{slbs}_n \leftarrow resto_{1:n-2}$
    $\mathsf{slbs} \leftarrow sílabas + \{resto_{n-1:n}\}$
  si no
    $\mathsf{slbs}_n \leftarrow sílabas_n + resto$

Para terminar, una vez ha terminado el bucle al llegar a la última letra, comprobamos si han quedado consonantes en *resto*, en cuyo caso debemos añadírselas a la sílaba final, salvo que se trate de una *y* en un texto grafémico, que podría diptongar con la anterior o hacer de núcleo silábico. Esto lo hacemos observando el tamaño de *resto* si este acaba en *y*, de forma que, si contiene un único elemento, sabemos que no hay consonantes intercaladas entre la última vocal y esta, por lo que añadiremos todo el contenido de *resto* a la última sílaba. Si hay más de un elemento en *resto*, sabemos que *y* va precedida de consonantes, por lo que la añadiremos los dos últimos elementos de *resto* como última sílaba de *sílabas* y todos aquellos elementos anteriores como coda de la sílaba precedente. En cualquier otro caso, añadimos las consonantes remanentes como coda a la última sílaba y retornamos la lista de sílabas procesada *sílabas*. Si retomamos el ejemplo anterior, al pasar por `DistribuyeConsonantes`, los resultados intermedios que se irían produciendo serían los siguientes.

```
0: word: []                    letter:      onset:          media: 0
1: word: []                    letter: S    onset: S        media: 0
2: word: ['Si']                letter: i    onset:          media: 0
3: word: ['Si']                letter: g    onset: g        media: 0
4: word: ['Si', 'güe']         letter: üe   onset:          media: 0
5: word: ['Si', 'güe']         letter: n    onset: n        media: 0
6: word: ['Si', 'güe']         letter: z    onset: nz       media: 1
7: word: ['Si', 'güen', 'za']  letter: a    onset:          media: 0
```

Ubicación del acento prosódico

A continuación definimos la función de encontrar la sílaba tónica en la palabra, a la que llamaremos `EncuentraTónica`. El único argumento que recibe es una lista de sílabas como la que hemos obtenido mediante los algoritmos descritos en las secciones anteriores. Emplearemos para ello las reglas de la ortografía española vistas en

Apartado 4.9. Primero examinamos el número de sílabas. Si es un monosílabo, la solución es trivial. Si este no es el caso, comprobamos si hay alguna marca gráfica de acentuación en la palabra. Aquí hemos indicado las tildes ortográficas por claridad, pero habría que considerar también las marcas de acento primario y secundario de las transcripciones. Si este es el caso, recorremos las sílabas hasta que la encontramos; la posición de la sílaba marcada, que será la que lleve el acento prosódico.

Podría ser que la palabra no tuviera marcas explícitas, por lo que veríamos si acaba en vocal y, de ser así, identificaríamos el acento prosódico en la penúltima sílaba. Si no acabara en vocal, observaríamos si acaba en ⟨n⟩, ⟨s⟩ ante vocal y, si se cumple, también asignaríamos el acento prosódico a la penúltima sílaba. En cualquier otro caso, la palabra sería aguda. Al salir de las bifurcaciones, la posición del acento prosódico está en *acento*, que devolvemos como salida de la función.

Algoritmo 29: Localizador de acento prosódico.

Método `EncuentraTónica(`*sílabas*`):`
  cuenta $\leftarrow |sílabas|$
  si $n = 1$ **entonces** acento $\leftarrow n$
  si no, si $\exists x \in sílabas_i \forall x \in \{á, é, í, ó, ú\}$ **entonces**
    acento $\leftarrow i$
  si no, si $sílabas_n$ **acaba con** $\{a,e,i,o,u,y\}$ **entonces**
    acento $\leftarrow n - 1$
  si no, si $sílabas_n$ **acaba con** $\{n, s\}$ **entonces**
    acento $\leftarrow n - 1$
  si no
    acento $\leftarrow n$
  devolver *acento*

Verificación de la precisión del silabeador

Para verificar la precisión del silabeador, probamos su implementación en Python con el corpus de palabras predefinida EDFU (Rosa y Pérez, 2020), que consta de 106.362 palabras y ya ha sido empleado como *gold standard* para estimar la precisión de la división silábica automática orientada a la escansión de versos.

Nuestro modelo obtiene una exactitud del 99,81 %, por encima del 98,06 % de Agirrezabal y col. (2017), el 98,35 % de Navarro-Colorado (2015), únicamente por detrás del 99,99 % de Rantanplan (Rosa y col., 2022). Este porcentaje responde diferentes agrupaciones en la distribución de grupos consonánticos de acuerdo con variedades regionales, como, por ejemplo, el grupo ⟨tl⟩, que en el corpus (350 entradas) se divide según la pronunciación americana *a·tlán·ti·co* y nuestro silabeador emplea la castellana *at·lán·ti·co*. Asimismo, hemos detectado en el corpus triptongos impropios, ya que

no son combinaciones de un diptongo creciente y otro decreciente, por lo que encontramos palabras del tipo [amambaiense] divididas como *a·mam·baien·se* en lugar de *a·mam·ba·ien·se*. El corpus adjudica una sílaba a las eses líquidas, mientras que nuestro silabeador no lo hace, ya que añade la sílaba epentética durante la transcripción. Descontando divisiones alternativas correctas, errores del corpus y eses epentéticas correctas por todo lo demás, se alcanzaría una precisión del 99,98 %.

No obstante, esta puntuación se obtiene usando la configuración por defecto, a saber, con el argumento `exceptions` puesto a `False`. Para la escansión métrica se pondrá a `True`, de modo que se hagan hiatos no explícitos. En este caso, la precisión respecto al corpus de palabras sería de 90,06 % incluyendo errores del corpus, siendo el 0,86 % de diferencia los hiatos no indicados implícitamente. Esto es, el corpus divide [huida] como `hui-da`, mientras que nuestro silabeador lo haría como `hu-i-da` si tenemos activadas las excepciones.

Para las pruebas de escansión automática de verso fijo, se aprecian diferencias distinguibles ente el uso de la división silábica peninsular (precisión del 97,51 %) o la americana (97,08 %), por lo que optamos, tanto por ser el uso lingüístico del autor como por la disponibilidad de informantes hablantes de esta variedad, por procesar usando las excepciones castellanas y luego tratar de hacer una sinéresis donde sea necesario en lugar del procedimiento contrario, abordar todas las palabras como regulares y dividirlas con diptongo e introducir una diéresis para ajustar el verso.

«A te convien tenere altro viaggio»,
rispuose poi che lagrimar mi vide,
«se vuo' campar d'esto loco selvaggio:

ché questa bestia, per la qual tu gride,
non lascia altrui passar per la sua via,
ma tanto lo 'mpedisce che l'uccide».

Dante Alighieri, *Comedìa*

Escansión automática 9

9.1 Introducción

Llegados aquí, disponemos de un texto ordenado por versos y útiles para transcribirlos. Ahora toca escandirlos, lo que implica encontrar el metro, ritmo y rima de cada uno de ellos. Por lo general, podemos suponer de antemano las características métricas de un determinado poema: en un soneto encontraremos endecasílabos con rima consonante, en un romance la rima será asonante en los pares, casi siempre de versos octosílabos, en una silva, será consonante, pero sus versos alternarán siete y once sílabas. En cualquiera de esos casos, podríamos hacer una estimación tentativa de esas características y acertar la mayoría de las ocasiones. Los textos dramáticos, por el contrario, no permiten esta simplificación, pues encontramos sin solución de continuidad, por ejemplo, un romance seguido de una silva, tras la que vienen redondillas, hexasílabos cantados y algunos endecasílabos sueltos intercalados. No puede hacerse una estimación preliminar, ya que el drama no ofrece unos parámetros fijos *a priori* que puedan emplearse para tal fin.

¿Cómo abordamos la polimetría? Para esta labor, la máxima *divide et impera* se hace más apropiada que nunca. Lo que necesitamos es descomponer el análisis polimétrico del texto en una suma de otros monométricos de cada una de sus unidades estróficas[120]. El problema que se presenta es que, para identificar correctamente las estrofas, debemos conocer la medida de los versos que los componen. Para enfrentarnos a esto, escandiremos en dos pasadas: en primer lugar, haremos una estimación no estrófica considerando el metro natural de cada verso —esto es, sin romper sinalefas

[120]Usaremos *estrófico* en un sentido laxo para simplificar, también y a riesgo de caer en la imprecisión terminológica cuando nos refiramos a poemas no estróficos.

obligatorias ni otras licencias extraordinarias— y las estimaciones de los versos precedentes más cercanos. Con una ponderación aproximada de la medida de los versos, los agrupamos por estrofas. Finalmente, evaluamos los versos en su contexto estrófico y escandimos de nuevo aquellos versos que no se ajusten a los requerimientos de aquel, según el esquema de Figura 9.1. Dado que la precisión es alta, el número de versos escandidos por segunda vez es una fracción mínima del total, por lo que el impacto en el tiempo requerido para procesarlos de nuevo no resulta exagerado.

En términos prácticos, esto se traduce en que, una vez que tomamos la información de la obra dispuesta según lo descrito en el capítulo anterior, esta ha de pasar por una cadena de módulos de procesamiento, en los que la salida de uno es la entrada del siguiente. En concreto, en la primera fase, la pieza a analizar, codificada en CSV, constituye los datos de entrada del primer módulo, que se encarga de separar por versos. Cada uno de ellos es procesado por un módulo de escansión que, además del verso, recibe solo una información contextual tentativa que inferida de los precedentes.

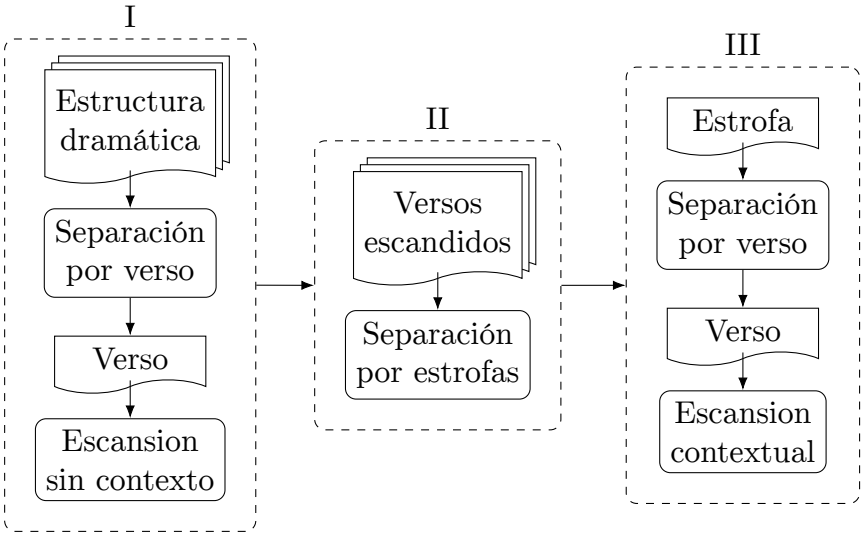


Figura 9.1: Esquema del proceso de escansión.

El producto de la primera fase es un resultado provisional que sirve como datos de entrada para la segunda. Aquí, otro módulo separa el texto en sus unidades estróficas a partir de la rima y la aproximación del metro. Con esto, disponemos ya de toda la información requerida para cada estrofa. Ahora no estamos analizando un texto polimétrico, sino sus estrofas, por lo que excluimos todos aquellos metros carentes de sentido en su entorno estrófico en el que se hallan. Por ejemplo, la presencia de versos sin ambigüedades —volveremos a esto después en detalle— permite establecer el contexto métrico de la estrofa: si uno de los versos acepta solamente una división en ocho sílabas, asumimos ese metro como propio de la estrofa; si tiene once, serán endecasílabos, pero si aparecen en esa misma estrofa versos de arte menor y rima

consonante, muy probablemente serán heptasílabos. Incluso si todos los versos son ambiguos, es posible aproximar con gran exactitud considerando que hay metros mucho más probables —octosílabos, endecasílabos, heptasílabos, hexasílabos— y que ciertas licencias de ajuste métrico —sinéresis, diéresis y dialefas entre átonas, por ejemplo— son excepcionales, por lo que los versos tienden a avenirse a las reglas. A partir de esto, adelantamos con un alto grado de precisión los detalles de cada verso.

De este modo, vamos analizando las estrofas en la fase tercera de la misma manera que lo habíamos hecho en la primera. A diferencia de entonces, ahora tenemos una estrofa y las características esperables de sus versos, por lo que aprovecharemos esta información para escandir nuevamente aquellos que se desvíen de lo esperado.

9.2 Separación por verso

Como dijimos, partimos de la tabla cuyo proceso de elaboración explicamos en el capítulo anterior (Cuadro 7.1). Lo que haremos será recorrer sus filas, examinar las características de la línea y escandir los versos cada vez que tengamos uno completo. Para simplificar tanto la composición como la comprensión del proceso a seguir, este tiene una estructura modular. Primero, explicaremos el algoritmo principal (Algoritmo 30), para lo que nos abstraeremos de las funciones `ActualizaTabla` y `ActualizaMetros`; por ahora, tomaremos ambas como sendas cajas negras que toman datos de entrada y devuelven una lista ordenada de posibles metros o añaden nuevos valores a la tabla, respectivamente. Volveremos sobre estas subrutinas una vez hayamos presentado la idea general.

Para comenzar, definimos dos variables *metros* y *musical* después de importar la biblioteca `libEscansión` (Apartado 9.3), que detallaremos más adelante. En la primera almacenamos una lista de metros ordenada según la probabilidad de que sea el metro del verso actual. Sin saber nada aún del texto, resulta plausible que el primer verso sea octosílabo, así como que hexasílabos cantados, endecasílabos o heptasílabos aparezcan en ocasiones y, esporádicamente, sus correspondientes quebrados. En la segunda variable tenemos un conjunto de cadenas de caracteres que suelen formar parte de la didascalia de personaje —por ejemplo, MÚSICO, MÚSICOS, MÚSICA— en partes cantadas; las usaremos para identificar esta partes e intentar en primer lugar un metro típico de canción. A verso_0, que lleva la cuenta de la numeración del verso anterior, le asignamos el valor cero para que preceda al del primer verso.

Empezamos a recorrer las filas de la tabla. Comprobamos que el número de verso es un entero positivo —recordemos que las acotaciones las habíamos numerado en

sentido inverso—. De esta manera, solo procesamos texto dramático. Si el valor es negativo, al tratarse de una acotación, no es necesario escandir el texto de esa fila, por lo que se incorpora con los valores métricos vacíos y se pasa a la siguiente.

Algoritmo 30: Escansión de los versos uno a uno.

```
Importar(libEscansión)
metros ←(8, 6, 11, 7, 4, 5, 3, 10, 12, 9, 14, 13, 15, 2)
musical ←{MÚS, TODOS, CANTA, CORO}
verso_0 ←0
para cada fila_i ∈ tabla hacer
    si fila_{i,verso} > 0 entonces
        si fila_{i,verso} > verso_0 entonces
            V ←fila_{i,texto}
            si musical_i ∈ fila_{i,personaje} entonces
                metros ←(6) ∪ metros \ {6}
            ActualizaTabla(i, V, fila_{i,personaje}, metros_0)
            si i = fila_{i−1} ∨ fila_{i,verso} < fila_{i+1,verso} entonces
                metros ←ActualizaMetros(fila_{i,sílabas}, metros)
            V_0 ← V
            personaje_0 ← fila_{i,personaje}
            verso_0 ← fila_{i,verso}
        si no
            V ←V_0 ⊕ fila_{i,texto}
            fila_{i,personaje} ← personaje_0 ⊕ fila_{i,personaje}
            ActualizaTabla(i, V, fila_{i,personaje}, metros)
            personaje_0 ←fila_{i,personaje}
            V_0 ← V
```

Si la fila contiene texto de un parlamento, hay que constatar si es el comienzo de un nuevo verso o su continuación. Para averiguarlo, comparamos el número de verso con el del verso precedente verso_0. Si el de esta fila es mayor, se trata de un nuevo verso. Tomamos el texto de la fila y se lo asignamos a V.

Antes de hacer la escansión, nos cercioramos de que ninguno de los elementos de musical sea parte del nombre del locutor. De serlo, estamos muy probablemente ante una parte cantada, por lo que actualizamos la lista de metros colocando 6 como la apuesta para el número de sílabas, en la primera posición[121]. Esto responde a que no daremos un metro fijo para escandir el verso, sino una lista ordenada de metros posibles, en función de lo que esperamos encontrar.

[121]Respecto a la notación del algoritmo en pseudocódigo, nótese que los paréntesis denotan una lista ordenada, por lo que la unión de dos de ellas es una concatenación que no cumple la propiedad conmutativa.

Con esto, llamamos a la función `ActualizaTabla` pasándole el índice de la fila, el texto, el personaje y la lista de metros. Seguidamente, nos aseguramos de que el verso actual no es el último de la tabla —en cuyo caso habría que terminar el proceso— y, si efectivamente no lo es, actualizamos el orden de la lista de posibles metros, considerando el número de sílabas del verso actual. Después de esto, guardamos para más tarde los valores de esta fila que indican el personaje, el texto y el número de verso en variables auxiliares, que hemos denotado con el subíndice 0.

Si, por el contrario, el número del verso actual no es mayor que el del último procesado, nos hallamos ante un verso compartido. Así, el texto del verso será la concatenación de la variable auxiliar donde habíamos almacenado las partes precedentes del verso y esta parte. Igualmente, combinaremos los personajes y, con los datos de las partes en uno solo, aplicamos `ActualizaTabla` para estos valores y guardamos en variables auxiliares los que vayamos a necesitar después. Para la combinación, usamos un carácter delimitador fácilmente reconocible y sin uso en la lengua española, de modo que, dado el caso, puedan separarse las partes del verso según la línea a la que correspondan y conocer el personaje que las pronuncia.

Veamos ahora el mecanismo interno de las dos subrutinas que hemos empleado. La primera, `ActualizaMetros`, compara las sílabas encontradas con una lista ordenada de las esperadas y actualiza esta última conforme al verso actual. Si el número de sílabas del verso corresponde al primer valor de la lista, la función no hace nada y devuelve la propia lista sin modificaciones. Si, por el contrario, se encuentra una discordancia, modifica la lista de la siguiente manera: si el verso es octosílabo, pone la lista a su valor inicial; si no lo es, pero tampoco es heptasílabo o endecasílabo, pone el número de sílabas encontradas en la primera posición de la lista; si no, el verso es bien endecasílabo o bien heptasílabo, por lo que pone ambos metros a comienzo de la lista de esperadas, puesto que suelen aparecer juntos, como por ejemplo en una silva.

Algoritmo 31: Escansión de los versos uno a uno.

Función `ActualizaMetros`(*encontradas, esperadas*)**:**
  si $encontradas \neq esperadas_0$ **entonces**
    si $encontradas = 8$ **entonces**
      esperadas $\leftarrow(8, 6, 11, 7, 10, 12, 5, 9, 14, 13, 4, 3, 2, 1)$
    si no, si $encontradas \notin \{11, 7\}$ **entonces**
      esperadas $\leftarrow(encontradas) \cup esperadas \setminus \{encontradas\}$
    si no
      esperadas $\leftarrow(11{,}7) \cup esperadas \setminus \{11{,}7\}$
  devolver *esperadas*

La segunda subrutina, que hemos llamado `ActualizaTabla`, recibe como argumento

la tabla a modificar y el índice de la fila correspondiente a la línea a procesar. Recibe también el texto y el personaje, ya que, de tratarse de un verso compartido, la fila solo indicaría la línea y el personaje que la pronuncia, sin aportar nada referente al resto de versos. Asimismo, le pasamos a la función la lista de metros esperados y, a continuación, empezamos a actualizar la tabla.

(15) y, en fin, si sabes que a pocos
días que hubo menester
su ingenio para instruirse,
catequizado en su fe,
hoy se baptiza; y hoy,
porque le venció o porque
le agasajó o porque uso
entre los cristianos es
poner al esclavo el nombre
del dueño, el de gran Muley
trueca en el de Baltasar,
y el apellido también
de Mahomet, su real stirpe,
en el de Loyola, a quien,
por un gran varón, cobró
amor —la causa no sé—,
¿cómo dudas que yo sienta,
sobre ser su maestro y ser
quien tan mal le dotrinó,
tan grande improperio ver
de nuestro Profeta? Y más
habiendo dado a entender
que el que quisiere seguirle
con él se quede y que el que
quiera volverse ya ahí tiene
la libertad y el bajel.

(Calderón, *El gran príncipe de Fez*, vv. 510-535)

Primero, actualizamos la columna correspondiente al personaje o conjunto de personajes que pronuncian el verso, pues es un dato que conocemos y no está sujeto a cambios. Hecho esto, limpiamos el texto para quitar espacios o puntuación superflua e introducir modificaciones *ad hoc* como, por ejemplo, tildar la última sílaba de algunas palabras que contravienen la regla cuando van en posición de rima, para ajustar la cantidad del verso. Veamos, por ejemplo, los versos 515, 523 y 533 del ejemplo 15. Los dos últimos, al ser monosílabos, no ofrecerían duda, pues consideraríamos el acento interno de la palabra que, al tener solo una sílaba, coincidiría también con la última y el verso es necesariamente agudo. El verso 515 no tiene una solución tan directa. Para escandirlo como octosílabo podríamos hacer dialefa entre *venció* y *o*, pero tendríamos una rima asonante o-e, cuando debería ser en e, como *fe* y *ser*. Así, necesitamos desplazar el acento a la última sílaba y respetar la sinalefa, de manera que tenemos un verso oxítono con siete sílabas gramaticales, que se ajusta al contexto. No es cuestión excepcional, al menos en lo que concierne a la palabra *porque*. Considérese que solo en *Eco y Narciso*, Calderón pone esta forma como aguda en doce rimas (vv. 338, 391, 591, 813, 1206, 339, 2184, 2188, 2385, 2443, 2834, y 3197).

Escandimos el verso considerando el texto y la lista de metros esperados. Con esto, actuaremos según la línea esté bien formada o no. Dicho de otro modo, intervendremos si la línea contiene algún signo que sugiera una intercalación editorial como, por ejemplo los corchetes. Representamos esta comprobación mediante la función `EsCorrecto`, que no descompondremos analíticamente, pues se limita a una serie de comparaciones sencillas y a devolver un valor booleano en función de su resultado.

Algoritmo 32: Actualización de la tabla.

Función `ActualizaTabla`(*tabla, i, texto, personaje, esperadas*)**:**
- $\mathsf{tabla}_{i,personaje} \leftarrow$personaje
- texto ←`Limpia`(*texto*)
- escansión ←slbs(texto, esperadas)
- **si** `EsCorrecto`(*text*) **es Verdadero entonces**
  - $\mathsf{tabla}_{i,síla} \leftarrow escansión_{ila}$
  - $\mathsf{tabla}_{i,núcl} \leftarrow escansión_{núcl}$
  - $\mathsf{tabla}_{i,ritm} \leftarrow escansión_{ritm}$
  - $\mathsf{tabla}_{i,cons} \leftarrow escansión_{cons}$
  - $\mathsf{tabla}_{i,ason} \leftarrow escansión_{ason}$
  - **si** $/ \in texto$ **entonces**
    - $\mathsf{tabla}_{i,ambi} \leftarrow$edición
  - **si no**
    - $\mathsf{tabla}_{i,ambi} \leftarrow escansión_{ambi}$
- **si no**
  - $\mathsf{tabla}_{i,\text{síla}} \leftarrow \text{esperadas}_0$
  - $\mathsf{tabla}_{i,núcl} \leftarrow \emptyset$
  - $\mathsf{tabla}_{i,ritm} \leftarrow \emptyset$
  - $\mathsf{tabla}_{i,cons} \leftarrow escansión_{cons}$
  - $\mathsf{tabla}_{i,ason} \leftarrow escansión_{ason}$
  - $\mathsf{tabla}_{i,ambi} \leftarrow escansión_{ambi}$
- **devolver** *tabla*

Por el contrario, el baremo para definir esas comparaciones resulta más interesante. La dificultad aquí radica en el grado de aceptabilidad de la intervención editorial. En ocasiones, la edición parte de fuentes defectuosas y el editor considera necesario corregir líneas con arreglo a su criterio filológico. Otras veces se han perdido fragmentos completos y el texto no proporciona información suficiente para completar los huecos, por lo que el editor se limita a señalar el espacio en blanco. En este último caso habría poco lugar a discusión: donde hay patrón, no manda marinero. Si el crítico no ha creído pertinente aventurar una solución, aceptaremos su decisión y marcaremos el verso

como ambiguo, indicando además el tipo de ambigüedad[122]. ¿Qué hacemos con las correcciones editoriales? En sentido estricto, no tenemos completa seguridad de que sea ese el texto del autor. Podríamos hacer como en el caso de las elipsis y marcar el verso como incompleto, pero confiaremos en el editor que, después de todo, es quien está más familiarizado con el texto; indicaremos, no obstante, la ambigüedad.

De esta manera, en Algoritmo 32, asignamos en la fila i los resultados de la escansión en las columnas correspondientes si el texto es aceptable. No obstante, cambiaremos el valor de la ambigüedad si se han encontrado, por ejemplo, corchetes. Si consideramos inválido el texto de la fila, marcaremos los valores de ritmo, sílabas y núcleo silábicos como dudosos. No obstante, guardaremos las rimas si disponemos de la última palabra del verso.

9.3 Escansión del verso

La escansión del verso la provee la biblioteca `libEscansión`[123]. Esta contiene algoritmos para, siguiendo una cadena de pasos, extraer la información métrica del verso. Primero, se toma el verso y se separa en palabras y estas, a su vez, en sílabas. Asimismo, se analiza el verso morfosintácticamente para discriminar entre las palabras átonas y las tónicas según la prosodia de la frase. Esto se hace mediante la evaluación de la categoría gramatical de cada palabra y su posición respecto a otras. Las palabras, divididas en sus sílabas métricas y con los acentos prosódicos marcados, se cotejan con la lista de metros esperados y la rima y, después, se aplican las licencias oportunas para que se ajuste al metro más plausible según los precedentes; si no es posible, se van probando sucesivamente los siguientes metros hasta que uno de ellos satisfaga la escansión del verso.

Necesitamos dos bibliotecas para llevar a cabo este proceso. Por un lado, una que analice fonológicamente las palabras y, por otro, una que proporcione información morfosintáctica sobre las que forman el verso. La primera, la hemos visto en el capítulo anterior. La segunda, se escapa con creces del propósito de este trabajo, por lo que la

[122]Aquí la describimos explícitamente por claridad, pero en la implementación en Python hemos optado por asignar simplemente un valor numérico a cada tipo de ambigüedad para agilizar la escritura del código y la visibilidad de la tabla.

[123]En la implementación en Python (Sanz-Lázaro, 2023b), hemos conservado también la ortografía española *escansión* para nombrarla en la documentación, pero le hemos añadido el prefijo *lib* y no *bib* para respetar la convención internacional de la nomenclatura. Por esta misma razón, el archivo se llama *libEscansion.py*, sin acento.

daremos por hecho en las explicaciones y emplearemos la biblioteca ficticia PLN[124] para ilustrar el proceso allí donde sea necesario[125].

La biblioteca `libEscansión` provee varias clases, pero solo dos de ellas son algorítmicas. Además, emplearemos clases de datos a modo de *contenedor* de variables (Algoritmo 34) tanto en la implementación en el computador como aquí. Aunque no es estrictamente necesario, resulta conveniente para evitar manipular por separado todas las variables que hacen referencia a una entidad única —una palabra o un verso—; de esta manera, agrupamos las sílabas del verso, núcleos vocálicos, asonancia, consonancia, número de sílabas, etc. en una variable compleja que contiene todas las simples. Si bien el impacto en el desarrollo del código no es grande, evita operaciones repetitivas, por lo que el algoritmo resulta más legible.

Algoritmo 33: Declaración de constantes y clase auxiliar.

```
Importar(PLN, fonemas)

sospechosos_habituales ←{/xueθ/, /suab/, /ruido/, /kɾuel/, /fiel/, /ruina/,
    /diabl/, /dios/, /kae/, /rios/, /biɾtuos/, /kɾio/, /ʰuid/ /poɾfiad}
perceptibilidad ←{⟨a⟩: 7, ⟨o⟩: 4, ⟨e⟩: 1, ⟨i⟩: -2, ⟨u⟩: -5,
    ⟨+TÓN⟩: +1, ⟨-SIL⟩: -1 }
trapecio ←{⟨i⟩: (-1, 1), ⟨e⟩: (-1, 0), ⟨a⟩: (0, -1), ⟨u⟩: (1, 1), ⟨o⟩: (1, 0)}
vocoides ←{/w/, /j/, /a̯/, /e̯/, i̯/, /o̯/, u̯/}
cerradas ←{⟨i⟩, ⟨u⟩}
abiertas ←{⟨a⟩, ⟨e⟩, ⟨o⟩}
vocales ←cerrados ∩ abiertos
vocálicos ←vocales ∩ vocoides
```

Empezamos, pues, por importar las bibliotecas requeridas. Seguidamente, definimos una variable que hemos llamado *sospechosos_habituales*, en la que guardamos aquellas palabras tendentes a ser objeto de diéresis en la poesía de la época (Llamas Martínez, 2020, pp. 55-64). A continuación, creamos un diccionario en el que asignamos a cada vocal un valor de acuerdo con su perceptibilidad (Navarro Tomás, 2004, sec. 25), así como correcciones si la vocal es tónica, por lo que sería más perceptible o no silábica, por lo que perdería perceptibilidad (Figura 9.2). Asimismo, definimos categorías de vocales abiertas y cerradas, cuya suma será el conjunto de las vocales,

[124]Es el acrónimo castellano *procesamiento del lenguaje natural*, traducción del inglés *Natural Language Processing* (*NLP*).

[125]En la implementación de Python hemos usado Stanza (Qi y col., 2020), pero podría emplearse cualquier otra biblioteca diseñada para el mismo propósito sin necesidad de alterar nuestro programa sustancialmente. Sea como fuere, aunque lo hayamos obviado en esta abstracción, con las bibliotecas necesitamos indicar el idioma del texto y, posiblemente, escoger un modelo lingüístico, así como definir cuáles de los análisis posibles deben llevarse a cabo.

así como los vocoides que, junto con la categoría anterior, forman otro conjunto que denominaremos, *grosso modo*, vocálico.

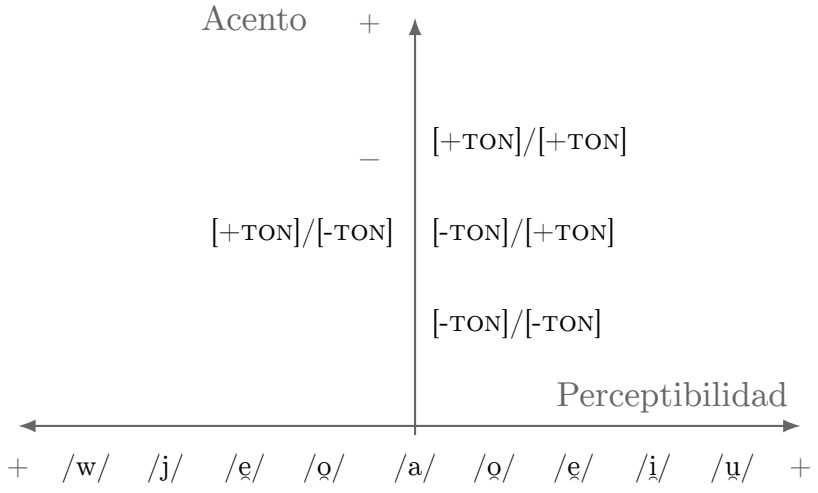


Figura 9.2: Perceptibilidad.

Para ayudarnos en la evaluación de uniones vocálicas, definimos una nueva variable. Esta la usaremos para establecer la posición de las vocales. Para ello, asignamos coordenadas a cada fonema según su punto de articulación en la cavidad bucal (Figura 9.3). Con esta configuración, la posición más central con apertura media sería el origen, el eje de abscisas representaría la frontalidad y la apertura se distribuiría a lo largo del eje de ordenadas. Estas coordenadas servirán más adelante para ordenar los desplazamientos entre fonemas. Hemos encontrado que un sistema de coordenadas ortonormal con valores aproximados de las posiciones produce buenos resultados. A la luz de los experimentos llevados a cabo para optimizar los resultados, las coordenadas $/i/ = (-1, 1)$, $/e/ = (-1, 0)$, $/a/ = (0, -1)$, $/u/ = (1, 1)$ y $/o/ = (1, 0)$ son suficientes para conseguir la escansión óptima del corpus de sonetos con el que hicimos las pruebas.

Nótese que los valores provistos no representan de manera exacta la relación con la articulación física de los sonidos. En otras palabras, no estamos asignando un valor numérico realmente, sino ordenando los pares vocálicos según la distancia entre sus componentes. Asimismo, la marcas solo resulta relevante a igualdad de condiciones entre varias uniones, como voto de calidad para decidirse por una u otra.

Además de estas variables requeridas por los algoritmos, definiremos otras por conveniencia, con las que nombrar colectivamente a vocales, vocoides y vocales cerradas. De este modo, nos referiremos después a ellas sin necesidad de volver a nombrarlas de nuevo de manera explícita.

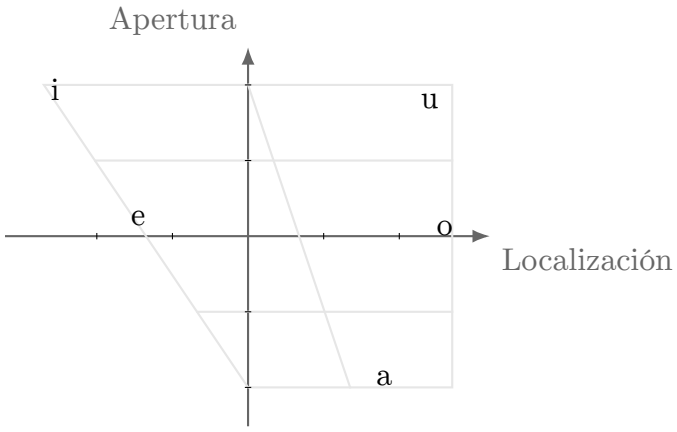


Figura 9.3: Vocales en el trapecio bucal.

A continuación definimos las clases auxiliares (algoritmo 34). La primera de ellas, a la que llamaremos clase `Palabra`, define objetos cuyos atributos son rasgos de la palabra. Estos son la propia palabra, su categoría gramatical, una lista de sus sílabas transcritas fonológicamente, rasgos sintáctico-gramaticales adicionales —persona y caso, por ejemplo— y si es tónica. La segunda, `Verso`, contiene la información que permite caracterizar el verso: sus sílabas, el tipo de ambigüedad, el recuento silábico y la rima, tanto asonante como consonante. Las sílabas están marcadas para distinguir entre vocales silábicas y no silábicas, de manera que representa implícitamente los núcleos vocálicos del verso y los acentos prosódicos, esto es, la caracterización propia del ritmo.

Algoritmo 34: Clases auxiliares.

```
Clase Palabra():
    escadena de caracteres
    categoría es cadena de caracteres
    fonemas es lista
    rasgos es diccionario
    dep es lista
    tónica es booleano

Clase Verso():
    slbs es lista
    ambi es entero
    recuento es entero
    asonancia es cadena de caracteres
    consonancia es cadena de caracteres
```

9.4 Prosodia gramatical

Con esto, abordaremos la primera clase funcional, que servirá para definir la prosodia del verso sin considerar las licencias poéticas. La definición es simple. La clase `Frase` recibe una línea de texto que preprocesa[126] para obtener el verso final ya listo para trabajar con él. En este, identificamos los rasgos de las palabras y, apoyándonos en ellos, buscamos los acentos prosódicos. El resultado lo dejamos en una variable que contiene una lista de palabras, cada una de las cuales es, a su vez, una lista de sílabas. Estas están representadas por su transcripción fonética. En cuanto a la prosodia, la sílaba acentuada no se marca en todas las palabras sino solo en las tónicas.

Algoritmo 35: Clase `Frase`.

```
Clase Frase():
    Método inicializa(línea):
        versolisto ←Preprocesa(línea)
        palabras ←EncuentraAcento(versolisto)
```

Veamos en qué consisten los métodos de la clase. En el preprocesado, alteramos las palabras de manera que resulte más conveniente su análisis. Por ejemplo, las palabras homógrafas de clases tónicas y átonas pueden ser analizadas por el programa incorrectamente, lo que provoca errores en el análisis del ritmo. Entre dos palabras átonas o entre dos tónicas, el problema es irrelevante, en tanto que no afecta a la prosodia desde nuestro punto de vista[127]. Así, empezaremos haciendo algunas correcciones *ad hoc*. Por ejemplo, el imperativo singular de *parar* plantea una situación problemática en tanto que el programa de análisis morfosintáctico tiende a confundirlo con la preposición. No obstante, tenemos al menos un caso del que se infiere la categoría gramatical gracias a la puntuación. En español, la preposición es el núcleo sintáctico de un sintagma preposicional que no suele romperse con la puntuación salvo para introducir apartes (16). Dadas las limitaciones estructurales que impone el verso, su empleo resulta más inusual que en el texto narrativo, Además, si el aparte no es retórico, este se señala mediante una acotación. Por lo tanto, si encontramos un *para* seguido de coma y un sintagma nominal, asumiremos que, probablemente, se trate del imperativo de *parar* con la coma propia del vocativo.

[126] En la implementación en Python, el resultado de preprocesar la línea se somete a un paso adicional antes de ser utilizado, ya que detectamos errores puntuales en el módulo de análisis morfosintáctico que conviene enmendar. Sirva esto también como elogio a la labor de los miembros del proyecto Stanza NLP, quienes resolvieron con premura todos los fallos que reportamos.

[127] Esto es una idealización en la que solo tenemos sílabas tónicas y átonas. En realidad, existe una gradación de la intensidad de la acentuación, por lo que existen diferencias entre dos palabras acentuadas o dos átonas. En esta diferenciación sí influye la categoría gramatical de la palabra y su función sintáctica.

(16) Para, digamos, eso.

Si se trata de esto, es posible engañar al programa haciéndole creer que la palabra es un nombre propio, por lo que será marcada como tónica. No supondría un problema que, como es común, el vocativo fuera otro nombre propio, ya que, al ir separados por coma el nombre verdadero y el impostado, no sería entendido como un nombre compuesto y, por lo tanto, el primero no perdería su calidad tónica.

Algoritmo 36: Preprocesado.

Método `Preprocesa(`*línea*`):`
  línea(⟨*para,*⟩) ←⟨Ppara,⟩, ∀⟨para⟩ ∈ *línea*
  símbolos ←{⟨(⟩: ⟨. ⟩, ⟨)⟩: ⟨. ⟩, ⟨—⟩: ⟨. ⟩, ⟨...⟩: ⟨. ⟩, ⟨«⟩: ⟨⟩, ⟨»⟩: ⟨⟩, ...
    ⟨õ⟩: ⟨o⟩, ⟨æ⟩: ⟨e⟩, ⟨à⟩: ⟨a⟩, ⟨ì⟩: ⟨e⟩, ⟨ì⟩: ⟨i⟩, ⟨ò⟩: ⟨o⟩, ⟨ù⟩, ...}
  si *línea* **es** ∅ **entonces**
    verso ←⟨∅⟩
  si no
    línea ←`CorrigeOrtotipografía(`*línea*`)`
    línea ←`PLN(`*línea*`)`
    verso ←$(palabra_0, ..., palabra_n), \forall palabra_i \in$ linea
  devolver `FijaRasgos(`*verso*`)`

A continuación, corregimos la ortotipografía para no tener que considerar situaciones excepcionales. Esto incluye asegurarse de que el espaciado sea apropiado —por ejemplo, que no haya dos espacios seguidos o entre una palabra y la puntuación de cierre o, por el contrario, ponerla si falta entre la puntuación de cierre y el inicio del siguiente grupo de palabras. Asimismo, simplificamos los signos ortográficos, tanto la puntuación como los diacríticos, para reducir el número de casos a considerar durante el procesado[128]. También transliteramos aquí marcas diacríticas ajenas a la ortografía española.

Solo resta aplicar el procedimiento de análisis morfosintáctico y almacenar los resultados en una lista ordenada según la posición de las palabras en el verso. Dado que recurrimos a una biblioteca de análisis morfosintáctico externa, necesitamos adaptar el formato a nuestras necesidades, por lo que introducimos un módulo `FijaRasgos` para dejar la salida de la subrutina en una forma más manejable. De nuevo, este módulo se trata de una abstracción que debe concretarse de manera diferente para cada caso específico[129]. Se trata de, a partir de la lista de palabras obtenida, componer

[128]En el algoritmo propuesto, hemos circunscrito el diccionario de sustituciones a una muestra porque la intención es ilustrar la idea y no detallar todos los casos. No obstante, se encuentra una forma más completa en la implementación para el computador en Apartado B.3.

[129]Por ejemplo, en Apartado B.3 para la biblioteca `Stanza`.

otra en la que cada uno de sus elementos sea un objeto de la clase `Palabra` con los rasgos necesarios.

Una vez dispuesta cada palabra del verso para que exhiba la información relevante, establecemos la prosodia del verso con arreglo a los acentos internos y la morfosintaxis de las palabras que lo componen. Este paso es crítico y, de su correcta solución junto con la evaluación de las sinalefas, depende determinar el ritmo métrico satisfactoriamente en la mayoría de los versos. Asimismo, es un paso que entraña ciertas dificultades de carácter técnico —al contrario que la resolución de uniones vocálicas, que es conceptual, como veremos—. La razón de las posibles complicaciones es la necesidad de categorizar las palabras del verso y considerar sus relaciones sintácticas. Si bien existen herramientas que automatizan la tarea, estas no son del todo precisas. Por lo tanto, aquí habremos de introducir una serie de soluciones al efecto para sortear fallos en el análisis gramatical. Lamentablemente, esto obliga a complicar este paso para contemplar una miríada de casos excepcionales que, de otra manera, habrían podido resolverse de una forma más elegante.

Algoritmo 37: Búsqueda de acentos prosódicos del verso (I).

```
Método EncuentraAcento(palabras):
    categorías_tónicas ←{a_i}_{i=1}^n, a_i ∈ categorías tónicas
    átonas ←{b_i}_{i=1}^n, b_i ∈ palabras átonas
    tónicas ←{c_i}_{i=1}^n, c_i ∈ palabras tónicas
    anterior ←∅
    unidades ←Falso
    /* Algoritmo 38 */
```

Aquí debemos crear listas de categorías gramaticales y palabras átonas y tónicas de acuerdo con lo expuesto en el capítulo que dedicamos a los aspectos teóricos de la métrica. Asimismo, definimos una variable booleana *unidades*, que emplearemos, en caso de encontrar numerales, para identificar los órdenes de magnitud superior a la unidad en palabras compuestas.

A continuación, procedemos a recorrer las palabras del verso en sentido inverso; esto es, desde la última a la primera. La razón para hacerlo así y no en su orden natural de lectura es que, en caso de compuestos de elementos de idéntica categoría, como numerales o nombres propios, el acento recae en la última palabra y son átonas las precedentes (Algoritmo 38). Por lo tanto, partimos de que la primera palabra que encontremos de las categorías susceptibles de agrupación es tónica con independencia de que forme parte de una estructura más compleja de palabras de su clase. De seguir la disposición natural del verso, al dar con una palabra semejante no podríamos

determinar de antemano si lleva o no acento, ya que se haría necesario cotejarla con las palabras subsiguientes para dilucidar si forma parte de una composición.

Partimos de la premisa de que la palabra es átona; esta suposición variará en función de las pruebas a las que la someteremos. En Algoritmo 38, definimos por conveniencia algunas variables descriptivas de los valores que les asignamos. Estas no son necesarias desde una perspectiva funcional, pero mejoran la legibilidad. Se comprueba primero si la palabra es la última del verso —la primera en nuestra lista de palabras invertida— , contiene vocales con tilde o es alguna de las que hemos definido como tónicas de manera inequívoca. En ambos casos, habremos de concluir que la palabra es tónica. De no cumplirse ninguno de las dos, constatamos que la palabra pertenezca al conjunto de átonas que definimos y, de ser cierto, no necesitamos hacer nada más salvo asignar un valor vacío a la palabra anterior para la próxima iteración, pues ya habíamos marcado la palabra como átona por defecto. Si tampoco es el caso, necesitaremos explorar los rasgos gramaticales de las palabras y su contexto sintáctico para encontrar sus características prosódicas. Dado que es necesario considerar un número significativo de posibilidades, lo detallaremos en un algoritmo aparte.

Algoritmo 38: Búsqueda de acentos prosódicos del verso (II).

```
palabras ←{b_1}_{i=1}^{n}, b_1 = a_{(n+1)−i}, a_i ∈ palabras
para palabra_i ∈ palabras hacer
    tónica ←Falso
    texto ←palabra_texto
    categoría ←palabra_categorí
    rasgos ←palabra_rasgos
    si i = 0 ∧ ∃x ∈ texto, x ∈ {á, é, í, ó, ú} ∧ texto ∈ tónicos entonces
        tónica ← Verdadero
    si no, si texto ∈ átonas entonces
        palabra_0 ← ⟨⟩
    si no
        /* Algoritmo 39 */
    palabra_tónica ←tónica
    categoría _0 ←categoría
```

Empezamos por marcar por defecto como acentuadas las palabras pertenecientes a categorías que, en principio, consideramos tónicas (Algoritmo 39). No obstante, modificaremos después esta clasificación preliminar si el contexto de la palabra lo requiere. Para ello, iniciamos una serie de comprobaciones, empezando por determinar si la palabra es un numeral. Sabemos que en números compuestos solo es tónica la última parte. Esto es, pongamos que tenemos, por ejemplo el número 333, la transcripción será /tɾes.θjen.tos tɾeɪ̯n.taɪ̯ ˈtɾes/. Debido a eso, empezamos por asegurarnos de que

la palabra anterior —recordemos que hemos invertido el orden, por lo que la palabra precedente en nuestro orden de verificación es la que sigue en la oración— no sea numeral. Si no lo es, la marcamos como tónica y, si es del orden de magnitud de las unidades, lo indicamos. La razón para esto es evitar que dos numerales independientes se evalúen como una agrupación. Esto es, por ejemplo, *uno, dos y tres*, al contrario que un número compuesto, tendría todas sus palabras acentuadas. En teoría, esto no está circunscrito a las unidades, ya que lo mismo sucedería con, por ejemplo *veinte y treinta*. Sin embargo, si los primeros casos son escasos, los segundos lo son más. Para refinar la clasificación podríamos proponer comprobar si estamos ante órdenes de magnitud equivalentes (*uno y dos*). Sin embargo, casos como *cuarenta y dos* necesitarían contextualización semántica. Esto es, estaríamos ante un único grupo con acento en *dos* si, por ejemplo, una persona está dando su edad, pero serían dos grupos acentuados si el interpelado se refiere a la suya (40) y la de su hijo (2). Asimismo necesitamos recurrir a la ortografía para diferenciar *trescientos treinta y tres* de *trescientos, treinta y tres*, la primera con un solo acento en *tres* y la segunda con los tres componentes acentuados al comenzar. Esto es empero discutible, ya que se cumple si aludimos a *333*, pero no si representa *300, 33*, en cuyo caso *treinta* sería átona, o *330 y 3*, en la que *trescientos* sería átona, pero *treinta* y *tres* tónicas

No todos los componentes de la unidad han de ser numerales por sí mismos. Por eso la siguiente comprobación es si la palabra es la conjunción *y* antes de numeral, lo que la convertiría en parte del grupo, no embargante lo dicho en el párrafo anterior. Si es así, forzamos su clasificación a numeral de manera que en la próxima iteración tenga en cuenta este contexto. Si tampoco es el caso, cotejamos la palabra con el conjunto de voces necesariamente átonas y, de estar en él, la marcamos como tal.

Si no se ha cumplido ninguna de las condiciones anteriores, la prueba consistirá en verificar si se trata de un *determinante*[130] y lo marcaremos como átono si no es indefinido o demostrativo. Si tampoco se da esta situación, comprobamos a continuación si se trata de un pronombre. De ser así, se hace necesaria una distinción más detallada. Primero miramos si es un pronombre objeto o reflexivo, en cuyo caso será átono. En el caso de que tuviéramos un verbo con pronombres clíticos, el etiquetador gramatical analizaría sus elementos por separado. Si se trata de un objeto indirecto o directo que no es parte de un sintagma preposicional, asumimos que es un pronombre

[130]En la implementación en Python, se confunde hasta cierto punto esta categoría con los pronombres. El programa encargado del etiquetado morfosintáctico cometía algunos errores, por lo que fue necesario introducir cambios para tenerlos en cuenta. No se entienda esto como una toma de postura a favor de la hipótesis del sintagma determinante (Bernstein, 2001) frente al nominal —una controversia sobre la que no tenemos elementos de juicio y dejamos a los expertos que sí los tienen—, sino como una mera cuestión práctica. En esta aproximación idealizada, por el contrario, damos por supuesto que el lector no cometerá esos fallos y nos atendremos a la gramática.

Algoritmo 39: Búsqueda de acentos prosódicos del verso (III).

```
si categoría ∈ categorías_tónicas entonces
    tónica ← Verdadero
si categoría = 'numeral' entonces
    si anterior ≠ 'numeral' entonces
        tónica ←Verdadero
        si texto ∈ unidades entonces
            unidad ←Verdadero
        si no
            unidad ←Falso
    si no, si texto ∈ unidades entonces
        si unidad es Verdadero entonces
            tónica ←Verdadero
        unidad ←Verdadero
    si no
        unidad ←Falso
si no, si texto = ⟨y⟩ ∧ anterior es numeral entonces
    categoría ←'numeral'
    tónica ←Falso
si no, si texto ∈ átonas entonces
    palabra_0 ← ⟨⟩
/* Algoritmo 40 */
```

Podría ser que tuviéramos un verbo con clítico clasificado como tal y, para ello, verificamos si la palabra acaba en un pronombre átono, pero sin ser toda ella uno, de manera que, si se cumple, tendremos un compuesto cuya primera parte es tónica. Otra posibilidad es que estemos ante un posesivo. En ese caso, debemos distinguir la clase, pues los artículos son átonos mientras que los pronombres son tónicos aun siendo homógrafos. Introduciremos también una condición que obligue a ignorar las operaciones subsiguientes en caso de tener un indefinido o un demostrativo, ya que estos son tónicos. Si la palabra no satisface las condiciones anteriores, miramos si actúa como objeto directo o indirecto y, de ser así, si es un pronombre átono. A continuación, buscamos amalgamados *con + migo/tigo/sigo* y, tras esto los relativos. En cualquier otro caso, la palabra permanece marcada como tónica.

La siguiente comprobación incluye sustantivos o fórmulas de tratamiento, cualesquiera que sean sus categorías gramaticales; esto es, incluimos aquí formas sustantivas como *don* o *doña* pero también sustantivos como *santo* o su variante apocopada *san*. La razón es que, en combinación con nombre propio, reproducen el comportamiento prosódico de otros nombres propios antepuestos: son átonos. Al igual que los numerales, solo el último nombre propio de una forma compuesta lleva acento. Como estamos leyendo el verso desde el final hacia el principio, crearemos una variable que indique si la palabra examinada anteriormente es un nombre propio, de manera que si este

Algoritmo 40: Búsqueda de acentos prosódicos del verso (IV).

```
si no, si categoría es determinante entonces
    si rasgos_tipo es ¬demostrativo ∨ rasgos_tipo es ¬indefinido entonces
        tónica ←Falso
si no, si categoría es pronombre entonces
    si rasgos_caso es objeto entonces
        si texto ∉ pronombres átonos entonces
            tónica ←Verdadero
        si no
            tónica ←Falso
    si no, si '-igo' ∈ texto entonces
        tónica ←Verdadero
    si no, si texto = ⟨donde⟩ ∨ rasgos_tipo ∈ {Int, Rel} entonces
        tónica ←Falso
/* Algoritmo 41 */
```

es el caso, la que estemos analizando tras ella será átona. En cualquier otro caso, no necesitaremos cambiar nada, ya que el sustantivo será palabra tónica.

Si no se ha cumplido tampoco la condición previa, la siguiente a comprobar es si la palabra es un adverbio. En caso de que efectivamente lo sea, la marcaremos como átona si funciona como conjunción o es parte de una compleja (Quilis, 1984/2013, p. 24).

Algoritmo 41: Búsqueda de acentos prosódicos del verso (V).

```
si no, si categoría es sustantivo ∨ texto ∈ tratamientos entonces
    si texto ∈ tratamientos ∧ anterior es nombre propio entonces
        tónica ←Falso
        anterior ←∅
si no, si categoría es adverbio entonces
    si texto es adverbio y texto ∈ adverbio entonces
        tónica ←Falso
anterior ←categoría
/* Algoritmo 42 */
```

Una vez hechas todas las comprobaciones, procedemos a marcar la conjunción *y*, de manera que podamos reconocerla en la transcripción fonológica aun sin disponer de información gramatical.

Tenemos una buena estimación del ritmo del verso, pero esta presenta algunas deficiencias. Por una parte, hemos analizado las palabras de manera individual y, por otra, no hemos tenido en cuenta las demandas del metro. En la lengua oral, las palabras no son independientes unas de otras, sino que producen un sonido continuo que difumina sus márgenes. Aquí volvemos a las nociones de fonética acústica que vimos en la parte teórica.

Algoritmo 42: Tratamiento de la conjunción *y*.

palabras $\leftarrow\{b_1\}_{i=1}^{n}, b_1 = a_{(n+1)-i}, a_i \in$ palabras
para $palabra_i \in palabras$ **hacer**
  si $i < n-1 \vee palabra_{texto} = \langle y \rangle$ **entonces**
    si $x \in vocales \forall x \in \{palabras_{i-1,fon._{n,m}}, palabras_{i+1,fon._{0,0}}\}$ **entonces**
      palabras $_{i,fon._0} \leftarrow \langle y \rangle$
devolver $\{palabra_i\}_{i=1}^{n}, \forall palabra \in palabras$

Si tomamos el fonograma de un enunciado, observamos fenómenos coarticulatorios entre palabras. Así, las sílabas fonéticas de, por ejemplo, *álbum abierto*, no se corresponden con las gramaticales (*al·bum·a·bier·to*) sino que se pronunciará como (*al·bu·ma·bier·to*). Si bien la unión entre vocal y consonante no tiene consecuencias en la escansión del verso, el contacto intervocálico altera a menudo la pronunciación. La sinalefa, cuyas características mencionamos en el capítulo dedicado a la fonología, y su efecto en el verso, en el dedicado al metro, reduce de forma natural la medida del verso. Por otra parte, si el metro lo requiere, el poeta ajustará el número de sílabas forzando diptongos mediante sinéresis o hiatos mediante diéresis. Así pues, debemos considerar estos fenómenos tomando las sílabas del verso como un flujo continuo y no como una serie de palabras aisladas e independientes.

9.5 Ajustes métricos

Para hacer esto, definimos una nueva clase `MetroVerso` hija de `Frase` que, por lo tanto, hereda los métodos y atributos de esta. Recordemos: *versolisto* y *palabras*, dos listas en el que la primera contiene información gramatical de cada palabra del verso y la segunda listas correspondientes a las sílabas de cada una de las palabras con los acentos prosódicos ya marcados. Asimismo, definimos una lista ordenada con metros, que se usará en caso de que no se proporcione de manera explícita el parámetro *esperadas* para establecer el orden de preferencia de los metros para los que se intentará escandir el verso. Esto es, intentaremos ajustar el verso para que tenga tantas sílabas como indica el primer elemento de la lista; si no es posible, como el segundo; después el tercero, etc.

Antes de definir los atributos de la clase, creamos una condición para asegurarnos de que el verso contiene al menos una palabra. Esto permite atajar ya aquí esta contingencia para evitar arrastrarla en el proceso, lo que complicaría su localización y obligaría a hacer excepciones más adelante.

Los atributos de la clase son las sinalefas, representadas en una lista de estas, así como la prioridad de cada una, obtenida de aplicar un método al efecto. Asimismo obtenemos una estimación de la longitud del verso en condiciones normales, esto es, haciendo las sinalefas obligatorias o, en nuestro caso, aquellas con una prioridad que supere un umbral determinado, como veremos más adelante. Otro método obtiene las características prosódicas del verso, a saber: las sílabas que lo componen, su rima y valores derivados como el número de sílabas y la rima parcial. Por último extraemos los núcleos silábicos del verso y traducimos estos a una notación binaria en función de si llevan o no acento para representar el patrón rítmico.

Algoritmo 43: Ajustes métricos.

```
Clase MetroVerso(Frase):
    comunes ←(6, 7, 8, 11, 9, 10, 12, 13, 5, 4, 3)
    Método inicializa(línea, esperadas ←comunes):
        si ∃palabra ∈ palabras entonces
            sinalefas ←EncuentraSinalefas(palabras)
            estimación, esperadas ←AjustaEsperadas(palabras, sinalefas,
              esperadas)
            verso ←AjustaMetro(palabras, esperadas)
            sílabas ←verso.sílabas
            ambigüedad ←verso.ambi
            asonancia ←verso.asonancia
            rima ←verso.consonancia
            cuenta ←cuenta.recuento
            núcleos ←EncuentraNúcleos(sílabas)
            ritmo ←EncuentraRitmo(núcleos)
        si no
            sinalefas, esperadas, sílabas ←[]
            estimación, cuenta ←0
            ambigüedad, asonancia, rima ←Falso
            núcleos, ritmo ←∅
```

Desglosemos antes de nada las funciones que hemos utilizado aquí, para, una vez tengamos las partes principales, hacer lo propio con las subrutinas que las componen y explorarlas analíticamente. Hacemos una estimación del número de sílabas del verso. Para ello necesitamos antes la cuenta de contactos vocálicos entre palabras para conocer las sinalefas potenciales. Si bien es posible que se dé una dialefa entre límites vocálicos de palabra adyacentes, hay casos en los que, como vimos en la parte teórica, esto contravendría la pronunciación natural del verso. Tales pares de sílabas deben de ser considerados *a priori* como una sola sílaba métrica y, solo de manera excepcional, se escandiría el verso atendiendo a su división silábica gramatical. Por lo

tanto, definimos un método dedicado a analizar el verso en busca de sinalefas posibles (Algoritmo 44).

Algoritmo 44: Búsqueda de vocales en contacto (I).

```
Método EncuentraSinalefas(palabra, h ←Falso):
    sinalefas ←()
    preferencia, desplazamiento ←0
    para palabra_i ∈ palabras hacer
        V, C, s ←palabra_0, palabras_{i−1,n}, Falso
        si i ≠ 0 ∧ ∀x ∈ {C_n, V_0}, x ∈ vocálicas entonces
            si i = 1 ∧ palabras_0 ∈ {i, o} ∧ V_0 ∈ vocales^{+TÓN} entonces
                preferencia ←preferencia − 8
            posición ←(idx − 1, |palabras_{idx−1} − 1)
            si palabra ∈ {(i), (e)} ∧ |palabras| > idx + 2 entonces
                si C_n ∈ vocálicas entonces
                    si palabras_{idx+1,0,0} ∈ vocálicas entonces
                        s ←Verdadero
            si no, si fonema ∈ vocálicas, ∀fonema ∈ {C_n, V_0} entonces
                perc ←perceptibilidad_{C_n}
                si |C| > 1 ∧ C_{−2} ∈ perceptibilidad entonces
                    previous_val ←perceptibilidad_{C_{−2}}
                si no
                    anterior ←perceptibilidad_{C_n}
                si |V|> 1 ∧ V_1 ∈ perceptibilidad entonces
                    siguiente ←perceptibilidad_{V_1}
                si no
                    siguiente ←perceptibilidad_{V_0}
                si anterior ≤ perc ∧ perc ≤ perceptibilidad_{V_0} ∨ anterior ≥
                  perc ∧ perc ≥ perceptibilidad_{V_0} ∧ siguiente ≤
                  perceptibilidad_{V_0} ∨ anterior ≤ perc ∧ perc >
                  perceptibilidad_{V_0} ∧ perceptibilidad_{V_0} ≥ siguiente entonces
                    s ←Verdadero
        si s entonces
            si ∀x^{−TÓN} ∈ {C, V_0} ∧ C_n = V_0 entonces
                si ∃x ∈ {['o'], ['y'], ['u'], ['e']} ∧ x ∈ {palabra, palabras_{i−1}}
                  entonces
                    preferencia ←preferencia − 1
                si no, si |V| > 1 ∧ V_1 ∈ {vocoides} entonces
                    preferencia ←preferencia − 2
            sinalefas ←sinalefas + (posición, PreferenciaSinalefa(C, V,
              preferencia), C, V)
        desplazamiento ←desplazamiento + |palabra|
    /* Algoritmo 45 */
```

Aquí se trata de componer una lista de posiciones donde es factible la sinalefa. Asimismo, hay que definir otra lista para contener la palabra precedente a la que estamos

evaluando. Antes de hacerlo, inicializamos a cero la preferencia de la sinalefa y la cuenta del desplazamiento dentro del verso. Con esto, empezamos a recorrer las palabras, que, recordemos, representábamos a su vez como listas de sílabas. Para simplificar, pensemos en la sinalefa como dos segmentos, uno inicial, compuesto de una posición no silábica más una nuclear, y uno final, compuesto de una posición nuclear junto a una no silábica, como la restricción de que una y solo una de las posiciones nucleares estará ocupada y ninguna de las partes estará completamente vacía. La posición nuclear solo acepta vocales puras, mientras que las no silábicas solo aceptan vocoides. Los posibles segmentos son $CN-V$, $C-NV$, $N-V$, $C-N$, donde V representa el segmento vocálico y C el semiconsonántico o semiocálico, sin preocuparnos ahora de cuál de ellos contiene el núcleo. La primera condición que establecemos es que, obviamente, la sinalefa no puede darse en la sílaba inicial del verso, pues no hay palabra precedente con la que pueda formarla. En cualquier otro caso, la palabra actual debe comenzar por sonido vocálico, como también vocálico debe ser el final de la anterior. Si esto es cierto, confirmamos que se trata de una sinalefa potencial para asignarle un rango de preferencia.

Primero marcamos la posición donde comienza la sinalefa, esto es, en la última sílaba de la palabra anterior. Para representarlo usaremos un binomio cuyo primer miembro es la palabra y el segundo la sílaba dentro de esta. A continuación, hacemos algunas verificaciones para tratar las conjunciones. Así, vemos si estamos en la segunda palabra del verso y esta va precedida de las conjunciones *o* o *y* y, además, la palabra actual comienza con sílaba tónica. Decrementaremos la preferencia si este es el caso.

Algoritmo 45: Búsqueda de vocales en contacto (II).

```
/* Viene de Algoritmo 45 */
para palabra_i ∈ palabras hacer
    si |palabra| > 1 entonces
        preferencia ←-12
        para sílaba_j ∈ palabra hacer
            si j > 0 entonces
                posición ←(i, j − 1)
                V ←sílaba
                C ←palabra_{j−1}
                si x ∈ vocálicas, ∀x ∈ {C, V_0} ∧ ¬(∃C^{+TÓN} ∧ idx + 1 =
                 |palabras| ∧ j + 1 = |word|) entonces
                    sinalefas ←sinalefas + (posición,
                     PreferenciaSinalefa(C, V, preferencia), C, V)
devolver DeMayorAMenor(sinalefas)
```

Si nos encontramos ante una conjunción copulativa y no hemos llegado a la última

sílaba, vemos si la palabra anterior acaba en sonido vocálico y, de ser cierto, marcaremos como objeto de una potencial sinalefa la unión de la palabra que estamos evaluando con la precedente. Si, por el contrario, no es una conjunción copulativa, pero es posible una sinalefa con la palabra precedente, tomamos el valor de la perceptibilidad del último fonema de la palabra anterior. Dado que la perceptibilidad de los fonemas de una unión vocálica ha de ser descendente en ambos sentidos desde el núcleo, hemos de verificar que la secuencia de perceptibilidad es válida para contemplar la sinalefa como aceptable. Para esto consideramos en la primera sílaba los valores de su última vocal y de la penúltima si la hubiera, así como la primera vocal de la segunda sílaba y la segunda de haberla. Si la perceptibilidad de esas vocales es ascendente, descendente o asciende y desciende, la unión vocálica es válida, por lo que marcaremos la sinalefa potencial como posible.

En caso de que se haga la sinalefa, corregiremos la preferencia para el caso especial de que ambas sílabas sean átonas y la unión sea de fonemas iguales. Si esta es la situación, disminuiremos la preferencia si una de las palabras implicadas es una conjunción. Si una de las sílabas es un diptongo, reduciremos más para primar las uniones simples sobre los triptongos. Ahora sí, añadimos a la lista de sinalefas la posición y el resultado de aplicar la función de resolver la preferencia (Algoritmo 46). También actualizamos el desplazamiento, que tendremos en cuenta para determinar la posición de posibles sinéresis.

Una vez encontradas las uniones entre palabras, evaluamos los contactos vocálicos interiores (Algoritmo 45). Como es natural, ignoraremos las monosílabas. Además, dado que es una unión que suele ser muy dura al oído, partiremos de un valor negativo significativamente bajo. Recorremos todas las sílabas y marcamos las posiciones de contacto vocálico intersilábico en las que ambas partes sean átonas, para añadirlas después a la lista de sinalefas tras corregir la unión vocálica atendiendo a sus componentes. Devolveremos la lista de sinalefas posibles ordenadas de mayor a menor preferencia.

Recordemos que hemos mencionado una función que evalúa la unión vocálica y devuelve el valor de su preferibilidad. Veamos, por tanto, qué hacemos para determinar ese valor. Como vemos en Algoritmo 46, necesitamos tres argumentos: las dos partes de la sinalefa potencial y el valor que damos a su preferencia. En principio, este último partiría de un valor neutro, ni positivo ni negativo, pero, como ocurre en el algoritmo anterior, en determinados casos conviene ajustarlo, por lo que hemos de considerar la manera de introducir una corrección externa a la evaluación.

Empezamos por determinar la distancia en la cavidad bucal entre los fonemas en

Algoritmo 46: Evaluación de la idoneidad de las sinalefas potenciales.

Método `PreferenciaSinalefa(`*C, V, preferencia=0*`):`
- distancia $\leftarrow distancia_{C,V}$
- voc $\leftarrow$vocálicas
- C $\leftarrow x, \forall x^{+\text{VOC}} \in C$
- V $\leftarrow y, \forall y^{+\text{VOC}} \in V$
- preferencia $\leftarrow$preferencia $- 2(|C| + |V| - 2 +$ distancia$)$
- **si** $\exists V^{+ASP}$ **entonces**
  - preferencia $\leftarrow$preferencia $- 2$
- **si** $\exists x^{-TÓN}, \forall x \in \{C, V\}$ **entonces**
  - preferencia $\leftarrow$preferencia $+ 4$
- **si no, si** $\exists x^{+TÓN} \in C \cup V$ **entonces**
  - preferencia $\leftarrow$preferencia $- 2$
  - **si** $\exists x^{+ALTA,+TÓN} \in C \cup V$ **entonces**
    - preferencia $\leftarrow$preferencia $- 1$
  - **si** $\exists x^{+TÓN}, x \in C \wedge x \in V$ **entonces**
    - preferencia $\leftarrow$preferencia $- 8$
- **si** $V_0 = \in \{y,\ o\}$ **entonces**
  - preferencia $\leftarrow$preferencia $- 1$
- **si** $C_n \in \{y,\ o\}$ **entonces**
  - preferencia $\leftarrow$preferencia $+ 1$
- **devolver** *preferencia*

contacto de las dos sílabas. Como habíamos dicho más arriba, podemos ordenar los desplazamientos vocálicos según una idealización de la distancia entre los componentes del diptongo. Nos quedaremos con los valores vocálicos sin tener en cuenta las consonantes al comienzo de la primera sílaba o al final de la segunda. Además, asignamos un valor preliminar a la preferencia, de modo que penalizamos los clústeres vocálicos más grandes. Una unión resultante en diptongo no verá modificada su plausibilidad, mientras que si una de las partes está formada de una combinación vocálica previa —lo que provocaría la unión de tres o más vocales—, tendremos una preferencia negativa. Asimismo, penalizamos también la distancia entre fonemas vocálicos: cuanto mayor sea esta, menos preferible será su unión.

Con esto, comenzamos a verificar si las condiciones que proponemos se cumplen. Comenzaremos por las aspiraciones. Dado que es una situación excepcional (Quilis, 1984/2013, p. 50), conviene darle un valor bajo, de modo que su peso no sea desproporcionado frente a otros factores. De esta manera, ante dos sinalefas potenciales equivalentes, quedará la que no tiene posibilidad de ser aspirada, pero, entre sinalefas no análogas, serán los otros factores los que determinen la resolución.

Proseguimos considerando la acentuación. Recordemos que las uniones entre átonas son las más propicias y entre tónicas las menos (Navarro Tomás, 2004, secs. 138-139).

Algoritmo 47: Ajuste de sílabas esperadas.

Método `AjustaEsperadas(`*sílabas, sinalefas, esperadas*`):`
- sнlf $\leftarrow|\{a_i\}_{i=1}^{n}|, a_i \in$ sinalefas$, a_{i,j} > -15$
- slbs $\leftarrow|$`PalabrasASílabas(`*sílabas*`)`$|$ + `EncuentraRima(`$sílabas_{n,cuenta}$`)`
- slbs $\leftarrow$slbs $-$ snlf
- **si** $expected_0 = slbs$ **entonces**
  - exp $\leftarrow()$
- **si no, si** $expected_0 = 8$ **entonces**
  - **si** $slbs \in \{6, 7, 8, 9\}$ **entonces**
    - exp $\leftarrow(8)$
  - **si no, si** $slbs < 6$ **entonces**
    - exp $\leftarrow(7, 8, 6, 5, 4, 3)$
  - **si no**
    - exp $\leftarrow(11)$
- **si no, si** $expected_0 \in \{7, 11\}$ **entonces**
  - **si** $slbs < 9$ **entonces**
    - exp $\leftarrow(7, 11)$
  - **si no, si** $slbs > 9$ **entonces**
    - exp $\leftarrow(11, 7, 8)$
  - **si no**
    - exp $\leftarrow(8, 7, 11)$
- **si no, si** $expected_0 = 6 \vee slbs \in \{5, 6, 7\}$ **entonces**
  - exp $\leftarrow(6, 8, 11, 7)$
- **si no**
  - exp $\leftarrow(8, 11, 7, 6)$
- **devolver** $slbs,\ exp + \{a_i\}_{i=1}^{n}, a_i \notin exp$

Así, dos sílabas átonas en contacto tendrán un valor positivo, mientras que una átona y una tónica lo tendrán negativo. Si, además, la vocal acentuada es alta, supondremos un valor más bajo, para primar una dialefa que reproduciría entre palabras el modelo del hiato dentro de una. Si ambas sílabas fueran tónicas, restaríamos un valor considerablemente alto a la potencial unión. Por último, consideramos el caso especial de la conjunción (Canellada y Madsen, 1987, p. 59; Navarro Tomás, 2004, sec. 49), a la que daremos preferencia como semiconsonante en diptongo creciente y la penalizaremos como semivocal en decreciente. Con esto, devolvemos el resultado de estas evaluaciones.

Definimos un nuevo procedimiento para ajustar el metro esperado (Algoritmo 47). La razón de esto es que, si bien los versos de ocho, once y siete sílabas son comunes, otros metros resultan más raros. Por ello, supondremos que, si las licencias métricas no lo impiden, el metro del verso será uno de aquellos y no otro menos frecuente. De esta manera, necesitaremos las sílabas de la palabra, la lista de sinalefas que proporciona la función descrita arriba y una lista de sílabas esperadas que dé la posibilidad de forzar

un metro dado si las circunstancias así lo aconsejan. En cualquier caso, quedarán solo las sinalefas naturales, ignorando aquellas que, por su dureza, tengan un valor de preferencia bajo. Como conocemos la posición de las uniones y su preferencia, no necesitamos trabajar con palabras, por lo que *aplanamos* la lista para producir otra de sílabas y sumar a su longitud la corrección que demanda la posición del acento de la rima (Algoritmo 48). El número de sílabas natural será el resultado de restar las uniones vocálicas suaves a las sílabas ortográficas con el ajuste que exige la rima.

Algoritmo 48: Reducción de lista de palabras a lista de sílabas.

Método `PalabrasASílabas(`*palabras*`):`
devolver $(s_{0,0}, s_{0,1}, ...s_{0,n}, s_{1,0}, ..., s_{n,m}), \forall s_j \in p, \forall p_i \in$ palabras

Si ese valor es igual al que ocupa la primera posición de la lista de metros esperados, no debemos hacer nada. Por el contrario, nuestra expectativa pasará a ser esta si tenemos entre seis y nueve sílabas naturales si esa posición es para un octosílabo. Si hay menos de seis, esperaremos 7, 8, 6, 5, 4 y 3, en ese orden. Si no, estamos ante diez o más sílabas, por lo que supondremos un endecasílabo.

Algoritmo 49: Ajustes métricos (I).

Método `EncuentraRima(`*palabra*`):`
desplazamiento $\leftarrow\{n:1, n-1:0, n-2:-1\}$
tón $\leftarrow n$
para $sílaba_i \in palabra_{n..0}$ **hacer**
  si $\exists fonema^{+TÓN} \in sílaba_i$ **entonces**
    tón $\leftarrow i$
    para $fonema_j \in sílaba$ **hacer**
      si $\exists fonema^{+TÓN}$ **entonces**
        rima $\leftarrow$palabra$_{i..n}$
        rima$_0$ $\leftarrow$sílaba$_{j..n}$
        InterrumpeIteración
    InterrumpeIteración
si $tón < n-2$ **entonces**
  tón $\leftarrow$n - 1
ason, cons $\leftarrow$rima
si $|rima| > 2$ **entonces**
  ason $\leftarrow$rima$_{n-2}$ + rima$_n$
ason $\leftarrow$ason $\cap$ vocales$^{+\text{SIL}}$
devolver
 {acento : tón, cuenta : corrección$_{\text{tón}}$, *parcial* : ason, total : cons}

Podría darse el caso de que estuviéramos esperando once o siete sílabas. Aquí, supondremos 7 y 11 si hay nueve sílabas o menos, y 11, 7 si es mayor. La causa es que estos

dos metros pueden concurrir en una misma forma poética, como, por ejemplo, en la silva. Si tenemos nueve, que está entre el endecasílabo y el heptasílabo, anteponemos 8 a ambos metros.

Una función tan elemental como importante es la que usamos para reducir las dimensiones de la lista (Algoritmo 48). Recordemos que trabajamos con una lista de palabras que, a su vez, es una lista de sílabas. Se trata, por tanto, de dos dimensiones. Esto resulta útil para distinguir los límites de las palabras, pero también tanto trabajar dentro de ellas como evaluar el contacto entre dos palabras. Sin embargo, una vez que ha cumplido su misión, nada nos impide reducirla a una dimensión y operar linealmente con una lista de sílabas. La función devuelve el resultado de recorrer el verso por palabras, tomar las sílabas de cada una y colocarlas en sucesión.

Algoritmo 50: Ajustes métricos (II).

```
Método EncuentraRima(palabra):
    desplazamiento ←{n : 1, n − 1 : 0, n − 2 : −1}
    tón ←n
    para sílaba_i ∈ palabra_{n..0} hacer
        si ∃fonema^{+TÓN} ∈ sílaba_i entonces
            tón ←i
            para fonema_j ∈ sílaba hacer
                si ∃fonema^{+TÓN} entonces
                    rima ←palabra_{i..n}
                    rima_0 ←sílaba_{j..n}
                    InterrumpeIteración
            InterrumpeIteración
    si tón < n − 2 entonces
        tón ←n - 1
    ason, cons ←rima
    si |rima| > 2 entonces
        ason ←rima_{n−2} + rima_n
    ason ←ason ∩ vocales^{+SIL}
    devolver
     {acento : tón, cuenta : corrección_tón, parcial : ason, total : cons}
```

Con las sílabas así dispuestas, interesa conocer las características del axis rítmico (Balbín Lucas, 1968, pp. 38-60). Para ello, codificamos un nuevo método que recibirá la última palabra del verso. En ella definimos el valor que sumaremos al cómputo total de sílabas en función de si esta palabra es aguda, llana o esdrújula. Asumiremos que todas las palabras son llanas, lo que enmendaremos en respuesta a las sílabas que encontremos, recorriéndolas desde el final hasta el principio. Si encontramos un núcleo vocálico tónico, esa sílaba constituye el axis rítmico. Al haber empezado por el final, evitamos la eventualidad de encontrarnos con un acento secundario en caso

de que lo hubiera y adelantar de manera incorrecta el axis. Dado que en la rima solo se considera desde el núcleo vocálico, tomamos las sílabas del verso desde el axis y sustituimos su primera sílaba, correspondiente a este, por sus fonemas desde el núcleo silábico. Con esto salimos del bucle porque ya tendríamos la rima preliminar que buscábamos.

Tras prevenirnos de posibles sobreesdrújulas, asignamos provisionalmente esta rima tentativa a la consonancia y la asonancia. Si es esdrújula, corregimos eliminando la sílaba interior, de manera que solamente se tenga en cuenta la acentuada y la última (Domínguez Caparrós, 2014, pp. 60-65). Con esto, resta quedarnos con los sonidos vocálicos en la rima parcial y devolver la posición del acento, la corrección que implica, la rima parcial y la total.

Algoritmo 51: Discriminación de núcleos vocálicos.

Método `EncuentraNúcleos(`*sílabas*`)`**:**
  para $sílaba_i \in$ `PalabrasASílabas(`*sílabas*`)` **hacer**
    $\mathsf{sílabas}_i \leftarrow \text{sílaba}_i \cap \text{vocales}^{+\text{SIL}}$
  devolver sílabas

Dejaremos el ajuste métrico para el final porque el algoritmo contiene cierto número de subrutinas no triviales y elaboráremos antes otras. En su lugar, localizaremos primero los núcleos vocálicos, lo que no puede ser más sencillo, ya que disponemos de todo cuanto necesitamos para llevarlo a cabo. Se trata simplemente de seleccionar una serie de caracteres previamente identificados para que permanezcan aquellos que tengan sendas marcas de vocálico y silábico, descartándose todos los demás (Algoritmo 51).

La identificación del patrón rítmico tampoco requiere operaciones complejas, por la misma razón que en los núcleos vocálicos. En este caso, la marca que interesa es la de tonicidad. Sin embargo, al no poder descartar elementos, pues la posición es relevante, debemos identificarlos de otra manera. Así, en Algoritmo 51, usaremos el símbolo + para denotar una sílaba tónica y el símbolo - para una átona.

Con el resto de comprobaciones preliminares satisfecho, ajustaremos el metro atendiendo a las reglas de la métrica mediante una nueva función (Algoritmo 53). Para esto contaremos con las sílabas del verso y una lista ordenada de los metros que esperamos encontrar. Por consiguiente, definimos las uniones posibles y, para ello, evaluaremos desde la segunda sílaba, ya que la primera palabra del verso carece de otra precedente con la que hacer sinalefa. Después, buscamos potengciales sinalefas mediante el método que habíamos definido (Algoritmo 44) y diéresis (Algoritmo 54), además de la rima del verso (Algoritmo 50). El metro tentativo que estimamos de

Algoritmo 52: Identificación del patrón rítmico.

```
Método MarcaRitmo(sílabas):
    metro ←''
    para sílaba ∈ sílabas hacer
        si ∃fonema^{+TÓN} ∈ sílaba entonces
            metro ←metro + '+'
        si no
            metro ←metro + '-'
    devolver metro
```

esta manera será la suma del número de sílabas más la corrección de la rima. Esto es, las sílabas obtenidas a partir de la ortografía, a la que, recordemos, sumaremos una adicional si el verso es oxítono y restaremos lo propio si es proparoxítono. Así, calculamos un desplazamiento frente a lo esperado o, dicho de otra manera, la diferencia entre el primer elemento de la lista de valores probables y el metro preliminar que acabamos de estimar. Esto indicará si faltan o sobran sílabas en el verso para alcanzar el número indicado por el primer elemento de la lista de metros esperados. Si no hemos indicado explícitamente los resultados que consideramos que deberían ser encontrados, ese primer elemento coincidirá con la estimación, por lo que el desplazamiento será nulo.

De esta manera, si ha sucedido lo anterior o, por cualquier otra razón, el número de sílabas coincide con el primer valor de la lista, el metro no será ambiguo. En otro caso, verificamos si la primera estimación menos el número de sinalefas potenciales coincide con lo esperado, por lo que el verso tampoco será ambiguo y se resolverá aplicando las sinalefas de la lista.

Si este tampoco es el caso, lo siguiente sería ver que la sustracción de las sinalefas potenciales a las sílabas ortográficas sigue siendo mayor que el metro esperado. Por lo tanto, buscamos una palabra esdrújula a final de hemistiquio y, además, extendemos el supuesto de Navarro Tomás (1991, p. 40) a los endecasílabos *a maiore*. De ser así, marcamos el verso como ambiguo, resolvemos el fenómeno de la cesura y aplicamos las sinalefas, sin olvidar que ahora tenemos una sílaba menos.

En caso de que ninguna de las posibilidades descrita se dé, marcamos ambigüedad, y comprobamos que el desplazamiento sea negativo, esto es, que la medida preliminar de sílabas sea mayor de lo esperado; haremos tantas sinalefas como indique la diferencia. De lo contrario, si el número de sílabas preliminar es menor de lo previsto, y, además, el verso medido tentativamente no baja de cinco sílabas —lo que apuntaría a un pie quebrado o un eco—, aplicamos tantos hiatos como indique el desplazamiento.

Con el verso corregido, volvemos a evaluar la rima y el metro. Si el resultado es mayor

Algoritmo 53: Ajuste *post hoc* del metro.

```
Método AjustaMetro(sílabas, esperados):
    slbs ←sílabas_{1..n}
    sinalefasp ←EncuentraSinalefas(slbs)
    diéresisp ←EncuentraDiéresis(slbs)
    rima ←EncuentraRima(slbs_n)
    metro ←|PalabrasASílabas(slbs)| + rima_cuenta
    desplazamiento ←esperados_0 − metro
    si desplazamiento = 0 entonces
        ambigüedad ←0
    si no, si metro − |sinalefasp| = esperados_0 entonces
        ambigüedad ←0
        slbs ←Sinalefas(slbs, -desplazamiento)
    si no, si metro − |sinalefasp| = esperados_0 entonces
        ambigüedad ←0
        slbs ←Sinalefas(slbs, -desplazamiento)
    si no, si metro − |sinalefasp| > esperados_0 entonces
        ambigüedad ←2
        slbs ←Sinalefas(slbs, -desplazamiento)
        hemistiquio ←CompruebaHemistiquios(silabas)
        si CompruebaHemistiquios(slbs) > 0 entonces
            slbs ←ResuelveLargo(slbs, hemistiquio)
            metro ←metro − 1
    si no
        ambigüedad ←1
        si desplazamiento < 0 ∧ |sinalefasp| ≥ | desplazamiento| entonces
            slbs ←Sinalefas(slbs, desplazamiento)
        si no, si esperados_0 < metro < 4 ∧ |diéresis| + metroesperados_0
         entonces
            slbs ←AplicaDiéresis(slbs, diéresis, desplazamiento))
    rima ←EncuentraRima(slbs_n)
    metro ←|PalabrasASílabas(slbs)| + rima_cuenta
    si metro > esperados_0 entonces
        devolver AjustaMetro(slbs, esperados_{1..n})
    si no, si metro < esperados_0 entonces
        devolver AjustaMetro(slbs, esperados_{1..n})
    si no
        devolver Verso(slbs, ambigüedad, metro, rima_parc., rima_tot.)
```

de lo esperado, devolvemos el producto de llamar recursivamente a este mismo método, pero pasándole la lista de versos esperados sin su primer valor para el siguiente nivel de recursión. Esto es, el metro a la cabeza de la nueva lista será el segundo de la que hemos estado usando. Si, por el contrario, el metro medido sigue siendo menor que ese valor, lo invocaremos también con la lista de metros actualizada y con las sílabas ajustadas. De no cumplirse tampoco, tenemos el verso resuelto y devolvemos

su caracterización de acuerdo con esto.

Uno de los valores en que nos apoyábamos era la posición de las diéresis potenciales. Esta la hallamos aplicando Algoritmo 54. El procedimiento es intuitivo: recorremos el verso por sus palabras y, estas últimas, por sus sílabas, en las que buscamos segmentos de vocal y vocoide. Haremos la prevención, eso sí, de que la diéresis ha de producirse ante sílaba tónica o, si es en sílaba tónica, el diptongo a romper ha de ser creciente. De encontrar uno, guardamos la posición en una lista, que devolveremos al terminar de recorrer así el verso.

Para aplicar las diéresis, definimos un método que recibe la lista de palabras, las posiciones de las diéresis potenciales y la diferencia entre las sílabas requeridas y las que se han encontrado (Algoritmo 55). Aplicamos una función para evaluar la preferencia de las diéresis (Algoritmo 56), de modo que obtengamos una lista ordenada de posiciones de las diéresis en el verso y la palabra según el grado de preferencia que les hemos concedido. Podemos recorrer esa lista tantas posiciones como sea necesario para satisfacer la diferencia de sílabas. Para simplificar los términos, guardamos la palabra en el verso en una variable, en otra el desplazamiento interno, así como la unión vocálica sujeta a una potencial ruptura. Definimos igualmente las partes del segmento de vocales en contacto. De esta forma, tenemos un núcleo silábico y, en al menos un lado de este, un fonema no silábico, semiconsonántico en un diptongo creciente o semivocálico en decreciente.

Algoritmo 54: Ubicación de diéresis potenciales.

Método `EncuentraDiéresis(`*palabras*`):`
  diptongos $\leftarrow()$
  para $palabra_i \in palabras$ **hacer**
    para $sílaba_j \in palabra$ **hacer**
      si $\exists x^{+\text{VOC}} y^{+\text{VOC}} \in sílaba$ **entonces**
        si $i + 1 < palabra_{i-acento} \vee sílaba_{vocoide} sílaba_{vocal^{+\text{TÓN}}}$ **entonces**
          diptongos $\leftarrow$diptongos $+ ((i, j))$
  devolver diptongos

La forma analítica de asignar un rango a las diéresis potenciales la desarrollamos en Algoritmo 56. Necesitamos la lista de palabras y las posiciones de las diéresis, que se indican mediante el número de palabra en el verso y su desplazamiento silábico dentro de la palabra correspondiente. Si la palabra es una de las que habitualmente suelen romper el diptongo, que habíamos definido (Algoritmo 33), colocamos la palabra en la primera posición de la lista y el resto tras ella. Al final hacemos un pequeño ajuste, de manera que, si hay dos diéresis posibles en una misma palabra, tenemos en

consideración el reajuste silábico de esta después de hacer la primera diéresis. Esto es, si en una palabra de tres sílabas tenemos diéresis potenciales en la primera y la segunda sílaba, hemos de cambiar la posición de la segunda diéresis hipotética tras haber hecho la primera.

Algoritmo 55: Aplicación de diéresis.

Método `AplicaDiéresis`(*palabras, diéresis, diferencia*)**:**
  preferencia ←`PreferenciaDiéresis`(*palabras, diéresis*)
  para $idx \in preferencia_{0..diferencia}$ **hacer**
    palabra, sílaba ←$palabras_{idx_{pal}}$, $palabra_{idx_{sil}}$
    dipt ←$aNc, \forall a^{+\text{ALTA}} N^{+\text{VOC}} c^{+\text{ALTA} \wedge \emptyset} \in$ sílaba
    C ←$x, x \in$ sílaba $\wedge\ x \prec aNc$
    V ←$y, y \in$ sílaba $\wedge\ x \succ aNc$
    si $\exists N \wedge \exists c$ **entonces**
      si $\exists N^{+TÓN}$ **entonces**
        N ←$N^{-\text{TÓN}}$
        a ←$a^{+\text{TÓN}}$
      parte1 ←$= C + N$
      parte2 ←$(c_0,..., c_{n-1}, c_n^{+\text{SIL}}) + V$
    si no
      parte1 ←$C + (a_0,..., a_{n-1}, a_n^{+\text{SIL}})$
      parte2 ←$N + V$
    diéresis ←"$parteuno\ parte2$"
    para $elemento_i \in palabra$ **hacer**
      si $i = idx_{sil}$ **entonces**
        elemento$_i$ ←$(idx_{sil}$, diéresis)
      palabra ←palabra + (elemento$_i$)
    palabras$_{idx_{pal}}$ **para** $elemento \in palabra$ **hacer**
    palabras$_{idx_{pal}}$ ←palabra$_0$ + palabra$_1$ + ... + palabra$_n$
  devolver palabras

En Algoritmo 53 nos ocupábamos de versos de arte mayor con esdrújula antes de la cesura, para contar solo dos sílabas desde el acento. La identificación de este fenómeno se hace mediante la función elaborada en Algoritmo 57. Nos aseguramos de que el verso es de arte mayor y, de ser así, lo recorremos. Con nueve o menos sílabas ortográficas se requerirían al menos dos diéresis para obtener un endecasílabo —que es el verso de arte mayor que se encuentra normalmente en una pieza teatral—, lo que resultaría muy forzado incluso para el poeta de criterio más laxo.

Ha de comprobarse la estructura silábica de cada palabra. La condición es que la penúltima sílaba antes de la cesura sea tónica y se encuentre después de la quinta sílaba en los endecasílabos *a minore* y de la séptima en los *a maiore*. Si bien Navarro

Algoritmo 56: Preferencia de las diéresis.

```
Método PreferenciaDiéresis(palabras, diéresis):
    preferencia ←()
    para idx ∈ diéresis_{n..0} hacer
        si palabras ∈ sospechosos_habituales entonces
            preferencia ←(idx) + preferencia
        si no
            preferencia ←preferencia + (idx)
    para segundo_i ∈ preferencia_{n..0} hacer
        para primero ∈ preferencia_{n−idx} hacer
            si primero_0 = segundo_0 ∧ primero_1 < segundo_1 entonces
                preferencia_{n−idx} ←(segundo_0, segundo_1 + 1)
    devolver preferencia
```

Tomás (1991, p. 40) discute únicamente los primeros, lo hacemos extensivo a los segundos, de modo que, si se da el caso y no hay otra posibilidad de ajuste, aplicaremos esta medida para reducir un dodecasílabo inconsistente.

Considerando lo dicho, si encontramos acento en la cuarta sílaba y, a su vez, esta es la antepenúltima de una palabra esdrújula, dejaremos de contar la penúltima sílaba de la palabra en el cómputo del verso. Si encontramos acento en la cuarta y no se da el caso, salimos igualmente, pues sería un verso *a minore* y la sexta resultaría irrelevante en lo que concierne al hemistiquio.

Algoritmo 57: Comprobación de esdrújula ante hemistiquio.

```
Método CompruebaHemistiquios(palabras):
    desplazamiento ←0; idx ←0
    corrección ←0
    para palabra_i ∈ palabras hacer
        si |sílabas(palabras)| > 9 entonces
            para sílaba_j ∈ palabra hacer
                si ∃x^{+VOC,+TÓN} ∈ sílaba ∧ sílaba_j + desplazamiento ∈ (3, 5)
                 entonces
                    si |palabra| − idy > 2 ∧ palabra_{n−1..n} ≠ -mente entonces
                        corrección ←idx
                    InterrumpeIteración
                i ←i + 1
            desplazamiento ←desplazamiento
        + |palabra|
    si no
        InterrumpeIteración
    devolver corrección
```

La corrección de palabra esdrújula ante la cesura se muestra en Algoritmo 58. Lo haremos mediante la eliminación de la penúltima sílaba de la palabra para que queden

únicamente sílabas con relevancia métrica. No cambiamos el texto original —este permanece invariable como referencia—, sino que trabajamos con su representación fonética para obtener información métrica.

Algoritmo 58: Reajuste en función del hemistiquio.

Método `ResuelveLargo(`*palabras, posición*`):`
  palabra ←palabras$_{\text{posición}}$
  palabra ←(palabra$_0$, ..., palabra$_{n-1}$) + (palabra$_n$)
  palabras$_{\text{posición}}$ ←palabra
  devolver palabras

La resolución de sinalefas se lleva a cabo mediante una función recursiva (Algoritmo 59). Recibe las sílabas, el desplazamiento y un contador. Si la diferencia entre sílabas esperadas y encontradas es positiva, calcula las sinalefas potenciales del verso, incrementa el contador, ajusta las sílabas con la primera sinalefa más probable —recordemos que la función de encontrar sinalefas las ordenaba—, y devolvemos una llamada recursiva a la propia función, con las sílabas ajustadas como argumento, la diferencia decrementada en uno por la sinalefa que hemos hecho y la cuenta actualizada. Si el desplazamiento es menor que uno, devolvemos las sílabas tal cual y daríamos la resolución por concluida.

Algoritmo 59: Sinalefas.

Método `Sinalefas(`*sílabas, diferencia, cuenta ←0*`):`
  si *desplazamiento* > 0 **entonces**
    potenciales ←`EncuentraSinalefas(`*sílabas*`)`
    cuenta ←cuenta + 1
    sílabas ←`AjustaSílabas(`*sílabas*, (*potenciales*$_0$)`)`
    devolver `Sinalefas(`*sílabas, diferencia-1, cuenta*`)`
  si no
    devolver *sílabas*

La parte de bajo nivel del ajuste (Algoritmo 60) requiere las sílabas y las sinalefas. Recordemos que identificábamos las sinalefas con una tupla[131] de la posición, preferencia, y las dos sílabas en contacto. De esta forma, definimos un conjunto con un elemento: una tupla correspondiente a las coordenadas de la primera sinalefa de la lista. Así, ante la eventualidad de una lista vacía, no intentaremos extraer un primer elemento inexistente, sino que creamos un conjunto vacío de coordenadas.

Hay que verificar que el conjunto no esté vacío. Si contiene un elemento, empezamos a ajustar. Para simplificar, definimos variables para las posiciones de la primera palabra

[131] Una *tupla* es una lista ordenada y finita de elementos.

y sílaba, así como para la primera palabra misma y la sílaba resultante de la unión de dicha palabra y la siguiente. En el algoritmo ilustrativo denotamos la palabra y sílaba de la izquierda con A y con B las de la derecha. Si el desplazamiento silábico en la palabra alcanza la última sílaba de la primera palabra, estamos ante una sinalefa. Por lo tanto, la parte de la derecha corresponderá a la primera sílaba de la siguiente palabra. Si la palabra de la izquierda no es polisílaba, señalamos las sílabas que preceden a la que ha de unirse; de lo contrario, no habrá tales sílabas. De la misma manera, señalamos también las sílabas que suceden a la que uniremos en la palabra de la derecha. Con esto obtenemos la resolución del diptongo, que desarrollaremos en Algoritmo 61.

Queda integrar esta nueva sílaba en el verso. Primero unimos todas las sílabas en una única palabra, con las posibles sílabas de la izquierda, seguidas del diptongo, así como las de la derecha si las hubiera. El nuevo verso lo conforman las palabras precedentes, la palabra resultante de la unión y las siguientes. Finalmente, reajustamos la posición para que todo concuerde.

Podría darse que la primera sílaba no fuera la última de una palabra. En ese caso, no estaríamos ante una sinalefa, sino una sinéresis, por lo que habría que ajustar la palabra internamente, pero no su posición dentro del verso. De cualquier modo, siempre que se haya cumplido la primera condición del algoritmo, lo volvemos a aplicar para obtener la lista de palabras, que es lo que finalmente devuelve la subrutina.

Hemos de encontrar asimismo la forma en que se realizan materialmente las uniones vocálicas ubicadas. Esto es, hay que determinar los fenómenos coarticulatorios que se dan entre las partes de la unión y modificar los sonidos en consecuencia (Algoritmo 61). Para ello tomamos los dos componentes del diptongo y comenzamos una serie de comprobaciones. Lo primero que verificamos es si las vocales en contacto son iguales, en cuyo caso elidiremos una. Si estas son diferentes, comprobamos si estamos ante un diptongo formado por la conjunción *y* seguido de palabra que comienza por *u*. En ese caso, no resolveremos como un diptongo, sino que la primera vocal adquiere calidad plenamente consonántica, como fricativa palatal sonora, que se coloca en ataque de la vocal siguiente. Si tampoco se cumple, constatamos que se trata de una unión de más de dos sonidos vocálicos, en cuyo caso, reordenaremos el grupo de manera que la vocal más perceptible adquiera rasgo silábico y las demás lo pierdan, desplazando el acento hasta el nuevo núcleo si fuera necesario.

Quedan los casos regulares. Recordemos que, entre dos vocales átonas, la segunda tenía precedencia. Si solo una era tónica, la átona perdía la calidad silábica y, si ambas llevaban acento, la perceptibilidad dirimía la cuestión. Así, comprobamos si la

Algoritmo 60: Ajuste silábico.

Método `AjustaSílabas(`*palabras, sinalefas*`)`**:**
- posiciones ←{sílaba$_{i,coord.}$}, ∀sílaba$_i$ ∈ sinalefas
- **si** *posiciones* ≠ ∅ **entonces**
  - palA ←$posiciones_{0,p}$
  - silA ←$posiciones_{0,s}$
  - palabra ←$palabras_{posiciones_{0,p}}$
  - unión ←$[sinalefas_{0,C}, sinalefas_{0,V}]$
  - **si** silA = $|palabra| - 1$ **entonces**
    - silB ←0
    - palB ←palA + 1
    - **si** $|words_{palA}| > 1$ **entonces**
      - C ←(palabras$_{\text{palA}_0}$, palabras$_{\text{palA}_1}$,..., palabras$_{\text{palA}_{n-1}}$)
    - **si no**
      - C ←∅
    - **si** $|words_{palB}| > 1$ **entonces**
      - V ←(palabras$_{\text{palB}_1}$, ..., palabras$_{\text{palB}_n}$)
    - **si no**
      - V ←∅
    - diptongo ←`AplicaSinalefas(`*unión*`)`
    - palabra ←C + diptongo
    - **si** $|palabras_{palB}| > 1$ **entonces**
      - palabra ←palabra + V
    - palabras ←(palabras$_{0..palA}$ + (palabra) + palabras$_{palB+1..n}$)
    - posiciones ←`AjustaPosicion(`$posiciones_{1..n}, posiciones_0, |C|$`)`
  - **si no**
    - silB ←silA + 1
    - C ←(palabras$_{\text{palA}_0}$, palabras$_{\text{palA}_1}$,..., palabras$_{\text{palA}_{silA}}$)
    - V ←(palabras$_{\text{palA}_{silB+1}}$, ..., palabras$_{\text{palA}_n}$)
    - diptongo ←`AplicaSinalefas(`*unión*`)`
    - palabra ←$C + (diptongo) + V$
    - palabras$_{palB}$ ←palabra
    - posiciones ←`AjustaPosicion(`$posiciones_{1..n}, posiciones_0, -1$`)`
  - palabras ←`AjustaSílabas(`*palabras, posiciones*`)`
- **devolver** *palabras*

primera es tónica y la segunda es átona o menos perceptible que la primera, en cuyo caso perderá la calidad silábica. Para cualquier otra situación, esto es, ambas tónicas, siendo la segunda más perceptible o las dos átonas, la primera parte se desilabiza y el núcleo queda en la segunda.

Necesitamos un último algoritmo para procesar el verso. Este debe corregir la lista de posiciones de las sinalefas para que tenga en cuenta los cambios realizados en la

Algoritmo 61: Aplicación de las sinalefas.

Método `AplicaSinalefas(`*diptongo*`):`
  C, V ←diptongo$_C$, diptongo$_V$
  si $C_n = V_0$ **entonces**
    diptongo ←$C_{0..n-1} + V$
  si no, si $\sum x, x \in vocalic > 2$ **entonces**
    diptongo ←`Perceptibility(`$C{+}V$`)`
  si no, si $C = y \vee (C = i) \wedge V_0 = u$ **entonces**
    diptongo ←$(\text{j}) + V$
  si no, si
   $\exists x^{+TÓN} \in C \wedge (\nexists x^{+TÓN} \in V \vee perceptibilidad_{C_n} > perceptibilidad_{V_0})$
   entonces
    diptongo ←$C^{+\text{SIL}} + V^{-\text{SIL}}$
  si no
    diptongo ←$C^{-\text{SIL}} + V^{+\text{SIL}}$
  devolver diptongo

distribución silábica (Algoritmo 62). Recibe tres parámetros: la lista de sinalefas y sinéresis del verso, la posición a tener en cuenta y el número de sílabas de la palabra precedente hasta la del diptongo, que llamaremos desplazamiento. Vamos recorriendo la lista de uniones y cotejamos el valor de cada elemento. Si es negativo —algo que indicaría una sinéresis, porque, de otra manera, habría recibido un entero positivo (ver Algoritmo 60)—, comparamos en cada iteración si la palabra correspondiente se encuentra en la posición indicada.

Algoritmo 62: Ajuste de la posición.

Método `AjustaPosicion(`*sinalefas, posición, desplazamiento*`):`
  para $sinalefa_i \in sinalefas$ **hacer**
    si $desplazamiento < 0$ **entonces**
      si $sinalefa_p = posición_p$ **entonces**
        si $sinalefa_s \geq posición_s$ **entonces**
          sinalefas$_{i,s}$ ←sinalefas$_{i,s} - 1$
    si no, si $sinalefa_p > posición_p$ **entonces**
      si $sinalefa_p = posición_p + 1$ **entonces**
        sinalefas$_{i,s}$ ←sinalefas$_{i,s}$ + desplazamiento
      sinalefas$_{i,p}$ ←sinalefas$_{i,p} - 1$
  devolver *sinalefas*

Si es el caso, comparamos las sílabas, decrementando en la lista de sinalefas si la sílaba es posterior a la de la posición dada porque, al hacerla, la palabra se reduciría en una sílaba y, con ella, las sílabas. Si la posición dada fuera posterior, no sería necesario hacer nada porque la sílaba de la iteración no cambiaría, al producirse tras ella la sinéresis.

Si la diferencia no es negativa, estamos ante una sinalefa, por lo que, además de corregir la palabra, necesitaremos hacer lo propio con esta respecto al verso. De esta manera, si la palabra de esta iteración es la que sigue a la de la posición dada, ajustamos el número de sílabas de aquella sumándoles el desplazamiento, pues pasan a formar parte de esta. En cualquier caso, decrementamos la posición de todas las palabras posteriores a la de la posición dada para tener en cuenta la unión de las palabras mediante la sinalefa.

Habíamos dispuesto la información dramática al modo de Cuadro 7.1. Ahora, esta incluiría columnas adicionales con datos sobre los versos (Cuadro 9.1), así como otras columnas, tantas como atributos obtenidos se deseen incluir para caracterizar los versos.

V	Texto	Síl.	Núcl.	Ason.	Cons.	Ritmo
-1	Sale en lo alto de un monte ROSAURA en hábito de hombre,de camino, y en representando los primeros versos va bajando.					
1	Hipogrifo violento,	7	ioIooEo	eo	ento	--+---+-
2	que corriste parejas con el viento,	11	eoIeaEaoeEo	eo	ento	--+--+---+-
3	¿dónde rayo sin llama,	7	OeAoiAa	aa	ama	+-+--+-
4	pájaro sin matiz, pez sin escama	11	AaoiaIEieAa	aa	ama	+----++--+-
5	y bruto sin instinto	7	iUoiiIo	io	into	-+---+-
6	natural, al confuso laberinto	11	auAaoUoaeIo	io	into	--+--+---+-
7	de esas desnudas peñas	7	EaeUaEa	ea	eɲas	+--+-+-
8	te desbocas, te arrastras y despeñas?	11	eeOaaAaieEa	ea	eɲas	--+--+---+-
9	Quédate en este monte,	7	EaeEeOe	oe	onte	+--+-+-
10	donde tengan los brutos su Faetonte;	11	oeEaoUoueOe	oe	onte	--+--+---+-
11	que yo, sin más camino	7	eOiAaIo	io	ino	-+-+-+-
12	que el que me dan las leyes del destino,	11	eeeAaEeeeIo	io	ino	---+-+---+-
13	ciega y desesperada,	7	EieeeAa	aa	ada	+----+-
14	bajaré la cabeza enmarañada	11	aaEaaEeaaAa	aa	ada	--+--+---+-
15	deste monte eminente	7	EeOeiEe	ee	ente	+-+--+-
16	que arruga el sol el ceño de la frente.	11	aUeOeEoeaEe	ee	ente	-+-+-+---+-
17	Mal, Polonia, recibes	7	AoOaeIe	ie	ibes	+-+--+-
18	a un extranjero, pues con sangre escribes	11	UeaEoeoAeIe	ie	ibes	+--+---+-+-
19	su entrada en tus arenas;	7	eAeuaEa	ea	enas	-+---+-
20	y apenas llega, cuando llega a penas.	11	aEaEaaoEaEa	ea	enas	-+-+---+-+-
21	Bien mi suerte lo dice;	7	EiEeoIe	ie	iθe	+-+--+-
22	mas ¿dónde halló piedad un infelice?	11	aOaOeAUieIe	ie	iθe	-+-+-++--+-
-22	(Sale CLARÍN, gracioso.)					
23	Di dos, y no me dejes	7	IOiOeEe	ee	exes	++-+-+-
24	en la posada a mí cuando te quejes;	11	eaoAaIaoeEe	ee	exes	---+-+---+-
25	que si dos hemos sido	7	eiOEoIo	io	ido	--++-+-
26	los que de nuestra patria hemos salido	11	oeeeaAAoaIo	io	ido	-----++--+-
27	a probar aventuras,	7	aoAaeUa	ua	uɾas	--+--+-
28	dos los que entre desdichas y locuras	11	OoeeeIaioUa	ua	uɾas	+----+---+-
29	aquí habemos llegado,	7	aIEoeAo	ao	ado	-++--+-
30	y dos los que del monte hemos rodado,	11	iOoeeOEooAo	ao	ado	-+---++--+-
31	¿no es razón que yo sienta	7	OaOeOEa	ea	enta	+-+-++-
32	meterme en el pesar y no en la cuenta?	11	eEeeeAiOaEa	ea	enta	-+---+-+-+-
...						

Cuadro 9.1: Tabla de atributos métricos.

9.6 Taxonomía versal

Aunque este trabajo aspira a extraer y presentar los datos y no clasificarlos, nos sobrepasaremos mínimamente para ofrecer una solución preventiva a modo de sugerencia. Con toda la información del verso, sería trivial la categorización del octosílabo y el endecasílabo según lo visto en la parte teórica. Necesitaríamos componer sendos diccionarios, cuyas claves fueran las distribuciones rítmicas posibles del tipo de verso, a cada una de las cuales asignaríamos un valor acorde a la categoría correspondiente. Para los endecasílabos usaríamos la taxonomía clásica y, para los octosílabos, una de las propuestas por Navarro Tomás (1991, pp. 274-275) o Varela Merino y col. (2005, pp. 149-158), por ejemplo. Consideraríamos solo hasta la última sílaba acentuada, puesto que las siguientes no afectan a ninguno de los modelos. Aunque cabría la posibilidad de hacerlo mediante reglas y decisiones, resultaría más complejo conceptualmente que la traducción directa desarrollada, pero emplazamos al lector a reflexionar sobre ello.

9.7 Verificación de la escansión rítmica

La precisión del método se probó comparándola con el corpus de sonetos anotado ADSO 100 (Navarro-Colorado, 2017), que instalamos usando la herramienta Averell (Díaz Medina y col., 2021). Este ha sido considerado el *gold standard* por otros estudios previos (Marco y col., 2021). Por el momento, es el único de su tamaño anotado de manera manual cuyo índice de acuerdo entre anotadores se encuentra publicado; concretamente, es del 96 % (Marco y col., 2021). Contiene 1.404 versos, puesto que «Pintando la vida de un señor mal ocupado» de Quevedo lleva estrambote. Empleamos silabación peninsular y epéntesis, que son las opciones por defecto.

Método	**Precisión (%)**
Gervás	70,88
Agirrezabal	90,84
Navarro-Colorado	94,45
Jumper	94,97
Rantanplan	96,23
libEscansión (nuestro)	*97,01*

Marco y col. (2021). Mejores resultados en cursiva.

Cuadro 9.2: Resultados en el corpus de Navarro-Colorado de 100 sonetos.

Resumido, nuestro algoritmo alcanza una precisión del 97,01 %, lo que supera lo obtenido hasta el momento (Cuadro 9.2). Sin embargo, conviene matizar esto porque

podría incluso mejorar si consideramos ciertas circunstancias. Por ejemplo, el corpus de pruebas contempla las interjecciones como átonas —algunas como *ay* y *oh* abundan en la colección de poemas—, mientras que nuestra aproximación las considera tónicas. Invirtiendo el comportamiento en Algoritmo 39, la precisión se eleva hasta el 98,65 %.

Aparte de las diferencias debidas a interjecciones, hay diecinueve versos discrepantes. Once de ellos son desavenencias causadas por erratas tipográficas (Cuadro 9.3, 1-5) o descuidos del anotador (Cuadro 9.3, 6-11). En el primer caso, la anotación manual refleja la forma correcta, si bien el verso escrito no lo hace, como en *Príamo* y los más sutiles diacríticos de *aún* y *dó*. Queda claro que el humano vence a la máquina en tanto que aquel reconoce la pronunciación correcta incluso si contradice la ortografía. El segundo caso va desde acentos ignorados en palabras léxicas, como la doble acentuación de los adverbios en *-mente* o dejando adjetivos como átonos, hasta lo contrario, marcando como tónicas palabras como el tratamiento que precede a un nombre.

	Verso
1	falte a Pr**i**amo tierra y falte fuego.
2	Si rico, ¿d**o** tus bienes vinculados?
3	**au**n no pudo, de lástima, dar muerte,
4	**au**n le quedaba brío y lozanía.
5	del muerto Lilio es; que **au**n no perdona
6	trocó mi mal en b*ie*n, mi pena en gloria.
7	este que, siendo s*o*lam**e**nte cero,
8	mil canas, está hecho un S**a**n Hilario.
9	y**a** que no se conserva, se previene
10	¡Oh s**a**nto bodegón! ¡Oh picardía!
11	Llámasme, *gran* Señor; nunca respondo.
12	con**fia***da en* la vista vencedora,
13	M**ae**str*o e*ra de esgrima Campuzano,
14	Yo v**i** **u**nos ojos bellos, qu*e hi*rieron
15	qu**ie**n promete buen fin a mi v*ia*je,
16	con l**a** **o**rej*a a*snina se conjura.
17	Este s**ua**v*e i*ncendio me sustenta,
18	breve pórfido sell*a e*n paz s**ua**ve;
19	P**a**ra cochero. El coche está en palacio.

En corpus en negrita, en algoritmo en cursiva.

Cuadro 9.3: Discrepancias.

Merece una mención especial la discrepancia 19 (Cuadro 9.3). Se trata de un error de anotación morfosintáctica en *para* producido por la ausencia de la coma vocativa. Este tipo de ambigüedades son difíciles de sortear, en tanto que, técnicamente, el programa

de análisis morfosintáctico no ha incurrido en fallo alguno; la forma adecuada ha de deducirse del contexto. Los siete versos restantes aceptarían distintas interpretaciones, por lo que son discrepancias puras, siendo facultativo decidirse por una entonación u otra.

Si nos quedamos solo con las discrepancias entre las distintas sensibilidades de los anotadores, dejando a un lado las erratas en el corpus o los deslices, nuestro algoritmo alcanzaría una precisión del 99,50 % respecto al corpus de pruebas.

9.8 Agrupación de versos

Las estrofas castellanas suelen estar formadas entre dos y diez versos. Están caracterizadas por el número de estos, así como la frecuencia y distribución de su cantidad y rima, como habíamos visto. En el caso de las agrupaciones no estróficas, al tener un cómputo de versos potencialmente ilimitado, su número es irrelevante, pero sí tiene importancia la cantidad de cada verso y el patrón con el que estos se distribuyen de acuerdo con sus características pertinentes.

La existencia de estrofas sugiere dos caminos para abordar el problema en la parte práctica. Por un lado, podríamos identificar las formas simples y, una vez hecho esto, agruparlas en otras complejas. Volviendo al ejemplo 85, encontraríamos los pareados por la rima y, después, los agruparíamos en una silva por el metro de los versos. Por otro parte, podríamos tratar de encontrar primero las agrupaciones complejas e ir probando con otras cada vez más sencillas en los versos restantes. Trataríamos de agrupar inicialmente los versos en poemas no estróficos, tras lo que haríamos lo propio en poemas estróficos para los versos restantes y terminaríamos por ordenar los versos que queden sin clasificar en estrofas, empezando con las de mayor número de versos y terminando con en el pareado.

Esta aproximación desde el verso a la estrofa bien podría beneficiarse de un análisis complementario en sentido contrario, de la estrofa al verso. Esto es, dado que las agrupaciones que encontraremos suelen estar formados de versos de tipos muy concretos y estos se combinan entre ellos observando usos establecidos relativamente predecibles, las posibilidades de escandir un verso se reducen significativamente una vez identificados algunos de los otros versos que componen la estrofa. Los versos adyacentes proporcionan indicios de peso para evaluar uno dado en la estrofa, pues es de esperar que las combinaciones posibles se ciñan a las descritas en las poéticas.

Disponemos de casi todos los elementos necesarios para identificar estrofas: longitud del verso y rima, tanto consonante como asonante, y podríamos deducir a partir de

ahí un patrón. Esto supondría ya medio camino para alcanzar el objetivo (Pérez Pozo y col., 2021). No obstante, los textos líricos asumen un elemento implícito al género del que carece el texto dramático. Mientras que los límites del poema están definidos, este tiene un principio y un final determinado, el texto teatral encadena formas métricas sin solución de continuidad. La cuestión se complica si tenemos en cuenta que, además, no todos los tipos son estróficos; no resulta infrecuente encontrar segmentos de extensión indefinida, como los romances. Por lo tanto, antes de tratar las estrofas debemos establecer primero sus fronteras. Esta necesidad motiva también la aproximación mediante reglas, ya que se presta a interconectar la segmentación y la clasificación.

Para la agrupación inicial partiremos de la premisa de que en una estrofa encontraremos diferentes rimas simples, *abab*, por ejemplo, o verso libre y rima en verso par $-a-a$. En una primera aproximación buscaremos estas estrofas y, una vez identificadas, trataremos de componer con ellas estrofas complejas donde sea posible.

En dos versos solo se da una rima *aa*; en tres, las combinaciones posibles ascienden a tres *aba*, *abb* y *aab*. Sin embargo, las dos últimas se dan exclusivamente en el contexto de una estrofa mayor. Con cuatro versos, podríamos encontrar rima abrazada o encadenada. Sería posible asimismo la rima circunscrita a los pares, probablemente asonante, y este patrón podría extenderse a los versos siguientes; tampoco resultaría extraña una rima gemela dentro de la estrofa —al final, normalmente—, en cuyo caso la habríamos detectado como dos pareados consecutivos. ´Las combinaciones de dos rimas se extienden a las estrofas de cinco versos, mientras que, en las de seis, encontramos variaciones de tres rimas, con la posible aparición de pareados, como, por ejemplo en un sexteto lira del tipo $aBaBcC$ o en la octava real siciliana $ABABABCC$.

Algoritmo 63: Agrupación de versos.

```
versos ←{}
para acto ∈ actos hacer
    segmento ←obra_acto
    versos_acto ←Estrofas(segmento)
    versos ←versos + versos_acto
obra_estrofa,estrofa_n ←versos_estrofas,cuenta
para acto ∈ actos hacer
    segmento ←obra_acto
    versos_acto ←Romances(segmento)
    versos ←versos + versos_acto
obra_estrofa,estrofa_n ←versos_estrofas,cuenta
```

Como indicamos, para abordar otras formas más extensas, como el ovillejo o la silva,

las analizaríamos en primera instancia como sus estrofas simples constituyentes y, en la segunda pasada, las agruparíamos en la estrofa compleja correspondiente. Así, partiremos de los treinta y seis versos de la sextina, los catorce del soneto, décimas, etc. En esta primera pasada no haremos distinciones precisas entre los tamaños de los versos, sino que solo consideraremos si son de arte mayor o menor. Por lo tanto, la silva de 102 versos con la que comienza *La vida es sueño* se reconocerá en esta aproximación inicial simplemente como una tirada de pareados.

Una vez identificadas las estrofas —o sus semiestrofas constitutivas—, buscaremos las agrupaciones asonantes en las series de versos que han quedado sin asignar a una agrupación. Aquí identificamos, por ejemplo, los romances, pero no podríamos distinguir aún entre un romancillo y un romance, ya que ambos son de arte menor. Tanto en esta como en la operación precedente, nos apoyaremos en las segmentaciones externas ya establecidas. Esto es, tratamos obras y, en su caso, actos, como unidades independientes, lo que evita, por ejemplo, tratar la tirada inicial de un segmento como la continuación de la final del precedente si las propiedades de los versos limítrofes son coincidentes hasta hacerlos susceptibles de confusión.

Algoritmo 64: Búsqueda de estrofas.

```
Método Estrofas(tabla):
    origen ←0
    estrofas ←{}
    nulo ←('', 0)
    est ←()
    mientras origen < tabla_n hacer
        i, estrofa ←BuscaEstrofa(tabla_origen..n)
        si i > 0 entonces
            si estrofa ∈ estrofas entonces
                estrofas_estrofa ←estrofas_estrofa + 1
            si no
                estrofas_estrofa ←1
            para idx in range(0,i) hacer
                est.append((estrofa, estrofas_estrofa))
            origen ←origen + i
        si no
            est.append(nulo)
            origen ←origen + 1
    devolver origen
```

A efectos de demostración, nos centraremos en algunas de las formas más relevantes. Consideraremos como tales las que Morley y Bruerton (1968, pp. 627-659) mencionan

en su apéndice de datos esenciales. Si bien, a partir de aquí, identificar otras agrupaciones es una extrapolación de lo que presentamos abajo a las particularidades de la estrofa.

Algoritmo 65: Definiciones de estrofas.

```
estrofas ←{
  36: {ABCDEFFAEBDCCFDABEECBFADDEACFBBDFECA: sextina},
  15: {},
  14: {
    ABBAABBACDCDCD: soneto, ABABABABCDCDCD: soneto,
    ABBAABBACDECDE: soneto, ABABABABCDECDE: soneto,
    ABBAABBACDEDCE: soneto, ABABABABCDEDCE: soneto,
    ABBAABBACDCEDE: soneto, ABABABABCDCEDE: soneto,
    ABBAABBACDEDEC: soneto, ABABABABCDEDEC: soneto,
    ABBAABBACDCCDC: soneto, ABABABABCDCCDC: soneto,
    ABBAABBACDEEDC: soneto, ABABABABCDEEDC: soneto,}
  11: {},
  10: {
    abbaaccddc: espinela, abbacddeed: décima, abbccaadda: décima,
    aa'bb'cc'cddc: ovillejo},
  9: {ab'cb'd'ed': seguidilla compuesta},
  8: {
    ABABABCC: octava real,
    abbaacca: copla, ABBAACCA: copla de arte mayor},
  7: {},
  6: {
    abcabc : sextilla, aabccb: sextilla, abbaba: sextilla,
    aBaBcC: sexteto lira, abCabC: sexteto lira, ABABCC: sexta rima,
    aBaBcC: estrofa alirada, c'abc': estrofa manriqueña, aabbba: zéjel},
  5: {
    aBabB: lira, ABABA: quinteto, ABBAB:quinteto,
    ababa: quintilla, abbab: quintilla, aabba: quintilla, abaab: quintilla},
  4: {
    AAAA: cuaderna vía, ABABA: serventesio, ABBA: cuarteto,
    abab: cuarteta, abba: redondilla},
  3: {ABA: terceto, AAA: trístico monorrimo},
  2: {aa: pareado_aa, AA: pareado_AA, Aa: pareado_Aa, aA: pareado_aA}
}
```

Definimos las posibilidades estróficas en Algoritmo 65, donde no contemplamos el número exacto de sílabas, sino si son de arte mayor o menor y, en el último caso, si tienen menos de seis sílabas, lo que llevaría a sospechar de un pie quebrado. Las dos primeras circunstancias se representan en el algoritmo mediante la notación tradicional, mientras que la última lleva la minúscula seguida del símbolo de prima. Dado que

las agrupaciones están sujetas a unas reglas más o menos estrictas, podemos pronosticar la estrofa con gran exactitud considerando solamente esta clasificación categórica de los versos. El recurso al diccionario en lugar de a otros métodos responde a que, gracias al razonablemente limitado número de posibles agrupaciones simples, resulta más que factible encontrar patrones por comparación, lo que redunda no solo en precisión sino en la velocidad del proceso. Si hubiera un número ilimitado de combinaciones o una cantidad demasiado alta de estas, otra aproximación se habría hecho necesaria.

Consideraremos que las claves del diccionario están ordenadas, de manera que 36 precede a 15, y este a 14. Incluimos claves con valores vacíos, como 15 u 11, ya que simplificará después los algoritmos. Los valores denotan el nombre que recibe una determinada combinación de versos. Esto admite afinar más o menos, como, por ejemplo, la distinción que hacemos entre la espinela y otras décimas.

Definimos en Algoritmo 66 la función para buscar estrofas. Esta recibe una lista de versos y otra con números de versos que espera encontrar en las estrofas. En primer lugar, reduciremos el segundo parámetro al número de versos disponibles para agrupar. Esto es, si solo tenemos tres versos, sería una pérdida de tiempo tratar de formar agrupaciones de cuatro o más. Preparamos también un resultado por defecto, con una lista de versos y un esquema de rimas vacíos. Asimismo, definimos un esquema correspondiente a un aquelindo de un número de versos generoso, ya que lo buscaremos también sobre la marcha. Dado que la peculiaridad de la forma dificulta llegar a ella de manera fortuita, daremos por hecho que una serie de versos coincidente constituye en efecto un perqué.

Si la subrutina espera encontrar versos, definimos una lista con las letras que asignaremos al esquema de la rima, y creamos listas vacías de palabras metros, así como un diccionario para las rimas. También inicializamos el patrón rítmico, que será por ahora una cadena de caracteres vacía. Con esto, empezamos a recorrer los versos desde el primero hasta el n-ésimo, siendo n igual al primer valor de la lista de versos que esperamos hallar. Tomamos la última palabra del verso, que es la que contiene el axis. Asignamos a la variable *rima* las sílabas de rima la lista de rima y actualizamos la lista de letras disponibles para el esquema en función de la rima actual.

Aquí comienza una serie de operaciones condicionales. La primera la usamos para determinar si hemos encontrado una sextina. Se trata de añadir la palabra final del verso a una lista de palabras si no se encuentra ya en ella. De tratarse de una, la lista contendrá como máximo seis palabras. Después, cotejamos si la consonancia del verso actual está en la lista de rimas y el patrón contiene menos de seis versos, en

Algoritmo 66: Búsqueda de estrofas.

Método `BuscaEstrofa(`*versos, nversos*`):`
- nversos ←$\{m_i\}, \forall m_i \in nversos, m_i \leq |versos|$
- resultado ←(0, '')
- aquelindo ←'abbccddeeffgghhiijjkkllmmnnooppqqrrs'
- **si** *nversos* **entonces**
  - lista ←'abcdefgh'
  - palabras, mtrs ←()
  - rimas ←{}
  - patrón ←∅
  - **para** $fila_i \in versos_{0..nversos_0}$ **hacer**
    - palabra ←$fila_{texto_n}$
    - lista, rima ←`AsignaEsquema(`*fila, lista, rimas*`)`
    - **si** $palabra \notin palabras$ **entonces**
      - $\{palabras_0, ..., palabras_n\} \cup \{palabra\}$
    - **si** $fila_{consonancia} \notin rimas \wedge |patron|_{\neq}^{may=min} < 6$ **entonces**
      - rimas$_{fila_{consonancia}}$ ←rima
    - **si no, si** $fila_i \in versos_{0..nversos_0}$ **entonces**
      - **InterrumpeIteración**
    - patrón ←patron + rima
    - mtrs ←$\{m_i\} \forall m_i \in nversos, m_i \leq |patrón \wedge m_i \notin mtrs$
    - **para** $i \in mtrs$ **hacer**
      - **si** $i < |patrón|$ **entonces**
        - siguiente ←$patrón_i$
      - **si no**
        - siguiente ←'x'
      - patrón ←$patrón_{0..i}$
      - exceso ←$|patron| - 1 - |patrón|\%2$
      - **si** $aquelindo_{0..exceso} = patron_{0..exceso} \wedge |patrón| > 5$ **entonces**
        - resultado ←$|patrón_{0..exceso}|$, 'aquelindo')
        - **InterrumpeIteración**
      - **si no, si** $patrón \in estrofas_i$ **entonces**
        - **si** $i = 36 \wedge |palabras| = 6$ **entonces**
          - resultado ←$(i, estrofas_{i,patrón})$
          - **InterrumpeIteración**
        - **si no, si** $|patrón| = 5$ **entonces**
          - **si** $siguiente^{MIN} = patrón_n$ **entonces**
            - resultado ←$(4, estrofas_{i,patrón_{0..3}})$
          - **si no**
            - resultado ←$(i, estrofas_{i,patrón})$
          - **InterrumpeIteración**
        - **si no**
          - resultado ←$(i, estrofas_{i,patrón})$
          - **InterrumpeIteración**
- **devolver** resultado

cuyo caso creamos una nueva clave en *rimas*, que denominamos con la consonancia del verso, a la que le asignamos la rima. Si no se cumple la condición anterior, pero la consonancia no está entre las rimas, se interrumpe el bucle, dado que no esperamos encontrar más de seis tipos diferentes de estas. Si, por el contrario, el algoritmo no entra en esta bifurcación, añadimos la rima al patrón. A continuación, reducimos la cantidad de versos a probar aquella igual al tamaño del patrón y las de menor número de versos: no tendría sentido buscar, por ejemplo, un soneto en un grupo de diez versos.

Algoritmo 67: Asignación de la letra del esquema de la rima.

```
Método AsignaEsquema(verso, lista, rimas):
    si verso_rima ∈ rimas entonces
        rima ←verso_rima
    si no
        rima ←lista_0
        lista ←lista_1..n
    si verso_sílabas > 8 entonces
        rima ←rima^+MAYÚS
    si no, si verso_síl < 6 entonces
        rima ←rima'
    devolver lista, rima
```

Con la lista de posibles estrofas reducida, empezamos a probar sus elementos. Antes de comenzar, extraemos la rima que sigue a las que estamos analizando. Ante de las operaciones generales, buscamos un aquelindo, como un caso particular. Lo hacemos verificando que el patrón tenga más de cinco versos y que se corresponda con el esquema que hemos definido anteriormente hasta el último de sus versos pares. De ser así, tenemos la solución, por lo que interrumpimos aquí las iteraciones. Si este no es el caso, recorremos los esquemas del diccionario de los mayores a los menores. El esquema más extenso es la sextina, que identificaremos por tener treinta y seis versos y seis palabras en rima en posiciones variables —no comprobamos que sea exactamente la correcta—. Terminamos y salimos si encontramos la forma, o continuamos evaluando, de lo contrario.

Vemos a continuación si la agrupación es de cinco versos. De serlo, si el verso que sigue al grupo rima con el quinto de este, nos quedamos con los cuatro primeros y asumimos que el último se unirá al siguiente. De no cumplirse, resolvemos con los cinco versos. En ambos casos, interrumpimos una vez resuelto. Para todos los demás supuestos, devolvemos la estrofa correspondiente al patrón.

Algoritmo 68: Búsqueda de patrones asonantes.

Método `BuscaAsonantes(`*obra*`)`**:**
- resultado $\leftarrow()$
- asonancia $\leftarrow\emptyset$
- versos, estrofa $\leftarrow 0$
- n $\leftarrow 1$
- **para** $fila_i \in obra$ **hacer**
  - **si** $fila_{estrofa}$ **entonces**
    - **si** *versos* **entonces**
      - **si** *versos* $>$ *4* **entonces**
        - estrofa $\leftarrow$estrofa $+ 1$
        - $resultado \leftarrow resultado + ('romance', poema) \forall j \in 0...versos$
      - **si no**
        - $resultado \leftarrow resultado + (\emptyset, 0) \forall j \in 0...versos$
      - versos $\leftarrow 0$
    - resultado $\leftarrow$resultado $+ (fila_{estrofa}, fila_{nestrofa})$
  - **si no**
    - **si** *versos* $=$ *0* **entonces**
      - asonancia $\leftarrow\emptyset$
    - **si no, si** *versos* mod2 **entonces**
      - **si** $\neg asonancia \vee fila_{ason} = asonancia$ **entonces**
        - asonancia $\leftarrow fila_{ason}$
      - **si no, si** *versos* $=$ *3* **entonces**
        - resultado $\leftarrow$resultado $+ [(\emptyset, 0), (\emptyset, 0)]$
        - versos $\leftarrow 1$
        - asonancia $\leftarrow fila_{ason}$
      - **si no**
        - estrofa $\leftarrow$estrofa $+ 1$
        - resultado $\leftarrow('romance', estrofa) \forall j \in (0...versos - 1)$
        - asonancia $\leftarrow fila_{ason}$
        - versos $\leftarrow 1$
    - versos $\leftarrow$versos $+ 1$
  - n $\leftarrow$n $+ 1$
- **si** *versos* **entonces**
  - **si** *versos* $>$ *5* **entonces**
    - estrofa $\leftarrow$estrofa $+ 1$
    - resultado $\leftarrow('romance', estrofa) \forall j \in (0...versos)$
  - **si no**
    - resultado $\leftarrow(\emptyset, 0) \forall j \in (0...versos)$
- **devolver** *resultado*

Para asignar una letra al esquema rítmico, necesitamos como argumentos el verso, las rimas existentes hasta ahora en la estrofa y la lista de letras disponibles (Algoritmo 67). Si la rima del verso ya se encuentra en la estrofa, le asignamos esa letra. Si no, le asignamos la primera letra de la lista de letras disponibles, que sacamos

inmediatamente de la lista. Ahora comprobamos el número de sílabas del verso. Si es de más de ocho, será de arte mayor, por lo que pondremos la letra correspondiente en mayúscula. Si tiene menos de seis, la marcaremos como tal y servirá para identificar versos de pie quebrado.

Con las formas estróficas de rima consonante identificadas, pasamos a buscar agrupaciones asonantadas. En concreto, interesa sobre todo identificar los romances, ya que es la variedad más habitual de este tipo en el teatro del Siglo de Oro. Recorremos los versos mientras verificamos a cada iteración si el verso tiene ya asignada una agrupación estrófica. De ser así, miramos si tenemos versos asonantes guardados —veremos cómo a continuación— y, si son más de cuatro, lo tomaremos por un romance y los marcaremos como tal, si no, no los marcaremos. Los versos con marca estrófica no se modifican.

V	Síl.	Núcl.	Ason.	Cons.	Ritmo	Estrofa	n.º
-1							
1	7	ioIooEo	eo	ento	--+---+-	pareado	1
2	11	eoIeaEaoeEo	eo	ento	--+--+---+-	pareado	1
3	7	OeAoiAa	aa	ama	+-+--+-	pareado	2
4	11	AaoiaIEieAa	aa	ama	+----++--+-	pareado	2
5	7	iUoiiIo	io	into	-+---+-	pareado	3
6	11	auAaoUoaeIo	io	into	--+--+---+-	pareado	3
7	7	EaeUaEa	ea	enas	+--+-+-	pareado	4
8	11	eeOaaAaieEa	ea	enas	--+--+---+-	pareado	4
9	7	EaeEeOe	oe	onte	+--+-+-	pareado	5
10	11	oeEaoUoueOe	oe	onte	--+--+---+-	pareado	5
11	7	eOiAaIo	io	ino	-+-+-+-	pareado	6
12	11	eeeAaEeeeIo	io	ino	---+-+---+-	pareado	6
13	7	EieeeAa	aa	ada	+----+-	pareado	7
14	11	aaEaaEeaaAa	aa	ada	--+--+---+-	pareado	7
15	7	EeOeiEe	ee	ente	+-+--+-	pareado	8
16	11	aUeOeEoeaEe	ee	ente	-+-+-+---+-	pareado	8
17	7	AoOaeIe	ie	ibes	+-+--+-	pareado	9
18	11	UeaEoeoAeIe	ie	ibes	+--+---+-+-	pareado	9
19	7	eAeuaEa	ea	enas	-+---+-	pareado	10
20	11	aEaEaaoEaEa	ea	enas	-+-+---+-+-	pareado	10
21	7	EiEeoIe	ie	iθe	+-+--+-	pareado	11
22	11	aOaOeAUieIe	ie	iθe	-+-+-++--+-	pareado	11
-22							
23	7	IOiOeEe	ee	exes	++-+-+-	pareado	12
24	11	eaoAaIaoeEe	ee	exes	---+-+---+-	pareado	12
25	7	eiOEoIo	io	ido	--++-+-	Pareado	13
26	11	oeeeaAAoaIo	io	ido	-----++--+-	pareado	13
27	7	aoAaeUa	ua	uɾas	--+--+-	pareado	14
28	11	OoeeeIaioUa	ua	uɾas	+----+---+-	pareado	14
29	7	aIEoeAo	ao	ado	-++--+-	pareado	15
30	11	iOoeeOEooAo	ao	ado	-+---++--+-	pareado	15
31	7	OaOeOEa	ea	enta	+-+-++-	pareado	16
32	11	eEeeeAiOaEa	ea	enta	-+---+-+-+-	pareado	16
...							

Cuadro 9.4: Ampliación de la tabla de atributos métricos.

Si no hay marca de estrofa, hacemos una agrupación intermedia de los versos. El primer verso impar pone la asonancia como vacía. Si es el primero par o la asonancia

del verso coincide con la de los pares previos, asignamos la asonancia del verso a la asonancia del romance. Si es el cuarto verso y no se cumple lo anterior, dejamos los dos primeros versos sin marcar y empezamos de nuevo la operación con el verso anterior y el cuarto ahora como segundo. En cualquier otro caso, incrementamos el número de la estrofa a $n+1$, y marcamos todos los versos almacenados salvo el último como el romance al $n+1$-ésimo. El verso final pasa a ser el primero de la nueva serie y el corriente señala la asonancia. En cualquier caso, a cada iteración incrementamos el contador de versos.

Una vez finalizado el bucle es posible que quede una tirada de versos sueltos con rima asonante. De ser así, si son más de cinco, los aceptaremos como romance, si no, los dejamos sin marcar. Al final, completaríamos los datos métricos de Cuadro 9.1 con información sobre las agrupaciones simples, tal y como refleja Cuadro 9.4.

La agrupación requiere al final una revisión para corregir casos excepcionales. Estos se categorizan como licencias poéticas, decisiones editoriales y errores propios del programa. Los primeros son intrínsecos de la obra, los segundos están motivados por una elección editorial y los terceros son interpretaciones erróneas del esquema versal. Entre las licencias, destaca la equivalencia de los sonidos /e/ e /i/, que vemos en los múltiples casos que encontramos en *La vida es sueño*, en los ejemplos 17, 18, 19 y 20. Abordarlo de raíz requeriría reconocer y tratar dos posibles fonemas para el grafema ⟨i⟩, que no determinaría el contexto prosódico del segmento, sino los segmentos independientes que corresponden a los versos adyacentes. Ha de abordarse, pues, en la segunda pasada.

(17) ¡Traición! Guardas desta torre,
que entrar aquí nos dejasteis,
pues que nos dais a escoger,
el prendernos es más fácil. (Calderón, *La vida es sueño*, vv. 287-290)

(18) examinar el prodigio
que entre estos peñascos yace!
¡Rendid las armas y vidas,
o aquesta pistola, áspid (vv. 301-304)

(19) Pero si ya ha sucedido
un peligro de quien nadie
se libró, porque el honor
es de materia tan fácil (vv. 445-448)

(20) que otra desdicha no sea;
que unas a otras suceden,¸

herederas de sí mismas.
A la imitación del fénix, (vv. 1830-1833)

De más difícil solución son las situaciones semejantes a la del estribillo del romance gongorino «Contando estaban sus rayos» (Góngora y Argote, 1966, n.° 72), que Calderón intercala en los versos 1218 y 1219 (en cursiva) del ejemplo 21 , así como en otras partes. Si bien no supondría un inconveniente insuperable corregir este caso en la segunda pasada, resulta aventurado asumir tanto el número de versos intercalados como de la continuación del romance y, por ello, predecir todas las situaciones posibles. Asimismo, como en el ejemplo, los versos divisorios, dada su naturaleza extraordinaria en el poema, podrían presentar particularidades de interés filológico, por lo que se haría de la necesidad virtud. Encontramos dos romances con la misma rima separados por un pareado, lo que llama la atención y aconseja un examen detallado. Por consiguiente, en este caso, consideramos preferible dejarlo tal cual y que sea el buen sentido humano quien decida.

(21) el silencio de su mano,
se cante por mí esta letra:
Clarín que rompe el albor
no suena mejor.
Tu queja está bien fundada;
yo satisfaré tu queja,
y en tanto sírveme a mí.
Pues ya Segismundo llega. (vv. 1216-1223)

Los problemas de orden editorial tienen solución sencilla recurriendo a otra fuente. Lo vemos en la palabra de rima del verso 1608 del ejemplo 22. En la edición de la que procede la versión electrónica que manejamos (Calderón de la Barca, 1636/1997b), la palabra *perfetas* prefiere modernizar la ortografía, por lo que la rima se hace asonante; el problema no se da con otras ediciones que sí respetan la forma original (Calderón de la Barca, 1636/2019). En 23, aparecen los versos 2.011 y 2.012 invertidos, de manera que altera el esquema versal del romance en *e-e*. Ninguno de ambos casos se trataría de un error analítico propiamente dicho, en tanto que se atendría a las directrices del editor.

(22) Yo en esferas perfectas,
llamando el sol a cortes los planetas, (vv. 1608-1609)

(23) villano y grosero amante.
No quiero que me le entregues;
porque yo tampoco quiero,
de que yo te le he pedido.
con tomarle, que me acuerdes (vv. 2008-2012)

Los errores propios del programa pueden aparecer cuando se suceden dos unidades independientes que comparten una o más rimas. Así, en el ejemplo 25, encontramos una redondilla con estribillo, que se interpreta incorrectamente como una sextilla.

(24) Piensa bien si hay otros modos...
Perderme de otra manera.
Pues has de perderte, espera,
hija, y perdámonos todos.

Si este día me viera
Roma, en los triunfos de su edad primera,
¡oh, cuánto se alegrara,
viendo lograr una ocasión tan rara
de tener una fiera
que sus grandes ejércitos rigiera, (vv. 2652-2661)

(25) Y así, aunque a libraros vais
de la muerte con hüir,
mirad que vais a morir,
si está de Dios que muráis.
Mirad que vais a morir,
si está de Dios que muráis. (vv. 3092-3097)

La segunda pasada resulta trivial, como se ve en Algoritmo 69. Recorremos la tabla por unidades estróficas y, por cada estrofa, comprobamos el metro de su verso mayor. Si este es once, escandimos de nuevo aquellos versos que se habían analizado como octosílabos, pero, en esta ocasión, priorizando la evaluación como heptasílabo. Si, por el contrario, el verso mayor es de ocho sílabas y encontramos heptasílabos, los evaluamos nuevamente, si bien anteponiendo ocho sílabas a siete en las preferencias. Finalmente, si el verso de más tamaño no sobrepasa las siete sílabas, es probable que se trate de versos cantados hexasílabos, por lo que lo intentaremos de esa manera.

Por último, importa encontrar las silvas, por ser esta forma poética a la que se recurre con cierta frecuencia en el teatro. Lo haremos primero buscando series de

Algoritmo 69: Segunda pasada de correcciones.

para $i \in obra_{estrofa,nestrf}$ **hacer**
  si $max(i_{síl}) = 11$ **entonces**
    `Escande(`*verso, (7, 11, 8, 7, 6)*`)` $\forall verso \in i, verso_{síl} \notin \{7, 11\}$
  si no, si $max(i_{síl}) = 8$ **entonces**
    `Escande(`*verso, (8, 7 ,6, 5, 4)*`)` $\forall verso \in i, verso_{síl} \neq 8$
  si no, si $max(i_{síl}) < 8$ **entonces**
    `Escande(`*verso, (6, 7, 8, 5, 4)*`)` $\forall verso \in i, verso_{síl} \neq 8$

heptasílabos y endecasílabos sin marcar, que agruparemos como una silva. Podríamos después asimismo buscar versos en agrupaciones menores para volver a evaluar los grupos como componentes de una secuencia mayor, como la serie de pareados con los que comienza *La vida es sueño*. Ahora ya sería posible añadir al corpus anotado la información métrica con la que nos proponíamos caracterizar sus versos. Con esto, terminaría nuestra tarea y tendríamos un corpus dispuesto para análisis de lectura distante de las características métricas y dramáticas de sus textos.

Y pues que yo soy criado
de paz, solamente os ruego
que consideréis, señores,
que de los yerros ajenos
no hay cosa como callar;
perdonadnos, pues, los nuestros.

Calderón, *No hay cosa como callar*

Conclusiones 10

10.1 Recapitulación

Las humanidades digitales han demostrado su valía a lo largo de las últimas décadas y, desde el mismo nacimiento de la disciplina con el *Index Thomisticus*, ha sido la filología el ámbito de estudio en el que más presentes se han hecho. Esto ha propiciado unas cotas de desarrollo incomparables con otras ramas humanísticas, en las que la aparición de métodos asistidos por computador ha sido más tardía. En cualquier caso, la computación aplicada a la filología no es solamente viable hipotéticamente, sino una realidad consolidada para abordar el examen de grandes corpus literarios y, de ahí, la idea que puso en marcha esta empresa.

Disponemos de herramientas que permiten escudriñar detalles en un inmenso volumen de texto. Podemos realizar en cuestión de minutos tareas que hubieran sido inconcebibles hace ocho décadas y económicamente inalcanzables treinta años atrás. Pretender que la ciencia filológica permanezca ajena al desarrollo tecnológico y científico —ajeno a la sociedad de su tiempo, al fin y al cabo— resultaría no solo tremendamente arrogante, sino que haría un flaco servicio a la disciplina, al privarla de usar todos los medios disponibles para la interpretación textual, sin importar si son digitales o tradicionales. El científico, también en su vertiente humanista, no se debe al método, sino al conocimiento.

Esto no quiere decir que la filología deba renunciar a su tradición hermenéutica, ni mucho menos. Muchas de las diversas escuelas de crítica literaria basadas en *close reading* tienen en su haber sólidos argumentos que las avalan como marco interpretativo. Esto, sin embargo, no debería estar reñido con abrazar los avances técnicos, tanto para complementar y dar más peso a las aproximaciones clásicas como para destapar nuevas cuestiones que han permanecido ocultas hasta ahora, enterradas bajo

una colosal montaña de fuentes, una mole tan descomunal que hacía humanamente imposible acercarse a los aspectos mínimos desde la totalidad textual. No se tome esto como un llamamiento a aceptar nuevas aproximaciones a despecho de la rigurosidad, pues esta debe ser clave para cualquier metodología, sea asistida por computador o no. Siempre sin olvidar que, en el centro de todo ello, se ubica el texto.

La práctica consecuente de la filología digital requiere, no obstante, conocer sus cualidades y limitaciones. Podemos usar una metodología *asistida* por datos para evaluar cierta información textual y emitir un juicio, considerar hipótesis o plantear nuevas cuestiones. Podemos, por otro lado, decidirnos por una metodología *guiada* por datos para obtener respuestas cuantitativas a nuestras cuestiones. En nuestro caso, hemos definido de antemano tanto las preguntas como los criterios exactos bajo los que considerar la solución, así como las condiciones objetivas en las que la buscaremos. De cualquier manera, esto ha condicionado tanto el tipo de datos empleados como los nuevos tipos propuestos.

En efecto, el modelo elegido ha determinado en gran medida el tipo de datos requeridos: no estructurados, semiestructurados o estructurados. En los dos últimos casos, se han tenido que reunir todos aquellos aspectos del texto necesarios para el estudio al que debía servir, las investigaciones del proyecto Sound and Meaning. En el caso de los datos no estructurados, hemos descrito ciertas herramientas más o menos complejas para poder utilizarlos de forma masiva en, por ejemplo, búsquedas de fragmentos textuales. Esto ha sido quizás una de las precondiciones que más peso ha tenido nuestro trabajo. Para satisfacer las demandas de ambas aproximaciones, ha sido necesario concebir una estructura de datos que permitiera estudios cuantitativos completamente digitalizados, pero que también proporcionara información útil para hacer inferencias cualitativas. En otras palabras, por una parte, los datos producidos deben ser aptos para su proceso inmediato mediante, por ejemplo, programas de estadística o *machine learning* y, por otra, han de permitir las búsquedas casuísticas particulares según unos criterios dados.

Para ello, hemos diseñado un sistema de datos estructurados. Este representa elementos textuales del drama y de la métrica, así como otros editoriales que permiten identificar la procedencia del texto adecuadamente. Hemos considerado también la eventualidad de que los textos se presenten en diversas codificaciones, por lo que, antes de tratar una fuente, nos hemos asegurado de unificar la forma de representación para que todos ellos estén dispuestos tal y como el programa espera encontrarlos.

A la hora de formalizar los presupuestos teóricos y, sobre todo, pensando en su modelización ulterior con un lenguaje de programación real, hemos tenido en cuenta que,

dado el tamaño del programa, habíamos de atenernos no solo a las reglas sintácticas propias de un lenguaje concreto, sino a unas buenas prácticas generales para mantener una complejidad asumible y evitar ralentizar el desarrollo. Dicho de otra forma: además de que funcione, hay que poner todas las facilidades para llegar a ello de una manera eficiente, rápida y segura.

En lo concerniente al teatro, hemos presentado los elementos necesarios para describir la estructura de la obra dramática. Lo hemos acometido partiendo del hecho teatral, de su posición en el mapa de los grandes géneros literarios, ofreciendo casi una panorámica de su esencia por contraste con la narrativa y la lírica. Una vez ubicado nuestro objeto de interés, hemos repasado sus características desde la dramatología para sentar las bases que han permitido descomponer la obra en sus elementos estructurales.

De esta visión general, hemos reducido el foco a la vertiente de la obra fijada por escrito, ya que esta es el objeto final de nuestra investigación. Aquí hemos revisado la relación de la estructura externa del texto con aquellos elementos escénicos y ortotipográficos mensurables que sirven de criterio para descomponerlo. Hemos discutido la forma de reconocer dichos elementos y hasta qué punto son suficientes sus criterios de identificación inferidos de la estructura externa para determinar aquellos sin necesidad de recurrir a la interna. Precisamente, definir los constituyentes que permiten asignar valores a estas variables es la parte fundamental de nuestro análisis digital del drama.

Dado que nos centramos en el texto escrito, presentamos las convenciones que este sigue, prestando especial atención a las ediciones críticas modernas y sus particularidades, pues son estas las que más garantías ofrecen en cuanto al respeto por las fuentes originales[132].

De todo esto, hemos identificado los segmentos más productivos, partiendo de elementos estructurales dramáticos mínimos, como son los parlamentos y las acotaciones externas. Si bien hemos podido marcar los actos, que están indicados de forma explícita, no ha sido así con secuencias de orden inmediatamente inferior. No obstante, nuestros resultados ofrecen información que facilita su identificación. En concreto, mediante el seguimiento de la entrada y salida de personajes a escena para tratar de aproximar los puntos en los que el escenario queda vacío. Asimismo, la segmentación métrica que hemos conseguido ayuda al análisis, si bien no todos los cambios métricos denotan una secuencia englobante y se requiere evaluar los elementos internos.

[132]En sentido estricto, no nos ocupamos de la edición crítica sino del texto fijado por esta.

Se han identificado los elementos que caracterizan al parlamento. Por un lado, sus atributos, que son el locutor o locutores que lo pronuncian, al igual que elementos modales opcionales, por ejemplo, indicaciones escénicas de aparte o cantado. Por otro, sus partes constituyentes o líneas, que también están indicadas explícitamente. Mientras que en las ediciones históricas la línea se corresponde con el verso y varios parlamentos pueden compartir una línea, este no es el caso en las modernas. Por fortuna, no ha supuesto gran problema porque los versos compartidos se señalan por medio del sangrado en la mayoría de las ediciones empleadas.

Hemos presentado los fundamentos para la escansión métrica de los versos teatrales y su ordenación en agrupaciones estróficas. Estos no difieren de los empleados en la lírica, por lo que, a todos los efectos, nos hemos conducido por las mismas reglas que se aplican a los poemas. Por ello, los tratados de poesía han sido el punto de partida para examinar las configuraciones métricas de las obras dramáticas. Nos hemos centrado en los metros más comunes, como son los octosílabos, endecasílabos, heptasílabos y hexasílabos, pero sin descuidar la posibilidad de versos quebrados ni la presencia de otros menos habituales, como los pentasílabos, siempre teniendo en cuenta que otras medidas son igualmente plausibles. El límite lo marca la prosodia española, por lo que los pasajes cantados pueden no corresponderse con el análisis, al atenerse a un ritmo acentual.

La unidad autosuficiente mínima de la métrica castellana es el verso, de ahí que nos hayamos ocupado de caracterizar la configuración que adoptan componentes subordinados. En concreto, las partes elementales tratadas son la sílaba y el axis métrico. Hemos estimado el tamaño y ritmo del verso, considerando los elementos definitorios de la sílaba métrica, como los fenómenos intersilábicos y la tonicidad. También hemos identificado la naturaleza y posición de las vocales de sílabas en contacto capaces de alterar la cantidad del verso y el axis métrico donde se resuelve la rima. Este es un criterio fundamental para combinar un verso con otros, puesto que altera el cómputo silábico respecto al gramatical según las vocales que queden a la derecha del eje. Abrimos también la posibilidad de clasificar versos de una cantidad dada, tanto endecasílabos en sus tipos tradicionales como octosílabos, conforme a diversas tipologías.

Los versos se asocian entre sí de acuerdo con su metro individual y rima en agrupaciones tanto estróficas como no. En el primer caso, hemos discriminado series de versos para formar una unidad intermedia, la estrofa. Esta es simple, como el cuarteto, o compuesta si está constituida por varias estrofas, como la décima, por ejemplo. Las estrofas se agrupan de forma que el conjunto constituye una unidad de orden superior, como un poema. Sin embargo, la configuración intermedia no es una condición

necesaria para la creación de una unidad poética de orden superior porque los propios versos también se agrupan independientemente para constituirla, como sucede en el caso de los romances. Esto lo hemos tenido también en cuenta.

El análisis teatral obliga a buscar patrones para, en primer lugar, aislar las diferentes unidades estróficas y superiores que componen la obra, dado que, al contrario de lo que ocurriría con un poema, las agrupaciones métricas se suceden en esta sin solución de continuidad. Además de la rima, hemos observado el metro de los versos, dado que las estrofas heterométricas no son arbitrarias, sino que suelen combinar tipos particulares de verso, como heptasílabos y endecasílabos. Sin embargo, el conjunto de combinaciones suele limitarse a unas pocas en la práctica. Disponer de estos segmentos métricos superiores como elementos individuales ha permitido clasificar las agrupaciones según los modelos vistos en la teoría.

Finalmente, hemos abordado la métrica con una perspectiva fonológica. Esto ha permitido representar textos escritos considerando los aspectos sonoros intrínsecos que de ellos se desprenden y aproximarlos a la pronunciación. Sin dejar de ser una idealización del habla humana, la fonología ha facilitado hacer una distinción clara entre las unidades textuales. Esto es, empleándola, podemos definir un modelo del signo lingüístico en la lengua oral. Esta idealización nos abstrae de las características físicas de las realizaciones particulares de los sonidos. Así, resulta factible obtener una generalización lingüísticamente relevante de la pronunciación del texto escrito. En tanto que partimos de textos escritos y no de muestras de habla producidas por un hablante, hemos transcrito —más bien transliterado— las fuentes.

Para construir este modelo, hemos vuelto sobre el concepto de fonema, que hemos tomado como unidad mínima de los textos. Lo hemos hecho, claro está, considerando solo el repertorio fonológico del español. La aproximación ha sido principalmente articulatoria porque nos interesaba representar con precisión y de forma concisa las unidades fonológicas que se infieren del texto. Sin embargo, ha sido obligado considerar algunas cuestiones fonéticas para resolver sistemáticamente determinadas secuencias de fonemas.

Por conveniencia, hemos entrado someramente en algunos aspectos de la fonética acústica. Si bien no la hemos usado en la representación del texto, ha sido una potente herramienta para evaluar muestras de lengua hablada como vía para resolver indeterminaciones. En caso de ambigüedad, hemos evitado recurrir a la autoelicitación; en su lugar, hemos observado textos de informantes para inferir la realización más neutra o, al menos, si es aceptable una forma determinada. Esto es, cuando el camino deductivo parece llegar a su fin, siempre podemos tomar el desvío inductivo

mediante la fonética acústica. Lamentablemente, esto también significa que, ante la presencia de varias alternativas válidas, hemos descartado todas excepto una para poder generalizar. Como ya dijimos antes, en la creación del modelo siempre se pierde algo respecto al referente real.

10.2 Respuestas a las preguntas de investigación

Habíamos comenzado este trabajo formulando tres preguntas de investigación: cómo automatizar el análisis de la estructura dramática, cómo escandir el verso dramático y cómo organizar toda esta información en un corpus. Veamos una por una y pormenorizadamente las respuestas que proponemos.

I. ¿Cómo automatizar el análisis de la estructura dramática?

En lo referente a la primera pregunta de investigación, hemos visto en los capítulos sexto y séptimo que la estructura dramática puede organizarse en función de sus elementos constituyentes. En concreto, hemos partido de entidades ya propuestas, a la manera descrita por el estándar XML-TEI. Sin embargo, esta disposición no se presta directamente a los exámenes de lectura distante porque, para considerar la complejidad estructural de la pieza dramática, requeriríamos trasladar la información a datos estructurados, esto es, en forma de tabla, que es el formato con el que funcionan internamente los programas estadísticos. Si bien existen paquetes que *leen* datos semiestructurados, como los representados en los susodichos archivos XML-TEI, estos se convierten previamente a un *marco de datos*, o sea, se tabulan. El problema que esto presenta es que las entidades están predeterminadas para tareas muy concretas, bien con información estadística del léxico de cada obra, como hacen los programas de estilometría, bien con relaciones escénicas, como es el caso de los paquetes dedicados a los estudios dramétricos. Esto supone una limitación que dificulta sobremanera los análisis métricos distantes, en tanto que no tiene en cuenta el metro como entidad dentro de la estructura dramática.

La solución propuesta consiste en codificar la información dramática considerando la línea como la unidad mínima, de modo que los datos adicionales relativos a esta puedan definirse en tantas columnas como se requiera. Para nuestro modelo, hemos planteado entidades de cuatro tipos. Primero, aquellas destinadas a los metadatos, que se aplican a la obra en su totalidad; esto es, en cada una de esas columnas, todas las filas correspondientes a una sola composición teatral contienen un valor idéntico. Este tipo de columnas incluyen información invariable como el autor de la pieza, el título y el subtítulo de esta, el género y el subgénero, el año de publicación, así como

la información relevante sobre la procedencia editorial de la fuente. En segundo lugar, tenemos elementos estructurales. Estos son externos, como el número de acto y de línea, o internos, como el número de parlamento. Tercero, tenemos elementos dramáticos, que incluyen al locutor y el propio texto de la línea. Finalmente, hay elementos poéticos; estos son el número de verso y de unidad poética en el que este se enmarca. La disposición tabular con arreglo a estas variables permitiría establecer redes de personajes como en los análisis dramétricos ordinarios, pero también segmentar la obra de forma externa e interna atendiendo a otros.

De esta manera, a cada línea le corresponde un número y, además, el propio texto, independientemente de si este se repite de nuevo en la obra, pues, a efectos de clasificación, serían considerados instancias distintas. Tendría, sin embargo, los mismos metadatos que el resto de la obra, así como el número de jornada del resto de líneas que se pronuncian en el mismo acto y el mismo número de parlamento y locutor que el resto de versos de dicho parlamento. El número de verso puede ser único, pero no necesariamente, ya que a veces se distribuye a lo largo de varias líneas —en teoría, sería posible distribuirlo en tantas como sílabas en las que podemos separarlo—, cuando el verso lo inicia un personaje y otro u otros terminan de recitarlo.

II. ¿Cómo escandir automáticamente el verso dramático?

Para contestar a la segunda pregunta, hemos construido la respuesta sobre la precedente, pues la identificación del verso dramático implica caracterizar primero las partes constituyentes externas del texto teatral. Una vez hecho esto, lo hemos procesado verso a verso considerando el contexto métrico y dramático. Para ello, hemos propuesto un modelo de escansión fonológico recursivo basado en prioridades. La base es el sistema *clásico* hasta ahora aplicado a la lírica, que consiste en un análisis morfosintáctico del verso para distinguir entre categorías gramaticales átonas y tónicas, división silábica y ajuste de las sílabas en función de un conjunto de reglas dado. Sin embargo, cada una de las partes de este enfoque es susceptible de mejoras en el procedimiento para llevar a cabo su propósito. Esto es, dar los mismos pasos empleando diferentes medios. En nuestro caso, hemos perfeccionado la división silábica y el tratamiento de las categorías gramaticales y hemos formulado reglas diferentes para el ajuste silábico. Asimismo, hemos cambiado la forma de representación del texto para trabajar sobre fonemas.

El sistema que planteamos parte de una transliteración fonológica de las palabras, incluyendo la distinción entre vocales silábicas y no silábicas. La aproximación grafémica tradicional implica un análisis fonológico inconsciente, por lo que resulta difícil

automatizarlo sin recurrir a un proceso fonológico implícito, con la complicación adicional de tener que ofuscarlo bajo subterfugios ortográficos. Trabajar con fonemas, por un lado, simplifica el desarrollo conceptual del método porque permite operar en términos fonológicos directamente sin necesidad de *traducir* a ortográficos; por otro lado, facilita el tratamiento informático de los caracteres, pues cada unidad textual mínima está representada por un único carácter digital. El cambio supone una ventaja, en tanto que, al distinguir entre vocales y vocoides, desaparecen las ambigüedades. Asimismo, se presta mejor a las generalizaciones que el análisis grafémico —por ejemplo, tratamos diptongos y sinalefas como uniones equivalentes de un núcleo silábico y una vocal no silábica, y no requiere prever excepciones en los segmentos consonánticos para tratar los dígrafos—. Por otra parte, permite emplear más de un acento prosódico, como es menester con los adverbios en *-mente*.

Otra mejora sustancial deriva del procedimiento para hacer la división silábica. Aunque la ortografía y la prosodia suelen ir de la mano las más de las veces, sus caminos se separan ocasionalmente. Si bien esto encuentra una solución editorial mediante el empleo de diacríticos (*crüel*), este no es siempre el caso y, de cualquier modo, no lo es para formas verbales excepcionales de uso común. El poeta escribe sus versos en el papel y no al contrario, por lo que poco sentido tiene ceñirse a la ortografía cuando esta contraviene la prosodia.

El complemento sintáctico al análisis morfológico supone otro avance. Numerales, nombres propios compuestos, fórmulas de tratamiento... Aunque resulten excepcionales en la lírica, son corrientes en el teatro. Por eso, ignorarlos no provocaría fallos marginales sin apenas efecto en el conjunto, sino errores recurrentes acumulados que desvirtuarían los metadatos del corpus anotado. Por lo tanto, su consideración resulta crucial para obtener resultados aceptables.

En cuanto al sistema de reglas de reajuste del verso, los resultados sugieren que el propuesto da un paso en la buena dirección. El modelo de prioridades combinado con la reevaluación en tiempo real alcanza una precisión que supera todos los sistemas de escansión del español publicados. Las ventajas del tratamiento recursivo por prioridades frente al análisis lineal del verso son evidentes: una vez que hemos hecho un reajuste versal, la configuración silábica es otra, por lo que resulta incoherente seguir analizándolo como si de la secuencia de sonidos inicial se tratase.

Una vez tenemos los versos con sus correspondientes metros, podemos entenderlos como una lista de elementos en los que buscar patrones. Estos responden a agrupaciones según el número de sílabas de los versos, por un lado, y, por otro —de vital importancia—, de acuerdo con su rima y el tipo de esta.

Por otra parte, el texto dramático, a diferencia del lírico, es polimétrico. Los distintos segmentos se suceden sin solución de continuidad, por lo que resulta imprescindible establecer sus límites para caracterizarlos. Al contrario que con la escansión de poemas, donde el principio y el fin de la unidad vienen dados, las agrupaciones de versos del texto dramático no presentan una separación explícita entre ellas. Por lo tanto, antes de clasificar, hay que segmentar el texto según las características comunes de los versos.

Los versos se agrupan en virtud de su caracterización métrica y de rima, conociendo que la mayoría de las estrofas son consonantes y las asonancias aparecen típicamente en romances, siendo las estrofas asonantadas excepcionales. En consecuencia, buscamos esquemas rítmicos consonantes que repitan el patrón de estrofas del mayor número de versos posible y, de no encontrarlas, probamos en orden sucesivo con las estrofas inmediatamente menores según el número de versos. Los versos consonantes se clasifican de este modo en estrofas simples. A continuación, evaluamos los versos que han quedado sin marcar. Primero cotejamos los metros asonantados con un número de versos fijo, tal como, por ejemplo, la cuarteta tirana y, después, determinamos los de longitud variable, como los romances. Una vez hechas las agrupaciones simples, se intenta unirlas en otras complejas donde sea posible. Esto es, el comienzo de *La vida es sueño* daría como resultado del paso anterior una tirada de cincuenta y un pareados de versos heptasílabos y endecasílabos. Tras esta segunda evaluación, la serie se agruparía como una única silva.

De aceptar cierto margen de error en la agrupación para el tamaño del verso, considerando la posibilidad de una escansión en falso —que, si bien infrecuente, no es imposible, dada nuestra aproximación a la polimetría—, puede aprovecharse el resultado para corregir posibles errores. Esto es, la rima ofrece la guía que seguir y el número de sílabas se considera una aproximación. Una vez agrupadas las estrofas, se deduce el número correcto de sílabas, tanto por las reglas poéticas como, mediante inducción, a partir de versos sin ambigüedades de la misma unidad. Con esto, pueden volver a escandirse aquellos que presenten un metro inusual, pero, en esta ocasión, forzando el número de sílabas en la escansión según el contexto.

III. ¿Cómo organizar la información resultante en un corpus anotado?

La última cuestión, si bien es tal vez la más sencilla de resolver en cuanto a la técnica, presenta mayor complejidad desde el punto de vista conceptual. No se trata solo de identificar entidades, sino de caracterizarlas de tal manera que su representación

no dé lugar a ambigüedades. El primer paso era convertir las ediciones digitales a archivos de texto, eliminando todo el contenido extratextual. Si bien esto permite hacer búsquedas de palabras de forma relativamente sencilla, los resultados son poco prácticos, en tanto que ubicarlos requiere valorar manualmente su posición dentro del texto. Peor aún, al haber limpiado la fuente de todo aquello ajeno al propio texto, se pierden datos como, por ejemplo, la paginación. Por lo tanto, esta primera aproximación valdría para identificar los textos que contienen ocurrencias de una búsqueda dada y hacer una estimación a ojo de su posición, con la que buscarla de vuelta en el texto de origen. Esto, ya de por sí, resulta de gran utilidad, pues agiliza la investigación cualitativa sobremanera. Permitiría incluso la cuantitativa, considerando, por ejemplo, el número de ocurrencias de una búsqueda, lo que se extendería a análisis más detallados, desde la complejidad léxica del texto a la estilometría. Sin embargo, detenerse aquí dejaría sin explotar todo el potencial que ofrecen las nuevas tecnologías.

Lo siguiente sería ir un paso más allá de las palabras y considerar sus relaciones. El ejemplo por antonomasia en el mundo de la edición digital es el estándar XML-TEI. Este caracteriza los elementos textuales mediante una jerarquía, por lo que un verso pertenece a un grupo de versos, estos, junto a la didascalia de personaje, el identificador del locutor y otros datos, a un parlamento, el parlamento a una escena, la escena a un acto, y el acto al cuerpo del texto. El problema que presenta es que requiere categorizar y ordenar los elementos explícitamente, por lo que se necesitaría procesar el texto de origen —mejor dicho, los textos, cientos, tal vez miles, de comedias, literalmente millones de versos— manualmente. La inviabilidad de esto es evidente, por lo que proponemos un pequeño rodeo. Se trata de preparar los archivos de texto de tal manera que, sin alterar en lo esencial el método ordinario, se reflejen los elementos textuales sin ambigüedades y, hasta cierto punto, de forma implícita, al imitar las convenciones editoriales de maquetación del drama. Así, es el propio texto de origen el que proporciona la información requerida. Sin que haya una diferencia notable en las posibilidades de uso finalista con respecto a los archivos de texto ordinarios, ahora es posible manipular y analizar la información dramática automáticamente y clasificar las entidades que contiene, lo que sí es un paso cualitativo sustancial.

Aunque los datos semiestructurados resultan convenientes para almacenar la información de partida, la metainformación y sus relaciones, no se prestan bien a tareas que demandan un trabajo intensivo de búsqueda o el cotejo masivo de datos. Precisamente, ese tipo de trabajos son los que requieren los análisis estadísticos o las bases de datos, por lo que ambos trabajan con estructuras tabulares. En consecuencia, se ha de traducir la disposición semiestructurada a una tabla secuencial, en la que las

relaciones entre entidades no se representen a través de la jerarquía, sino mediante los valores de sus columnas.

Nuestra aproximación consiste en tomar aquí como unidad mínima no el parlamento o las líneas, sino el verso y, a idéntico nivel jerárquico, las acotaciones y las líneas en prosa y ecos, que se considerarían *versos* carentes de metro, así como locutor y parlamento, según el caso, que se numerarían aparte. A cada una de estas unidades se le asigna una fila, que se caracteriza en las columnas mediante variables cuantitativas y categóricas. De esta manera, unas columnas representan la información metatextual, como el autor, título o datos de la edición; otras columnas contienen información estructural, como el número de acto, de parlamento (o acotación) y de verso; otras, información dramática, como las líneas y el personaje o personajes que la pronuncian y, por último, aquellas que proporcionan información métrica, como el ritmo, metro, rima, estrofa a la que pertenece y el tipo de esta. La realización física de la estructura es un archivo CSV, ya que este formato es compatible tanto con los paquetes de estadística como las bases de datos más comunes.

Poniendo todo junto

En resumen, la cadena de trabajo que proponemos comienza con la corrección semi-manual de cada uno de los textos de origen, con la peculiaridad de que el resultado ha de facilitar la distinción entre entidades textuales. Esto permite identificar los versos y su contexto para crear una tabla a partir de ellos. Una de estas columnas contiene el verso en cuestión, que puede analizarse mediante PLN para añadir la información métrica en nuevas columnas.

Esta escansión se lleva a cabo mediante el análisis morfosintáctico para identificar los acentos prosódicos del verso. Después, se corrige el metro mediante la aplicación de las licencias poéticas. Este proceso se basa en asignar un grado de preferencia a los contactos vocálicos y resolver potenciales uniones o separaciones recursivamente. La preferencia se confiere según criterios ortológicos. La resolución consiste en el ajuste versal atendiendo al contacto vocálico de preferencia más alta, tras lo que, si el metro sigue sin cuadrar, vuelve a evaluarse el verso ajustado, y se repite el proceso hasta obtener el metro esperado; si no es posible, el proceso comienza de nuevo desde el principio pero con otro metro objetivo. Los versos examinados se agrupan finalmente atendiendo a la información obtenida. La tabla resultante de procesar un número suficiente de obras —casi medio millar en el proyecto Sound and Meaning a la fecha de redactar estas últimas líneas—, permite realizar análisis de corpus en función de todas las variables reflejadas en ella.

10.3 Cuestiones abiertas y unas palabras finales

Durante el transcurso de este trabajo han surgido ciertas cuestiones, unas a raíz de problemas a resolver, otras como producto inesperado de los experimentos. Algunas de ellas tienen cierto interés por sí mismas, por lo que no podíamos poner fin a este trabajo sin notar los interrogantes que abren para —esperemos— ser explorados en el futuro como merecen. Mencionaremos las tres que más nos han llamado la atención.

Primero, durante las largas cavilaciones para abordar la resolución de las sinalefas, se hizo necesario considerar también aquellas que ocurren en versos compartidos. Como resulta evidente, no se da un desplazamiento articulatorio entre dos cavidades bucales, por lo que la situación ha de resolverse de otra manera. Esto condujo a la realización de análisis fonéticos acústicos, que revelaron diversas técnicas escénicas para enfrentarse a la peliaguda situación. Entre los hallazgos más interesantes destaca que algunas de estas estrategias entraban en abierta contradicción con la forma poética del verso, al hacer alteraciones espurias en el recuento silábico. Esto aconseja recopilar un número mayor de muestras para poder llevar a cabo un estudio descriptivo de las prácticas teatrales al respecto. Asimismo, da pie a reflexionar sobre la aproximación al fenómeno más apropiada, respetuosa con el espíritu de la obra y acorde a la sensibilidad del poeta.

En segundo lugar, al examinar los textos, hemos encontrado otro fenómeno que se observa muy especialmente también en los versos compartidos. Se trata de la presencia de acentos contiguos. Si bien en la lírica estos se prestan mejor o peor a ser reducidos donde convenga según las necesidades del metro y el gusto del poeta, este no es siempre el caso del texto dramático. En una serie de réplicas y contrarréplicas monosilábicas, por ejemplo, resulta no únicamente difícil privar de la calidad tónica a las líneas allí donde lo requiera el metro, sino que, de hacerlo, se correría el riesgo de socavar el efecto dramático de los parlamentos. Resulta difícil reconciliar métrica y acción en estos casos porque demandan acciones contradictorias. ¿Cuál de las dos ha de satisfacer la resolución del metro? Cabría asimismo plantearse la cuestión desde el punto de vista compositivo, preguntarse qué idea podría tener el autor cuando pergeñó el verso.

Tercero y último, durante las pruebas, aquellas obras que presentaban errores de edición, transmisión o composición que afectaban a los versos producían inconsistencias fácilmente visibles en los resultados, por lo que se planteó la posibilidad de crear un paquete dotado de una interfaz de usuario sencilla que permitiera a cualquiera, aun sin conocimientos técnicos, llevar a cabo análisis automáticos y cotejar los resultados. Esto permitiría en cuestión de segundos encontrar metros y rimas inconsistentes, lo

que podría agilizar la edición de textos, al menos en lo métrico, cuando la ocasional palabra mal puesta o enmendada altera el recuento silábico o el esquema de la rima.

Arribamos por fin a puerto. Atrás quedan las ideas y conceptos que hemos discutido y llevado a la práctica para marcar el rumbo que trazaron los interrogantes formulados al zarpar. Durante esta travesía, hemos ido esbozando propuestas en cada singladura, las hemos refinado en la siguiente y, por último, las hemos compendiado ya desbastadas en estas páginas postreras. Ha sido un largo y arduo viaje, este que ahora acabamos. Concluimos con la esperanza de que, más allá del mucho o poco interés que este trabajo pueda suscitarle al lector —que de más ha hecho llegando hasta aquí—, le sirva al menos como acicate para volver a explorar el teatro clásico español con la idea de contemplarlo desde una perspectiva diferente, nueva o antigua.

Apéndices

Resúmenes A

Zusammenfassung

Ziel dieser Doktorarbeit ist es, der spanischen Philologie neue Methoden, Werkzeuge und Proofs of Concept zur Verfügung zu stellen, welche die digitalhumanistischen Studien des spanischen Theaters der Frühneuzeit erleichtern. Insbesondere entwickelt es Analysemethoden für polymetrisch versifizierte dramatische Werke des Goldenen Zeitalters, ausgehend von einer theoretischen Basis dramatologischer, poetischer und phonologischer Elemente, um diese als Algorithmen und Datenstrukturen zu formalisieren, die für die Implementierung in Computerprogramme geeignet sind. Es greift weniger auf Digital Humanities als methodischen Rahmen zurück, sondern sicht diese als Empfänger der Ergebnisse. Da die Eignung der vorgeschlagenen Formalisierung getestet wird, indem sie in reale Programme übersetzt wird, berücksichtigt das Design Programmiertechniken sowie die Verarbeitung natürlicher Sprache und Computerlinguistik. Der in dieser Arbeit vorgestellte Ansatz zielt darauf ab, die Sammlung und Klassifizierung von Daten zu erleichtern. Erstens schlägt es Ontologien und Klassen von Textelementen und ihre Beziehungen in Stücken des spanischen Goldenen Zeitalters vor; zweitens beschreibt es automatische Verarbeitungsmethoden, um sie zu extrahieren; schließlich wird untersucht, wie die resultierenden Informationen in Datenstrukturen organisiert werden können, welche für die weitere digitale Analyse geeignet sind.

Schlagwörter: literarische Analyse, Fernlesen, Digital Humanities, digitale Philologie, Spanisches Goldenes Zeitalter, Natural Language Processing, automatische Skansion

Abstract

This dissertation aims at equipping Spanish Philology with novel methods, tools, and proofs of concept to facilitate Digital Humanities studies of Spanish plays. In particular, it develops methods to analyse Spanish Golden Age versified plays. It departs from the theoretical grounds of dramatology, poetry, and phonology to formalise its elements as algorithms and data structures suitable for implementation as software. It resorts to Digital Humanities not as much as a methodological framework but as a target audience for the outcomes. Since the formalisation's suitability is tested by translating it into actual computer programmes, the design considers programming techniques, natural language processing and computational linguistics. The approach presented in this work intends to expedite data retrieval and classification. Firstly, it proposes ontologies and classes of textual elements in Golden Age plays and their interrelations; secondly, it describes automatic processing methods to extract them; finally, it explores how to organise the resulting information into data structures suitable for further digital analysis.

Keywords: literary analyisis, distant reading, digital humanities, digital philology, Spanish Golden Age, Siglo de Oro, Natural Language Processing, metrical scansion

Resumen

Esta tesis doctoral aspira a proveer a la filología española de nuevos métodos, herramientas y pruebas de concepto que faciliten los estudios humanísticos digitales del teatro aurisecular. En concreto, desarrolla métodos de análisis de obras dramáticas versificadas polimétricas del Siglo de Oro. Parte de una base teórica de elementos dramatológicos, poéticos y fonológicos, para formalizar estos como algoritmos y estructuras de datos aptos para su implementación en programas informáticos. Contempla asimismo las humanidades digitales no tanto como marco metodológico sino como receptor de los resultados. En tanto que la idoneidad de la formalización propuesta se somete a prueba mediante su traducción a programas reales, el diseño considera técnicas de programación, procesamiento del lenguaje natural y lingüística computacional. La aproximación que presenta este trabajo pretende facilitar la recolección y clasificación de datos. Primero, propone ontologías y clases de características textuales y sus relaciones en las obras teatrales áureas; segundo, describe métodos de procesamiento automático para extraer dichos elementos; finalmente, explora cómo organizar la información resultante en estructuras de datos aptas para análisis digitales ulteriores.

Palabras clave: análisis literario, lectura distante, humanidades digitales, filología digital, Siglo de Oro, Procesamiento del Lenguaje Natural, escansión automática

Listados de código B

B.1 Preprocesado

Listado `tei2txt.py`

```
1  #!/usr/bin/env python3
2  # v.0.0.1
3  # Fernando Sanz-Lázaro <fernando.sanz-lazaro@univie.ac.at>
4  import sys
5  import re
6  import os
7  from bs4 import BeautifulSoup as bs
8  import pandas as pd
9
10
11 basename = sys.argv[1].split('.')[0]
12
13
14 with open(sys.argv[1], "r", encoding="utf-8") as f:
15     soup = bs(re.sub(r'[\s\n]+', ' ', f.read()), 'html.parser')
16
17 teiheader = soup.teiheader
18 filedesc = teiheader.filedesc
19 titlestmt = filedesc.titlestmt
20 if titlestmt.find('title', attrs={'type': 'main'}):
21     title = titlestmt.find('title', attrs={'type': 'main'}).get_text().strip()
22 else:
23     title = titlestmt.find('title', attrs={'type': None}).get_text().strip()
24 subtitle = titlestmt.find('title', attrs={'type': 'sub'})
25 if subtitle:
26     subtitle = titlestmt.find(
27         'title', attrs={
```

```
28              'type': 'sub'}).get_text().strip()
29  else:
30      subtitle = ''
31
32
33  persname = titlestmt.persname.get_text().strip()
34  persname = re.sub('[\n ]{2,}', ' ', persname)
35  author_ids = titlestmt.find_all('idno')
36
37  #
38  publicationstmt = filedesc.publicationstmt
39  authority = publicationstmt.authority.get_text().strip()
40  editdate = publicationstmt.authority.get_text().strip()
41
42
43  sourcedesc = filedesc.sourcedesc
44
45  origdate = sourcedesc.find(re.compile('^date$', re.I),
46                             attrs={'type': re.compile('^print$', re.I)})
47  if 'when' in origdate:
48      origdate = origdate['when']
49  else:
50      origdate = ''
51  originalsource = sourcedesc.find(re.compile('^bibl$', re.I),
52                                   attrs={'type':
53                                          re.compile('^originalSource$', re.I)},
54                                   recursive=True).get_text().strip()
55
56
57  particdesc = teiheader.particdesc
58  listperson = teiheader.listperson
59  for tag in listperson.find_all(re.compile('^person$', re.I)):
60      name = tag.persname.get_text().strip()
61      id = tag['xml:id']
62      sex = tag['sex']
63      #print(f'name id sex')
64
65
66  textclass = teiheader.textclass
67  keywords = textclass.keywords
68  genretitle = keywords.find('term',
69                             attrs={'type': 'genreTitle'}).get_text().strip()
70  genreusubitle = keywords.find('term', attrs={'type': 'genreSubtitle'})
71  genresubitle = ''
72
```

```
text_file = f'<a>{persname}\n<t>{title}\n<g>{genretitle}\n'\
    f'<s>{genresubitle}\n<o>{originalsource}\n<f>{origdate}\n'
text = soup.find('text')
front = text.front
castlist = front.castlist
lista_personajes = ''

if castlist:
    for personaje in castlist.find_all(re.compile('^castItem', re.I)):
        if len(castlist) > 0:
            lista_personajes += '*'
        lista_personajes += personaje.get_text().strip()
    text_file += f'<el>{lista_personajes.strip("*")}\n'

body = soup.body
actos = body.find_all(re.compile('^div$', re.I))

if len(actos) < 1:
    actos = [body]
n = 1
tab = '\t'
for acto in actos:
    if len(actos) > 1:
        text_file += f'<j>{n}\n'
        n += 1
    for entrada in acto.findChildren(recursive=False):
        if entrada.name == 'stage':
            text_file += f'<i>{entrada.text.strip()}\n'
        elif entrada.name == 'sp':
            who = entrada.attrs['who'].strip()
            if entrada.speaker:
                speaker = entrada.speaker.text.strip()
            elif entrada.stage:
                speaker =  entrada.stage.text.strip()
            else:
                speaker = who
            text_file += f'{speaker} {who}\n'
            if entrada.lg:
                entrada = entrada.lg
            lines = entrada.findChildren(recursive=False)
            lineout = ''
            for line in lines:
                texto = line.text.strip()
                if line.name == 'stage':
                    lineout = f'<i>{texto}\n'
```

```
118                 elif line.name == 'l':
119                     if line.has_attr('part') and line['part'] != 'I':
120                         tab += '\t'
121                     lineout = f'{tab}{texto}\n'
122                 else:
123                     pass
124                 text_file += f'{lineout}'
125                 if line.has_attr('part') and line['part'] == 'F':
126                     tab = '\t'
127 
128         else:
129             pass
130 
131 with open(f'{basename}_conv.txt', 'w') as f:
132     f.write(text_file)
```

Listado `normal.py`

```
1  #!/usr/bin/env python3
2  import sys
3  import re
4  import os
5  import pandas as pd
6 
7  act = speech = nspeech = nstage = verse = 0
8  p = ''
9  author = title = genre = subgenre = gender = s = edition = date = 'ø'
10 
11 character = 'STAGE DIR.'
12 rtag = re.compile(r'^\s*<(.+)>')
13 rtext = re.compile(r'\s*<.+>(.*)$')
14 df = pd.DataFrame(columns=['Author',
15                            'Title',
16                            'Subtl',
17                            'Genre',
18                            'Subgenre',
19                            'Edition',
20                            'Date',
21                            'Act',
22                            'Speech',
23                            'Verse',
24                            'Character',
25                            'Type',
26                            'Gender',
27                            'Text'])
28 names_list = {}
```

```
29
30
31 def assign_gender(name):
32     namex = re.sub(r'(.*)\d.*', '\1', name).strip()
33     if namex in names_list:
34         return names_list[namex]
35     elif any(x.endswith(('AS', 'ª', 'ª')) or x.startswith('DOÑA ')
36              for x in (name, namex)):
37         sx = 'F'
38     elif any(x.endswith(('O', 'OS', '°', 'º')) or x.startswith('DON ')
39              for x in (name, namex)):
40         sx = 'M'
41     else:
42         sx = 'X'
43     names_list[namex] = sx
44     return names_list[namex]
45
46
47 with open(sys.argv[1], 'r', encoding='utf-8') as f:
48     for line in f.readlines():
49         flag = False
50         if rtag.match(line):
51             type_ch = rtag.search(line).group(1)
52             text = rtext.search(line).group(1)
53             if type_ch.lower() == 'j':
54                 act += 1
55             elif type_ch.lower() == 't':
56                 title = text.strip()
57             elif type_ch.lower() == 'a':
58                 author = text.strip()
59             elif type_ch.lower() == 'g':
60                 genre = text.strip()
61             elif type_ch.lower() == 'o':
62                 edition = text.strip()
63             elif type_ch.lower() == 'f':
64                 date = text.strip()
65             elif type_ch.lower() == 's':
66                 subgenre = text.strip()
67             elif type_ch.lower() == 'i':
68                 flag = True
69                 verse = -abs(verse)
70                 nstage -= 1
71                 if line.startswith('\t'):
72                     character = 'int. STAGE DIR.'
73                 else:
```

```
                    character = 'STAGE DIR.'
                speech = nstage + 0
                gender = 'ø'
            elif type_ch.lower() in ('p', 'e'):
                flag = True
                verse = -abs(verse)
                character = p
                gender = s
                text = str(line).strip()
            else:
                verse = abs(verse)
        else:
            verse = abs(verse)
            if re.match(r'^\s*$', line):
                continue
            elif not re.match(r'^\t+', line):
                p = line.split('#')[0].strip().upper()
                nspeech += 1
                s = assign_gender(p)
            elif re.match(r'^\t\S', line):
                flag = True
                verse += 1
                character = p
                gender = s
                text = str(line).strip()
            else:
                flag = True
                character = p
                gender = s
                text = str(line).strip()
            speech = nspeech + 0
        linea = {'Author': author, 'Title': title, 'Genre': genre,
                 'Subgenre': subgenre, 'Edition': edition, 'Date': date,
                 'Act': act, 'Speech': speech, 'Verse': verse,
                 'Character': character, 'Gender': gender, 'Type': '',
                 'Text': text}
        if flag is True:
            df = pd.concat([df, pd.DataFrame(linea, index=['row1'])])
df.to_csv(f'{os.path.splitext(sys.argv[1])[0]}.csv', index=False)
```

B.2 Transcripción fonológica

Listado `fonemas.py`

```
import re
```

```
2  from silabeador import Syllabification
3  from dataclasses import dataclass
4
5
6  @dataclass
7  class Values:
8      words: list
9      syllables: list
10
11
12 class Transcription:
13     def __init__(self, sentence, mono=False, exceptions=1,
14                  epenthesis=False, aspiration=False, rehash=False,
15                  stress='"'):
16         self.sentence = self.__clean(sentence, epenthesis)
17         self.__exceptions = exceptions
18         if rehash:
19             self.sentence = self.make_rehash(self.sentence)
20         self.phonology = self.transcription_fnl(self.sentence, mono,
21                                                 aspiration)
22         self.phonetics = self.transcription_fnt(self.phonology)
23         self.sampa = self.ipa2sampa(self.phonetics, stress)
24
25     @staticmethod
26     def __clean(raw_sentence, epen):
27         letters = {'b': 'be', 'c': 'θe', 'ch': 'ʧe', 'd': 'de', 'f': 'efe',
28                    'g': 'ge', 'h': 'haʧe', 'j': 'jota', 'k': 'ka', 'l': 'ele',
29                    'll': 'eʎe', 'm': 'eme', 'n': 'ene', 'p': 'pe', 'q': 'ku',
30                    'r': 'erre', 's': 'ese', 't': 'te', 'v': 'ube',
31                    'w': 'ubedoble', 'x': 'ekis', 'z': 'θeta'}
32         symbols = ['(', ')', '¿', '?', '¡', '!', '«', '»', '“', '”', '‘', '’',
33                    '[', ']', '-', '…', ',', ';', ':', "'", '.', '-', '"', '-']
34         diacritics = {'à': 'á', 'è': 'é', 'ì': 'í', 'ò': 'ó', 'ù': 'ú',
35                       'æ': 'e', 'ä': '_a', 'ë': '_e', 'ï': '_i',  'ö': '_o',
36                       'ã': 'á', 'õ': 'ó', 'â': 'a', 'ê': 'e', 'î': 'i',
37                       'ô': 'o', 'û': 'u', 'ç': 'θ'}
38         raw_sentence = raw_sentence.lower()
39         for char in letters:
40             if re.search(rf'\b{char}\b', raw_sentence):
41                 raw_sentence = re.sub(rf'\b{char}\b', letters[char],
42                                       raw_sentence)
43         for match in letters:
44             raw_sentence = re.sub(rf'\b{match}\b', letters[char], raw_sentence)
45         for char in symbols:
46             if char in raw_sentence:
```

```
47                raw_sentence = raw_sentence.replace(char, ' ')
48            for char in diacritics:
49                if char in raw_sentence:
50                    raw_sentence = raw_sentence.replace(char, diacritics[char])
51            if epen:
52                raw_sentence = re.sub(r'\bs((?![aeiouáéíóú]))', r'es\1',
53                                      raw_sentence)
54            return raw_sentence
55
56        @staticmethod
57        def make_rehash(sentence):
58            vowels = 'aeioujwăĕŏ'
59            for idx, syllable in enumerate(sentence):
60                if idx > 0 and len(syllable) > 1:
61                    if syllable[0].lower() in vowels and \
62                            sentence[idx-1][-1].lower() not in vowels:
63                        sentence[idx] = sentence[idx - 1][-1] + syllable
64                        sentence[idx - 1] = sentence[idx - 1][:-1]
65            return sentence
66
67        def transcription_fnl(self, sentence, mono, aspiration):
68            diacritics = {'á': 'a', 'à': 'a', 'ä': 'a',
69                          'é': 'e', 'è': 'e', 'ë': 'e',
70                          'ú': 'u', 'ù': 'u', 'ü': 'u',
71                          'í': 'i', 'ì': 'i', 'ï': 'i',
72                          'ó': 'o', 'ò': 'o', 'ö': 'o',
73                          '_': ''}
74            consonants = {'w': 'b', 'v': 'b', 'z': 'θ', 'ñ': 'ɲ', 'x': 'ks',
75                          'j': 'x', 'r': 'ɾ', 'R': 'r', 'ce': 'θe', 'cé': 'θé',
76                          'që': 'θë', 'ci': 'θi', 'cí': 'θí', 'cï': 'θï',
77                          'cj': 'θj', 'ch': 'ʧ', 'c': 'k', 'qu': 'k', 'll': 'ʎ',
78                          'ph': 'f', 'hie': 'ʝe', 'h': ''}
79            sentence = re.sub(r'(?:([nls])r|\br|rr)', r'\1R', sentence)
80            if aspiration:
81                sentence = re.sub(r'\bh', 'ʰ', sentence)
82            for consonant in consonants:
83                if consonant in sentence:
84                    sentence = sentence.replace(consonant, consonants[consonant])
85            if 'y' in sentence:
86                sentence = re.sub(r'\by\b', 'i', sentence)
87                sentence = re.sub(r'uy\b', 'wi', sentence)
88                sentence = re.sub(r'y\b', 'j', sentence)
89                sentence = sentence.replace('y', 'ʝ')
90                for key, value in diacritics.items():
91                    if key in 'áéíóú':
```

```
92                      sentence = re.sub(rf'{value}j̵\b', f'{key}i', sentence)
93              sentence = re.sub(r'j̵((?![aeiouáéíóú]))', r'i\1', sentence)
94          if 'g' in sentence:
95              for reg in [[r'g([eiéíiëï])', r'x\1'],
96                          [r'g[u]([eiéíëï])', r'g\1']]:
97                  sentence = re.sub(reg[0], reg[1], sentence)
98              sentence = re.sub(r'gü([eiéí])', r'gw\1', sentence, re.IGNORECASE)
99              sentence = re.sub(r'gu([aoáó])', r'gw\1', sentence, re.IGNORECASE)
100         transcription = self.__split_variables(sentence, mono)
101         for key, value in diacritics.items():
102             transcription.words = [word.replace(key, value)
103                                    for word in transcription.words]
104             transcription.syllables = [syllable.replace(key, value)
105                                        for syllable in transcription.syllables]
106         return transcription
107
108     def __split_variables(self, sentence, mono):
109         words = []
110         syllables_sentence = []
111         for word in sentence.split():
112             if len(word) > 5 and word.endswith('mente'):
113                 syllabification = Syllabification(word=word[:-5],
114                                                   exceptions=self.__exceptions,
115                                                   ipa=True,
116                                                   h=True)
117                 syllables = syllabification.syllables
118                 if len(syllables) > 1:
119                     syllables = syllables + ['ˌmen', 'te']
120                     stress = syllabification.stress - 2
121                     word = word.replace('mente', 'ˌmente')
122                 else:
123                     syllables = syllables + ['ˈmen', 'te']
124                     stress = -2
125             else:
126                 syllabification = Syllabification(word,
127                                                   exceptions=self.__exceptions,
128                                                   ipa=True,
129                                                   h=True)
130                 syllables = syllabification.syllables
131                 stress = syllabification.stress
132             syllables = self.__diphthongs(syllables)
133             syllables[stress] = f'ˈ{syllables[stress]}'
134             word = ''
135             for syllable in syllables:
136                 if not mono and len(syllables) == 1:
```

```
137                     syllable = syllable.strip('ˈ')
138                 syllables_sentence.append(syllable)
139                 word += syllable
140             words.append(word)
141         return Values(words, syllables_sentence)
142 
143     def __diphthongs(self, syllables):
144         for idx, syllable in enumerate(syllables):
145             if re.search(r'[aeioáéó][ui]', syllable):
146                 syllable = re.sub(r'([aeoáéó])i', r'\1j', syllable)
147                 syllable = re.sub(r'([aeioáéó])u', r'\1w', syllable)
148             if re.search(r'[ui][aeiouáéó]', syllable):
149                 syllable = re.sub(r'i([aeoáéó])', r'j\1', syllable)
150                 syllable = re.sub(r'u([aeoiáéó])', r'w\1', syllable)
151             syllables[idx] = syllable
152         return syllables
153 
154     def transcription_fnt(self, phonology):
155         words = ' '.join(phonology.words)
156         syllables = '-'.join(phonology.syllables)
157         return Values(self.__fsubstitute(words), self.__fsubstitute(syllables))
158 
159     @staticmethod
160     def __fsubstitute(words):
161         words = words.replace('b', 'β').replace('d', 'ð').replace('g', 'ɣ')
162         words = re.sub(r'([mnɲ ˆ])(\-*)β', r'\1\2b', words)
163         words = re.sub(r'([mnɲlʎ ˆ])(\-*)ð', r'\1\2d', words)
164         words = re.sub(r'([mnɲ ˆ])(\-*)ɣ', r'\1\2g', words)
165         words = re.sub(r'θ(\-*)([bdgβðɣmnɲlʎrɾ])', r'ð\1\2', words)
166         words = re.sub(r's(\-*)([bdgβðɣmnɲlʎrɾ])', r'z\1\2', words)
167         words = re.sub(r'f(\-*)([bdgβðɣmnɲʎ])', r'v\1\2', words)
168         allophones = {'nb': 'mb', 'n'b': 'm'b', 'nf': 'ɱf', 'n'f': 'ɱ'f',
169                       'nk': 'ŋk', 'n'k': 'ŋ'k', 'ng': 'ŋg', 'n'g': 'ŋ'g',
170                       'nx': 'ŋx', 'n'x': 'ŋ'x', 'xu': 'χu', 'xo': 'χo',
171                       'xw': 'χw', 'n-b': 'm-b', 'n-'b': 'm-'b', 'n-f': 'ɱ-f',
172                       'n-'f': 'ɱ-'f', 'n-k': 'ŋ-k', 'n-'k': 'ŋ-'k',
173                       'n-g': 'ŋ-g', 'n-'g': 'ŋ-'g', 'n-x': 'ŋ-x',
174                       'n-'x': 'ŋ-'x'}
175         if any(allophone in words for allophone in allophones):
176             for key, value in allophones.items():
177                 words = words.replace(key, value)
178         return words.replace('-', ' ').split()
179 
180     @staticmethod
181     def ipa2sampa(ipa, sampastr):
```

```
182         ipa = Values(ipa.words.copy(), ipa.syllables.copy())
183         transliteration = {'β': 'B', 'ð': 'D', 'ɣ': 'G', 'ʎ': 'L', 'r': 'rr',
184                            'ɾ': 'r', 'ɱ': 'M', 'ŋ': 'N', 'ɲ': 'J', 'ʧ': 'tS',
185                            'ʝ': 'y', 'χ': '4', 'θ': 'T',
186                            ''': sampastr, 'ˌ': '%'}
187         for key, value in transliteration.items():
188             for idx, word in enumerate(ipa.words):
189                 ipa.words[idx] = word.replace(key, value)
190             for idx, syllable in enumerate(ipa.syllables):
191                 ipa.syllables[idx] = syllable.replace(key, value)
192         return ipa
```

Listado `silabeador.py`

```
1  import re
2
3
4  class Syllabification:
5      __vowels = 'aeiouáéíóúäëïöüàèìòùAEIOUÁÉÍÓÚÄËÏÖÜÀÈÌÒÙ'
6      __close = 'iuIU'
7
8      def __init__(self, word, exceptions=1, ipa=False, h = False, epen = False,
9                   tl = False):
10         self.__ipa = ipa
11         self.__h = h
12         if tl:
13             self.__tl = True
14         else:
15             self.__tl = False
16         if epen:
17             word = self.__epenthesis(word)
18         if self.__ipa:
19             self.__vowels += 'jw'
20             self.__close += 'jw'
21         if exceptions > 0:
22             if exceptions > 1:
23                 self.__hiatus = True
24             else:
25                 self.__hiatus = False
26             self.__word = self.__make_exceptions(word)
27             self.__word = self.__latin(self.__word)
28         else:
29             self.__word = word
30         self.syllables = self.__syllabify(self.__word)
31         self.stress = stressed_s(self.syllables)
32
```

```
33
34      def __make_exceptions(self, word):
35          caparros = {'fie':'fi_e', 'sua':'su_a', 'rui': 'ru_i'}
36          if self.__hiatus and any(word.startswith(x) for x in caparros):
37              word = word.replace(word[:3], caparros[word[:3]])
38          from importlib import resources
39          lines = resources.read_text('silabeador', 'exceptions.lst')
40          nouns = lines.splitlines()
41          nouns = [n.strip().split() for n in nouns if n.strip() and not
        n.startswith('#')]
42          for noun in nouns:
43              word = re.sub(rf'{noun[0]}', rf'{noun[1]}', word)
44          return word
45
46      @staticmethod
47      def __epenthesis(word):
48          liquidae = ('sch', 'sc', 'st', 'sp', 'sf','sb', 'sm', 'sn')
49          if word.startswith(liquidae):
50              for onset in liquidae:
51                  if word.startswith(onset):
52                      lonset = len(onset)
53                      if word[len(onset)] in 'aeiouáéíóúrl':
54                          onset = 'es_' + onset[1:]
55                      else:
56                          onset = 'e'+onset+'_'
57                      word = onset + word[lonset:]
58                      break
59          return word
60
61      def __latin(self, verbum):
62          flexiones = ('um', 'em', 'at', 'ant', 'it', 'unt', 'am')
63          # 'int', 'ent', it', 'et', 'at', 'ur' 'int', 'ent'
64          dictionarium = {'a': 'á', 'e': 'é', 'i': 'í', 'o': 'ó', 'u': 'ú'}
65          diphthongi = {'ae': 'æ', 'oe': 'œ'}
66          if verbum.lower().endswith(flexiones) and not any(x in verbum.lower()
67                                                            for x in 'áéíóú'):
68              verbum = verbum.lower()
69              for flexio in flexiones:
70                  if verbum.endswith(flexio):
71                      verbum = re.sub(fr'{flexio}\b', f'_{flexio}', verbum)
72                      break
73              for clavis, pretium in diphthongi.items():
74                  verbum = verbum.replace(clavis, pretium)
75              syllabae = self.__syllabify(verbum)
76              if len(syllabae) == 1:
```

```
77                  pass
78              elif len(syllabae) == 2 or (len(syllabae) > 1 and (
79                  any(dipht in syllabae[-2] for dipht in diphthongi.values()) or
80                  len([x for x in dictionarium.keys() if x in syllabae[-2]]) > 1
        or
81                  not syllabae[-2].endswith(tuple(dictionarium.keys()))):
82                  for clavis, pretium in dictionarium.items():
83                      if clavis in syllabae[-2]:
84                          syllabae[-2] = syllabae[-2].replace(clavis, pretium)
85                          break
86              elif any(dipht in syllabae[-3] for dipht in diphthongi.values())
        or (
87                  len([x for x in dictionarium.keys() if x in syllabae[-3]]) > 1
        or
88                  not syllabae[-3].endswith(tuple(dictionarium.keys()))):
89                  for clavis, pretium in dictionarium.items():
90                      if clavis in syllabae[-3]:
91                          syllabae[-3] = syllabae[-3].replace(clavis, pretium)
92                          break
93              else:
94                  for i in dictionarium:
95                          syllabae[-2] = syllabae[-2].replace(i, dictionarium[i])
96                          break
97              verbum = syllabae
98          return verbum
99
100     def __syllabify(self, letters):
101         slbs = []
102         if type(letters) is str:
103             foreign_lig = {'à': 'a', 'è': 'e', 'ì': 'i', 'ò': 'o', 'ù': 'u',
104                         'ã': 'a', 'ẽ': 'e', 'ĩ': 'i', 'õ': 'o', 'ũ': 'u',
105                         'ﬁ': 'fi', 'ﬂ': 'fl'}
106             word = re.sub(r'\W', '', letters)
107             word = ''.join([letter if letter not in foreign_lig
108                         else foreign_lig[letter] for letter in letters])
109             slbs = self.__split(word)
110             slbs = self.__join(slbs)
111             letters = [x.strip() for x in slbs]
112         return letters
113
114     def __split(self, letters= '', word = []):
115         dipht = re.search(rf'(?:[qg][wuü](?:[eé](?:h*[{self.__close}])'
116                           '{,1}|i(?:h*[aeoáéó]){,1}|í)|'
117
        fr'[{self.__close}](?:h*[aáoóeéi])(?:h*[{self.__close}])'
```

```
118                         '{,1}|'
119                         rf'[aáoóeéiíɪ](?:h*[{self.__close}]))\b', letters)
120         digraph = ('ll', 'ch', 'rr')
121         if dipht:
122             dipht = dipht.group().strip('gq')
123             word = self.__split(letters.removesuffix(dipht), [dipht]+ word)
124         elif letters.endswith(digraph):
125             word = self.__split(letters[:-2], [letters[-2:]] + word)
126         elif letters:
127             word = self.__split(letters[:-1], [letters[-1]]+ word)
128         return word
129 
130     def __join(self, letters):
131         indivisible_onset = ('pl', 'bl', 'fl', 'cl', 'kl', 'gl', 'll',
132                              'pr', 'br', 'fr', 'cr', 'kr', 'gr', 'rr',
133                              'dr', 'tr', 'ch', 'dh', 'rh', 'th',
134                              'βl', 'ɣl',
135                              'βɾ', 'pɾ', 'fɾ', 'kɾ', 'gɾ', 'ɣɾ', 'dɾ', 'ðɾ',
136                              'tɾ', 'bɾ', 'tʃ', 'gw', 'ɣw')
137         if self.__tl:
138             indivisible_onset += ('tl',)
139         indivisible_coda = ('ns', 'bs', 'nz', 'βs', 'bz', 'βz', 'nd', 'rt',
140                             'st', 'ff', 'ls', 'lz', 'zz', 'll', 'nt', 'rs',
    'ɾs',
141                             'ch', 'nk', 'nc', 'lk', 'sh', 'nt', 'sch', 'mp',
    'rd')
142         word = []
143         onset = ''
144         if self.__ipa:
145             symbols = ''
146         else:
147             symbols = 'jw'
148         for letter in letters:
149             if letter == '_':
150                 if len(word) > 0:
151                     word[-1] += onset
152                     onset = ''
153             elif all(x.lower() not in self.__vowels for x in letter):
154                 if onset.endswith('y'):
155                     if onset == 'y' and word[-1][-1] in 'AOEÁÓÉaoeáóé':
156                         word[-1] += onset
157                     else:
158                         word += [onset]
159                     onset = letter
160                 else:
```

```
161                     onset += letter
162                     if len(word) > 0:
163                         media = len(onset) // 2
164                     if self.__h and onset.endswith('h'):
165                         media = 0
166             elif len(onset) <= 1 or len(word) == 0:
167                 word += [onset+letter]
168                 onset = ''
169             elif onset.endswith(indivisible_onset):
170                 if len(word) > 0:
171                     word[-1] += onset[:-2]
172                     word += [onset[-2:] + letter]
173                 else:
174                     word += [onset + letter]
175                 onset = ''
176             elif onset.startswith(indivisible_coda) and (len(onset) > 2):
177                 word[-1] += onset[:2]
178                 word += [onset[2:] + letter]
179                 onset = ''
180             elif (onset[-1] in 'dðfkt' and
181                   onset[-2] in 'bβcθkdðfgɣkmɱɲñpqstvwxχz' + symbols
182                   or onset[-1] in 'gɣ' and onset[-2] in 'cθtk' + symbols
183                   or onset[-1] in 'lmɱ' and onset[-2] in 'mɱl'
184                   or onset[-1] in 'cθ' and onset[-2] in 'kc'):
185                 word[-1] += onset[:-1]
186                 word += [onset[-1] + letter]
187                 onset = ''
188             else:
189                 word[-1] += onset[:media]
190                 word += [onset[media:] + letter]
191                 onset = ''
192         if onset:
193             if onset.endswith('y') and len(onset) == 1:
194                 word[-1] += onset
195             elif onset.endswith('y'):
196                 word[-1] += onset[:-2]
197                 word += [onset[-2:]]
198             else:
199                 word[-1] += onset
200         return word
201
202
203 def stressed_s(slbs):
204     if len(slbs) == 1:
205         stress = -1
```

```
206     elif any(k in 'áéíóúÁÉÍÓÚ' for k in ''.join(slbs)):
207         for x in  range(-len(slbs), 0, 1):
208             if any(k in 'áéíóúÁÉÍÓÚ' for k in slbs[x]):
209                 stress = x
210                 break
211     elif slbs[-1][-1] in 'aeiouAEIOUy':
212         stress = -2
213     elif slbs[-1][-1] in 'nsNS' and slbs[-1][-2] in 'aeiouAEIOU':
214         stress = -2
215     else:
216         stress = -1
217     return stress
218
219
220 def syllabify(word, exceptions=1, ipa= False, h=False, epen=False, tl=False):
221     return Syllabification(word, exceptions, ipa, h, epen, tl).syllables
222
223
224 def tonica(word, exceptions=1, ipa= False, h=False, epen=False, tl=False):
225     return Syllabification(word, exceptions, ipa, h, epen, tl).stress
```

Lista de excepciones

```
###### -uir verbs
# Infinitive forms
([^fqgck]|\b)ui([rr])\b        \1u_i\2
([^fqgck]|\b)uid([oa]s{,1})\b \1u_id\2
# Presente
([^fqg]|\b)ui(mos|s)\b \1u_i\2
# Indefinido
([^fqg]|\b)u(i|iste|isteis)\b \1u_\2
# Futuro !
([^qg]|\b)ui([rr])(é|ás|á|emos|éis|an)\b \1u_i\2\3
# Condicional !
([^qg]|\b)ui([rr])í(a|as|amos|ais|an)\b u_\1i\2í\3
##### Verbos en -uar
# Infinitives
([^gk])ua([rr])\b \1u_a\2
([^gkc])uad((?:[oa]s*)*)\b \1u_ad\2
([^gkc]{2,})uand((?:[oa]s*)*)\b \1u_and\2
([^gkc]{2,})uant((?:[ea]s*)*)\b \1u_ant\2
# Presente
```

```
([^gkc])uamos\b u_amos
([^gkc])uáis\b u_áis
# Imperfecto
([^gkc])uab(a|as|ais|an)\b \1u_ab\2
([^gkc])uábamos\b \1u_ábamos
# Indefinido
([^gkc])u(é|aste|ó|asteis|aron|aron)\b \1u_\2
# Futuro !
([^gkc])ua([rr])(é|ás|á|emos|éis|an)\b \1u_a\2\3
# Condicional !
([^gkc])ua([rr])í(a|as|amos|ais|an)\b \1u_a\2í\3
# Subjuntivo, presente
([^gkc])uemos\b \1u_emos
# Imperfecto subjuntivo
([^gkc])ua([rr])(a|as|ais|an)\b \1u_a\2í\3
([^gkc])uá([rr])amos\b \1u_á\2amos
# Futuro subjuntivo
([^gkc])ua([rr])(e|es|a|eis|en)\b \1u_a\2\3
([^gkc])uá([rr])emos\b \1u_á\2emos
# -iar
# Infinitives
(\b.{1,2}|.*f)ia([rr])\b \1i_a\2
(\b.{1,2}|.*f)iad((?:[oa]s*)*)\b \1i_ad\2
(\b.{1,2}|.*f)iand((?:[oa]s*)*)\b \1i_and\2
(\b.{1,2}|.*f)iant((?:[ea]s*)*)\b \1i_ant\2
# Presente
(\b.{1,2}|.*f)iamos\b \1i_amos
(\b.{1,2}|.*f)iáis\b \1i_áis
# Imperfecto
(\b.{1,2}|.*f)iab(a|as|ais|an)\b \1i_ab\2
(\b.{1,2}|.*f)iábamos\b \1i_ábamos
# Indefinido
(\b.{1,2}|.*f)i(é|aste|ó|asteis|aron|aron)\b \1i_\2
# Futur !
(\b.{1,2}|.*f)ia([rr])(é|ás|á|emos|éis|an)\b \1i_a\2\3
# Condicional !
(\b.{1,2}|.*f)ia([rr])í(a|as|amos|ais|an)\b \1i_a\2í\3
# Subjuntivo, presente
```

```
(\b.{1,2}|.*f)iemos\b i\1_emos
# Imperfecto subjuntivo
(\b.{1,2}|.*f)ia([ɾr])(a|as|ais|an)\b \1i_a\2í\3
(\b.{1,2}|.*f)iá([ɾr])amos\b \1i_á\2amos
(en|des)[bv]ia \1bi_a
([kc]on|desa)fia \1fi_a
# Futuro subjuntivo
(\b.{1,2}|.*f)ia([rɾ])(e|es|a|eis|en)\b \1i_a\2\3
(\b.{1,2}|.*f)iá([ɾr])emos\b \1i_á\2emos
a(r{1,2})ie([ɾr]) a\1i_e\2
a([kc])uos a_\1uos
a([kc])u([ií]) a\1u_\2
([iu])ángulo \1_ángulo
([iu])anual \1_anu_al
a(r{1,2})ie([ɾr]) a\1i_e\2
bienio bi_enio
biombo bi_ombo
bo(qu|k)ian(ch|ʧ) bo\1ian\2
\bb([ɾr])io b\1i_o
([kc])lient \1li_ent
ian([zθ])a i_an\1a
(cr|kɾ)uel \1u_el
(cr|kɾ)i([ao]) \1i_\2
diablo di_ablo
#diálog
#\bdia([rɾ])i di_a\1i
#\bdia([kc]) di_a\1
([^g])üi \1u_í
\bdi([aáu].*(?:\w*[aeiouáéíóúäëïöü]\w*){3,}) di_\1
die([cθ])io(ch|ʧ)o die\1i_o\2o
fe(r{1,2})o([bv])ial fe\1o\2i_al
\bfia fi_a
#fianza
fo([rɾ])tuito fo\1tu_ito
#gorrión
\bg(u{,1})i([óo])n g\1i_\2n
hiato hi_ato
#(h{,1})uid \1u_id
```

```
#huir hu_ir
io(ch|ʧ)o\b i_o\1o
([jx])esuit \1esu_it
\bmanio mani_o
miasma mi_asma
#piano
p([rɾ])ior p\1i_or
\bri([ée])ndo ri_\1ndo
santua([rɾ])io santu_a\1io
\b([sS])i([oó])n \1i_on
tia([rɾ])a ti_a\1a
envi([aeioóé]) envi_\1
embi([aeioóé]) embi_\1
\bt([rɾ])i([aáe]) t\1i_\2
t([ɾ])ru(h{,1}[aá])n t\1u_\2n
uos([oa]s*) u_os\1
(^[áéíóú]*[aeiou])([uea]m|[aie]t|[aeui]nt|ur)\b \1_\2
([ck]ur*)i([ck])(.lum) \1í\2ulum
([Dd])o(m).num \1ó\2inum
abi([tm])u(rs) ábi\1u\2
abi([tm])u(rs) ábi\1u\2
ebi([tm])u(rs) ébi\1u\2
bi([tm])u(rs) ébi\1u\2
#\b([Rr])e([hui][^n]) \1e_\2
#\b([Pp][rɾ])e([iuh]) \1e_\2
\b([Tt][rɾ])a(n*)sh \1a\2s_h
\b([dD])esh \1es_h
\b([aie])nh \1n_h
\b([Pp])anh \1an_h
\b([Ss])inh \1in_h
f([rl])ue f\1ue
```

B.3 Escansión versal

Listado `libEscansion.py`

```
1 import re
2 import stanza
```

```
from math import sqrt
from fonemas import Transcription
from dataclasses import dataclass
version = '1.0.0pre5'  # 5/07/2023

processor_dict = {'tokenize': 'ancora', 'mwt': 'ancora', 'pos': 'ancora',
                  'ner': 'ancora', 'depparse': 'ancora'}
conf = {'lang':  'es', 'processors': processor_dict, 'download_method': 'None'}
nlp = stanza.Pipeline(**conf, logging_level='ERROR')
usuals = ('xueθ', 'suab', 'kruel', 'fiel', 'ruina', 'diabl', 'dios', 'kae',
          'rios', 'birtuos', 'krio', 'ʰuid', 'porfiad')

values = {'A': 7, 'a': 6, 'ă': 5,
          'O': 4, 'o': 3, 'ŏ': 2,
          'E': 1, 'e': 0, 'ĕ': -1,
          'I': -2, 'i': -3, 'j': -4,
          'U': -5, 'u': -6, 'w': -7,
          'y': -2, 'X': -999, '': -1000}
trapez = {'i': (-1, 1), 'e': (-1, 0), 'a': (0, -1), 'u': (1, 1),
          'j': (-1, 1), 'ĕ': (-1, 0), 'ă': (0, -1), 'w': (1, 1),
          'y': (-1, 1), 'o': (1, 0), 'ŏ': (1, 0)}
non_syllabic = {'a': 'ă', 'e': 'ĕ', 'i': 'j', 'o': 'ŏ', 'u': 'w',
                'A': 'ă', 'E': 'ĕ', 'I': 'j', 'O': 'ŏ', 'U': 'w',
                'j': 'j', 'w': 'w', 'ă': 'ă', 'ĕ': 'ĕ', 'ŏ': 'ŏ',
                'y': 'ɟ'}
indeed_syllabic = {'ă': 'a', 'ĕ': 'e', 'j': 'i', 'ŏ': 'o', 'w': 'u'}
glides, close, med = 'wjăĕŏ', 'IUiuy', 'AEOaeo'
vowels, vocalic = close + med, glides + close + med
allvoc = vocalic + 'ʰ'


@dataclass
class Features:
    text: str
    pos: str
    phon: list
    feats: dict
    dep: list
    ton: bool


@dataclass
class VerseFeatures:
    slbs: list
    amb: int
```

```
48      count: int
49      asson: str
50      cons: str
51
52
53  class PlayLine:
54
55      def __init__(self, line, adso=False):
56          self.line = line
57          self.adso = adso
58          if len(line) > 0:
59              self.__fixed_verse = self.__fix_line(self.__preprocess(self.line))
60              self.words = self.__find_prosodic_stress(self.__fixed_verse)
61          else:
62              self.words = []
63
64      def __preprocess(self, transcription):
65          transcription = re.sub(r'[Pp]ara\,', 'Ppara,', transcription)
66          symbols = {'(': '.', ')': '.', '-': '.', '…': '.', '‘': ' ', '’': ' ',
67                     ';': '.', ':': '.', '?': '.', '!': '.', '"': ' ', '_': ' ',
68                     'õ': 'o', 'æ': 'ae', 'à': 'a', 'è': 'e', 'ì': 'i', 'ò': 'o',
69                     'ù': 'u', '«': ' ', '»': ' ', '-': '.', '“': ' ', '”': ' ',
70                     "'": ' ', '.': '. '}
71          if transcription != transcription:
72              transcription = 'mama mama mama mama'
73          else:
74              for i in symbols:
75                  transcription = transcription.replace(i, f'{symbols[i]} ')
76              transcription = re.sub(r'\s*\.+(\w)', r',\1', transcription)
77              transcription = re.sub(r'\s*\,+(\w)', r',\1', transcription)
78              transcription = re.sub(r'\[|\]|¿|¡|^\s*[\.\,]', '', transcription)
79              transcription = re.sub(r'\s*\.[\.\s]+', ', ', transcription)
80              transcription = transcription.strip()
81          verse = nlp(f'{transcription}')
82          transcription = []
83          for sentence in verse.sentences:
84              usadas = []
85              for word in sentence.words:
86                  if word.parent.id not in usadas:
87                      usadas.append(word.parent.id)
88                      word.text = word.parent.text
89                      transcription.append(word)
90                  word.text = word.text.strip('.')
91          return transcription
92
```

```
93      def __fix_line(self, line):
94          for word in line:
95              words = [word for word in line if word.pos != 'X']
96          while words[-1].pos == 'PUNCT':
97              if any(x.isalpha() for x in words[-1].text):
98                  words[-1].pos = 'ADJ'
99              else:
100                 words = words[:-1]
101
102         if len(words) > 1:
103             if words[-1].parent.text == words[-2].parent.text and \
104                     words[-1].text != words[-2].text:
105                 words[-2].text = words[-2].parent.text
106                 words.pop()
107         return self.__set_features(words)
108
109     def __set_features(self, words):
110         feats = []
111         for word in words:
112             if word.text.isalpha():
113                 if any(x in word.text for x in 'äëïöü'):
114                     exc = 2
115                 else:
116                     exc = 1
117                 feats.append(Features(word.text,
118                                       word.upos,
119                                       Transcription(word.text,
120                                                     mono=True,
121                                                     epenthesis=True,
122                                                     aspiration=True,
123                                                     stress='',
124                                                     exceptions=exc
125                                                     ).phonology.syllables,
126                                       self.__parse_feats(word.feats),
127                                       word.deprel,
128                                       False))
129         return feats
130
131     @staticmethod
132     def mark_prosodic_stress(word):
133         for idx, syllable in enumerate(word.phon):
134             if syllable.startswith("'") or syllable.startswith("ˌ"):
135                 if word.ton is True:
136                     syllable = syllable.translate(
137                         str.maketrans("aeiou", "AEIOU"))
```

```
138             word.phon[idx] = syllable.replace("'", '').replace('ˌ', '').strip()
139         return word
140 
141     def __fix_word(self, word):
142         if word.pos == 'PUNCT' and any(x.isalpha() for x in word.text):
143             word.pos = 'ADJ'
144         return(word)
145 
146     def __find_prosodic_stress(self, words):
147         stressed_pos = ['ADV', 'NOUN', 'PROPN',
148                         'DET.Dem', 'DET.Int', 'DET.Ind', 'PRON.Com',
149                         'PRON.Nom', 'PART', 'INTJ', 'ADJ', 'VERB', 'AUX']
150         conjat = ['y', 'e', 'ni', 'o', 'u',
151                   'que', 'quien', 'quienes',
152                   'pero', 'sino', 'mas', 'aunque', 'aun'
153                   'pues', 'porque', 'como', 'conque', 'si',
154                   'cual', 'cuales', 'do',
155                   'cuanto', 'cuanta', 'cuantos', 'cuantas',
156                   'donde', 'tan',
157                   'cuando', 'como',
158                   'mi', 'tu', 'su', 'mis', 'tus', 'sus']
159         tonicos = ['agora', 'yo', 'vos', 'es', 'soy', 'voy',
160                    'sois', 'vais', 'ti', 'nosotros', 'vosotros', 'ellos',
161                    'nosotras', 'vosotras', 'ellas', 'ella',
162                    'todo', 'toda', 'todos', 'todas', 'cada',
163                    'aqueste', 'aquesta', 'aquestos', 'aquestas',
164                    'aquese', 'aquesa', 'aquesas', 'aquesos',
165                    'este', 'esta', 'esto', 'estos', 'estas',
166                    'ese', 'esos', 'esa', 'esas', 'eso',
167                    'aquel', 'aquella', 'aquellos', 'aquellas',
168                    'tuyo', 'tuyos', 'tuya', 'tuyas',
169                    'suyo', 'suya', 'suyos', 'suyas']
170         numbers = ['uno', 'una', 'dos', 'tres', 'cuatro', 'cinco', 'seis',
171                    'siete', 'ocho', 'nueve']
172         courtesy = ['don', 'doña', 'sor', 'fray',
173                     'santo', 'san', 'santa', 'gran']
174 
175         PoSerrors = ['mas', 'ei']
176         ant, units = '', False
177         for idx, word in enumerate(words[::-1]):
178             word = self.__fix_word(word)
179             ton = False
180             txt = word.text.lower()
181             pos = word.pos
182             fts = word.feats
```

```
183             if idx == 0:
184                 ton = True
185             elif txt in conjat:
186                 pass
187             elif txt.lower() in ['oh', 'ay']:  # ADSO 100
188                 ant = ''
189                 if not self.adso:
190                     ton = True
191             elif any(x in txt for x in 'áéíóú'):
192                 ton = True
193             else:
194                 if pos == 'NUM':
195                     if ant != 'NUM':
196                         ton = True
197                         if txt in numbers:
198                             units = True
199                         else:
200                             units = False
201                     elif txt in numbers:
202                         if units is True:
203                             ton = True
204                         units = True
205                     else:
206                         units = False
207                 elif txt == 'y' and ant == 'NUM':
208                     pos = 'NUM'
209                 elif pos == 'DET':
210                     nonstressed = ['me', 'te', ' le', 'nos', 'les',
211                                    'lo', 'la', 'los', 'las']
212                     if any(txt.endswith(x) for x in nonstressed):
213                         if txt not in nonstressed:
214                             ton = True
215                     elif fts['PronType'] == 'Poss' or fts['Poss'] == 'Yes':
216                         if txt in ['mi', 'tu', 'su', 'mis', 'tus', 'sus']:
217                             pass
218                         elif any([txt.startswith(x)
219                                   for x in ('nuestr', 'vuestr')]):
220                             if ant not in ['PROPN', 'NOUN', 'ADJ']:
221                                 ton = True
222                         else:
223                             ton = True
224                     elif fts['PronType'] in ['Dem', 'Ind', 'Tot']:
225                         ton = True
226                     elif 'Definite' in fts:
227                         if fts['Definite'] != 'Def':
```

```
                            ton = True
                elif pos in ['PROPN', 'NOUN']:
                    if not (txt.lower() in courtesy and ant == 'PROPN') and \
                            txt not in PoSerrors:
                        ton = True
                elif pos == 'ADV':
                    if txt in ('tan', 'medio', 'aun') or (txt in 'ya' and
                                                          ant == 'SCONJ'):
                        pass
                    else:
                        ton = True
                elif pos == 'PRON':
                    if txt in tonicos:
                        ton = True
                    elif any(x in txt for x in ['ar', 'er', 'ir']):
                        ton = True
                    elif txt.endswith('igo'):
                        ton = True
                    elif fts['PronType'] == 'Int,Rel':
                        pass
                    elif any(txt.startswith(x) for x in ('nuestr', 'vuestr')):
                        ton = True
                    elif 'Nom' not in fts['Case'] and \
                            fts['PronType'] == 'Prs' and \
                            txt not in ['ti', 'mí'] and fts['Poss'] != 'Yes':
                        pass
                    else:
                        ton = True
                elif txt in tonicos or pos in stressed_pos:
                    ton = True
            word.ton = ton
            words[idx] = self.mark_prosodic_stress(word)
            if txt not in ('y'):
                ant = pos
            else:
                ant = ''
        words.reverse()
        for idx, word in enumerate(words):
            if idx < len(words) - 1 and word.text == 'y':
                if all(x.lower() in 'aeiouwj'
                       for x in (words[idx - 1].phon[-1][-1],
                                 words[idx + 1].phon[0][0])):
                    words[idx].phon[0] = 'y'
        return [word.phon for word in words if word.pos != 'PUNCT']

```

```
273     @staticmethod
274     def __parse_feats(feats):
275         dd = {}
276         if feats:
277             elements = feats.split("|")
278             for element in elements:
279                 feature = element.split("=")
280                 dd[f'{feature[0]}'] = feature[1]
281         else:
282             dd['None'] = 'None'
283         if 'Case' not in dd:
284             dd['Case'] = ''
285         if 'Poss' not in dd:
286             dd['Poss'] = 'No'
287         return dd
288 
289 
290 class VerseMetre(PlayLine):
291     most_common = [6, 7, 8, 11, 10, 9, 14, 12, 5, 15, 4]
292 
293     def __init__(self, line, expected_syl=False, adso=False):
294         PlayLine.__init__(self, line, adso)
295         if len(self.words) > 0:
296             natural = len(self.__flatten(self.words)) + self.__find_rhyme(
297                 self.words[-1])['count']
298             self.synaloephas = self.__find_synaloephas(self.words)
299             normalsyn = [a for a in self.synaloephas if a[1] > -15]
300             self.natural = natural - len(normalsyn)
301             self.estimate, self.expected_syl = self.__adjust_expected(
302                 self.words, self.synaloephas, expected_syl)
303             self.__verse = self.__adjust_metre(self.words, self.expected_syl)
304             self.syllables = self.__verse.slbs
305             self.ambiguity = self.__verse.amb
306             self.asson = self.__verse.asson
307             self.rhyme = self.__verse.cons
308             self.count = self.__verse.count
309             self.nuclei = self.find_nuclei(self.syllables)
310             self.rhythm = self.find_rhyhtm(self.nuclei)
311         else:
312             self.synaloephas = self.syllables = self.expected_syl = []
313             self.estimate = self.count = 0
314             self.ambiguity = self.asson = self.rhyme = False
315             self.nuclei = self.rhythm = ''
316             self.natural = 0
317 
```

```
318     def __find_synaloephas(self, words, h=False):
319         synaloephas, ant = [], ['X']
320         preference = offset = 0
321         for idx, word in enumerate(words):
322             coda = word[0].replace('ʰ', '')
323             onset = ant[-1]
324             s = False
325             if idx != 0 and all(x.lower() in allvoc
326                                 for x in (coda[0], onset[-1])):
327                 position = [idx - 1, len(words[idx - 1]) - 1]
328                 if idx == 1 and words[0] in (['i'], ['o'])\
329                         and coda[0] in 'AEIOU':
330                     preference -= 8
331                 if word in [[x] for x in 'ei'] and len(words) > idx + 2:
332                     if not words[idx+1][0][0] in allvoc:
333                         s = True
334                 else:
335                     val = values[onset[-1]]
336                     if len(onset) > 1 and onset[-2] in values:
337                         previous_val = values[onset[-2]]
338                     else:
339                         previous_val = values[onset[-1]]
340                     if len(coda) > 1 and coda[1] in values:
341                         next_val = values[coda[1]]
342                     else:
343                         next_val = values[coda[0]]
344                     if (previous_val <= val and val <= values[coda[0]]) or (
345                             previous_val >= val and
346                             val >= values[coda[0]] and
347                             next_val <= values[coda[0]]) or (
348                             previous_val <= val and
349                             val > values[coda[0]] and
350                             values[coda[0]] >= next_val):
351                         s = True
352             if s:
353                 if (coda[0] + onset[-1]).islower() and coda[0] == onset[-1]:
354                     if any(x in (['o'], ['y']) for x in (word, ant)):
355                         preference -= 1
356                     elif len(coda) > 1 and coda[1] in 'jwăĕŏ':
357                         preference -= 2
358                 synaloephas.append((position,
359                                     self.__synaloepha_pref(onset, coda,
360                                                            preference),
361                                     onset, coda))
362             ant = word
```

```
363             offset += len(word)
364         for idx, word in enumerate(words):
365             if len(word) > 1:
366                 preference = -12
367                 for idy, syllable in enumerate(word):
368                     if idy > 0:
369                         position = [idx, idy - 1]
370                         coda = syllable
371                         onset = word[idy - 1]
372                         if all(x in allvoc
373                                for x in [onset[-1], coda[0]]) and not (
374                                    onset[-1].isupper()
375                                    and idx + 1 == len(words)
376                                    and idy + 1 == len(word)):
377                             synaloephas.append((position,
378                                                 self.__synaloepha_pref(
379                                                     onset, coda)
380                                                 + preference - 2, onset, coda))
381         synaloephas.sort(key=lambda x: x[1], reverse=True)
382         return synaloephas
383 
384     def __adjust_expected(self, syllables, synaloephas, expected):
385         snlf = len([a for a in synaloephas if a[1] > -15])
386         slbs = len(self.__flatten(syllables)) + (
387             self.__find_rhyme(syllables[-1])['count'])
388         slbs = slbs - snlf
389         if expected:
390             if expected[0] == 7:
391                 exp = [7, 11]
392             elif expected[0] == 11:
393                 exp = [11, 7]
394             else:
395                 exp = expected[:1]
396         elif slbs in (6, 7, 8, 11):
397             expected = [6, 7, 8, 11, 10, 9, 14, 12, 5, 15, 4]
398             exp = [slbs]
399         else:
400             if slbs > 10:
401                 exp = [11]
402             elif slbs > 8:
403                 exp = [8]
404             else:
405                 exp = [slbs]
406             expected = [8, 11, 7, 6, 10, 9, 14, 12, 5, 15, 4]
407         return (slbs, exp + [a for a in expected if a not in exp])
```

```
408
409     def __adjust_metre(self, syllables, expected):
410         potential_synaloephas = self.__find_synaloephas(syllables)
411         potential_hiatuses = self.__find_hiatuses(syllables)
412         rhyme = self.__find_rhyme(syllables[-1])
413         len_rhyme = len(self.__flatten(syllables)) + rhyme['count']
414         offset = expected[0] - len_rhyme
415         sllbls = syllables[0:]
416         if offset == 0:
417             ambiguous = 0
418         elif len_rhyme - len(potential_synaloephas) == expected[0]:
419             ambiguous = 0
420             syllables = self.__synaloephas(syllables, -offset)
421         elif len_rhyme - len(potential_synaloephas) > expected[0]:
422             ambiguous = 2
423             syllables = self.__synaloephas(syllables, -offset - 1)
424             hemistich = self.__test_hemistich(syllables)
425             if self.__test_hemistich(syllables) > 0:
426                 syllables = self.__resolve_long(syllables, hemistich)
427                 len_rhyme -= 1
428         else:
429             ambiguous = 1
430             if offset < 0 and len(potential_synaloephas) >= -offset:
431                 syllables = self.__synaloephas(syllables, -offset)
432             elif (len_rhyme < expected[0] > 4 and
433                   len(potential_hiatuses) + len_rhyme >= expected[0]):
434                 expected = expected[:1] + expected
435                 syllables = self.__apply_hiatus(syllables,
436                                                 potential_hiatuses,
437                                                 offset)
438         rhyme = self.__find_rhyme(syllables[-1])
439         len_rhyme = len(self.__flatten(syllables)) + rhyme['count']
440         if len_rhyme > expected[0]:
441             verse = self.__adjust_metre(sllbls, expected[1:])
442         elif len_rhyme < expected[0]:
443             verse = self.__adjust_metre(sllbls, expected[1:])
444         else:
445             rep = {'y': 'i', 'Y': 'I', 'ppA': 'pA'}
446             for i, s in enumerate(syllables):
447                 if s[0] in rep.keys():
448                     syllables[i][0] = rep[s[0]]
449             verse = VerseFeatures(syllables,
450                                   ambiguous,
451                                   len_rhyme,
452                                   rhyme['assonance'],
```

```
453                                 rhyme['consonance'])
454         return verse
455
456     @staticmethod
457     def find_nuclei(syllables):
458         return ''.join([nucleus for syl in [phon for clstr in syllables
459                                             for phon in clstr]
460                         for nucleus in syl if nucleus in vowels])
461
462     @staticmethod
463     def find_rhyhtm(syllables):
464         metre = []
465         for syllable in syllables:
466             if any(phoneme.isupper() for phoneme in syllable):
467                 metre.append('+')
468             else:
469                 metre.append('-')
470         return ''.join(metre)
471
472     def __synaloepha_pref(self, onset, coda, preference=0):
473         distance = self.__vowel_distance(onset, coda)
474         onset = ''.join([x for x in onset if x in allvoc])
475         coda = ''.join([x for x in coda if x in allvoc])
476         preference -= 2*(len(onset) + len(coda) - 2 + distance)
477         if coda.startswith('ʰ'):
478             preference -= 2
479             coda = coda.strip('ʰ')
480         if coda.islower() and onset.islower():
481             preference += 4
482         elif not (coda + onset).islower():
483             preference -= 2
484             if any(y in x for y in 'UI' for x in (coda, onset)):
485                 preference -= 1
486             if (coda + onset).isupper():
487                 preference -= 8
488         if coda[0] in 'yo':
489             preference -= 1
490         if onset[-1] in 'yo':
491             preference += 1
492         return preference
493
494     @staticmethod
495     def __find_hiatuses(words):
496         diphthongs = []
497         for idx, word in enumerate(words):
```

```
498             ton = 0
499             for idy, syllable in enumerate(reversed(word)):
500                 if any(x.isupper() for x in syllable):
501                     ton = len(word) - idy - 1
502                     continue
503             for idy, syllable in enumerate(word):
504                 if rg := re.search(r'([wj][AEOIaeoi])|([AEOIaeoi][wj])',
505                                    syllable):
506                     if idy < ton or (rg.group(1) and
507                                      not rg.group(1).islower()):
508                         diphthongs.append((idx, idy))
509         return diphthongs
510
511     def __apply_hiatus(self, words, hiatuses, difference):
512         sem2voc = {'j': 'i', 'w': 'u', 'J': 'I', 'W': 'U'}
513         preference = self.__hiatus_preference(words, hiatuses)
514         for idx in preference[:difference]:
515             word = words[idx[0]]
516             syllable = word[idx[1]]
517             diphthong = re.search(r'([jw]*)([aeiouAEIOUjw])([jw]*)', syllable)
518             if len(diphthong.group()) > 1:
519                 onset = syllable.split(diphthong.group())[0]
520                 coda = syllable.split(diphthong.group())[1]
521                 semiconsonant = diphthong.group(1)
522                 nucleus = diphthong.group(2)
523                 semivowel = diphthong.group(3)
524                 if semivowel and nucleus:
525                     if nucleus.isupper():
526                         nucleus, semivowel = nucleus.lower(), semivowel.upper()
527                     first = onset + nucleus
528                     second = semivowel.replace(semivowel[-1],
529                                                sem2voc[semivowel[-1]]) + coda
530                 else:
531                     first = onset + semiconsonant.replace(
532                         semiconsonant[-1], sem2voc[semiconsonant[-1]])
533                     second = nucleus + coda
534                 hiatus = f'{first} {second}'
535                 word = [(index, element) if index != idx[1] else
536                         (idx[1], hiatus) for index, element in enumerate(word)]
537                 words[idx[0]] = re.split(' +', ' '.join([element[1]
538                                                          for element in word]))
539         return words
540
541     def __test_hemistich(self, word_list):
542         offset = i = correction = 0
```

```
        for idx, word in enumerate(word_list):
            if len(self.__flatten(word_list)) > 9:
                for idy, syllable in enumerate(word):
                    if any(x in syllable for x in 'AEIOU') \
                            and idy + offset in (3, 5):
                        if len(word) - idy > 2 and word[-2:] != ['mEn', 'te']:
                            correction = idx
                        break
                    i += 1
                offset += len(word)
            else:
                break
        return correction

    def __synaloephas(self, syllables, offset, count=0):
        if offset > 0:
            potential_synaloephas = self.__find_synaloephas(syllables)
            count += 1
            syllables = self.__adjust_syllables(syllables,
                                                potential_synaloephas[:1])
            syllables = self.__synaloephas(syllables, offset-1, count)
        return syllables

    @staticmethod
    def __resolve_long(words, position):
        words[position] = words[position][:-2] + [words[position][-1]]
        return words

    @staticmethod
    def __find_rhyme(word):
        offset, tonic = {-1: 1, -2: 0, -3: -1}, -1
        for idx, syllable in enumerate(word[::-1]):
            if not syllable.islower():
                tonic = -idx - 1
                for jdx, phoneme in enumerate(syllable):
                    if phoneme.isupper():
                        coda = word[(idx+1)*-1:]
                        coda[(idx+1)*-1] = syllable[jdx:]
                        break
                break
            else:
                coda = word[-2:]
        if tonic < -3:
            tonic = -2
        if len(coda) > 2:
```

```
588             assonance = ''.join([syl.lower() for syl in [coda[i]
589                                                          for i in (0, -1)]])
590         else:
591             assonance = ''.join([syl.lower() for syl in coda])
592         assonance = ''.join([phoneme for phoneme in assonance
593                              if phoneme in vowels])
594         consonance = ''.join([phoneme.lower() for phoneme in coda])
595         return {'stress': tonic, 'count': offset[tonic],
596                 'assonance': assonance, 'consonance': consonance}
597
598     def __adjust_syllables(self, words, synaloephas):
599         synaloephas_list = [syllable[0] for syllable in synaloephas]
600         if synaloephas_list:
601             i_word1 = synaloephas_list[0][0]
602             i_syllable1 = synaloephas_list[0][1]
603             word = words[synaloephas_list[0][0]]
604             l_word = len(word)
605             joint = [synaloephas[0][-2], synaloephas[0][-1]]
606             if i_syllable1 == l_word - 1:
607                 i_syllable2 = 0
608                 i_word2 = i_word1 + 1
609                 if len(words[i_word1]) > 1:
610                     onset = words[i_word1][:-1]
611                 else:
612                     onset = []
613                 coda = words[i_word2][1:]
614                 diphthong = [self.__apply_synaloephas(joint)]
615                 word = onset + diphthong
616                 if len(words[i_word2]) > 1:
617                     word = word + coda
618                 words = (words[:i_word1] + [word] + words[i_word2+1:])
619                 synaloephas_list = self.__adjust_position(
620                     synaloephas_list[1:], synaloephas_list[0], len(onset))
621             else:
622                 i_syllable2 = i_syllable1 + 1
623                 onset = words[i_word1][:i_syllable1]
624                 coda = words[i_word1][i_syllable2+1:]
625                 diphthong = self.__apply_synaloephas(joint)
626                 word = onset + [diphthong] + coda
627                 words[i_word1] = word
628                 synaloephas_list = self.__adjust_position(
629                     synaloephas_list[1:], synaloephas_list[0], -1)
630             words = self.__adjust_syllables(words, synaloephas_list)
631         return words
632
```

```
633     @staticmethod
634     def __perception(chain, may=False):
635         max = -999
636         for x in chain:
637             if x in non_syllabic and values[x] > max:
638                 max = values[x]
639         chain = list(chain)
640         for i, x in enumerate(chain):
641             if x in non_syllabic:
642                 if values[x] < max:
643                     chain[i] = non_syllabic[x]
644                 else:
645                     chain[i] = indeed_syllabic[x]
646         chain = ''.join(chain)
647         if may:
648             for j in 'aeo':
649                 chain = chain.replace(j, j.upper())
650         return chain
651 
652     def __apply_synaloephas(self, diphthong):
653         diphthong[1] = diphthong[1].replace('ʰ', '')
654         onset, coda = diphthong[0], diphthong[1]
655         onsetb = ''.join([non_syllabic[x] if x in non_syllabic else x
656                           for x in onset])
657         codab = ''.join([non_syllabic[x] if x in non_syllabic else x
658                          for x in coda])
659         if non_syllabic[onset[-1]] == non_syllabic[coda[0]]:
660             if coda[0] in glides or onset[-1].isupper():
661                 diphthong = onset + coda[1:]
662             else:
663                 diphthong = onset[:-1] + coda
664         elif len([x for x in onset + coda if x in non_syllabic]) > 2:
665             tonic = not (onset + coda).islower()
666             diphthong = self.__perception(onsetb + codab.replace('ʝ', 'j'),
667                                           tonic)
668         elif onset == 'y' or (onset == 'i' and coda.startswith('u')):
669             diphthong = 'ʝ' + coda
670         elif not onset.islower() and (coda.islower() or
671                                       values[onset[-1]] > values[coda[0]]):
672             diphthong = onset + codab.replace('ʝ', 'j')
673         else:
674             diphthong = onsetb + coda
675         return diphthong
676 
677     @staticmethod
```

```
678     def __hiatus_preference(words, hiatuses):
679         preference = []
680         for idx in hiatuses[::-1]:
681             if ''.join(words[idx[0]]).lower().startswith(usuals):
682                 preference = [idx] + preference
683             else:
684                 preference += [idx]
685         for idx, second in enumerate(reversed(preference)):
686             for first in preference[:-idx-1]:
687                 if first[0] == second[0] and first[1] < second[1]:
688                     preference[-idx-1] = (second[0], second[1]+1)
689         return preference
690 
691     @staticmethod
692     def __flatten(thelist):
693         return [item for sublist in thelist for item in sublist]
694 
695     @staticmethod
696     def __vowel_distance(onset, coda):
697         onset = onset.lower()
698         coda = coda.strip('ʰ').lower()
699         distance = sqrt(pow(abs(trapez[onset[-1]][0] - trapez[coda[0]][0]), 2)
700                         + pow(abs(trapez[onset[-1]][1] -
701                                   trapez[coda[0]][1]), 2))
702         return distance
703 
704     @staticmethod
705     def __adjust_position(word_list, position, offset):
706         for idx, word in enumerate(word_list):
707             if offset < 0:
708                 if word[0] == position[0]:
709                     if word[1] >= position[1]:
710                         word_list[idx][1] -= 1
711             elif word[0] > position[0]:
712                 if word[0] == position[0] + 1:
713                     word_list[idx][1] += offset
714                 word_list[idx][0] -= 1
715         return word_list
```

B.4 Agrupación estrófica

Listado `stanzas.py`

```
1  #!/usr/bin/python
2 
```

```
3  import pandas as pd
4  from tabulate import tabulate
5
6
7  entrada = 'corpus.csv'
8  df = pd.read_csv(entrada)
9  df = df.loc[(df['Title'] == 'La vida es sueño') & (df['Genre'] == 'Tragedia')]
10
11
12 def asignaesquema(verso, lista, rimas):
13     quebradas = {'a': 'á', 'b': 'v', 'c': 'k', 'd': 'ð', 'e': 'é'}
14     if verso.Consonance in rimas:
15         rima = rimas[verso.Consonance]
16     else:
17         rima = lista[0]
18         lista = lista[1:]
19     if verso.Syllables > 9:
20         rima = rima.upper()
21     elif verso.Syllables < 6:
22         rima = quebradas[rima]
23     return lista, rima
24
25
26 def buscaestrofa(versos, metros=[36, 14, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2]):
27     metros = [m for m in metros if m <= len(versos)]
28     resultado = (0, '')
29     aquelindo = 'abbccddeeffgghhiijjkkllmmnnooppqqrrs'
30     if metros:
31         estrofas = {36: {'ABCDEFFAEBDCCFDABEECBFADDEACFBBDFECA': 'sextina'},
32                     15: {},
33                     14: {'ABBAABBACDCDCD': 'soneto', 'ABABABABCDCDCD':
       'soneto',
34                          'ABBAABBACDECDE': 'soneto', 'ABABABABCDECDE':
       'soneto',
35                          'ABBAABBACDEDCE': 'soneto', 'ABABABABCDEDCE':
       'soneto',
36                          'ABBAABBACDCEDE': 'soneto', 'ABABABABCDCEDE':
       'soneto',
37                          'ABBAABBACDEDEC': 'soneto', 'ABABABABCDEDEC':
       'soneto',
38                          'ABBAABBACDCCDC': 'soneto', 'ABABABABCDCCDC':
       'soneto',
39                          'ABBAABBACDEEDC': 'soneto', 'ABABABABCDEEDC':
       'soneto'},
40                     11: {},
```

```
41                  10: {'abbaaccddc': 'espinela',
42                       'abbacddeed': 'décima', 'abbccaadda': 'décima',
43                       'aábvckcddc': 'ovillejo'},
44                  9: {'avcvðeð': 'seguidilla compuesta'},
45                  8: {'ABABABCC': 'octava real',
46                      'abbaacca': 'copla',
47                     'ABBAACCA': 'copla de arte mayor'},
48                  7:{}, #seguidilla,
49                  6: {'abcabc' : 'sextilla', 'aabccb': 'sextilla', 'abbaba':
    'sextilla',
50                      'aBaBcC': 'sexteto lira', 'abCabC': 'sexteto lira',
51                      'ABABCC': 'sexta rima',
52                      'aBaBcC': 'estrofa alirada',
53                      'abkabk': 'estrofa manriqueña',
54                      'aabbba': 'zéjel'},
55                  5: {'aBabB': 'lira', 'ABABA': 'quinteto',
    'ABBAB':'quinteto',
56                  'ababa': 'quintilla', 'abbab': 'quintilla', 'aabba':
    'quintilla',
57                      'abaab': 'quintilla'},
58                  4: {'AAAA': 'cuaderna', 'abab': 'cuarteta', 'abba':
    'redondilla',
59                      'ABABA': 'serventesio', 'ABBA': 'cuarteto'},
60                  3: {'ABA': 'terceto', 'AAA':'trístico monorrimo'},
61                  2: {'aa': 'pareado', 'AA': 'pareado', 'Aa': 'pareado',
    'aA': 'pareado'}}
62          lista = 'abcdefgh'
63          palabras = mtrs = []
64          rimas = {}
65          patron = ''
66          for i, row in versos[:metros[0]].iterrows():
67              palabra = row.Text.split()[-1].lower().strip('.!,¡¿?;:')
68              lista, rima = asignaesquema(row, lista, rimas)
69              if palabra not in palabras:
70                  palabras.append(palabra)
71              if row.Consonance not in rimas and len(set(patron.lower())) < 6:
72                  rimas[row.Consonance] = rima
73              elif row.Consonance not in rimas:
74                  break
75              patron += rima
76          mtrs = [m for m in metros if m <= len(patron) and m not in mtrs]
77          for i in mtrs:
78              if i < len(patron):
79                  siguiente = patron[i]
80              else:
```

```
81                siguiente = 'x'
82            patron = patron[:i]
83            exceso = len(patron) % 2
84            if aquelindo.startswith(patron[len(patron) - 1 - exceso]) and \
85                    len(patron) > 5:
86                exceso = len(patron) % 2
87                resultado = (len(patron) - exceso, 'aquelindo')
88                break
89            elif patron in estrofas[i]:
90                if (i == 36 and len(palabras) == 6):
91                    resultado = (i, estrofas[i][patron])
92                    break
93                elif len(patron) == 5:
94                    if siguiente.lower() == patron[-1]:
95                        resultado = (4, estrofas[4][patron[:4]])
96                    else:
97                        resultado = (i, estrofas[i][patron])
98                    break
99                else:
100                   resultado = (i, estrofas[i][patron])
101                   break
102               #elif siguiente in patron:
103               # continue
104
105    return resultado
106
107 def Estr(dataframe):
108    orig = 0
109    estrofas = {}
110    nulo = ('', 0)
111    est = []
112    while orig < len(dataframe):
113        i, estrofa = buscaestrofa(dataframe[orig:])
114        if i > 0 and orig < len(dataframe):
115            if estrofa in estrofas:
116                estrofas[estrofa] += 1
117            else:
118                estrofas[estrofa] = 1
119            for index in range(0,i):
120                est.append((estrofa, estrofas[estrofa]))
121            orig += i
122        else:
123            #for index in range(0,1):
124            est.append(nulo)
125            orig += 1
```

```
126     return est
127
128
129 def romances(obra):
130     #cuarteta asonantada
131     resultado = []
132     estrofa = obra.loc[obra['Stanza'] == 'romance']['nStanza'].max()
133     if estrofa != estrofa:
134         estrofa = 0
135     asonancia = ''
136     versos = 0
137     n = 1
138     for i, row in obra.reset_index().iterrows():
139         if row.Stanza:
140             if versos:
141                 if versos > 4:
142                     estrofa += 1
143                     [resultado.append(('romance', estrofa)) for j in range(0,
        versos)]
144                 else:
145                     [resultado.append(('', 0)) for j in range(0, versos)]
146                 versos = 0
147             resultado.append((row.Stanza, row.nStanza))
148         else:
149             if versos == 0:
150                 asonancia = ''
151             elif versos % 2:
152                 if not asonancia or row.Assonance == asonancia:
153                     asonancia = row.Assonance
154                 elif versos == 3:
155                     resultado += [('', 0), ('', 0)]
156                     versos = 1
157                     asonancia = row.Assonance
158                     #asonancia = obra.iloc[i-1,:]['Assonance']
159                 else:
160                     estrofa += 1
161                     [resultado.append(('romance', estrofa)) for j in range(0,
        versos-1)]
162                     asonancia = row.Assonance
163                     versos = 1
164             versos += 1
165         n += 1
166     if versos:
167         if versos > 3:
168             estrofa += 1
```

```
169             [resultado.append(('romance', estrofa)) for i in range(0, versos)]
170         else:
171             [resultado.append(('', 0)) for i in range(0, versos)]
172     return resultado
173 
174 df = pd.read_csv(entrada)
175 actos = [1,2,3]
176 df = df.loc[(df['Title'] == 'La vida es sueño') & (df['Genre'] == 'Tragedia')
        & (df['Act'].isin(actos))]
177 df.drop(['Author', 'Title', 'Subtl', 'Genre', 'Speech', 'Subgenre', 'Edition',
178          'Date','Type', 'Gender','Ambiguous', 'Nuclei', 'Rhythm',
179          'sentiment_neu', 'sentiment_pos', 'sentiment_neg']
180         , axis=1, inplace=True)
181 e = []
182 segmento = df.loc[df['Act'].isin(actos)]
183 estrofas = Estr(segmento)
184 e += estrofas
185 
186 df["Stanza"] = [x[0] for x in e]
187 df["nStanza"] = [x[1] for x in e]
188 
189 df2 = df.loc[df['Act'].isin(actos)]
190 e = []
191 for acto in actos:
192     segmento = df2.loc[df2['Act'] == acto]
193     rom = romances(segmento)
194     e += rom
195 df2["Stanza"] = [x[0] for x in e]
196 df2["nStanza"] = [x[1] for x in e]
197 
198 from tabulate import tabulate
199 print(tabulate(df2,headers='keys', showindex=False))
```

B.5 Codificación XML-TEI

Listado `txt2tei.py`

```
1 #!/usr/bin/env python3
2 import re
3 from unidecode import unidecode
4 import pandas as pd
5 import lxml.etree as etree
6 from lxml.etree import Element, SubElement, parse
7 from datetime import datetime
8 import sys
```

```
import chardet
if sys.version_info < (3, 9):
    # importlib.resources either doesn't exist or lacks the files()
    # function, so use the PyPI version:
    import importlib_resources
else:
    # importlib.resources has files(), so use that:
    import importlib.resources as importlib_resources
import os.path

#
# Please, edit this part if necessary
editor = 'Fernando Sanz-Lázaro'
authority = 'University of Vienna, Institute of Romance Languages '\
    'and Literatures'
publisher_id = ''
publisher_name = ''
licence = 'CC BY 3.0'
licence_url = 'https://creativecommons.org/publicdomain/by/3.0/'
hodie = datetime.today().strftime('%Y-%m-%d')
nunc = datetime.today().strftime('%H:%M:%S')
authority = ''
publisher = ('', '', '')
licence = 'CC BY'
licence_url = 'https://creativecommons.org/publicdomain/BY/3.0/'
xml_model = '<?xml-model href="https://dracor.org/schema.rng" type="applicati'\
        'on/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>'
#


def make_tree(title, subtitle, author, genre, subgenre, source, date,
              authority, publisher, licence, speakers_list):
    root = Element('TEI',
                   xmlns='http://www.tei-c.org/ns/1.0')
    root.set('{http://www.w3.org/XML/1998/namespace}lang', 'es')
    tree = etree.ElementTree(root)
    tei_header = SubElement(root, 'teiHeader')
    # fileDesc
    file_desc = SubElement(tei_header, 'fileDesc')
    title_stmt = Element('titleStmt')
    title_stmt = make_title(title_stmt, title, subtitle)
    title_stmt = make_authors(title_stmt, author)
    title_stmt = make_editor(title_stmt, editor)
    file_desc.append(title_stmt)
    # publicationStmt
```

```
54      publication_stmt = SubElement(file_desc, 'publicationStmt')
55      publication_stmt = make_edition(
56          publication_stmt, authority, publisher, licence)
57      file_desc.append(publication_stmt)
58      # srcDesc
59      source_desc = SubElement(file_desc, 'sourceDesc')
60      make_source(source_desc, source)
61      file_desc.append(source_desc)
62
63      # profileDesc
64      profile_desc = SubElement(tei_header, 'profileDesc')
65      par = SubElement(profile_desc, 'particDesc')
66      par.append(make_participants(speakers_list))
67      text_class = SubElement(profile_desc, 'textClass')
68      gen = SubElement(
69          SubElement(
70              text_class,
71              'keywords'),
72          'term',
73          type='genreTitle')
74      gen.text = genre
75
76      # revisionDesc
77      tei_header.append(make_revision())
78      # Construct text and body
79      standoff = SubElement(root, 'standOff')
80      standoff = make_standoff(standoff, date)
81
82      SubElement(root, 'text')
83      return tree
84
85
86  def parse_date(date):
87      if not date or len(date) == 0:
88          date = {'notbefore': '1550', 'notafter': '1699'}
89      elif '-' in date:
90          date = {'notbefore': date.split('-')[0],
91                  'notafter': date.split('-')[1]}
92      elif '?' in date:
93          date = {'when': date.strip('?'), 'cert': 'medium'}
94      else:
95          date = {'when': date, 'cert': 'medium'}
96      return date
97
98
```

```
 99 def make_title(title_stmt, title, subtitle):
100     if len(subtitle) > 0:
101         titlea = SubElement(title_stmt, 'title', type='main')
102         titleb = SubElement(title_stmt, 'title', type='sub')
103         titleb.text = subtitle
104     else:
105         titlea = SubElement(title_stmt, 'title')
106     titlea.text = title
107     return title_stmt
108
109
110 def make_authors(title_stmt, author):
111     title_stmt.append(author)
112     return title_stmt
113
114
115 def make_editor(title_stmt, editor):
116     resp_stmt = SubElement(title_stmt, 'respStmt')
117     SubElement(
118         resp_stmt,
119         'resp').text = 'Automatic generation of the XML/TEI by'
120     SubElement(resp_stmt, 'persName').text = editor
121     return title_stmt
122
123
124 def make_edition(publication_stmt, authority, publisher, licence):
125     SubElement(publication_stmt, 'authority').text = authority
126     pub = SubElement(publication_stmt, 'publisher')
127     pub.set('{http://www.w3.org/XML/1998/namespace}id', publisher[0])
128     pub.text = publisher[1]
129     SubElement(publication_stmt, 'idno', type='URL').text = publisher[2]
130     SubElement(publication_stmt, 'date', when=f'{hodie}T{nunc}')
131     lic = SubElement(SubElement(publication_stmt, 'availability'), 'licence')
132     SubElement(lic, 'ab').text = licence
133     SubElement(lic, 'ref').text = licence_url
134     return publication_stmt
135
136
137 def make_source(source_desc, source):
138     bibld = SubElement(source_desc, 'bibl', type='digitalSource')
139     source = [x.strip() for x in source.split('*')]
140     original = ''
141     web = ''
142     url = ''
143     if len(source) > 0:
```

```
144         original = source[0]
145         if len(source) > 1:
146             web = source[1]
147             if len(source) > 2:
148                 url = source[2]
149     SubElement(bibld, 'name').text = web
150     SubElement(bibld, 'idno', type='URL').text = url
151     biblo = SubElement(source_desc, 'bibl', type='originalSource')
152     SubElement(biblo, 'title').text = original
153     return source_desc
154 
155 
156 def make_standoff(stand_off, date):
157     list_event = SubElement(stand_off, 'listEvent')
158     SubElement(list_event, 'date', date, type='print')
159     SubElement(list_event, 'date', date, type='written')
160     return stand_off
161 
162 
163 def make_revision():
164     rev = Element('revisionDesc')
165     first_commit = SubElement(SubElement(rev, 'listChange'),
166                               'change', when=f'{hodie}T{nunc}')
167     first_commit.text = 'Converted with txt2tei'\
168         '<https://github.com/fsanzl/txt2tei>'
169     return rev
170 
171 
172 def make_participants(speakers_list):
173     list_person = Element('listPerson')
174     collective = ['UNOS', 'OTROS', 'UNAS', 'OTRAS', 'HOMBRES', 'MUJERES',
175                   'MÚSIC', 'CORO', 'VOCES', 'SOLDADOS']
176     for character in speakers_list:
177         name = speakers_list[character][0]
178         pid = character.strip('#')
179         sexo = speakers_list[character][1]
180         if any([x in name for x in collective]):
181             person = 'personGrp'
182         else:
183             person = 'person'
184         char = SubElement(list_person, person, sex=sexo)
185         char.set('{http://www.w3.org/XML/1998/namespace}id', pid)
186         SubElement(char, 'persName').text = name
187     return list_person
188 
```

```
189
190 def assign_gender(name):
191     if any(x in name for x in 'ºª'):
192         sx = 'MALE'
193     elif any(x in name for x in 'ªª'):
194         sx = 'FEMALE'
195     else:
196         name = re.sub(r'(.*)\d.*', '\1', name).strip()
197         if any(name.startswith(ambiguous)
198                for ambiguous in ['MÚSIC', 'ACOMPAÑAMIENT']):
199             sx = 'UNKNOWN'
200         elif any(name.endswith(morf) for morf in
201                  ['A', 'AS', 'ª', 'ª', 'DOÑA ']):
202             sx = 'FEMALE'
203         elif any(name.endswith(morf) for morf in
204                  ['O', 'OS', 'º', 'DON ', 'OR', 'EL', 'UD', 'ÍAS']):
205             sx = 'MALE'
206         else:
207             sx = 'UNKNOWN'
208     return sx
209
210
211 def make_id(name):
212     name = re.sub(r'(\d)\.[ª ºªº]', r'\1', name)
213     return f'#{unidecode(name.lower().replace(" ", "-"))}'
214
215
216 def find_characters(play_file):
217     try:
218         nombresdf = pd.read_csv('sexos.csv')
219     except Exception:
220         nombresdf = pd.DataFrame(columns=['Personaje', 'Sexo', 'Comprobado'])
221     names = []
222     characters_dict = {}
223     indirect = ['ÉL', 'ELLA', 'ELLOS', 'ELLAS', 'TODOS', 'TODAS']
224     # 'UNOS', 'OTROS', 'UNAS', 'OTRAS',
225     # 'UNO', 'OTRO', 'UNA', 'OTRA']
226     collective = ['LOS ', 'LAS ']
227     while True:
228         line = play_file.readline().split('#')[0]
229         if re.search(r'^[\[A-ZÁ-Úa-zá-ú0-9]', line):
230             characters = grouped(line)
231             for character in characters:
232                 character = character.strip().replace('[', '').replace(']', '')
233                 if 'MÚSIC' in character:
```

```
234                     character = 'MÚSICOS'
235                 if character not in [item for sublist in [names, indirect]
236                                      for item in sublist] and not any(
237                                          character.startswith(art)
238                                          for art in collective):
239                     names.append(character)
240         if not line:
241             break
242     plurals = []
243     for name in names:
244         name_clean = re.sub(r'(.*)\s+\d.*$', r'\1', name)
245         if re.search(r'\d\.*[ª ºªº]*$', name):
246             plurals += [f'{name_clean}{morpheme}'
247                         for morpheme in ['S', 'ES']]
248     names = [name for name in names if name not in plurals]
249     pairs = []
250     for name in names:
251         name_clean = re.sub(r'(.*)\s+\d.*$', r'\1', name)
252         if name_clean in nombresdf['Personaje'].unique():
253             sex = nombresdf.loc[nombresdf['Personaje'] ==
254                                 name_clean].iloc[0]['Sexo']
255         else:
256             sex = assign_gender(name_clean)
257         pairs.append((name, sex))
258     for pair in pairs:
259         characters_dict[make_id(pair[0])] = [pair[0], pair[1]]
260     return characters_dict
261
262
263 def parse_cast(line):
264     cast = Element('castList')
265     characters = line.split('*')
266     if '|' in characters[0]:
267         title = characters[0].split('|')
268         characters = title[1:] + characters[1:]
269         title = title[0]
270         SubElement(cast, 'head').text = title
271     for character in characters:
272         char = character.split(',')
273         item = SubElement(cast, 'castItem')
274         SubElement(item, 'role').text = char[0]
275         if len(char) > 1:
276             SubElement(item, 'roleDesc').text = char[1]
277     return cast
278
```

```
279
280 def split_singulars(character, characters_list):
281     collective_list = []
282     if character.endswith('s'):
283         base = character.rstrip('s').rstrip('e')
284     for char in characters_list:
285         if char.startswith(base) and char != character:
286             collective_list.append(char)
287     if len(collective_list) < 1:
288         collective_list = [character]
289     return collective_list
290
291
292 def last_characters(on_stage, character, characters_list):
293     if character.endswith('s'):
294         collective_list = split_singulars(character, characters_list)
295     else:
296         collective_list = [character]
297     for i in collective_list:
298         if i in on_stage:
299             on_stage.remove(i)
300         on_stage.append(i)
301     return on_stage
302
303
304 def grouped(character):
305     characters = character.upper().replace(' E ', ' Y '). strip().split(', ')
306     return [x.strip() for x in characters[:-1] + characters[-1].split(' Y ')]
307
308
309 def parse_collective_name(character, on_stage, characters_list):
310     todos = [
311         x for x in on_stage
312         if x in characters_list and characters_list[x][1] == 'MALE']
313     todas = [
314         x for x in on_stage
315         if x in characters_list and characters_list[x][1] == 'FEMALE']
316     groups = {'TODOS': (-1, False), 'TODAS': (-1, True),
317               'LOS DOS': (2, False), 'LOS 2': (2, False), 'ELLOS': (2, False),
318               'LAS DOS': (2, True), 'LAS 2': (2, True), 'ELLAS': (2, True),
319               'LOS 3': (3, False), 'LOS TRES': (3, False),
320               'LAS 3': (3, True), 'LAS TRES': (3, True),
321               'LOS 4': (4, False), 'LOS CUATRO': (4, False),
322               'LAS 4': (4, True), 'LAS CUATRO': (4, True)}
323     if character in groups:
```

```
324         if not groups[character][1]:
325             if all([characters_list[x][1] == 'FEMALE'
326                     for x in on_stage[-groups[character][0]:]]):
327                 collective = todos[-1:] + todas[-groups[character][0] + 1:]
328             else:
329                 collective = on_stage[-groups[character][0]:]
330         else:
331             collective = todas[-groups[character][0]:]
332     else:
333         collective = [x for x in on_stage]
334     return collective
335
336
337 def parse_pronoun(pronoun, on_stage, characters_list):
338     if pronoun == 'ÉL':
339         sex = 'MALE'
340     else:
341         sex = 'FEMALE'
342     return [[person for person in on_stage
343              if characters_list[person][1] == sex]][-1]
344
345
346 def parse_exit(on_stage, line):
347     line = line.strip().replace('<i>', '')
348     if re.search('[Va][n]*se', line, re.IGNORECASE):
349         if any(character[0] in line.upper() for character in on_stage):
350             for character in on_stage:
351                 if character[0] in line.upper():
352                     on_stage = [x for x in on_stage if x[0] != character[0]]
353         elif line.upper().startswith('VASE'):
354             on_stage = on_stage[:-1]
355         elif line.upper().startswith('VANSE'):
356             on_stage = on_stage[:-2]
357     return on_stage
358
359
360 def parse_speech(ln, nextl, nl, p=False):
361     tabs = ln.count('\t')
362     tabsn = nextl.count('\t')
363     text = re.search('(?:<.*>)*(.*)', ln.strip()).group(1)
364     if p:
365         line = Element('p')
366     else:
367         line = Element('l')
368         if tabsn > 1:
```

```
369             if tabs > 1:
370                 line.set('part', 'M')
371             else:
372                 line.set('part', 'I')
373             line.set('n', f'{nl + 1}')
374         else:
375             nl += 1
376             # if nl line.set('n', f'nl')
377             if tabs > 1:
378                 line.set('part', 'F')
379     line.text = text
380     return (line, nl)
381
382
383 def compare_sexes(pids, characters_list):
384     return 'UNKNOWN'
385
386
387 def parse_author(name):
388     if os.path.isfile('authors.xml'):
389         fauthors = parse('authors.xml')
390     else:
391         with importlib_resources.path('txt2tei', 'authors.xml') as dat:
392             fauthors = parse(dat)
393     names = name.split()
394     if len(names) > 1:
395         cert = 'medium'
396     else:
397         cert = 'high'
398     authors = [a for a in fauthors.xpath('author')
399                if any(b.text == names[0] for b in a.xpath('persName/*'))]
400     if authors:
401         author = [a for a in authors if a.get('cert') == cert][0]
402     else:
403         author = Element('Author')
404         b = SubElement(author, 'persName')
405         b.text = name
406     return author
407
408
409 def parse_name(ln, characters_list, on_stage):
410     if '#' in ln:
411         splitted = ln.split('#')
412         ids = splitted[1:]
413         speakers = splitted[0].strip().upper()
```

```
414         pids = [f'#{pid.strip()}' for pid in ids]
415         speaker_info = [speakers, pids, '']
416         characters_list.append(speaker_info)
417     else:
418         pids = []
419         ip = []
420         speakers = ln.upper()
421         speakers_list = grouped(speakers)
422         for speaker in speakers_list:
423             speaker = speaker.strip().replace('[', '').replace(']', '')
424             ip = [make_id(speaker)]
425             if 'MÚSIC' in speaker:
426                 ip = ['#musicos']
427             elif any(speaker.startswith(x)
428                      for x in ['TODO', 'TODA', 'LOS ', 'LAS ',
429                                'ELLOS', 'ELLAS']):
430                 ip = parse_collective_name(speaker, on_stage, characters_list)
431             elif speaker in ['ÉL', 'ELLA']:
432                 ip = parse_pronoun(speaker, on_stage, characters_list)
433             else:
434                 pass
435             for i in ip:
436                 if i not in pids:
437                     pids.append(i)
438         for x in pids:
439             on_stage = last_characters(on_stage, x, characters_list)
440     ppids = ' '.join(pids)
441     speech = Element('sp', who=f'{ppids}')
442     SubElement(speech, 'speaker').text = speaker
443     return (speech, on_stage)
444
445
446 header_labels = {'t': 'title', 'tt': 'subtitle', 'a': 'author',
447                  'g': 'genre', 's': 'subgenre', 'o': 'source',
448                  'f': 'date', 'x': 'comment'}
449 rev_header_labels = dict((v, k) for k, v in header_labels.items())
450 body_labels = {'j': 'act', '<e>': 'echo', '<p>': 'prose',
451                '<i>': 'stage_direction', '<x>': 'comment'}
452 subtitle = ''
453 author = ''
454 genre = ''
455 subgenre = ''
456 # date = ''
457 source = ''
458 on_stage = []
```

```
459 dramatis_personae = Element('null')
460 roman = {1: 'I', 2: 'II', 3: 'III', 4: 'IV'}
461 reference = {'1º': 'I', '1.º': 'I', '2º': 'II', '2.º': 'II', '3º': 'III',
462              '3.º': 'III', '4º': 'IV', '4.º': 'IV', 'PRIMERO': 'I',
463              'SEGUNDO': 'II', 'TERCERO': 'III', 'CUARTO': 'IV',
464              '1ª': 'i', '1.ª': 'i', '2ª': 'ii', '2.ª': 'ii', '3ª': 'iii',
465              '3.ª': 'iii', '4ª': 'iv', '4.ª': 'iv', 'PRIMERA': 'i',
466              'SEGUNDA': 'ii', 'TERCERA': 'iii', 'CUARTA': 'iv'}
467
468
469 def unixfy(filename):
470     with open(filename, 'rb') as file:
471         raw = file.read(32)
472         encoding = chardet.detect(raw)['encoding']
473     return encoding
474
475
476 def main(input_arguments=sys.argv):
477     input_file = input_arguments[1]
478     output = f'{input_file.rsplit(".", 1)[0]}.xml'
479     enc = unixfy(input_file)
480     with open(input_file, 'r', encoding=enc) as f:
481         characters_list = find_characters(f)
482         f.seek(0)
483         lines = f.readlines()
484     author = date = title = n = sp = ''
485     for line in lines:
486         if line.startswith('<'):
487             label = re.search(r'^\s*<(\w*)>', line.strip()).group(1)
488             if label in header_labels.keys():
489                 header_labels[label] = line.strip(f'<{label}>').strip()
490             else:
491                 break
492         else:
493             break
494     if header_labels['f']:
495         date = parse_date(header_labels['f'])
496     else:
497         date = parse_date('')
498     author = parse_author(header_labels['a'])
499     title = header_labels['t']
500     subtitle = header_labels['tt']
501     genre = header_labels['g']
502     subgenre = header_labels['s']
503     source = header_labels['o']
```

```
504     tree = make_tree(
505         title,
506         subtitle,
507         author,
508         genre,
509         subgenre,
510         source,
511         date,
512         authority,
513         publisher,
514         licence,
515         characters_list)
516     text = tree.xpath('/TEI/text')[0]
517     front = SubElement(text, 'front')
518     body = SubElement(text, 'body')
519     count = 0
520     actn = 0
521     scenen = 1
522     speakers_list = []
523     act = SubElement(body, 'div', type='act', n='1')
524     scene = SubElement(act, 'div', type='scene', n='1')
525
526     for idx, line in enumerate(lines):
527         next_line = ''
528         if not any(line.startswith(x) for x in header_labels):
529             if line.strip() in reference.keys():
530                 print(f'{input_file}:\tPlease, edit manually'
531                       '<sp who="#character"> for '
532                       f'{line.strip()} in {output} verse {n}\n')
533             if line.startswith('<el>'):
534                 dramatis_personae = parse_cast(line)
535                 front.append(dramatis_personae)
536             elif line.startswith('<j>') or line.startswith('<q>'):
537                 on_stage = []
538                 if line.startswith('<j>'):
539                     head = line.replace('<j>', '').strip()
540                     actn += 1
541                     scenen = 1
542                     if actn > 1:
543                         act = SubElement(body, 'div', type='act', n=f'{actn}')
544                         scene = SubElement(act, 'div', type='scene',
545                                            n=f'{scenen}')
546                     if head and len(head) > 1:
547                         SubElement(act, 'head').text = head
548                 else:
```

```
549                     scenen += 1
550                     if scenen > 1:
551                         scene = SubElement(act, 'div', type='scene',
552                                            n=f'{scenen}')
553             elif line.strip().startswith('<i>'):
554                 text = f'{line.replace("<i>","").strip()}'
555                 if line.startswith('\t'):
556                     SubElement(sp, 'stage').text = text
557                 else:
558                     stage = SubElement(scene, 'stage')
559                     stage.text = text
560                     parse_exit(speakers_list, line)
561             elif re.search(r'^[A-ZÁ-ÚÜ0-9\[\.]+', line):
562                 parsed_line = parse_name(line, characters_list, on_stage)
563                 sp = parsed_line[0]
564                 scene.append(sp)
565                 on_stage = parsed_line[1]
566             elif line.startswith('\t') or line.strip().startswith('<p>'):
567                 if line.strip().startswith('<p>'):
568                     p = True
569                 else:
570                     p = False
571                 for i in list(range(1, 9)):
572                     if idx+i < len(lines):
573                         if any([x in lines[idx+i] for x in ['<i>', '<x>']]):
574                             pass
575                         elif lines[idx+i].startswith('\t'):
576                             next_line = lines[idx+i]
577                             break
578                         else:
579                             pass
580                     else:
581                         next_line = '\tFinal'
582                 parsed_line = parse_speech(line, next_line, count, p)
583                 code = parsed_line[0]
584                 count = parsed_line[1]
585                 sp.append(code)
586     tree.write(output, doctype=xml_model, encoding='UTF-8', pretty_print=True)
587
588
589 main()
```

Ejemplos de uso C

Transcripción

Vamos a mostrar un ejemplo de aplicación de las implementaciones en Python de los algoritmos propuestos. Con esto demostramos cómo hemos conseguido abstraernos de un problema relativamente complejo para convertirlo en una caja negra en la que, en una situación real, introducimos por un lado lo que deseamos analizar y obtenemos el análisis por el otro, sin tener que preocuparnos del proceso, lo descrito arriba, que lleva al resultado. Para esto, tomaremos la biblioteca informática Fonemas (Sanz-Lázaro, 2023a) para Python, que aplica los algoritmos elaborados en el capítulo dedicado al análisis fonológico.

Ejemplo de uso

Para facilitar la instalación y uso del software, este se encuentra publicado bajo licencia libre GNU General Public License (GPL), tanto su código fuente como en un paquete ya preparado para instalar. Esto podemos hacerlo directamente por medio del gestor de paquetes de Python desde la línea de órdenes[133] del sistema operativo, con la prevención de que hay que instalar el paquete junto a la biblioteca Silabeador (Sanz-Lázaro, 2023d), de la que depende, como veremos un poco más adelante.

```
$ pip install silabeador fonemas
```

Una vez hecho esto, ya tenemos esta biblioteca disponible para hacer transcripciones y podemos cargarla en nuestros programas para emplearla cuando necesitemos hacerlo. La forma más simple de hacerlo es arrancar el intérprete interactivo de Python y

[133]Con frecuencia se emplea *línea de comandos*, que es un calco del inglés *command line*.

crear una instancia de la clase `Transcription`[134]. Ilustraremos con un ejemplo cómo hacemos una sencilla transcripción de, verbigracia, *verbigracia*. El procedimiento se limita a introducir las dos instrucciones que mostramos a continuación.

```
$ python
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from fonemas import Transcription
>>> texto_transcrito = Transcription('verbigracia')
```

Veamos qué hemos hecho. Primero pedimos al sistema operativo que inicie un intérprete interactivo de Python. Esto lo hacemos en la línea de órdenes mediante el mandato que aparece tras el símbolo del dólar. Como respuesta, el intérprete nos facilita datos sobre la versión y nos informa de cómo podemos ampliar la información. Tras esto, nos presenta tres signos seguidos de *mayor que*, que son la manera en que nos invita a introducir instrucciones de Python. Así lo hacemos, en este caso, importando primero la clase `Transcription` del módulo `fonemas`. Esto tiene como objetivo poder emplearla en la siguiente instrucción. En esta, creamos el objeto `texto_transcrito`, que es una instancia de la clase `Transcription`, y le pasamos el argumento `'verbigracia'`. El objeto que hemos creado contiene la información fonológica y fonética que requerimos.

Si deseamos acceder a esta información fonológica y fonética de la palabra, podemos hacerlo leyendo el valor de los atributos del objeto `texto_transcrito` que hemos creado. Así, en `texto_transcrito.phonology.words` encontraremos el texto dividido por palabras y en `texto_transcrito.phonology.syllables` dividido por sílabas. De manera análoga tenemos las transcripciones fonéticas en AFI y SAMPA en los subatributos `words` y `syllables` de sendos atributos `texto_transcrito.phonetics` y `texto_transcrito.sampa`, respectivamente.

```
>>> texto_transcrito.phonology.words
['berbi'gɾaθja']
>>> texto_transcrito.phonology.syllables
['berˈ', 'bi', ''gɾa', 'θja']
>>> texto_transcrito.phonetics.words
['βerβi'ɣɾaθja']
>>> texto_transcrito.phonetics.syllables
['βerˈ', 'bi', ''ɣɾa', 'θja']
>>> texto_transcrito.sampa.words
```

[134]En la implementación material con Python hemos optado por emplear el inglés para maximizar el número de personas que pueden examinar el código fuente de forma intuitiva en el repositorio público en el que está albergado.

```
['BerBi"GraTja']
>>> texto_transcrito.sampa.syllables
['Ber', 'bi', '"Gra', 'Tja']
```

El ejemplo anterior no necesita mayor explicación. De esta manera transcribimos y accedemos a las transcripciones manualmente; de la misma manera, los programas que emplean la biblioteca Fonemas se limitan a hacer llamadas equivalentes y, como nosotros en esta primera prueba, permanecen ajenos a las operaciones que esta lleva a cabo para llegar al resultado.

Silabación

Para facilitar el uso del modelo de pruebas de Python, la biblioteca proporciona dos funciones adicionales que pueden ser usadas de manera inmediata sin necesidad de crear explícitamente un objeto de la clase `Syllabification` (equivalente a la clase `Silabación` que hemos visto). Estas son las funciones `syllabify()` y `tonica()`, que pueden ser invocadas rápidamente y devuelven las sílabas de una palabra y la posición de la sílaba tónica, respectivamente. Se limitan a hacer una llamada a `Syllabification` y devolver el atributo correspondiente como hemos desarrollado en detalle arriba, y permiten indicar si se han de omitir las excepciones, que están activadas por defecto.

Ejemplo de uso

El módulo de silabación es independiente del transcriptor fonológico, por lo que, al igual que este, podemos instalarlo y usarlo por separado. El proceso de instalación es el mismo que con la otra biblioteca.

```
$ pip install silabeador
```

Podemos probar que todo funciona como es debido invocando el intérprete interactivo de Python pidiéndole que divida las palabras de una frase en sus sílabas y nos devuelva estas y la posición de la sílaba acentuada.

```
$ python
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> words = 'El gran teatro del mundo de Pedro Calderón de la Barca'
>>> from silabeador import Syllabification
>>> [(s.syllables, s.stress) for s in [Syllabification(w) for w in
sentence.split()]]
```

```
   [(['El'], -1), (['gran'], -1), (['te', 'a', 'tro'], -2), (['del'],
-1), (['mun', 'do'], -2), (['de'], -1), (['Pe', 'dro'], -2), (['Cal',
'de', 'rón'], -1), (['de'], -1), (['la'], -1), (['Bar', 'ca'], -2)]
```

Como dijimos, para la versión en Python hemos dotado a la clase de la capacidad de operar de distintas maneras a las que necesitamos en nuestro trabajo. Para ello, podemos pasarle los argumentos adicionales `exceptions`, `ipa` y *h*. El primero de estos argumentos sirve para indicar si la silabación debe hacerse ignorando las excepciones. Esto puede facilitar una división silábica más fiel a la pronunciación en determinadas variedades del español contemporáneo. El segundo argumento, es imprescindible para permitir el uso general del módulo silabeador, pues su valor determina si el texto va a leerse como grafémico o fonémico. Cabe pensar que la división silábica ortográfica encontrará más aplicaciones que la fonémica. Por ese motivo, aunque aquí asumamos que el texto es fonémico, en el diseño en Python la opción por defecto es grafémica y se necesita indicar de forma explícita que el valor de `ipa` es `False` al crear un objeto de la clase. La tercera opción solo es relevante para textos grafémicos, pues modifica el comportamiento de la división silábica en grupos con *h* posconsonántica, de modo que si se activa, la división silábica será ortográfica (*in·hi·bir*) y, si no se indica nada, será fonológica (*i·nhi·bir*).

Escansión métrica

Finalmente, la biblioteca de escansión de los versos proporciona la clase `VerseMetre`. Esta biblioteca pone en uso todo lo desarrollado antes para escandir versos. La manera más simple de usarla es es importar la biblioteca que contiene la clase, de la que creamos una instancia. El único parámetro obligatorio es el verso a escandir. Adicionalmente, en caso de que el metro dependa del entorno, puede recibir de manera opcional una lista de posibles metros conforme a las expectativas. De esta manera, si estamos analizando un soneto, antepondremos once sílabas, pero al encontrar endecasílabos en una obra teatral, pondremos once y siete delante; procederemos de forma análoga según lo que convenga en cada situación.

Ejemplo de uso

El módulo de escansión depende de los dos anteriores, por lo que estos habrán de instalarse de no encontrarse ya en el sistema para satisfacer los requisitos del programa de escansión. El proceso de instalación es el mismo que con las otras dos bibliotecas.

```
$ pip install libEscansion
```

El funcionamiento es ciertamente sencillo. Solo se requiere invocar la clase para crear una instancia de ella. Los atributos de este objeto contienen la descripción métrica del verso.

```
$ python
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from libEscansion import VerseMetre
>>> línea = 'Alma a quien todo un dios prisión ha sido'
>>> verso = VerseMetre(línea)
>>> verso.rhyme
'ido'
>>> verso.asson
'io'
>>> verso.rhythm
'+--+++-+++-'
>>> verso.nuclei
'AaeOUOiOAIo'
>>> verso.syllables
[['Al', 'ma'], ['kjen'], ['tO', 'dŏUn'], ['djOs'], ['pɾi', 'sjOn'],
['ʰA'], ['sI', 'do']]
>>> verso.synaloephas
[([0, 1], 4.0, 'ma', 'a'), ([3, 1], -4.0, 'do', 'Un')]
>>> verso.count
11
>>> verso.natural
11
```

Manipulando el parámetro de metros esperados, se altera el comportamiento de la escansión. Por ejemplo, si en lugar de estar analizando un soneto esperáramos encontrar un verso alejandrino, indicaríamos al crear el objeto que trate de forzar la escansión a catorce sílabas.

```
>>> verso = VerseMetre(línea, [14, 8, 11, 7, 6, 5, 4])
>>> verso.syllables
[['Al', 'ma'], ['a'], ['kjen'], ['tO', 'do'], ['Un'], ['djOs'],
['pɾi', 'si', 'On'], ['ʰA'], ['sI', 'do']]
>>> verso.count
14
>>> verso.natural
11
```

Al contrario que en el caso anterior, en el que la escansión sigue las reglas al pie de la letra, ahora cobra sentido el último atributo, pues indica el resultado de la escansión

natural. Dado que hemos forzado la escansión a catorce sílabas, esta difiere de aquella. En nuestro caso, no usamos la biblioteca de forma interactiva, sino que va integrada en un *script*, que se encarga de escandir los versos y guardar los resultados donde corresponde.

Análisis dramático cuantitativo manual D

Durante la investigación para esta tesis nos familiarizamos dentro del proyecto Sound and Meaning con los análisis cuantitativos del léxico y el metro de corpus teatrales. Sin embargo, faltaba la perspectiva de potenciales investigadores de las características dramáticas de los textos para tener el cuadro completo. Por ello, decidimos analizar una pieza teatral cuantitativamente de manera completamente manual para conocer así las necesidades de tales análisis y, con ello, las soluciones digitales que podemos aportar. Estimamos que resulta de interés y lo hemos incluido en este apéndice, pues muestra mediante el ejemplo las dificultades que presenta la extracción, recopilación y tratamiento de datos sobre el drama sin ayuda técnica. Se limita tan solo la primera jornada de *La vida es sueño*[135] (Calderón de la Barca, 1636/2019), pues resulta más que suficiente para evidenciar los requerimientos de este tipo de pruebas. En el futuro —ahora sí, con ayuda del computador—, podría extenderse para hacer un análisis de la comedia entera, así como de otras, para encontrar patrones de interés.

Iremos presentando las magnitudes y aplicándolas al texto para observar los elementos

[135]Nos guiamos por la edición de Ciriaco Morón para aprovechar la segmentación en escenas,pues usa la división de Hartzenbusch (Calderón de la Barca, 1636/1848). Esta resulta útil para aclarar los conceptos, incluso si toma decisiones como disponer las escenas VII y VIII no solo dividiendo en dos un romance, sino partiendo sin contemplaciones el parlamento de Clotaldo. Tanto las dos primeras ediciones publicadas en 1636, la una en Zaragoza (Calderón de la Barca, 1636b) y en Madrid la otra (Calderón de la Barca, 1636a), como la edición que Vera Tassis (Calderón de la Barca, 1685) curó a partir de ambas, contemplan las líneas de Clotaldo como un único parlamento. A falta de manuscritos u otras versiones de autoridad equivalente, parece que la edición de Ruano de la Haza (Calderón de la Barca, 1924/1992) se ajusta más tanto al texto como a los criterios objetivos que hemos visto, pues reparte la jornada en dos cuadros separados por la salida al escenario de Astolfo y Estrella, en coincidencia con cambios métrico y espaciales. Sin embargo, aunque nos inclinamos por el criterio de Ruano de la Haza, la distribución de Hartzenbusch resulta óptima a efectos ilustrativos. En este caso, el fin justifica los medios.

que nos permiten hallarlas. Esos elementos son precisamente nuestra meta. El análisis que haremos sigue el modelo del seminal trabajo de Dinu (1968). En este se lleva a cabo una aproximación matemática al análisis del texto dramático de la manera que más tarde Romanska (2015, p. 446) denominaría *dramétrica*.

Presencia en escena

Empezando por análisis globales, Marcus (1973, pp. 292-310) propone una serie de magnitudes, que enumeraremos y describiremos brevemente. La primera es la *densidad escénica*, cuyo cálculo requiere definir primero una *matriz de presencia* de la escena. En esta, cada línea representa un personaje, cada columna una escena, y en las celdas se indica la presencia o ausencia del personaje en la escena correspondiente. Lo ilustramos con la matriz de presencia de Figura 7.1, que representa las escenas en que se divide la primera jornada de *La vida es sueño*. A la clasificación binaria de Marcus le hemos añadido los matices de si el personaje habla o no y si lo hace desde dentro.

En la escena primera solo están presentes dos personajes, Rosaura y Clarín, que dialogan alternándose en el turno de palabra. En la siguiente, tenemos también a Segismundo en la torre, de ahí que no tenga presencia mientras está dentro, pero sí diálogo, cuando pronuncia «¡Ay mísero de mí! ¡Ay infelice!». En la siguiente oímos a Clotaldo dentro, a pesar de que no sale hasta su segunda intervención; lo hace junto a un grupo indeterminado de soldados, si bien es aquel el que sostiene el peso del diálogo, apoyado por Segismundo y, en menor medida, por Clarín y Rosaura. El quinto parlamento de la escena lo pronuncia alguien indeterminado *dentro*. En la cuarta escena, ya sin Segismundo, salvo un par de breves intervenciones de Clarín y una interrupción de un soldado, conversan Rosaura y Clotaldo. En la quinta escena hablan Astolfo y Estrella en presencia de soldados y las damas de la infanta, que no intervienen. Lo mismo pasa en la mayor parte de la sexta escena, aunque esta sucede en presencia de Basilio y su acompañamiento, que le ponen fin. Rosaura y Clarín vuelven en la séptima escena a compartir tablas con Clotaldo y Basilio, aunque dejan a estos dos todos los diálogos, sin intervenir. Sí lo hacen, por el contrario, en la octava y última escena de la jornada, Clarín, de forma fugaz al principio, y Rosaura, en un largo diálogo con Clotaldo, hasta que Rosaura y Clarín salen antes de terminar el acto, al que pone fin Clotaldo con un monólogo.

La densidad escénica se define como la razón entre el número de personajes presentes en una escena y el número total de personajes. Por consiguiente, la densidad escénica de la primera escena, en la que participan dos personajes de los diez posibles, es

0,2 mientras que la escena sexta tiene 0,6 (Cuadro D.1). Además de disponer de la distribución y de los personajes presentes en cada segmento, necesitaremos saber también el número total de personajes envueltos en el texto a analizar.

Cuadro D.1: Densidad escénica.

Escena	**Personajes**	**Densidad**
I	2	0,2
II	3	0,3
III	5	0,5
IV	4	0,4
V	4	0,4
VI	6	0,6
VII	4	0,4
VIII	3	0,3

Marcus define después los tipos de relaciones entre los personajes. Para ello, propone asignarle a cada uno de ellos un conjunto de números naturales, correspondientes a las escenas en las que están presentes. Sean los personajes x e y de una obra o un fragmento de una obra, el conjunto de cuyas escenas denominaremos A. A los personajes x e y les asignamos sendos conjuntos Ax y Ay, cuyos elementos son los números correspondientes a las escenas en las que están presentes x e y, respectivamente. Si $Ax = Ay$, llamaremos *concomitantes* a los personajes x e y. Diremos que el personaje x está *dominado* por el personaje y si $Ax \subset Ay$ y, al contrario, y *dominará* al personaje x si $Ay \subset Ax$. Si no se da ninguna de las dos situaciones, hablaremos de dos personajes *escénicamente independientes*, que suele ser la relación más habitual.

Dentro de los personajes independientes, encontramos diferentes categorías. Si $Ax \cap Ay = \emptyset$, esto es, x e y no coinciden en escena, decimos que son *alternativos*. Si, además, $|Ax \cup Ay| = |A|$, o lo que es lo mismo, si en todas las escenas de la obra aparece o bien el personaje x o bien el personaje y, serán además *complementarios*.

Los datos de los que ya disponíamos permiten llevar a cabo estos cálculos para estimar los valores de las magnitudes vistas. Si aplicamos esto a los personajes de la primera jornada de *La vida es sueño*, obtenemos los conjuntos siguientes, cuyos cardinales mostramos en Figura D.1.

$$A_{\text{ROS}} = \{1, 2, 3, 4, 7, 8\}$$
$$A_{\text{CLAR}} = \{1, 2, 3, 4, 7, 8\}$$
$$A_{\text{SEGIS}} = \{2, 3\}$$
$$A_{\text{CLOT}} = \{3, 4, 7, 8\}$$
$$A_{\text{SOL}} = \{3, 4, 5, 6\}$$
$$A_{\text{AST}} = \{5, 6\}$$
$$A_{\text{ESTR}} = \{5, 6\}$$
$$A_{\text{DAMAS}} = \{5, 6\}$$
$$A_{\text{BAS}} = \{6, 7\}$$
$$A_{\text{ACOM}} = \{6\}$$

Tomemos por ejemplo a Rosaura. Vemos que $A_{\text{ROS}} = A_{\text{CLAR}}$, por lo que esta y Clarín son concomitantes en esta jornada. Se verifica que $A_{\text{SEGIS}} \subset A_{\text{ROS}}$, por lo que Rosaura domina a Segismundo. Ninguna de estas condiciones se cumple si comparamos a Rosaura con Clotaldo, por lo que cabe afirmar que son independientes. Sin embargo, $A_{\text{ROS}} \cap A_{\text{CLOT}} = \{3, 7{,}8\}$, por lo que no son alternativos. Por otra parte, $A_{Ros} \cap A_{Acom} = \emptyset$, por lo que, en este caso, Rosaura y el acompañamiento de Basilio sí serían alternativos. Sin embargo, dado que $|A_{\text{ROS}}| = 6$, $|A_{\text{ACOM}}| = 1$ y $|A| = 8$, no serían complementarios, puesto que $|A_{\text{ROS}} \cup A_{\text{ACOM}}| < |A|$. En el caso de Astolfo, $A_{\text{ROS}} \nsubseteq A_{\text{AST}}$ y $A_{\text{ROS}} \not\subset A_{\text{AST}}$, por lo que son independientes. Asimismo, $A_{\text{ROS}} \cap A_{\text{AST}} = \emptyset$, por lo que son alternativos y, además, $|A_{\text{AST}}| = 2$, por lo que $|A_{\text{ROS}} \cup A_{\text{AST}}| = |A|$, lo que hace a Rosaura y Astolfo personajes complementarios en esta jornada.

Otra forma de definir la relación entra un personaje x y un personaje y es encontrar su *distancia escénica*. Esta se define como una *cadena de x a y*, una secuencia de personajes $z_1, z_2, ..., z_n$ que cumplen que $z_1 = x_1$, $z_n = y$ y z_i y z_{i+1} no son alternativos para ningún valor de i, siendo $1 \leq i \leq n$. Así, tomemos a Segismundo y Basilio, dos personajes alternativos. La distancia $d(\text{SEGIS}, \text{BAS}) = 2$: de Segismundo a Clotaldo y, de este, a Basilio. Si no es posible definir una cadena entre dos personajes —por ejemplo, si intentamos buscar una cadena en sentido contrario, de Basilio a Segismundo—, diremos que la distancia escénica es infinita. La mayor de las distancias escénicas entre personajes se denomina *diámetro escénico*[136] de la obra. En este caso, también sería 2, al ser esta la mayor distancia posible. Sea como fuere, nos sigue bastando con los elementos que ya habíamos encontrado: segmentos y personajes de cada segmento.

[136]Es traducción literal. En el original se denomina *Durchmesser*.

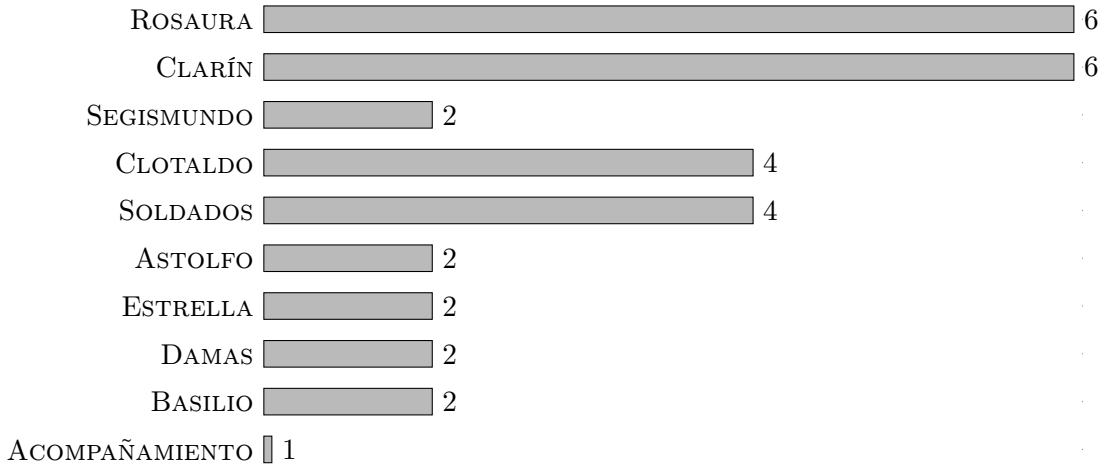


Figura D.1: Presencia en las distintas configuraciones.

Personajes en el conjunto

Las magnitudes anteriores describen características relativas de un personaje respecto a otro, pero no los ubican en el conjunto de la obra. Para hacer esto, nos valdremos del valor resultante del sumatorio de las distancias escénicas del personaje x de cada uno del resto de los personajes (D.1).

$$d(x) = \sum d(x,y) \tag{D.1}$$

De esta manera, vemos en Cuadro D.2 las distancias entre los personajes de la primera columna y los demás, que ocupan cada una de las columnas siguientes y, al final, la distancia d. Nótese que esta última magnitud para un personaje respecto de sí mismo es nula. En este caso, el menor valor d lo tienen los soldados: acabamos de toparnos con el primer problema. Efectivamente, los datos que hemos obtenido no son suficientes. La trampa está en que —cabe esperar— los soldados que acompañan a Clotaldo y los que van con Astolfo son personajes diferentes, incluso si el texto no lo menciona explícitamente o los recursos de la representación obligan a que los mismos actores interpreten ambos acompañamientos. Aquí, como en casi todo, conviene aplicar una buena dosis de sentido común antes de lanzarse a las interpretaciones literales. Al asignar personajes sin considerar la evidencia interna, hemos dado con una de las principales dificultades, si no la principal, de las que encontraremos para automatizar la extracción de los datos del texto. La situación contraria es también habitual, esto es, personajes que se nombran de maneras diferentes: Músicos o

MÚSICA, pero también, especialmente en autos sacramentales, personajes alegóricos que reciben diferentes nombres según el momento de la obra y su función dentro de esta. El formato XML-TEI resuelve esto mediante la asignación de un identificador único independiente del nombre textual a cada personaje, por lo que puede haber personajes diferentes con el mismo nombre o un mismo personaje nombrado de diferentes maneras, siempre que a cada uno, con independencia de como se le llame, le corresponda un y solo un identificador.

x	y										$d(x)$	$\delta(x)$
	ROS.	CLA.	SEG.	CLO.	SOL.	AST.	EST.	DAM.	BAS.	ACO.		
ROS.	0	1	1	1	1	2	2	2	1	2	11	2
CLAR.	1	0	1	1	1	2	2	2	1	2	11	2
SEGIS.	1	1	0	1	1	2	2	2	2	2	14	2
CLOT.	1	1	1	0	1	2	2	2	1	2	13	2
SOL.	1	1	1	1	0	1	1	1	1	1	9	1
AST.	2	2	∞	2	1	0	1	1	1	1	∞	∞
EST.	2	2	∞	2	1	1	0	1	1	1	∞	∞
DAMAS	2	2	∞	2	1	1	1	0	1	1	∞	∞
BAS.	1	1	∞	1	1	1	1	1	0	1	∞	∞
ACOM.	2	2	∞	2	1	1	1	1	1	0	∞	∞

Cuadro D.2: Distancia escénica.

Otra posibilidad es obtener un valor δ, siendo $0 \leq \delta(x) \leq d(x)$. En este caso, tomamos la mayor distancia entre un personaje x y cualquiera de los otros personajes. Así, $\delta(\text{ROS}) = 2$ mientras que $\delta(\text{AST}) = \infty$. Lo formalizamos mediante la ecuación D.2.

$$\delta(x) = \max_{x \neq y} d(x,y) \tag{D.2}$$

La siguiente magnitud a considerar es el número de escenas en que dos personajes x e y aparecen juntos, que denominamos $\gamma(x,y)$. De manera análoga a los cálculos de las magnitudes absolutas que hemos hecho en las ecuaciones D.1 y D.2, definiremos también aquí (D.3) un índice que dé una idea de la tendencia de un personaje a compartir escena con otros.

$$\gamma(x) = \sum_{x \neq y} \gamma(x,y) \tag{D.3}$$

Hemos recopilado en Cuadro D.3 los valores de γ de los personajes, tanto relativos para con cada uno de los otros como absolutos, en la última columna. Vemos, por ejemplo, que $\gamma(\text{ROS,CLAR}) = 6$, ya que Rosaura y Clarín aparecen juntos en seis escenas o $\gamma(\text{ROS,CLOT}) = 4$, que es el número de veces que Rosaura comparte escenario con Clotaldo en esta jornada.

x	y										$\gamma(x)$	$\lambda(x)$
	Ros.	Cla.	Seg.	Clo.	Sol.	Ast.	Est.	Dam.	Bas.	Aco.		
Ros.	-	6	2	4	2	0	0	0	1	0	15	21
Clar.	6	-	2	4	2	0	0	0	1	0	15	21
Segis.	2	2	-	1	1	0	0	0	0	0	6	8
Clot.	4	4	1	-	2	0	0	0	1	0	12	16
Sol.	2	2	1	2	-	2	2	2	1	1	15	19
Ast.	0	0	0	0	2	-	2	2	1	1	8	10
Est.	0	0	0	0	2	2	-	2	1	1	8	10
Damas	0	0	0	0	2	2	2	-	1	1	8	10
Bas.	1	1	0	1	1	1	1	1	-	1	8	10
Acom.	0	0	0	0	1	1	1	1	1	-	6	7

Cuadro D.3: Copresencia escénica.

Este marcador está relacionado con lo que en la teoría de la información se denomina *distancia de Hamming*. Se define esta como el número de posiciones con valores divergentes que diferencian dos palabras válidas en un código dado. Por ejemplo, la distancia de la palabra *casa* respecto a *cal**a*** es 1, con ***p**ala*** es 2, con ***p**alo*** es 3 y con ***pelo*** es 4. Análogamente, definimos un código binario de palabras de tamaño igual al número de escenas de la obra y con tantas palabras como personajes intervienen en ella. Para hacerlo, retornemos una vez más a la primera jornada de *La vida es sueño*, cuyos valores recogemos en Cuadro D.4[137].

	VIII	VII	VI	V	IV	III	II	I
Rosaura	1	1	0	0	1	1	1	1
Clarín	1	1	0	0	1	1	1	1
Segismundo	0	0	0	0	0	1	1	0
Clotaldo	1	1	0	0	1	1	0	0
Soldados	0	0	1	1	1	1	0	0
Astolfo	0	0	1	1	0	0	0	0
Estrella	0	0	1	1	0	0	0	0
Damas	0	0	1	1	0	0	0	0
Basilio	0	1	1	0	0	0	0	0
Acompañamiento	0	0	1	0	0	0	0	0

Cuadro D.4: Presencia escénica.

Como medida para calcular la distancia entre dos personajes, Marcus propone hallar primero la *diferencia simétrica* de la distancia que separa ambos conjuntos. Definimos esta operación para A y B como $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Esto es, la unión de los elementos que A no comparte con B y viceversa. Para nuestro propósito, esto corresponde al conjunto de escenas en las que aparece uno u otro personaje, pero no

[137]La disposición en orden inverso de las escenas en el cuadro es arbitraria. Obedece a un propósito ilustrativo, a saber: que la caracterización binaria de cada personaje imite a la convención matemática del valor posicional de los dígitos y facilitar así la analogía.

los dos juntos. La distancia de Hamming será el cardinal del conjunto resultante de aplicar la diferencia simétrica, lo que denotamos formalmente según la ecuación D.4.

$$h(A,B) = |A \triangle B| \tag{D.4}$$

Esta medida por sí misma resulta un tanto confusa. Por ejemplo, Segismundo mantiene la equidistancia con el resto de personajes, con excepción del acompañamiento de Basilio (Cuadro D.5). De las seis escenas de Rosaura, dos las comparte con Segismundo, por lo que su distancia de Hamming es 4; lo mismo que con Astolfo, pues cada uno tiene dos escenas en las que no coinciden. Sin embargo, conservaremos esta medida, pues va a resultar de gran utilidad en los cálculos que siguen.

	ROS.	CLA.	SEG.	CLO.	SOL.	AST.	EST.	DAM.	BAS.	ACO.
ROS.	-	0	4	2	6	8	8	8	6	7
CLAR.	0	-	4	2	6	8	8	8	6	7
SEGIS.	4	4	-	4	4	4	4	4	4	3
CLOT.	2	2	4	-	4	6	6	6	4	5
SOL.	6	6	4	4	-	2	2	2	4	3
AST.	8	8	4	6	2	-	0	0	2	1
EST.	8	8	4	6	2	0	-	0	2	1
DAMAS	8	8	4	6	2	0	0	-	2	1
BAS.	6	6	4	4	4	2	2	2	-	1
ACOM.	7	7	3	5	3	1	1	1	1	-

Cuadro D.5: Distancia de Hamming.

En efecto, si emprendemos la aproximación contraria, esto es, calcular no el alejamiento, sino el grado de proximidad, la distancia de Hamming demuestra su utilidad. La proximidad entre los personajes x e y es la diferencia entre el número de escenas en las que aparecen uno o ambos y la distancia de Hamming que los separa (D.5).

$$\gamma(x,y) = |Ax \cup Ay| - h(x,y) \tag{D.5}$$

Hemos llegado por otro camino al resultado que obteníamos observando la *copresencia escénica* (Cuadro D.3), lo que permite hilar más fino al comparar los personajes. Si bien esto parece un asunto trivial, en la práctica supone dos aproximaciones diferentes. La copresencia demanda evaluar la presencia escénica de ambos personajes de forma concurrente, mientras que esta nueva forma implica hallar sucesivamente los valores individuales de cada personaje y hacer el cálculo al final. En cualquier caso, si volvemos a los resultados relativos que vistos allí, destaca, por un lado, que Rosaura y Clarín sean los personajes más próximos. Pero tal vez llama más la atención la

equidistancia que mantiene Basilio para con todos los personajes excepto uno, que no es otro que su hijo Segismundo.

Densidad

Otra medida que propone Marcus es la *densidad de la escena*, esto es, el número de personajes que intervienen en ella. Para nuestra jornada, $\alpha_{\text{I}} = 2$, $\alpha_{\text{II}} = 3$, $\alpha_{\text{III}} = 6$, $\alpha_{\text{IV}} = 5$, $\alpha_{\text{V}} = 6$, $\alpha_{\text{VI}} = 9$, $\alpha_{\text{VII}} = 4$ y $\alpha_{\text{VIII}} = 3$, si asumimos —por conveniencia— que cada personaje colectivo está interpretado por dos actores. Los cambios cuantitativos en las configuraciones escénicas se obtienen aplicando la fórmula de la ecuación D.6, en la que μ representa la media de personajes por escena.

$$\beta(n) = |\mu - \alpha(n)| \tag{D.6}$$

Si tomamos las consideraciones hechas para los personajes colectivos, $\mu = 4{,}75$, por lo que, por ejemplo, $\beta_{\text{I}} = 2{,}75$ y $\beta_{\text{VI}} = 4{,}25$, es obligado considerar otro aspecto. Pensemos, por ejemplo, en un monólogo: dado que el sumatorio para calcular γ excluye la copresencia de un personaje consigo mismo —que ocurre en todas las escenas en las que interviene—, una escena monologada no tendría influencia. Sin embargo, sabemos que este tipo de escenas —que denotaremos como $\alpha'(x)$—, lejos de ser secundarias, tienden a realzar el protagonismo del personaje que interviene. Marcus propone una nueva medida (D.7) como respuesta. En este caso, esta incluso corrige nuestro desliz con los soldados a la hora de establecer los personajes con más relevancia escénica (Cuadro D.3).

$$\lambda(x) = \sum_{y} \gamma(x,y) \tag{D.7}$$

Aproximación probabilística

Una manera diferente de caracterizar elementos de la obra teatral es mediante expresiones probabilísticas. Así, Dinu (1968) sugiere partir de la frecuencia de un personaje X dado $f_i = \frac{n_i}{N}$, donde N es el número total de escenas y n_i el número de escenas en las que interviene el personaje x. Un número de escenas suficiente da ocasión de calcular la probabilidad $P(X_i)$ de que el personaje x_i intervenga en una escena aleatoria dada. Además, sabemos que esta probabilidad se aproxima a la frecuencia de las intervenciones de dicho personaje (D.8).

$$P_i = P(X_i) \approx f_i \tag{D.8}$$

Extendemos esto a la copresencia de dos personajes X_i y X_j. La probabilidad P_{ij} de que ambos personajes coincidan en una escena es el producto de las probabilidades de que aparezca en escena cada uno de ellos (D.9).

$$P_{ij} = P(X_i \cap X_j) = P(X_i).P(X_j) \approx f_i \cdot f_j = \frac{n_i n_j}{N^2} \tag{D.9}$$

De esto se deduce el número de escenas en las que los personajes X_i y X_j coincidirán, según la ecuación D.10.

$$\bar{n}_{ij} = N \cdot P_{ij} = N \cdot P_i \cdot P_j \approx N \cdot f_i \cdot f_j = \frac{n_i n_j}{N} \tag{D.10}$$

Hemos recogido en Cuadro D.6 los valores correspondientes a la primera jornada de *La vida es sueño*. Por ejemplo, según esta, la probabilidad de que en una escena al azar aparezca Rosaura es del 75 %, mientras que la probabilidad de encontrar al acompañamiento de Basilio apenas alcanza el 12,5 %. Según la estadística, encontrarnos a Rosaura y Clarín juntos debería ser una cuestión de cara o cruz, lo que, como sabemos, no es así.

X_i	P_i	X_j									
		Ros.	Cla.	Seg.	Clo.	Sol.	Ast.	Est.	Dam.	Bas.	Aco.
Ros.	0,750	-	0,563	0,188	0,375	0,375	0,188	0,188	0,188	0,188	0,094
Clar.	0,750	0,563	-	0,188	0,375	0,375	0,188	0,188	0,188	0,188	0,094
Segis.	0,250	0,188	0,188	-	0,125	0,125	0,063	0,063	0,063	0,063	0,031
Clot.	0,500	0,375	0,375	0,125	-	0,250	0,125	0,125	0,125	0,125	0,063
Sol.	0,500	0,375	0,375	0,125	0,250	-	0,125	0,125	0,125	0,125	0,063
Ast.	0,250	0,188	0,188	0,063	0,125	0,125	-	0,063	0,063	0,063	0,031
Est.	0,250	0,188	0,188	0,063	0,125	0,125	0,063	-	0,063	0,063	0,031
Damas	0,250	0,188	0,188	0,063	0,125	0,125	0,063	0,063	-	0,063	0,031
Bas.	0,250	0,188	0,188	0,063	0,125	0,125	0,063	0,063	0,063	-	0,031
Acom.	0,125	0,094	0,094	0,031	0,063	0,063	0,031	0,031	0,031	0,031	-

Cuadro D.6: Probabilidad P_{ij} de copresencia.

Para reflejar que $\bar{n}_{ij}$ difiere del número real n de escenas en las que coinciden los personajes, se define otra medida. Mediante la comparación de ambos valores se estima en qué grado difiere la copresencia de dos personajes de lo que sería esperable, lo que arroja información sobre su relación escénica. Para esto, hay que hallar un valor que describa la diferencia entre copresencia real y esperada de los personajes X_i y X_j, al que llamaremos λ_{ij} (D.11)

$$\lambda_{ij} = n_{ij} - \bar{n}_{ij} \tag{D.11}$$

Extraemos una medida en términos absolutos resolviendo la ecuación D.11 de un personaje respecto a todos los demás y sumamos sus valores. Lo expresamos con la ecuación D.12.

$$\sigma_i = \sum_{i \neq i} \lambda_{ij} \tag{D.12}$$

Si $\lambda < 0$, los personajes concurren por debajo de lo predecible, lo que da pie a sospechar que el poeta evita su coincidencia por alguna razón. Lo contrario, si $\lambda > 0$, bien podría ser indicio de que la pareja tiene una relación especial. Ilustrémoslo volviendo a nuestro ejemplo. Rosaura y Clarín coinciden en seis escenas, pero, atendiendo a lo anterior, $P_{\text{Ros,Clar}} = 4{,}5$, lo que pone de relieve su cercanía ($\lambda_{\text{Ros,Clar}} > 0$). Por el contrario, si comprobamos lo esperado de Rosaura y Astolfo ($P_{\text{Ros,Ast}} = 1{,}5$) con lo que realmente sucede ($\alpha_{\text{Ros,Ast}} = 0$), verificamos su distancia ($\lambda_{\text{Ros,Clar}} < 0$). Solo en el caso de Clotaldo y Segismundo se cumple $\alpha_{ij} = P_{ij}$ ($\lambda_{\text{Ros,Clar}} = 0$), lo que apunta a un posicionamiento escénico neutro de un personaje respecto al otro. El valor de σ da una idea de la implicación de un personaje en diferentes planos del conflicto, de manera que valores bajos sugieren una tendencia unilateral mientras que los valores altos apuntarían a una relación escénica presente en planos de conflicto.

X_i	X_j										σ_i
	Ros.	Cla.	Seg.	Clo.	Sol.	Ast.	Est.	Dam.	Bas.	Aco.	
Ros.	-	1,5	0,5	1	-1	-1,5	-1,5	-1,5	-0,5	-0,75	-3,75
Clar.	1,5	-	0,5	1	1	-1,5	-1,5	-1,5	-0,5	-0,75	-3,75
Segis.	0,5	0,5	-	0	0	-0,5	-0,5	-0,5	-0,5	-0,25	-1,25
Clot.	1	1	0	-	0	-1	-1	-1	0	-0,5	-1,5
Sol.	-1	-1	0	0	-	1	1	1	0	0,5	1,5
Ast.	-1,5	-1,5	-0,5	-1	1	-	1,5	1,5	0,5	0,75	0,75
Est.	-1,5	-1,5	-0,5	-1	1	1,5	-	1,5	0,5	0,75	0,75
Damas	-1,5	-1,5	-0,5	-1	1	1,5	1,5	-	0,5	0,75	0,75
Bas.	-0,5	-0,5	-0,5	0	0	0,5	0,5	0,5	-	0,75	0,75
Acom.	-0,75	-0,75	-0,125	-0,5	0,5	0,75	0,75	0,75	0,75	-	1,375

Cuadro D.7: Valores λ_{ij}.

Lo visto ahora con pares de personajes se traslada a configuraciones escénicas complejas. Definimos $\varphi_i = f_i$ si $X_i \in M_s$ o $\varphi_i = 1 - f_i$ en caso contrario. Llamamos p al número total de personajes y M_s al conjunto de todos los que aparecen en la escena s.

$$P_s \approx \prod_{i=1}^{p} \varphi_i \tag{D.13}$$

Volvamos a nuestro ejemplo de *La vida es sueño* para ilustrar estas propuestas. A partir de una matriz de instancias de presencia escénica, como la de Figura 7.1, se establece la dominancia de los personajes. Hasta aquí, habíamos considerado la matriz como si fuera booleana, con dos únicos valores posibles para cada celda: presente y ausente. Ahora introduciremos algunos matices. Además de considerar el tiempo en que los personajes están sobre el escenario, diferenciaremos los momentos en los que habla el propio personaje de aquellos en los que es otro el que pronuncia sus líneas. De esta manera, representamos la preponderancia del personaje dentro de la escena además de su presencia.

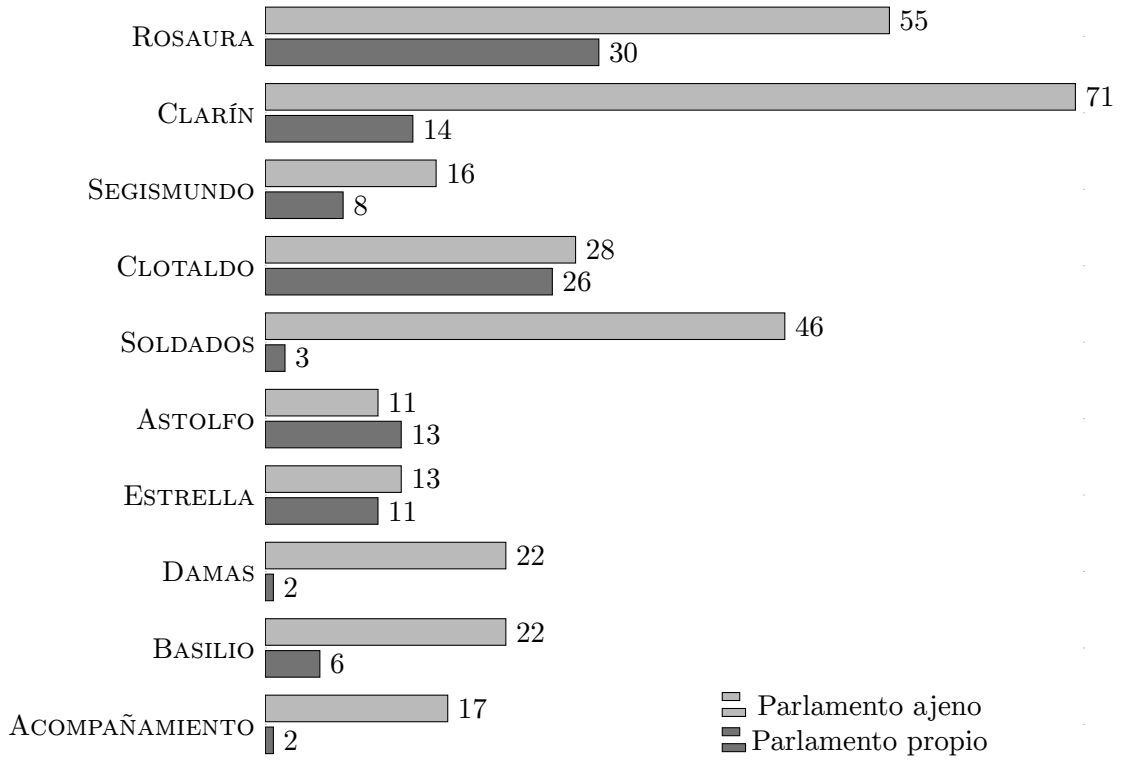


Figura D.2: Dominancia en la primera jornada.

Nos hacemos una mejor idea si comparamos, por ejemplo, la relación entre el tiempo en que un personaje tiene la palabra respecto a los parlamentos que pasa en escena mientras hablan otros personajes (Figura D.2). Cabe aquí preguntarse si todos los

parlamentos tienen el mismo valor, pues cuantos más personajes pronuncien una línea, más se diluye la importancia que esta confiere a quien la declama. Lo apreciamos al final de la séptima escena, cuando todos los presentes piden primero a Basilio un heredero y, después, aclaman al monarca. En este caso, quien analice el texto debe tomar una decisión e incluir esos versos sin más, excluirlos del recuento o darles un valor ponderado. El texto procesado debe proporcionar la información para identificar estos parlamentos para actuar de una manera u otra.

Podemos ir más allá de considerar la presencia y dominancia escénica absoluta de un personaje si lo relacionamos con otros personajes, de manera que se establezca una red de asociaciones entre ellos. Aquí, por ejemplo, parece claro que se repiten las relaciones que habíamos observado al analizar por escena, algo que revela una fuerte relación entre Rosaura y Clarín, lo que ya habíamos visto antes al ver las relaciones por escena.

	Ros.	Clar.	Segis.	Clot.	Sol.	Ast.	Estr.	Damas	Bas.	Acom.
Ros.	-	95	23	52	25	0	0	0	9	0
Clar.	95	-	23	52	28	0	0	0	9	0
Segis.	23	23	-	10	10	0	0	0	0	0
Clot.	52	52	10	-	25	0	0	0	9	0
Sol.	25	25	10	25	-	24	24	24	19	19
Ast.	0	0	0	0	24	-	24	24	19	19
Estr.	0	0	0	0	24	24	-	24	19	19
Damas	0	0	0	0	24	24	24	-	19	19
Bas.	9	9	0	9	19	19	19	19	-	19
Acom.	0	0	0	0	19	19	19	19	19	-

Cuadro D.8: Copresencia escénica por parlamento.

Si descendemos hasta el nivel de los parlamentos, es posible tener en cuenta distintas configuraciones. Por ejemplo, hasta que Clotaldo entra con los soldados en la tercera escena, hay tres parlamentos, uno de ellos recitado desde dentro. No consideraremos que estos están presentes y en silencio en ese intervalo, pero sí que Rosaura, Clarín y Segismundo callan mientras Clotaldo habla desde dentro.

A la vista de esto, pensemos en cuántas de las anteriores tareas podría delegar el estudioso en el computador y el tiempo que eso le ganaría para concentrarse en la interpretación de los resultados.

Bibliografía

Fuentes primarias

Calderón de la Barca, P. (1636a). La vida es sueño. En *Parte treynta de comedias famosas de varios autores* (1r-26v). María de Quiñones.

Calderón de la Barca, P. (1636b). La vida es sueño. En *Parte treynta de comedias famosas de varios autores* (pp. 127-173). Hospital Real y General de Nuestra Señora de Gracia.

Madrid: Biblioteca Nacional de España, *RES/117*. 1642. https : / / www . cervantesvirtual.com/nd/ark:/59851/bmcsj209

Calderón de la Barca, P. (1685). La vida es sueño. En J. de Vera Tassis (Ed.), *Primera parte de comedias* (pp. 1-49). Francisco Sanz.

Calderón de la Barca, P. (1795). *El secreto a voces*. Librería de Quiroga. https: //www.cervantesvirtual.com/nd/ark:/59851/bmck07q5

Calderón de la Barca, P. (1848). La vida es sueño. En J. E. Hartzenbusch (Ed.), *Comedias de don Pedro Calderón de la Barca* (pp. 1-19). M. Rivadeneyra. (Fecha inicial de publicación 1636)

Calderón de la Barca, P. (1861). *Bien vengas, mal si vienes solo* (Á. M. Dacarrete, Ed.). José Cuesta. (Fecha inicial de publicación 1635)

Calderón de la Barca, P. (1970). *Amar después de la muerte o el Tuzaní de las Alpujarras* (M. Zugasti, Ed.). Espasa Calpe. (Fecha inicial de publicación 1676)

Calderón de la Barca, P. (1983). El toreador [Entremés]. En E. Rodríguez y A. Tordera (Eds.). Castalia.

Calderón de la Barca, P. (1989). *La fiera, el rayo y la piedra* (A. Egido, Ed.). Cátedra. (Fecha inicial de publicación 1664)

Calderón de la Barca, P. (1992). *La primera versión de* La vida es sueño (J. M. Ruano de la Haza, Ed.). Liverpool University Press. (Fecha inicial de publicación 1924)

Calderón de la Barca, P. (1995a). *El año sancto de Roma* (I. Arellano y Á. L. Cilveti, Eds.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1650)

Calderón de la Barca, P. (1995b). *El nuevo hospicio de pobres* (I. Arellano, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1668)

Calderón de la Barca, P. (1997a). *El primer blasón de Austria* (V. Roncero, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1634-1635)

Calderón de la Barca, P. (1997b). *La vida es sueño* (E. Rodríguez Cuadros, Ed.; 18.ª ed.). Espasa-Calpe. (Fecha inicial de publicación 1636)

Calderón de la Barca, P. (1997c). *Sueños hay que verdad son* (M. D. McGaha, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1670)

Calderón de la Barca, P. (1999a). *El diablo mudo* (C. C. García Valdés, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1660)

Calderón de la Barca, P. (1999b). *El divino Orfeo* (J. E. Duarte, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1634)

Calderón de la Barca, P. (1999c). *A María el corazón* (I. Arellano, I. Adeva, F. Crosas y M. Zugasti, Eds.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1664)

Calderón de la Barca, P. (2000). *El veneno y la triaca* (J. M. Escudero, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación ca. 1634)

Calderón de la Barca, P. (2005a). *El verdadero Dios Pan* (F. Antonucci, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1670)

Calderón de la Barca, P. (2005b). *La vacante general* (P. Ibáñez, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1647)

Calderón de la Barca, P. (2006a). *El sacro Pernaso* (A. Rodríguez-Rípodas, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1659)

Calderón de la Barca, P. (2006b). *Las espigas de Ruth* (C. Buezo, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1663)

Calderón de la Barca, P. (2007). *El viático cordero* (J. M. Escudero, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1665)

Calderón de la Barca, P. (2009). *Los alimentos del hombre* (M. Zugasti, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1676)

Calderón de la Barca, P. (2010a). ¿Cuál es mayor perfección, hermosura o discreción? En J. M. Ruano de la Haza (Ed.), *Verdadera quinta parte de comedias* (pp. 127-173). Biblioteca Castro. (Fecha inicial de publicación ca. 1650)

Calderón de la Barca, P. (2010b). Eco y Narciso. En S. Neumeister. (Ed.), *Cuarta parte de Comedias* (pp. 129-226). Fundación José Antonio de Castro. (Fecha inicial de publicación 1661)

Calderón de la Barca, P. (2010c). Hado y divisa de Leonido y Marfisa. En J. M. Ruano de la Haza (Ed.), *Verdadera quinta parte de comedias* (pp. 127-173). Biblioteca Castro. (Fecha inicial de publicación 1680)

Calderón de la Barca, P. (2011). *El pintor de su deshonra* (A. K. Paterson, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1644-1646)

Calderón de la Barca, P. (2012a). *Judas Macabeo* (F. Rodríguez Gallego, Ed.). En F. Rodríguez Gallego (Ed.), *Parte treynta de comedias famosas de varios autores*. Universidad de Navarra; Iberoamericana; Vervuert. (Fecha inicial de publicación 1682)

Calderón de la Barca, P. (2012b). La vida es sueño. Primera versión. En F. Plata Parga (Ed.), La vida es sueño*, edición crítica de las dos versiones del auto y de la loa*. Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1924)

Calderón de la Barca, P. (2013a). *La cena del rey Baltasar* (A. Sánchez Jiménez y A. J. Sáez, Eds.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1635)

Calderón de la Barca, P. (2013b). *La redención de cautivos* (M. Trambaioli, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1670-1677)

Calderón de la Barca, P. (2014a). *La cura y la enfermedad* (I. Arellano y E. Reichenberger, Eds.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1647-1658)

Calderón de la Barca, P. (2014b). *A secreto agravio, secreta venganza* (S. Fernández Mosquera, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1637)

Calderón de la Barca, P. (2016a). *El santo rey don Fernando (segunda parte)* (C. Pinillos, Ed.). Universidad de Navarra; Edition Reichenberger. (Fecha inicial de publicación 1671)

Calderón de la Barca, P. (2016b). *La banda y la flor* (J. Castro Rivas, Ed.). Universidad de Navarra; Iberoamericana; Vervuert. (Fecha inicial de publicación 1668)

Calderón de la Barca, P. (2019). *La vida es sueño* (C. Morón, Ed.; 42.ª ed.). Cátedra. (Fecha inicial de publicación 1636)

Calderón de la Barca, P. (2021). *El gran teatro del mundo* (I. Arellano, Ed.). Universidad de Navarra. (Fecha inicial de publicación 1655)

Calderón de la Barca, P. (2022). *Los empeños de un acaso* (S. Kroll, Ed.). Edition Reichenberger. (Fecha inicial de publicación 1661)

Calderón de la Barca, P. (en prep.). *El gran príncipe de Fez* (F. Rodríguez Gallego, Ed.). (Fecha inicial de publicación 1669)

Calderón de la Barca, P. (en prensa). *Eco y Narciso* (M. Greer, F. Rodríguez-Gallego y S. Kroll, Eds.). Edition Reichenberger. (Fecha inicial de publicación 1661)

Cáncer y Velasco, J., Moreto, A. y Matos Fragoso, J. de. (2019). El hijo pródigo. En D. Gavela (Ed.), *Comedias de Agustín Moreto. Obras escritas en colaboración.* Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1750)

Cervantes Saavedra, M. de. (1997). *El rufián dichoso* (F. Sevilla Arroyo, Ed.; 20.ª ed.). Castalia. (Fecha inicial de publicación 1615)

Cervantes Saavedra, M. de. (2003). *La entretenida* (R. Schevill y A. Bonilla, Eds.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación ca. 1607)

Cervantes Saavedra, M. de. (2004). *Los trabajos de Persiles y Segismunda* (C. Romero Muñoz, Ed.; 14.ª ed.). Cátedra. (Fecha inicial de publicación 1617)

Cervantes Saavedra, M. de. (2016a). *El ingenioso hidalgo don Quijote de la Mancha* (J. J. Allen, Ed.; 35.ª ed., Vol. 1). Cátedra. (Fecha inicial de publicación 1605)

Cervantes Saavedra, M. de. (2016b). Prólogo al lector. En N. Spadaccini (Ed.), *Entremeses* (20.ª ed., pp. 93-96). Cátedra. (Fecha inicial de publicación 1615)

Cueva, J. de la. (1924). Los siete infantes de Lara. En F. A. de Icaza (Ed.), El infamador*,* Los siete infantes de Lara *y* Ejemplar poético (pp. 131-184). Ediciones de la Lectura. (Fecha inicial de publicación 1579)

Cueva, J. de la. (1997). La muerte del rey don Sancho y reto de Zamora. En J. Matas Caballero (Ed.), *La muerte del rey don Sancho y reto de Zamora. Comedia del degollado* (pp. 145-216). Universidad de León. (Fecha inicial de publicación 1618)

Enríquez Gómez, A. (2019). Engañar para reinar. En A. García González y A. Gutiérrez Gil (Eds.), *Comedias de Antonio Enríquez Gómez* (pp. 39-149). Universidad de Castilla-La Mancha.

Góngora y Argote, L. de. (1966). *Obras completas de Luis de Góngora y Argote* (J. Millé y Giménez e I. Millé y Giménez, Eds.). Aguilar.

Lehmann, J., Milling, C. y Fischer, F. (2022, 10 de junio). *CalDraCor (Calderón Drama Corpus)*. https://github.com/dracor-org/caldracor

Mira de Amescua, A. (2007a). *La casa de Austria* (A. de la Granja, Ed.). Universidad de Granada/Diputación de Granada. (Fecha inicial de publicación ca. 1617)

Mira de Amescua, A. (2007b). *Los pastores de Belén* (A. M. Martín Conteras, Ed.). Universidad de Granada/Diputación de Granada.

Molina, T. de. (2003a). La peña de Francia. En L. Vázquez (Org.), *Obras completas. Cuarta parte de comedias* (pp. 509-616). Instituto de Estudios Tirsianos. (Fecha inicial de publicación 1635)

Molina, T. de. (2003b). Santo y sastre. En J. Garau (Org.), *Obras completas. Cuarta parte de comedias* (pp. 633-738). Instituto de Estudios Tirsianos. (Fecha inicial de publicación 1635)

Molina, T. de. (2014). *El condenado por desconfiado* (L. Galván, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1626)

Moreto, A. (2016). No puede ser el guardar una mujer. En M. L. Lobato y M. Ortega (Eds.), *Comedias de Agustín Moreto. Segunda parte de comedias* (pp. 47-246). Edition Reichenberger. (Fecha inicial de publicación 1659)

Moreto, A. (2018). *La cena del rey Baltasar* (A. Vara, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1648)

Oleza Simó, J., Ferrer Valls, T., Miñano Martínez, E., Calvo Rigual, C., Tronch Pérez, J., Calvo García de Leonardo, J. J., Teruel Pozas, M., Badía Herrera, J., Souto, L. C., Martínez Rubio, J., Revenga, N., Barreda Villafranca, C., Burgos Segarra, G., Martínez Fernández, Á., Ros Ferrer, V. y Nuckols, A. (2022). *Base de Datos y Argumentos del Teatro de Lope de Vega. ARTELOPE*. https://artelope.uv.es/

Rey de Artieda, A. (2014). *Los amantes* (T. Ferrer Valls, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1581)

Rojas Zorrilla, F. de. (2009). *Obligados y ofendidos* (F. Pedraza y R. González Cañal, Eds.). Ediciones de la Universidad de Castilla-La Mancha. (Fecha inicial de publicación ca. 1635)

Rojas Zorrilla, F. de. (2014). Lo que son mujeres. En F. Pedraza y R. González Cañal (Eds.), *Obras completas IV. Segunda parte de comedias* (pp. 55-167). Ediciones de la Universidad de Castilla-La Mancha. (Fecha inicial de publicación 1645)

Schiller, F. (2001). *Die Räuber* (C. Grawe, Ed.). Reclam. (Fecha inicial de publicación 1781)

TC/12 Red del Patrimonio Teatral Clásico Español. (2014). *Canon 60*. https://www.cervantesvirtual.com/obra/canon-60-la-coleccion-esencial-del-tc12-teatro-clasico-espanol/

Madrid: Biblioteca Nacional de España, *VITR/7/5*. 1613. https://www.cervantesvirtual.com/nd/ark:/59851/bmc1j9v2

Vega Carpio, L. de. (1617). La dama boba. En *Doze comedias de Lope de Vega sacadas de sus originales por el mismo* (pp. 256-275). Viuda de Alonso Martín de Balboa, a costa de Alonso Pérez. https://www.cervantesvirtual.com/nd/ark:/59851/bmcxp7j7

Vega Carpio, L. de. (1982). *La discreta enamorada* [A. Guirau (dir).]. http://teatroteca.teatro.es/opac?id=00006142

Vega Carpio, L. de. (1992). *El caballero de Olmedo* (R. Maestre, Ed.). Aguaclara. (Fecha inicial de publicación ca. 1620)

Vega Carpio, L. de. (1996). *La Dorotea* (J. M. Blecua, Ed.). Cátedra. (Fecha inicial de publicación 1632)

Vega Carpio, L. de. (1998). *Fuente Ovejuna* (R. Froldi, Ed.; 19.ª ed.). Espasa-Calpe. (Fecha inicial de publicación 1619)

Vega Carpio, L. de. (2010a). *El perro del hortelano* (A. Carreño, Ed.). Espasa-Calpe. (Fecha inicial de publicación 1618)

Vega Carpio, L. de. (2010b). *Poder vencido y amor premiado* (J. Checa, Ed.). En J. Checa (Ed.). R. Valdés Gázquez y M. Morrás (Coords.), *Comedias de Lope de Vega. Parte X*. Milenio/Universitat Autònoma de Barcelona. (Fecha inicial de publicación 1618)

Vega Carpio, L. de. (2014a). *El anzuelo de Fenisa* (L. Gómez Canseco, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación 1617)

Vega Carpio, L. de. (2014b). *El remedio en la desdicha* (R. Ippolito, Ed.). En R. Ippolito (Ed.). N. Fernández (Org.), *Comedias. Parte XIII*. Gredos. (Fecha inicial de publicación 1604)

Vega Carpio, L. de. (2014c). Los locos de Valencia. En C. Peña López (Ed.), *Comedias de Lope de Vega. Parte XIII* (pp. 167-328). Gredos. (Fecha inicial de publicación 1590-1595)

Vega Carpio, L. de. (2015a). *El villano en su rincón* (G. Serés, Ed.). Biblioteca Virtual Miguel de Cervantes. (Fecha inicial de publicación ca. 1614)

Vega Carpio, L. de. (2015b). *La dama boba* (M. Presotto, Ed.). Prolope; Alma Mater Studiorum - Università di Bologna. https://doi.org/10.6092/UNIBO/LADAMABOBA (Fecha inicial de publicación 1613)

Vega Carpio, L. de. (2019). *El castigo sin venganza* (14.ª ed.). Cátedra. (Fecha inicial de publicación 1634)

Vega Carpio, L. de. (2023). *La francesa Laura* (Á. Cuéllar y G. Vega García-Luengos, Eds.). Gredos.

Fuentes secundarias

Agirrezabal, M., Alegría, I. y Hulden, M. (2017). A comparison of feature-based and neural scansion of poetry. *RANLP 2017 – Recent Advances in Natural Language Processing*, 18-23. https://doi.org/10.26615/978-954-452-049-6_003

Aichinger, W. (2012). Imagen, imagen soñada y romance en Calderón. Algunas consideraciones generales. *Iberoromania*, *75-76*, 121-141. https://doi.org/10.1515/ibero-2012-0023

Aichinger, W. (2013). *Pintura viva. Einführung in die Bedeutung, Funktionen und Ausdrucksmittel literarischer Texte*. Turia + Kant.

Aichinger, W. (2015). Introducción a *El secreto a voces*. Trama, artificio, momento histórico. En W. Aichinger, S. Kroll y F. Rodríguez-Gallego (Eds.), *Pedro Calderón de la Barca,* El secreto a voces (pp. 1-146). Edition Reichenberger.

Aichinger, W. (2023). Los bautizados de socorro de Pedro Bernardo (Ávila). Un momento de transición en el registro de la muerte neonatal. *Memoria Y Civilización*, *26*(1), 231-253. https://doi.org/10.15581/001.26.010

Alarcos Llorach, E. (1964). *Fonología española* (4.ª ed.). Gredos.

Andrews, T. L. (2013). The third way: Philology and critical edition in the Digital Age. *Variants*, *10*, 61-76. https://doi.org/10.7892/boris.43071

Antonucci, F. (2010). La segmentación métrica, estado actual de la cuestión. *Teatro de palabras. Revista sobre teatro áureo*, *4*, 77-97. https://www.uqtr.ca/teatro/teapal/TeaPalNum04Rep/TeaPal04Antonucci.pdf

Antonucci, F. (2017). Escena, cena, paso: las particiones internas al acto o jornada en textos teatrales y teóricos del Siglo de Oro. En A. Cayuela y M. Vuillermoz (Eds.), *Les mots et les choses du théâtre. France-Italie-Espagne, XVI^{e}-XVII^{e} siècles* (pp. 111-121). Droz.

Antonucci, F. (2018). Una nueva herramienta para el estudio del teatro clásico español: Calderón Digital. Base de datos, argumentos y motivos del teatro de Calderón. *Bulletin of the Comediantes*, *70*(1), 79-95. https://doi.org/10.1353/boc.2018.0004

Antonucci, F. (2023). Propuestas para una reconsideración del conjunto de la polimetría calderoniana. En A. Sánchez Jiménez, C. López Lorenzo, A. J. Sáez y S. J. A. (Eds.), *La actualidad de los estudios del Siglo de Oro* (pp. 3-31). Edition Reichenberger. https://doi.org/10.59010/9783967280494_001

Arellano, I. (2007). *Editar a Calderón*. Iberoamericana; Vervuert.

Aristóteles. (2010). Poética. En V. García Yebra (Ed. & Trad.), *Poética de Aristóteles. Edición trilingüe* (V. García Yebra, Ed. & Trad.; pp. 125-239). Gredos. (Fecha inicial de publicación 336 a.e.c.)

AuTeSO. (2023). Consultado el 17 de julio de 2023, desde http://theatheor-fe.netseven.it/

Baehr, R. (1997). *Manual de versificación española* (K. Wagner y F. López Estrada, Trad.; Vol. 25). Gredos. (Fecha inicial de publicación 1973)

Balbín Lucas, R. de. (1968). *Sistema de métrica castellana* (2.ª ed.). Gredos.

Bay-Cheng, S. (2017). Digital historiography and performance. *Theatre Journal*, *68*(4), 507-527. https://doi.org/10.1353/tj.2016.0104

Bello, A. (1981). *Estudios filológicos. Vol. 1. Principios de la ortología y métrica de la lengua castellana y otros escritos* (S. Gili Gaya, Ed.; *Vol. 1*). (Fecha inicial de publicación 1835)

Benjamin, W. (1978). *Ursprung des deutschen Trauerspiels* (R. Tiedemann, Ed.). Suhrkamp. (Fecha inicial de publicación 1928)

Benjamin, W. (2013). *Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit* (3.ª ed.). Suhrkamp. (Fecha inicial de publicación 1935)

Bernstein, J. B. (2001). The DP hypothesis: Identifying clausal properties in the nominal domain. En M. Baltin y C. Colins (Eds.), *The handbook of contemporary syntactic theory* (pp. 536-561). Blackwell.

Berry, D. M. (2011a). The computational turn: Thinking about digital humanities. *Culture Machine*, *12*, 1-22. https://doi.org/10.2337/DB11-0751

Berry, D. M. (Ed.). (2011b). *Understanding the digital humanities*. Palgrave.

Berry-Rogghe, G. L. M. y Crawford, T. D. (1973). Developing a machine-independent concordance program for a variety of languages. En A. J. Aitken, R. W. Bailey y N. Hamilton-Smith (Eds.), *The computer and literary studies* (pp. 309-16). Edinburgh University Press.

Bettels, J. y Bishop, E. a. (1993). Unicode: A universal character code. *Digital Technical Journal*, *5*(3), 21-31.

Bidwell-Steiner, M. (2013). Bespoke Spanish passions: Calderon's *Médico de su honra* against the backdrop of Early Modern doctrines of affect. *Zeitsprünge. Forschungen zur Frühen Neuzeit*, *17*(4), 433-452.

Bird, S., Klein, E. y Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Blecua, A. (2001). *Manual de crítica textual*. Castalia. (Fecha inicial de publicación 1983)

Bloom, H. (1995). *The Western Canon. The books and school of the ages*. Riverhead Books.

Boersma, P. y Weenink, D. (2022). *Praat: Doing phonetics by computer* (Ver. 6.2.21). http://www.praat.org/

Booth, A. D. (1958). Aspects of translation. En *Studies in communication* (pp. 88-104). Secker-Warburg.

Brackett, M. H. (2014). *Data resource data: A comprehensive data resource understanding*. Technics Publications.

Bradley, R. (2004). *The ultimate computing glossary for advanced level*. Nelson Thornes.

Brandwood, L. (1956). Analysing Plato's style with an electronic computer. *Bulletin of the Institute of Classical Studies*, *3*(1), 45-54. https://doi.org/10.1111/j.2041-5370.1956.tb00155.x

Burdick, A., Drucker, J., Lunenfeld, P., Presner, T. y Schnapp, J. (2012). *Digital_Humanities*. The MIT Press.

Bureau of the Budget. (1962). *Automatic data processing. Glossary*. Executive Office of the President.

Burton, D. M. (1981). Automated concordances and word indexes: The Fifties. *Computers and the Humanities*, *15*(1), 1-14. https://doi.org/10.1007/BF02404370

Busa, R. (1951). *Sancti Thomae Aquinatis hymnorum ritualium varia specimina concordantiarum: Primo saggio di indici di parole automaticamente composti e stampati da macchine IBM a schede perforate*. Fratelli Bocca.

Busa, R. (1965). An inventory of fifteen million words. En J. B. Bessinger, Jr., S. M. Parrish y H. F. Arader (Eds.), *Literary data processing conference proceedings, yorktown heights, 1964* (pp. 64-78). The Modern Language Association Materials Centre.

Busa, R. (2019). The use of punched cards in linquistic analysis. En J. Nyhan y M. Passarotti (Eds.), *One origin of digital humanities* (pp. 39-57). Springer. https://doi.org/10.1007/978-3-030-18313-4_2

Butterfield, A., Ngondi, G. E. y Kerr, A. (2016). *A dictionary of computer science*. Oxford University Press.

Calderón Rivera, Á. (1991). Vigencia del diptongo hispánico /ui̯/. *Verba Hispanica*, *1*, 69-88.

Calvo Tello, J. (2019). Diseño de corpus literario para análisis cuantitativos. *Revista de Humanidades Digitales*, *4*, 115-135. https://doi.org/10.5944/rhd.vol.4.2019.25187

Cameron, D. (2011). Evolution, science and the study of literature: A critical response. *Language and Literature*, *20*(1), 59-72. https://doi.org/10.1177/0963947010391126

Canellada, M. J. y Madsen, J. K. (1987). *Pronunciación del español. Lengua hablada y literaria*. Castalia.

Cantero Serena, F. J. (2002). *Teorìa y análisis de la entonación*. Edicions Universitat Barcelona.

Caramuel Lobkowitz, J. (2007). *Primer cálamo de Juan de Caramuel que pone ante los ojos la rítmica. Tomo II. Rítmica* (I. Paraiso, Ed.; A. Carrera, J. A. Izquierdo y C. Lozano, Trad.). Universidad de Valladolid. (Fecha inicial de publicación 1665)

Caramuel Lobkowitz, J. (2011). Epístola XXI (P. Conde Parrado, Trad.). *Cuadernos de teatro clásico*, *27*. El Siglo de Oro habla de Lope, 280-323. (Fecha inicial de publicación 1668)

Carreño, A. (2019). Esta edición. En *El castigo sin venganza* (14.ª ed., pp. 97-100). Cátedra.

Carvallo, L. A. de. (1997). *Cisne de Apolo* (A. Porqueras Mayo, Ed.). Edition Reichenberger. (Fecha inicial de publicación 1602)

Casariego Castiñeira, P. (2018). Introducción. En *Nadie fíe su secreto* (pp. 11-68). Iberoamericana; Vervuert.

Ceccarelli, L. (1999). *Prosodia y métrica del latín clásico* (R. Carande Herrero, Trad.). Pblicaciones de la Universidad de Sevilla.

Chitoran, I. y Hualde, J. I. (2007). From hiatus to diphthong: The evolution of vowel sequences in Romance. *Phonology*, *1*, 37-75. https://doi.org/10.1017/S095267570700111X

Chomsky, N. y Halle, M. (1968). *Sound pattern of English*. Harper & Row.

Clavería Nadal, G. (2018). Contribución a la historia de los procesos de adaptación en los préstamos del español moderno. *Actas del X Congreso Internacional de Historia de la Lengua Española*, *1*, 157-191.

Clegg, J. H. y Willis, C. F. (2018). *Manual de fonética y fonología españolas*. Routledge.

Cohen, M. (1999). *The sentimental education of the novel*. Princeton University Press.

Corneille, P. (1862). Discours des trois unités d'action, de jour, et de lieu. En C. J. Marty-leveaux (Ed.), *Ouvres de P. Corneille* (pp. 98-122). Librairie de L. Hachette et Cie. (Fecha inicial de publicación 1660)

Crivellari, D. (2013). Marcas autoriales de segmentación en las comedias autógrafas de Lope de Vega: estudio y análisis. Edition Reichenberger.

Crymble, A. (2021). *Technology and the historian. Topics in the digital humanities*. University of Illinois Press.

Cuéllar, Á. (2023). La Inteligencia Artificial al rescate del Siglo de Oro: transcripción y modernización automática de mil trescientos impresos y manuscritos teatrales.

Hipogrifo. Revista de literatura y cultura del Siglo de Oro, *11*(1), 101-115. https://doi.org/10.13035/H.2023.11.01.08

Cuéllar, Á. y Vega García-Luengos, G. (2017). *ETSO: Estilometría aplicada al Teatro del Siglo de Oro*. Consultado el 28 de mayo de 2020, desde http://etso.es/

Cuéllar, Á. y Vega García-Luengos, G. (2023). *La francesa Laura*. El hallazgo de una nueva comedia del Lope de Vega último. *Anuario Lope de Vega. Texto, literatura, cultura*, *29*, 131-198. https://doi.org/10.5565/rev/anuariolopedevega.492

Debian. (2022). *Debian GNU/Linux* (Ver. 11.5). https://www.debian.org/

Dennerlein, K. (2015). Measuring the average population densities of plays. a case study of Andreas Gryphius, Christian Weise and Gotthold Ephraim Lessing. *Semicerchio. Rivista di poesia comparata*, *53*(2), 80-88.

Díaz Medina, A., Pérez Pozo, Á. y Rosa, J. de la. (2021). *Averell: A corpus management tool to transform poetic corpora into a JSON format compliant with the POSTDATA ontology* (Ver. 1.2.2). Zenodo. https://doi.org/10.5281/zenodo.5702404

Díaz Rengifo, J. (2012). *Arte poética española* (Á. Pérez Pascual, Ed.). Edition Reichenberger. (Fecha inicial de publicación 1592)

Digital Música Poética. (2023). Consultado el 17 de julio de 2023, desde https://digitalmp.uv.es/consulta/

Dinu, M. (1968). Structures linguistiques probabilistes dans l'étude du théâtre. *Cahiers du linguistique théorique et apliquée*, *5*, 29-46.

Dixon, V. (1985). The uses of polymetry: An approach to editing the Comedia as a verse drama. En F. P. Casa y M. McGaha (Eds.), *Editing the Comedia* (pp. 104-125). Michigan Romance Studies.

Dixon, V. (1994). The study of versification as an aid to interpreting the Comedia: Another look at some well-known plays by Lope de Vega. En C. Ganelin y H. Mancing (Eds.), *The Golden Age Comedia. text, theory and performance* (pp. 384-402). Purdue University Press.

Domínguez Caparrós, J. (1985). *Diccionario de métrica española*. Paraninfo.

Domínguez Caparrós, J. (2005). *Elementos de métrica española*. Tirant Lo Blanch.

Domínguez Caparrós, J. (2012). Prosodia rìtmica de vocales contiguas en interior de palabra. *Rhythmica. Revista Española de Métrica Comparada*, *10*, 11-44. https://doi.org/10.5944/rhythmica.13067

Domínguez Caparrós, J. (2014). *Métrica española*. Universidad Nacional de Educación a Distancia.

Donatus, A. (1838). Argumenti loco haec habet. En R. Klotz (Ed.), *Comœdiae* (pp. 3-6). E. B. Schwickert. (Fecha inicial de publicación ca. 350)

Donatus, A. (1840). Argumentum in *Adelphos* Terenti. En R. Klotz (Ed.), *Comœdiae* (pp. 3-6). E. B. Schwickert. (Fecha inicial de publicación ca. 350)

Drucker, J. (2021). *The digital humanities coursebook: An introduction to digital methods for research and scholarship*. Routledge.

Ebenhoch, M. (2015). Spätbarocke Kanzelrhethorik und aufklärische Homiliekritik in José Francisco de Islas *Fray Gerundio de Campazas*. En M. Calderón y G. Marko (Eds.), *Religion un Sprache. Religion and Language* (pp. 307-320). Peter Lang.

Eder, M., Rybicki, J. y Kestemont, M. (2016). Stylometry with R: A package for computational text analysis. *R Journal*, *8*(1), 107-121. https://doi.org/10.32614/RJ-2016-007

Ehrlicher, H. (2008). Peregrinación, mercado del mundo y la economía del auto sacramental: de Valdivielso a Calderón. En M. Tietz y G. Arnscheidt (Eds.), *Calderón y el pensamiento ideológico y cultural de su epoca. XIV Coloquio Anglogermano sobre Calderón. Heidelberg, 24-28 de julio de 2005* (pp. 191-208). Franz Steiner.

Ehrlicher, H. (2021). Narradores no fiables al servicio del imperio. Micropolítica y escritura táctica en las autobiografías de soldados del siglo XVII (Alonso de Contreras, Catalina de Erauso y Miguel de Castro). En S. Friedrich, W. Nitsch y C. Wehr (Eds.), *Figuraciones literarias del poder político en el Siglo de Oro* (pp. 345-367). Brill; Fink. https://doi.org/10.30965/9783846764862_020

Ehrlicher, H. (2023). Los límites de la alegoría del auto sacramental: comparación de dos controversias barrocas en Quevedo y Calderón. *Iberorromania*, *97*, 113-126. https://doi.org/10.1515/iber-2023-2002

Ehrlicher, H. y Lehmann, J. (2019). Datenerhebung als epistemologisches Labor : Überlegungen am Beispiel der virtuellen Forschungsumgebung *Revistas culturales 2.0*. En M. Huber, S. Krämer y C. Pias (Eds.), *Forschungsinfrastrukturen in den digitalen Geisteswissenschaften: wie verändern digitale Infrastrukturen die Praxis der Geisteswissenschaften? (Symposienreihe „Digitalität in den Geisteswissenschaften“)* (pp. 40-57). CompaRe.

Ehrlicher, H., Lehmann, J., Reiter, N. y Willand, M. (2020). La poética dramática desde una perspectiva cuantitativa: la obra de Calderón de la Barca. *Revista de Humanidades Digitales*, *5*, 1-25. https://doi.org/10.5944/rhd.vol.5.2020.27716

Ellegård, A. (1962). *Who was Junius?* Almquist & Wiksel.

Escobar Varela, M. (2021). *Theater as data: Computational journeys into theater research*. University of Michigan Press.

Esgueva Martínez, M. (2004). Vocales en contacto: la sinalefa. *Rhythmica*, *2*(2).

Esteban Rodríguez, S. (2022). Despoblación y cambios en el tipo de hábitat en las áreas rurales de Aragón (España). En C. Navarro Gómez, Á. R. Ruiz Pulpón, F. Velasco Caballero y J. Castillo Abella (Eds.), *Actas del I Congreso interdisciplinar sobre despoblación. Diagnóstico, territorio y gobierno local. Ciudad Real, 22 y 23 de septiembre de 2022* (pp. 25-42). Instituto de Derecho Local de la Universidad Autónoma de Madrid.

Evanthius. (1838). De tragœedia et comœdia. En R. Klotz (Ed.), *Comœdiae* (pp. xvii-xvi). E. B. Schwickert. (Fecha inicial de publicación ca. 350)

ExplosionAI. (2023). *spaCy* (Ver. 3.6). https://spacy.io/

Fant, G. (1970). *Acoustic theory of speech production* (R. Jakobson y C. H. Van Schooneveld, Eds.). Mouton. (Fecha inicial de publicación 1960)

Fernández Guillermo, L. (2007). Polimetría y variaciones métricas en el teatro de Lope de Vega. En B. Mariscal y B. López de Mariscal (Eds.), *Actas del XV Congreso de la Asociación Internacional de Hispanistas* (pp. 167-180). Asociación Internacional de Hispanistas/Tecnológico de Monterrey/El Colegio de México.

Fernández Guillermo, L. (2021). *Lope de Vega. El arte de la versificación teatral*. Ediciones del Lirio.

Fernández Rodríguez, D. (2021). El ritmo acentual en el Siglo de Oro: algunos romances líricos del joven Góngora (con Lope al trasluz). *Bulletin of Hispanic Studies*, *98*, 1-22. https://doi.org/10.3828/bhs.2021.1

Ferrer Valls, T. (2013). CATCOM: A database on performances of Spanish classical theater. *Journal for Early Modern Cultural Studies*, *13*(4), 160-163. https://doi.org/10.1353/jem.2013.0059

Ferrer Valls, T. (2023). El proyecto ASODAT indaga sobre el patrimonio teatral a partir de una federación de bases de datos. *Revista PH*, *109*, 10-12. https://doi.org/10.33349/2023.109.5374

Fischer, F., Börner, I., Göbel, M., Hechtl, A., Kittel, C., Milling, C. y Trilcke, P. (2019). Programmable corpora: Introducing DraCor, an infrastructure for the research on European drama. *Proceedings of DH2019: "Complexities", Utrecht, July 9–12, 2019*. https://doi.org/10.5281/zenodo.4284002

Fradejas Rueda, J. M. (2022a). *Cuentapalabras. Estilometría y análisis de texto con R para filólogos*. http://www.aic.uva.es/cuentapalabras/

Fradejas Rueda, J. M. (2022b). De editor analógico a editor digital (S. Bazzaco, Ed.). *Historias Fingidas*, *1*. Humanidades Digitales y estudios literarios hispánicos, 39-65. https://doi.org/10.13136/2284-2667/1108

Freytag, G. (1894). *Die Technik des Dramas* (7.ª ed.). S. Hirzel.

Frye, N. (1971). *Anatomy of criticism. Four essays*. Princeton University Press. (Fecha inicial de publicación 1957)

Fundación Biblioteca Virtual Miguel de Cervantes. (2021). *Biblioteca Virtual Miguel de Cervantes*. http://www.cervantesvirtual.com

García Adánez, I. (2005). Reflexiones sobre la composición fonética y el lenguaje como material de la composición. En A. M. Reboul (Ed.). A. Gimber y A. I. Fernández Valbuena (Colabs.), *Palabra y música* (pp. 95-104). Universidad Complutense de Madrid.

García Barrientos, J. L. (1991). *Drama y tiempo*. Consejo Superior de Investigaciones Científicas.

García Barrientos, J. L. (2020). *Anatomía del drama: una teoría fuerte del teatro*. Punto de Vista Editores.

García-Reidy, A. (2019). La presencia escénica de Andrés de Claramonte en el Siglo de Oro a partir de las bases de datos Manos y CATCOM. *Bulletin of the Comediantes*, *71*(1), 135-154. https://doi.org/10.1353/boc.2019.0009

Gavin, M. (2022). *Literary mathematics. Quantitative theory for textual study*. Stanford University Press.

Gervás, P. (2000). A logic programming application for the analysis of Spanish verse. *Computational Logic - CL 2000, First International Conference*, 1330-1344. https://doi.org/10.1007/3-540-44957-4_89

GIC. (2023). *Grupo de Investigación Calderón de la Barca*. Consultado el 14 de julio de 2023, desde http://www.usc.es/es/investigacion/grupos/calderon

Gilabert, G. (2014). *El arte dramático de Bances de Candamo: poesía y música en las comedias del ocaso del Siglo de Oro* (Tesis doctoral). Universitat de Barcelona. http://hdl.handle.net/10803/286632

Glyph & Cog. (2022). *Text extraction*. https://www.glyphandcog.com/textext.html

Gondomar Digital. La colección teatral del conde de Gondomar. (2017). Consultado el 17 de julio de 2023, desde https://gondomar.tespasiglodeoro.it

Gottschall, J. (2008). *Literature, science, and a new humanities*. Palgrave Macmillan.

Gottschall, J. (2010). Evolutionary literary studies: A foreword to Norbert Francis's review article 'A modest proposal'. *Language and Literature*, *19*(3), 301-304. https://doi.org/10.1177/0963947010368317

Greer, M. R. y García-Reidy, A. (Dirs.). (2022). Manos. Base de datos de manuscritos teatrales áureos. http://www.manos.net

Grupo de investigación teatral DICAT. (2023). Consultado el 17 de julio de 2023, desde https://dicat.uv.es/elgrupodicat

Grupo de Investigración Siglo de Oro. (2020). *Autos sacramentales completos de Calderón*. https://www.unav.edu/web/griso/publicaciones/autos-sacramentales-completos-de-calderon

Hanssen, F. (1957). La seguidilla. *Anales de la Universidad de Chile*, *107-108*, 184-221. (Fecha inicial de publicación 1909)

Haralambous, Y. y Dürst, M. (2019). Unicode from a linguistic point of view. En Y. Haralambous (Ed.), *Proceedings of Graphemics in the 21st Century, Brest 2018* (pp. 167-183). Fluxus Editions.

Hawking, S. (1995). *A brief history of time.* Bantam.

Hayles, N. K. (2011). How we think: Transforming power and digital technologies. En D. M. Berry (Ed.), *Understanding the digital humanities* (pp. 42-66). Palgrave. https://doi.org/10.1057/9780230371934_3

Hegel, G. W. F. (2016). *Vorlesungen über die Ästethik* (10.ª ed., Vol. 3). Suhrkamp. (Fecha inicial de publicación 1835)

Hernández-Lorenzo, L. (2022a). Análisis de corpus poéticos con *Litcon*. *Revista de Humanidades Digitales*, *104-126*, 137-166. https://doi.org/10.5944/rhd.vol.7.2022

Hernández-Lorenzo, L. (2022b). Introducción al panorama internacional de las Humanidades Digitales. *Revista de Humanidades*, *46*, 137-166. https://revistas.uned.es/index.php/rdh/article/view/32464

Hernández-Lorenzo, L. y Byszuk, J. (2022). Challenging stylometry: The authorship of the Baroque play *La Segunda Celestina*. *Digital Scholarship in the Humanities*, *38*(2), 1-15. https://doi.org/10.1093/llc/fqac063

Herrero Prado, J. L. (1996). *Métrica española: teoría y práctica*. Ediciones del Orto.

Hiergeist, T. (2018). Hörner, die die Welt bedeuten. Stierkampf und Theater in Pedro Calderón de la Barcas *entremés* „El toreador". En S. Frisch, E. Fritz y R. Rieger (Eds.), *Spektakel als ästhetische Kategorie. Theorien und Praktiken* (pp. 179-193). Wilhelm Fink. https://doi.org/10.30965/9783846762684_009

Hilborn, H. W. (1938). *A chronology of the plays of Don Pedro Calderón de la Barca*. University of Toronto Press.

Hockey, S. (2004). The history of Humanities Computing. En S. Schreibman, R. Siemens y J. Unsworth (Eds.), *A companion to digital humanities* (pp. 3-19). Blackwell. https://doi.org/10.1002/9780470999875.ch1

Hoover, D. L. (2013). Quantitative analysis and literary studies. En R. Siemens y S. Schreibman (Eds.), *A companion to digital literary studies* (pp. 517-533). Wiley-Blackwell.

Hopper, G. M. (1978). Keynote address. En R. L. Wexelblat (Ed.), *History of programming languages I* (pp. 16-20). Association for Computing Machinery.

Horaz. (1986). *Ars poetica/Die Dichtkunst*. Reclam. (Fecha inicial de publicación ca. 19 a.e.c.)

Hualde, J. I. (1991). On Spanish syllabification. En H. Campos y F. Martínez-Gil (Eds.). Georgetown University Press.

Ilsemann, H. (1998). *Shakespeare Disassembled: Eine quantitative Analyse der Dramen Shakespeares*. Peter Lang.

Ilsemann, H. (2013). Quantitativ-statistische Dramenanalyse: Welche Aussagekraft haben Häufigkeitsverteilungen der Replikenlängen? *Jahrbuch für Computerphilologie*. http://computerphilologie.digital-humanities.de/jg09/Ilsemann.html

Ingarden, R. (1972). *Das literarische Kunstwerk* (4.ª ed.). Max Niemeyer. (Fecha inicial de publicación 1931)

Jakobson, R. (1962). Phoneme and phonology. En *Roman Jakobson Selected Writings: Phonological Studies* (pp. 231-233). Mouton & Co./'s-Gravenhage.

Josselson, H. (1953). *The Russian word count*. Wayne University Press.

Kayser, W. (1992). *Das sprachliche Kunstwerk. Eine Einfürung in die Literaturwissenschaft* (20.ª ed.). Francke Verlag. (Fecha inicial de publicación 1948)

Kroll, S. (2019a). Filología digital para el estudio de la cultura y literatura del Siglo de Oro (2014-2017). *Etiópicas. Revista de letras renacentistas*, *15*, 1-21. https://aiso-asociacion.org/wp-content/uploads/2019/12/aiso_1.pdf

Kroll, S. (2019b). La sonoridad de la culpa y de la gracia en los autos sacramentales de Calderón: el caso de *La vida es sueño*. *Hipogrifo. Revista de literatura y cultura del Siglo de Oro*, *7*(2), 433-444. https://doi.org/10.13035/H.2019.07.02.34

Kroll, S. (2020). Voces del horror y del amor: el sentido de las asonancias calderonianas. *Arte Nuevo: Revista de Estudios Áureos*, *7*, 300-326. https://doi.org/10.7203/tirant.23.19179

Kroll, S. y Sanz-Lázaro, F. (2022). Romances teatrales entre Mira de Amescua, Calderón y Lope: ritmo, asonancia y cuestiones de autoría. *Revista de Humanidades Digitales*, *7*(1), 1-18. https://doi.org/10.5944/rhd.vol.7.2022.31620

Kroll, S. y Sanz-Lázaro, F. (2023). Ritmo, autoría y género: nuevas perspectivas sobre teatro lopesco desde las humanidades digitales. *Anuario Lope de Vega. Texto, literatura, cultura*, *29*, 351-375. https://doi.org/10.5565/rev/anuariolopedevega.491

Lapesa, R. (2008). *Historia de la lengua española*. Gredos. (Fecha inicial de publicación 1981)

Lehmann, J. y Padó, S. (2022). Clasificación de tragedias y comedias en las comedias nuevas de Calderón de la Barca. *Revista de Humanidades Digitales*, *7*, 80-103. https://doi.org/10.5944/rhd.vol.7.2022

Lessig, L. (2004). *Free culture. The nature and future of creativity*. Penguin Group.

Llamas Martínez, J. (2020). Teoría y práctica en la poesía de Quevedo: la *compositio* fonética y silábica. *La Perinola*, *24*, 49-69. https://doi.org/0.15581/017.24.49-69

López-Gonzalo, E., Olaszy, G. y Németh, G. (1993). Improvements of the Spanish version of the MULTIVOX text-to-speech system. *Third European Conference on Speech Communication and Technology, EUROSPEECH 1993, Berlin, Germany, September 22-25, 1993*, *2*, 869-872.

Ludwig Tiecks Bibliothek. Anatomie einer romantisch-komparatistischen Büchersammlung [Dirección: Achim Hölter]. (2018). Consultado el 28 de junio de 2023, desde https://tieck-bibliothek.univie.ac.at

Lukács, G. (2009). Die Theorie des Romans. (Fecha inicial de publicación 1920)

Luque Moreno, J. (2010). Lope de Vega, versificador latino. *Boletín de la Biblioteca de Menéndez Pelayo*, *86*, 109-144.

Lutz, M. y Ascher, D. (2013). *Learning Python* (5.ª ed.). O'Really.

MacKenzie, C. E. (1980). *Coded character sets: History and development*. Addison Wesley.

Marco, G., Rosa, J. de la, Gonzalo, J., Ros, S. y González-Blanco, E. (2021). Automated metric analysis of Spanish poetry: Two complementary approaches. *IEEE Access*, *9*, 51734-51746. https://doi.org/10.1109/ACCESS.2021.3069635

Marco Remón, G. y Gonzalo Arroyo, J. (2021). Escansión automática de poesía española sin silabación. *Procesamiento del Lenguaje Natural*, *66*, 77-87. https://doi.org/10.26342/2021-66-6

Marcos Marín, F. A. (1994). *Informática y humanidades*. Gredos.

Marcus, S. (1973). *Mathematische Poetik* (E. Mândroiu, Trad.). Athenäum.

Martinet, A. (1970). *La linguistique sychronique* (3.ª ed.). Presses Universitaires de France. (Fecha inicial de publicación 1965)

Martínez, M. A. E. (2004). Vocales en contacto: la sinalefa. *Rhythmica. Revista Española de Métrica Comparada*, *2*, 87-108. https://doi.org/10.5944/rhythmica.6403

Martínez Celdrán, E. (2004). Castilian Spanish. *Journal of the International Phonetic Association*, *34*(2), 201-210. https://doi.org/10.1017/S0025100304001732

Martínez Celdrán, E., Fernández Planas, A. M. y Carrera Sabaté, J. (2003). Castilian Spanish. *Journal of the International Phonetic Association*, *33*(2), 255-259. https://doi.org/10.1017/S0025100303001373

Martínez Zuviría, G. y Herrera, A. (1938). Sobre el endecasílabo. *Boletín de la Academia Argentina de las Letras*, *6*(23-24), 389-396.

McKinney, W. (2018). *Python for data analysis: Data wrangling with Pandas, NumPy and Python* (2.ª ed.). O'Really.

Menéndez Pelayo, M. (1951). Advertencia preliminar. En E. Sánchez Reyes (Ed.), *Bibliografía hispano-latina clásica*. Consejo Superior de Investigaciones Científicas. (Fecha inicial de publicación 1908)

Milner, A. (2021). El diptongo/hiato como rasgo contrastivo: un estudio perceptual con hablantes de herencia de español. *Studies in Hispanic and Lusophone Linguistics*, *14*, 459-496. https://doi.org/10.1515/shll-2021-2052

Minsky, M. (1975). A framework for representing knowledge. En P. Winston (Ed.), *The psychology of computer vision* (pp. 211-277). McGraw-Hill.

Montgomery, G. y Hubbard, L. A. (Eds.). (1957). *Concordances to the poetical works of John Dryden*. University of California Press.

Monzó, C. (2019). *Poética de la acotación en la dramaturgia de Calderón de la Barca* (Tesis doctoral). Universitat de València. https://hdl.handle.net/10550/72198

Monzó, C. (2020). Sobre la identidad del receptor en las acotaciones del teatro de Calderón. Arqueología y reconstrucción. En A. Martínez Pereira, M. D. Martos, E. Borrego Gutiérrez e I. Osuna Rodríguez (Eds.), *En la villa y corte. TRIGESIMA AUREA: Actas del XI Congreso de la Asociación Internacional Siglo de Oro (Madrid, 10-14 de julio de 2017)* (pp. 557-566). Universidad Nacional de Educación a Distancia/Universidad Complutense de Madrid.

Monzó, C. (2021). La formulación de la comedia nueva a través de las acotaciones de Francisco Tárrega: conciencia dramatúrgica y autoridad autorial. *Anuario Lope de Vega. Texto, literatura, cultura*, *27*, 87-121. https://doi.org/10.5565/rev/anuariolopedevega.398

Moran, S. y Cysouw, M. (2018). *The Unicode cookbook for linguists: Managing writing systems using ortography profiles*. Language Science Press.

Moretti, F. (2000a). Conjectures on world literature. *New Left Review*, *1*, 54-68. https://newleftreview.org/issues/ii1/articles/franco-moretti-conjectures-on-world-literature

Moretti, F. (2000b). The slaughterhouse of literature. *Modern language quarterly (Seattle)*, *61*(1), 207-228.

Moretti, F. (2003). Graphs, maps, trees: Abstract models for literary history - 1. *New Left review*, *24*, 67-93.

Moretti, F. (2004). Graphs, maps, trees: Abstract models for literary history - 2. *New Left review*, *26*, 79-103.

Moretti, F. (2008). The novel: History and theory. *New Left Review*, *52*, 111-124. https://newleftreview.org/issues/ii52/articles/franco-moretti-the-novel-history-and-theory

Moretti, F. (2017). *Patterns and interpretation*. Stanford University. https://litlab.stanford.edu/LiteraryLabPamphlet15.pdf

Moretti, F. y Sobchuk, O. (2019). Hidden in plain sight. *New Left review*, *118*.

Morley, S. G. y Bruerton, C. (1968). *Cronología de las comedias de Lope de Vega: Con un examen de las atribuciones dudosas, basado todo ello en un estudio de su versificación estrófica* (M. R. Cartes, Trad.). Gredos.

Morton, A. Q. (1965). *The authorship of the Pauline epistles: A scientific solution*. University of Saskatchewan.

Mosteller, F. y Wallace, D. L. (1963). Inference in an authosrhip problem: A comparative study of discrimination methods applied to the authorship of the disputed 'Federalist' papers. *Journal of the American Staistical Association*, *58*(302), 275-309.

Muñiz Cachón, C. y Cuevas Alonso, M. Á. (2012). Grados de sonorización de la consonante /s/ en el español de Asturias. *Revista de Filoloxía Asturiana*, *3*(3-4), 291-304.

Musiphron [Mark Akinside]. (1746). The ballance of poets. En R. Dodsley (Ed.), *The Museum: Or, the literary and historical register* (pp. 165-169).

National Telecommunication Information Administration. (1997). *Telecommunications: Glossary of telecommunications terms*. Government Institutes.

Navarro Tomás, T. (1946). *Estudios de fonología española*. Syracuse University Press.

Navarro Tomás, T. (1957). *Manual de pronunciación española* (8.ª ed.). Consejo Superior de Investigaciones Científicas.

Navarro Tomás, T. (1991). *Métrica española*. Labor.

Navarro Tomás, T. (2004). *Manual de pronunciación española* (28.ª ed.). Consejo Superior de Investigaciones Científicas.

Navarro Tomás, T. (2014). El octosílabo y sus modalidades. En *Los poetas en sus versos: desde Jorge Manrique a García Lorca* (pp. 35-66). Centro para la edición de los clásicos españoles. (Fecha inicial de publicación 1973)

Navarro-Colorado, B. (2015). A computational linguistic approach to Spanish Golden Age sonnets: Metrical and semantic aspects. *Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature*, 105-113. https://doi.org/10.3115/v1/W15-0712

Navarro-Colorado, B. (2017). A metrical scansion system for fixed-metre Spanish poetry. *Digital Scholarship in the Humanities*, *33*(1), 112-127. https://doi.org/10.1093/llc/fqx009

Navarro-Colorado, B., Ribes Lafoz, M. y Sánchez, N. (2016). Metrical annotation of a large corpus of Spanish sonnets: Representation, scansion and evaluation. En N. C. (Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard,

J. Mariani, H. Mazo, A. Moreno, J. Odijk y S. Piperidis (Eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

Nebrija, A. de. (1981). *Gramática de la lengua castellana* (A. Quilis, Ed.; Vol. 2). Editora Nacional. (Fecha inicial de publicación 1492)

Nietzsche, F. (1982). *Homer und die klassische Philologie* (G. Colli y M. Montinari, Eds.; Vol. 5). De Gruyter. (Fecha inicial de publicación 1869)

Oleza Simó, J. (2010). A propósito de la estructura, la significación y la técnica de la Comedia Nueva. *Teatro de Palabras*, *4*, 99-137. https://www.uv.es/entresiglos/oleza/pdfs/a_proposito_estructura_Comedia_Nueva.pdf

Oleza Simó, J. (2013). A description of ARTELOPE: A database containing information, plots, and texts related to Lope de Vega's theater. *Journal for Early Modern Cultural Studies*, *13*(4), 150-151. https://doi.org/10.1353/jem.2013.0041

Online Etymological Dictionary. (2022). Douglas Harper. https://www.etymonline.com

Open Source Initiative. (2007). *The Open Source Definition* (1.9). https://opensource.org/osd

Papelier, G. (2005). Le Trésor de la langue française (TLF). En J. Borboulon (Ed.), *L'aventure Gamma 60*. Hermès Science Publications.

Paraíso, I. (2000). La métrica española en su contexto románico. Arco.

Patsch, H. (1966). Friedrich Schlegels „Philosophie der Philologie" und Schleiermachers frühe Entwürfe zur Hermeneutik: Zur Frühgeschichte der romantischen Hermeneutik. *Zeitschrift für Theologie und Kirche*, *63*(4), 434-472.

Pérez, J. C. y Vidal, E. (1991). Un sistema de conversión de texto a voz para castellano. *Procesamiento del Lenguaje Natural*, *11*, 197-207.

Pérez Pozo, Á., Rosa, J. de la, Ros, S., González-Blanco, E., Hernández, L. y Sisto, M. de. (2021). A bridge too far for artificial intelligence?: Automatic classification of stanzas in Spanish poetry. *Journal of the Association for Information Science and Technology*, *73*(2), 258-267. https://doi.org/10.1002/asi.24532

Piles, R. de. (1708). *Cours de peinture par principes avec un balance des peintres*. Jacques Etienne.

Pino, M. (1995). Manual de codificación textual para los corpus CREA y CORDE. Normas de marcación SGML según las recomendaciones de la TEI.

Pino, M. y Santalla, M. P. (1996). Codificación de la infomación morfosintáctica de corpus en lenguage SGML. *Procesamiento del Lenguaje Natural*, *19*, 101-117.

Platon. (2011). *Platon: Werke in acht Bänden* (G. Eigler, Ed.; F. Schleiermacher, Trad.; 6.ª ed., Vol. 4). Wissenschaftliche Buchgesellschaft. (Fecha inicial de publicación 376 a.e.c.)

Plecháč, P., Bobenhausen, K. y Hammerich, B. (2018). Versification and authorship attribution. A pilot study on Czech, German, Spanish, and English poetry. *Studia Metrica et Poetica*, *5*(2), 29-54. https://doi.org/10.12697/smp.2018.5.2.02

Popper, K. (1989). *Conjectures and refutations* (5.ª ed.). Routledge. (Fecha inicial de publicación 1963)

Popper, K. (2002). *The logic of scientific discovery* (2.ª ed.). Routledge. (Fecha inicial de publicación 1935)

Potter, R. G. (Ed.). (1989). *Literary computing and literary criticism: Theoretical and practical essays on theme and rhetoric*. University of Pennsylvania Press.

Potter, R. G. (1991). Statistical analysis of literature: A retrospective on 'Computers and the Humanities', 1966–1990. *Computers and the Humanities*, *25*(6), 401-429. https://doi.org/10.1007/bf00141190

Presotto, M. (2018). La edición crítica digital del teatro del Siglo de Oro. *Cuadernos AISPI. Estudios de lenguas y literaturas hispánicas*, *11*, 31-46. https://doi.org/10.14672/0.2018.1426

Presotto, M. (2023). Ecdótica y Humanidades Digitales en el estudio del Teatro clásico español: una experiencia. *Revista de Humanidades Digitales*, *8*, 1-13. https://doi.org/10.5944/rhd.vol.8.2023.37267

Prolope. (2023). *Grupo de investigación sobre Lope de Vega de la Universidad Autónoma de Barcelona*. Consultado el 14 de julio de 2023, desde https://prolope.uab.cat/

Prospect. (1966). *Computers and the Humanities*, *1*(1), 1-2. https://doi.org/10.1007/BF00188009

PROTEO. (2023). *Moretianos.com*. Consultado el 14 de julio de 2023, desde http://www.moretianos.com/

Qi, P., Zhang, Y., Zhang, Y., Bolton, J. y Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. https://doi.org/10.18653/v1/2020.acl-demos.14

Quilis, A. (2013). *Métrica española* (17.ª ed.). Ariel. (Fecha inicial de publicación 1984)

Quilis, A. (2019). *Tratado de fonología y fonética españolas*. Gredos. (Fecha inicial de publicación 1993)

R Core Team. (2019). *R: A language and environment for statistical computing* (Ver. 3.6.2). Viena, R Foundation for Statistical Computing. https://www.R-project.org/

Real Academia Española y Asociación de Academias de la Lengua Española. (2010a). *Nueva gramática de la lengua española. Manual*. Espasa.

Real Academia Española y Asociación de Academias de la Lengua Española. (2010b). *Ortografía de la lengua española*. Espasa.

Real Academia Española y Asociación de Academias de la Lengua Española. (2012). *Ortografía básica de la lengua española*. Espasa.

Richens, R. H. y Booth, A. D. (1955). Some methods of mechanized translation. En W. N. Locke y A. D. Booth (Eds.), *Machine translation of languages*. Greenwood Press.

Rio Riande, M. G. del. (2014). De qué hablamos cuando hablamos de humanidades digitales. En L. Cantamutto, M. G. d. Rio Riande y G. Striker (Eds.), *Actas de las I Jornadas Nacionales de Humanidades Digitales: Culturas, Tecnologías, Saberes* (pp. 31-41). Asociación Argentina de Humanidades Digitales. https://doi.org/10.5281/zenodo.2545557

Rio Riande, M. G. del y González Blanco García, E. (2015). Introducción a las humanidades digitales [Material didáctico sistematizado]. https://www.aacademica.org/gimena.delrio.riande/115

Ríos Mestre, A. (1999). La transcripción fonética automática del diccionario electrónico de formas simples flexivas del español: estudio fonológico en el léxico. *Estudios de Lingüística del Español*, *4*. http://elies.rediris.es/elies4/

Rodríguez Crespo, M. Á., Escalada Sardina, J. G. y Torre Toledadno, D. (1998). Conversor texto-voz multilingüe para español, catalán, gallego y euskera. *Procesamiento del Lenguaje Natural*, *23*, 16-23.

Rojo, G. (2016). Citius, maius, melius: del CREA al CORPES XXI. En J. Kabatek (Ed.), *Lingüística de corpus y lingüística histórica iberorrománica* (pp. 197-212). De Gruyter. https://doi.org/10.1515/9783110462357-010

Romanska, M. (2015). Drametrics: What dramaturgs should learn from mathematicians. En M. Romanska (Ed.), *The Routledge companion to dramaturgy* (pp. 438-447). Routledge.

Rommel, T. (2004). Literary studies. En S. Schreibman, R. Siemens y J. Unsworth (Eds.), *A companion to digital humanities* (pp. 88-96). Blackwell.

Rosa, J. de la y Pérez, Á. (2020). *Linhd-postdata/edfu: First version of EDFU syllabification corpus* (Ver. 1.0.0). https://doi.org/10.5281/zenodo.3898684

Rosa, J. de la, Pérez, Á., Hernández, L., Ros, S. y González-Blanco García, E. (2020). Rantanplan, fast and accurate syllabification and scansion of Spanish poetry. *Procesamiento del Lenguaje Natural*, *65*, 83-90. https://doi.org/10.26342/2020-65-10

Rosa, J. de la, Pérez, Á., Hernández, L., Ros, S. y González-Blanco Garcìa, E. (2022). *Poetrylab: An open source toolkit for the analysis of Spanish poetry corpora* (Ver. 1.0.1-dev). https://postdata.linhd.uned.es/results/poetrylab/poetrylab-app/

Rosa, J. de la, Pérez Pozo, Á., Ros, S. y González-Blanco García, E. (2023). ALBERTI, a multilingual domain specific language model for poetry analysis. *SEPLN 2023: 39th International Conference of the Spanish Society for Natural Language Processing*. https://doi.org/10.48550/arXiv.2307.01387

Ruano de la Haza, J. M. (1994). Una nota sobre la división en cuadros. En J. M. Ruano de la Haza y J. J. Allen (Eds.), *Los teatros comerciales del siglo XVII y la puesta en escena de la Comedia* (pp. 291-294). Castalia.

Ruano de la Haza, J. M. (2007). Los espacios de *Don Juan*. *Hecho Teatral*, *7*, 127-145.

Russell, B. (1976). *Human knowledge: Its scope and limits*. George Allen & Unwin. (Fecha inicial de publicación 1948)

Russell, D. B. (1967). *COCOA – A word count and concordance generator for Atlas*. Atlas Computer Laboratory.

Saavedra Molina, J. (1944). El octosílabo castellano. *Anales de la Universidad de Chile*, *53-54*, 65-229.

Salvador Plans, A. (1982). El grafema *H* en los tratadistas del Siglo de Oro. *Anuario de Estudios Filológicos*, *5*, 167-178.

Sánchez, F. y Porqueras Mayo, A. (1972). *Preceptiva dramática española del Renacimiento y el Barroco*. Gredos.

Sánchez Jiménez, A. (2014). Métrica y secreto en Lope de Vega: el endecasílabo encadenado en la *Parte XI (El amigo hasta la muerte y El mayordomo de la duquesa de Amalfi)*. *Criticón*, *122*, 117-129.

Sánchez Jiménez, A. (2017). Acentos contiguos en los romances de la *Arcadia* (1598), de Lope de Vega. *Atalanta. Revista de las letras barrocas*, *5*(1), 5-61. https://doi.org/10.14643/51A

Sánchez Jiménez, A. (2020). Peculiaridades métricas de la poesía no dramática de Calderón: los desplazamientos acentuales. *Anuario Calderoniano*, *13*, 297-315.

Sánchez Jiménez, A. y Sáez, A. J. (2013). Introducción al auto *La cena del rey Baltasar*. En *La cena del rey Baltasar* (pp. 11-114). Edition Reichenberger; Universidad de Navarra.

Sanz-Lázaro, F. (2021). Complejidad léxica de artículos editoriales de la prensa española: una selección de cuatro cabeceras. *Revista de Humanidades Digitales*, *6*, 85-100. https://doi.org/10.5944/rhd.vol.6.2021.30861

Sanz-Lázaro, F. (2022a). Gregorio González en su pícaro: algunas cuestiones sobre la datación del *Buscón* a partir del *Guitón*. *Hipogrifo. Revista de literatura y cultura del Siglo de Oro*, *10*(1), 717-736.

Sanz-Lázaro, F. (2022b). *Txt2tei* (Ver. 1.0.3). https://github.com/fsanzl/txt2tei

Sanz-Lázaro, F. (2022c). Gregorio González, sumulista de primer año. *Avisos de Viena*, *3*, 96-104. https://journals.univie.ac.at/index.php/adv/article/view/6590/6630

Sanz-Lázaro, F. (2023a). *Fonemas* (Ver. 2.0.16). https://github.com/fsanzl/fonemas

Sanz-Lázaro, F. (2023b). *LibEscansión* (Ver. 1.0.0). https://github.com/fsanzl/libEscansion

Sanz-Lázaro, F. (2023c). Planteamientos digitales del drama aurisecular: automatización del análisis métrico de obras teatrales. En I. González Cabeza, É. Redruello Vidal y R. de la Varga Llamazares (Eds.), *La escritura en el tiempo. Investigaciones en torno a la literatura hispánica* (pp. 109-126). Universidad de León.

Sanz-Lázaro, F. (2023d). *Silabeador* (Ver. 1.1.11). https://github.com/fsanzl/silabeador

Sanz-Lázaro, F. (en prensa). Del fonema al verso: una caja de herramientas digitales de escansión teatral. *Revista de Humanidades Digitales*, *8*. https://doi.org/https://doi.org/10.5944/rhd.vol.8.2023

Schaeffer, J.-M. (1995). Énonciation théâtrale. En O. Ducrot y J.-M. Schaeffer (Eds.), *Nouveau dictionnaire encyclopédique des sciences du langage* (pp. 612-621). Seuil.

Schlegel, F. (1957). *Friedrich Schlegel: Literary Notebooks, 1797–1801* (H. Eichner, Ed.). University of London. (Fecha inicial de publicación 1804-1806)

Schmidt, T., Burghardt, M., Dennerlein, K. y Wolff, C. (2019). Katharsis – A tool for computational drametrics. *Book of Abstracts*. https://dev.clariah.nl/files/dh2019/boa/0584.html

Scholes, R. y Wulfman, C. (2008). Humanities Computing and Digital Humanities. *South Atlantic Review*, *73*(4), 50-66.

Schreibman, S., Siemens, R. y Unsworth, J. (Eds.). (2004a). *A companion to digital humanities*. Blackwell.

Schreibman, S., Siemens, R. y Unsworth, J. (2004b). The digital humanities and humanities computing: An introduction. En S. Schreibman, R. Siemens y J. Unsworth (Eds.), *A companion to digital humanities* (pp. xxiii-xxvii). Blackwell.

Schubart, C. F. D. (1792). Kritische Skala der vorzüglichsten deutschen Dichter. En E. L. Posselt (Ed.), *Archiv für ältere und neuere, vorzüglich Teutsche Geschichte, Staatsklugheit und Erdkunde* (pp. 164-172).

Spoerhase, C. (2018). Rankings: A prehistory. *New Left Review*, *114*, 99-112.

Stallman, R. M. (2002). Why 'free software' is better than 'open source'. En *Free software, free society* (2.ª ed., pp. 57-62). GNU Press. (Fecha inicial de publicación 1998)

Stowasser, J. M., Petschenig, M. y Skutsch, F. (1998). *Stowasser: Lateinisch – deutsches Schülwörterbuch*. Verlag Hölder-Pichler-Tempsky.

TEI. (2005). History. Consultado el 13 de junio de 2022, desde https://tei-c.org/about/history/

TEI Consortium. (2022). TEI Guidelines. https://tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf

TOP500.org. (2022). *Top 500. The list*. Consultado el 29 de noviembre de 2022, desde https://www.top500.org/statistics/details/osfam/1/

Torre, E. (1999). *El ritmo del verso*. Universidad de Murcia.

Torre, E. (2000). *Métrica comparada*. Universidad de Sevilla.

Torreblanca, M. (1978). El fonema /s/ en la lengua española. *Hispania*, *61*(3), 498-503.

Torrente, Á. (2016). Del corral al coliseo: armonías del teatro áureo. En Á. Torrente (Ed.), *Historia de la música en España e Hispanoamerica* (pp. 321-434). Fondo de Cultura Económica.

Trubetzkoy, N. S. (1939). *Grundzüge der Phonologie*. Cercle Linguistique de Copenhague/Ministère de l'Instruction Publique de la République Tchéco-Slovaque.

Trubetzkoy, N. S. (1969). *Principles of phonology* (C. A. M. Baltaxe, Trad.). University of California Press.

Türschmann, J. (2013). Transdisziplinarität und Transmedialität. En F. Brand, F. Schaller y H. Völker (Eds.), *Transdisziplinarität. Bestandsaufnahme und Perspektiven. Beiträge zur THESIS-Arbeitstagung im Oktober 2003 in Göttingen* (pp. 177-191). Universitätsverlag Göttingen.

Ubersfeld, A. (1996). *Lire le théâtre*. Belin. (Fecha inicial de publicación 1977)

Underwood, T. (2019). Dear humanists: Fear not the digital revolution. *The Chronicle of Higher Education*, *65*. https://www.chronicle.com/article/dear-humanists-fear-not-the-digital-revolution/

Unsworth, J. (2002). What is humanities computing and what is not? *Jahrbuch für Computerphilologie*, *4*, 71-84. http://computerphilologie.digital-humanities.de/jg02/unsworth.html

Urzáiz Tortajada, H., González Martínez, J., Cienfuegos Antelo, S., Sánchez Jiménez, R., San José Lera, J., Presotto, M., Marcos Rodríguez, E. y Ferreira Berrocal, J. (2023). *Censuras y licencias en manuscritos e impresos teatrales (CLEMIT)*. Consultado el 17 de julio de 2023, desde http://www.clemit.es/

Valdés, J. de. (2015). *Diálogo de la lengua* (C. Barbolani, Ed.; 11.ª ed.). Cátedra. (Fecha inicial de publicación 1737)

Valdés Gázquez, R. (2104a). Anhelos, realidades y sueños ante la perspectiva y urgencia de la edición crítica digital: reflexiones desde un grupo de investigación. *Anuario Lope de Vega*, *20*, 1-46. https://doi.org/10.5565/rev/anuariolopedevega.86

Valdés Gázquez, R. (2104b). Edición digital y edición crítica. *Anuario Lope de Vega*, *20*, i-iii. https://doi.org/10.5565/rev/anuariolopedevega.91

van Leuwen, H. C. (1989). A developement tool for linguistic rules. *Computer Speech and Language*, *3*(1), 83-104. https://doi.org/10.1016/0885-2308(89)90016-8

Vaquero Sánchez, A. (1999). La lengua española en el contexto informático. *Revista de enseñanza y tecnología*, *13*, 6-13.

Vaquero Sánchez, A. y Joyanes Aguilar, L. (1985). *Informática: glosario de términos y siglas. Diccionario inglés-español-español-inglés*. McGraw-Hill.

Varela Merino, E., Moíno Sánchez, P. y Jauralde Pou, P. (2005). *Manual de métrica española*. Castalia.

Vega Carpio, L. de. (2006). *Arte nuevo de hacer comedias* (E. García Santo-Tomás, Ed.). Cátedra. (Fecha inicial de publicación 1609)

Vitse, M. (1988). *Éléments pour une théorie du théâtre espagnol du XVII^e siècle*. France-Ibérie Recherche.

Vitse, M. (1996). Sobre los espacios en *La dama duende*: el cuarto de don Manuel. *Revista del Instituto de Lengua y Cultura Españolas*, *2*(12), 337-356. (Fecha inicial de publicación 1985)

Vitse, M. (1998). Polimetría y estructuras dramáticas en la comedia de corral del siglo XVII: el ejemplo de *El burlador de Sevilla*. En Y. Campbell (Ed.), *El escritor y la escena VI. Estudios sobre teatro español y novohispano de los Siglos de Oro* (pp. 45-63). Universidad Autónoma de Ciudad Juárez.

Vitse, M. (2010). «Partienda est comœdia»:la segmentación frente a sí misma. *Teatro de Palabras*, *4*, 19-75.

Wells, J. (1997). SAMPA computer readable phonetic alphabet. En D. Gibbon, R. Moore y R. Winski (Eds.), *Handbook of standards and resources for spoken language systems* (pp. 684-732). De Gruyter.

Wilhelm, T., Burghardt, M. y Wolff, C. (2013). To see or not to see – an interactive tool for the visualization and analysis of Shakespeare plays. En R. Franken-

Wendelstorf, E. Lindinger y J. Sieck (Eds.), *Kultur und Informatik: Visual worlds & interactive spaces* (pp. 175-185). Werner Hülsbusch.

Wilkinson, M., Dumontier, M., Aalbersberg, I., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J., Da Silva Santos, L., Bourne, P., Bouwman, J., Brookes, A., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C., Finkers, R., ... y Monsr, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*. https://doi.org/10.1038/sdata.2016.18

Williamsen, V. G. (1978). The structural function of polymetry in the Spanish Comedia. En A. V. Ebersole (Ed.), *Perspectivas de la Comedia* (pp. 33-47). Estudios de Hispanofilia.

Wimsatt, W. K. y Beardsley, M. C. (1954). The intentional fallacy. En *The verbal icon* (pp. 3-20). University of Kentucky Press.

Winter, T. N. (1999). Roberto Busa, S.J., and the invention of the machine-generated concordance. *The Classical Bulletin*, *1*(75), 3-20.

Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice-Hall.

Wisbey, R. (1967). *Vollständige Verskonkordanz zur „Wiener Genesis“. Mit einem rückläufigen Wörterbuch zum Formenbestand*. Erich Schmidt Verlag.

Wisbey, R. y Bott, M. F. (1968). Literary and Linguistic Computing Centre, Cambridge. En J. Stindlová y Z. Skoumalová (Eds.), *Les machines dans la linguistique* (pp. 39-48). De Gruyter. https://doi.org/10.1515/9783112315354-009

W^{3}Techs. (2023). *Comparison of the usage statistics of linux vs. windows for websites*. Consultado el 14 de julio de 2023, desde https://w3techs.com/technologies/comparison/os-linux,os-windows

Zentrum für Datenverarbeitung. (2023). *TUSTEP* (Ver. 2023). Tubinga, Eberhard Karls Universität Tübingen. https://www.tustep.uni-tuebingen.de/