

**Árboles, Gráficos y Matrices de Datos. Codificación en TEI de un Corpus de
Interacciones Parlamentarias con Python**

Marilina Pisani

Facultad de Filosofía y Letras, Universidad Autónoma de Barcelona

Máster en Humanidades y Patrimonio Digitales

Edición 2021-2022

Dra. Núria Bel Rafecas

Universidad Pompeu Fabra

Año de defensa: 2022

Resumen

Inspirada en el concepto de “lectura distante” (Moretti, 2005), en este trabajo se propone una metodología para el etiquetado en TEI de un corpus de interacciones parlamentarias a través de una serie de funciones programadas en Python. Mediante la construcción de una matriz de datos y otras herramientas de visualización de la información, se propone un análisis del corpus que posibilita un etiquetado bien formado y validado por el esquema PARLA-CLARIN. De esta manera, se ofrece una solución al desafío de enriquecer un gran conjunto de datos (Schöch, 2013). Se reflexiona, además, sobre el valor que comportan las competencias en programación y el lugar que ocupan en las humanidades digitales. Un lugar que —a diferencia de lo que ocurre con la anotación manual en los lenguajes de marcado (O’Sullivan et al., 2015)— continúa generando controversias aún hoy (Ramsay, 2013).

Palabras clave: TEI, Python, corpus, ParlaMint, humanidades digitales

Agradecimientos

Este trabajo no habría sido posible sin la formación provista por la Universidad Autónoma de Barcelona y el acompañamiento de mi tutora, Nuria Bel Rafecas. Tampoco sin los recursos brindados por la Universidad Pompeu Fabra, institución donde realicé las prácticas que dieron lugar a esta investigación.

Del mismo modo, la infraestructura del proyecto ParlaMint y las interacciones con sus miembros enriquecieron el desarrollo de este trabajo desde su inicio.

Quiero agradecer también a mis compañeras y compañeros del Máster, por el apoyo constante durante este año de estudios. A la inmensa comunidad en línea de Python, sin la cual no habría comenzado a escribir. A Juan, por acompañarme y aconsejarme sobre el código. A mi mamá, por no dejar de animarme.

A mi papá, por haberme enseñado a programar.

Índice

1. Introducción	5
1.1. Programación Fácil y Amigable	6
1.2. Las Humanidades Digitales: “Un Mosaico de Iniciativas”	8
1.3. Las Guías Directrices del Etiquetado de Corpus	9
1.3.1. El Proyecto ParlaMint	9
1.3.2. La Iniciativa de Codificación de Texto	11
2. Planteamiento del Problema	13
2.1. Definición del Problema	13
2.2. Objetivos	14
2.3. Justificación	15
3. Estado de la Cuestión	16
3.1. Metodologías Utilizadas para el Etiquetado en TEI de Corpus ParlaMint	16
3.2. Propuestas de Automatización del Etiquetado en TEI	19
4. Marco Metodológico	22
4.1. Guía para Comenzar un Proyecto en TEI	22
4.2. Validación de un Corpus ParlaMint	23
4.3. Fundamentos de la Programación	24
5. Codificar en TEI un Corpus de Interacciones Parlamentarias con Python: Los Discursos del Parlamento de Cataluña (2015-2020)	27
5.1. Presentación de la Propuesta Metodológica	27
5.2. Herramientas Utilizadas	30
5.3. Primer Paso: Construcción de un DataFrame	31
5.4. Segundo Paso: Análisis del DataFrame	32
5.4.1. Descripción y resumen	33
5.4.2. Visualizaciones	35
5.5. Tercer Paso: el Etiquetado Distante	37
5.5.1. Técnicas principales	38
5.5.2. Ejemplos de Aplicación	39
5.5.2.1. Comentarios del Transcriptor	39
5.5.2.2. Idiomas	42
5.5.2.3. Intervinientes	43
5.5.2.4. Divisiones	44
5.6. Cuarto Paso: Del DataFrame al Árbol XML/TEI	45
6. Conclusiones	48
Apéndice	52
Referencias	53

1. Introducción

Hace poco más de diez años, en un panel sobre la historia y el futuro de las Humanidades Digitales, Stephen Ramsay (2013) preguntó “¿hay que saber programar?”, y —sabiendo que su respuesta iba a enojar a la mitad de las personas de la sala— respondió: “I’m a tenured professor of digital humanities and I say ‘yes’” [Soy profesor titular de humanidades digitales y digo ‘sí’] (p. 240). Su lectura ese día reavivó un debate en el núcleo de las Humanidades Digitales, después de todo, no es casual que antes se conocieran por el nombre de “Humanidades Computacionales”. Un giro que, como explica del Río Riande (2015), surge en pos de permitir la reflexión humanística sobre lo digital en lugar de limitar la actividad a la aplicación de herramientas informáticas (p. 14).

Como la aplicación y la reflexión suelen ocurrir juntas, la pregunta sigue vigente: ¿hasta qué punto el saber programar es exigible de los humanistas digitales? Quizás el debate se resuelve fácilmente si atendemos a lo que sucede en otras disciplinas. Los astrónomos, por ejemplo, utilizan herramientas que otros construyen, como un telescopio. ¿Necesita el astrónomo saber cómo se construye el telescopio para ser considerado astrónomo? ¿Por qué el caso del humanista debería ser diferente? ¿Son los programas informáticos para el humanista una mera herramienta cuya construcción puede externalizar, o esa construcción —en algunos casos— comporta conocimiento valioso?

A partir de una encuesta a 96 académicos en el área —con el objetivo de determinar el nivel en el que los estudiosos de las humanidades digitales programan activamente y cómo ven la importancia de dichas actividades— O’Sullivan et al. (2015) señalan que sólo una pequeña mayoría —el 52.1%— considera a la programación parte de su trabajo (p. 147). Advierten, no obstante, que no existe acuerdo en lo que se considera “programación”: muchos de los encuestados destacan actividades como el uso de los lenguajes de marcado, en particular HTML y XML, frente al uso de lenguajes de programación dinámicos más sofisticados (p. 147).

Esto resulta cierto si atendemos a la buena aceptación que tienen las guías TEI (Text Encoding Initiative, o en español “Iniciativa de Codificación de Texto”), un esquema de marcado que se ha convertido —desde su creación en 1987— en el estándar técnico de referencia para la representación de contenido textual en las humanidades.

La anterior analogía con el telescopio acaso es acertada pero por otro motivo. Analizando el impacto que tuvo su desarrollo durante el siglo dieciséis Masterman (1962) señalaba que, el telescopio, al ampliar la gama completa de lo que sus poseedores podían ver y hacer, al final, fue un factor que cambió toda su imagen del mundo (p. 38). La autora luego reflexiona sobre el impacto que podrá tener la nueva computadora digital, cuya capacidad de procesar datos “is so great as to make of it the telescope of the mind” [es tan grande como para convertirla en el telescopio de la mente] (Masterman, 1962, p. 39). Sus palabras resuenan a lo que Moretti define como “lectura distante”: una forma de conocer donde el texto sufre un proceso deliberado de “reducción y abstracción” que permite visualizar sus características globales (Moretti, 2005, p. 1).

Siguiendo estas reflexiones, la motivación del presente trabajo es indagar qué visión —telescópica o *distante*— le aportará al humanista digital saber “escribir código”. Para ello, se presenta una propuesta para la codificación —siguiendo las guías TEI— de un corpus de discursos parlamentarios a través de un código en Python, uno de los lenguajes de programación más valorados del momento.

1.1. Programación Fácil y Amigable

En las primeras páginas de *Automate the Boring Stuff with Python* (Automatizar las Cosas Aburridas con Python), Sweigart presenta de manera muy simple un problema muy frecuente: muchas personas pasan horas haciendo clic y tecleando para realizar tareas repetitivas, sin saber que la máquina que utilizan podría hacer su trabajo en segundos si le dieran las instrucciones adecuadas (Sweigart, 2015, p. 2). Programar es, para el autor,

“simply the act of entering instructions for the computer to perform” [Simplemente el acto de introducir instrucciones para que el ordenador las realice] (p. 3).

Creado en 1991 por Guido van Rossum, Python es actualmente uno de los lenguajes de programación de código abierto más valorados por informáticos y no informáticos. En el sitio oficial se declara que Python es “amigable” —existe una amplia comunidad en línea que ofrece instructivos, cursos y respuestas a los interrogantes más específicos— y “fácil de aprender” —una gran cantidad de libros introductorios a la programación lo utilizan en sus lecciones—. Según su creador, su éxito se debe posiblemente a la facilidad del lenguaje para ser ampliado o extendido (Venners, 2003). Esto significa que cualquiera puede escribir partes de código que cumplen ciertas funciones y que otros pueden volver a utilizar, algo que se conoce como “librerías”.

“Python” refiere tanto al lenguaje de programación como al interpretador que lee el código y realiza las instrucciones (Sweigart, 2015, p. 4). Existen diversas interfaces web que permiten crear código, incluir texto e imágenes y compartir el trabajo a través del navegador. Google Research, por ejemplo, lanzó Colaboratory (Colab) como una alternativa para la colaboración en proyectos de programación en Python.

Otro motivo que explica la gran aceptación de este lenguaje es su versatilidad. Actualmente es utilizado para diversas aplicaciones como desarrollo web, estudios científicos, educación o sistemas comerciales.

Python también goza de cierta popularidad dentro de las humanidades digitales. Existen libros introductorios (Kokensparger, 2018), instructivos en línea (“The Programming Historian”) e incluso canales de youtube (“Programming for Digital Humanities”). Estos recursos presentan posibilidades para el humanista si usa el lenguaje de programación para sus estudios: desde técnicas para la minería de texto hasta, por ejemplo, herramientas de visión por computador. Esta diversidad de propuestas responde a que las humanidades digitales son, como señala Río Riande, “un mosaico de iniciativas” (Río Riande, 2015, p. 8).

1.2. Las Humanidades Digitales: “Un Mosaico de Iniciativas”

Proponer una definición exhaustiva de las humanidades digitales parece ser una tarea pendiente —acaso imposible— desde que el campo comenzó a configurarse hace varias décadas. Este fenómeno es manifiesto frente a la pluralidad de publicaciones sobre la cuestión (Gold, 2011; Kirschenbaum, 2010; Vanhoutte et al., 2013; Río Riande y Gonzalez Blanco, 2015) y debates aún abiertos (Gold, 2012; Gold y Klein, 2019; Stommel y Kim, 2018). La tarea se vuelve aún más difícil cuando atendemos no sólo a la diversidad de los objetos de estudio o metodologías de trabajo sino a las diferentes situaciones sociopolíticas del gran abanico geográfico en donde se desarrolla la actividad (Río Riande, 2015, p. 32).

Siguiendo a González-Blanco (2013), los principales campos de investigación en Humanidades Digitales son: la aplicación de las bases de datos a la ordenación de materiales, el etiquetado de textos (basados en XML y con el estándar TEI), la digitalización del patrimonio, la estadística y los sistemas de visualización de datos (p. 56). Dentro de estos últimos, encontramos al concepto de “lectura distante”. Impulsado a partir de la publicación de *Graphs, Maps and Trees* (Moretti, 2005, puede traducirse como “Gráficos, Mapas y Árboles”), consiste en la interpretación de grandes cantidades de datos a través de distintas representaciones. Al ser tomadas prestadas de las ciencias, algunas críticas denuncian que estas herramientas gráficas esconden en realidad sesgos epistemológicos en relación a su transparencia e independencia del observador (Drucker, 2011, párr. 6). En este sentido, Drucker demanda que no sean adoptadas sin un examen crítico y sin toda la fuerza de los conocimientos teóricos (párr. 7).

Algunos humanistas digitales utilizan herramientas informáticas motivados por el tamaño del objeto de sus estudios, demasiado grande para ser analizado con las técnicas tradicionales. Un ejemplo de este tipo de objetos son los corpus, a los que Rojo (2021) define como:

“un conjunto de (fragmentos de) textos, orales o escritos, producidos en condiciones naturales, conjuntamente representativos de una lengua o una variedad lingüística,

en su totalidad o en alguno(s) de sus componentes, que se almacenan en formato electrónico y se codifican con la intención de que puedan ser analizados científicamente” (Rojo 2014a, 371 como se citó en Rojo, 2021, p. 1).

En el siguiente apartado se ofrece una introducción a los estudios de corpus de discursos parlamentarios y su abordaje a través del etiquetado en TEI.

1.3. Las Guías Directrices del Etiquetado de Corpus

Durante la última década, dada su disponibilidad y riqueza de contenido, los estudios de corpus basados en la actividad parlamentaria han crecido notablemente. Según Truan y Romary (2022), los corpus parlamentarios comportan especial valor por dos razones. En primer lugar, las sesiones ya están transcritas por un equipo de taquígrafos profesionales familiarizados con los procedimientos parlamentarios, así como con los diputados, lo que permite al investigador centrarse en otros niveles de transcripción y anotación (párr. 7). En segundo lugar, al estar en la interfaz entre los datos hablados y los escritos, el discurso parlamentario da acceso a una amplia gama de características del discurso (párr. 9).

1.3.1. El Proyecto ParlaMint

A nivel europeo, un caso de este tipo de estudios es el proyecto [ParlaMint](https://www.clarin.eu/parlamint) (ParlaMint: Towards Comparable Parliamentary Corpora)¹, que contribuye a la creación de corpus multilingües comparables y uniformemente anotados de sesiones parlamentarias, centrados principalmente en la pandemia de COVID-19 (Erjavec et al., 2022, párr. 4). Atendiendo al problema de la falta de interoperabilidad entre corpus lingüísticos codificados de distintas maneras —en ausencia de un criterio compartido— el proyecto ofrece orientaciones y define un esquema basado en las Directrices TEI².

Erjavec et al. (2022) añaden otras razones por las que los corpus de los discursos parlamentarios resultan de interés. Al tratarse de transcripciones del lenguaje hablado

¹ Accesible a través de: <https://www.clarin.eu/parlamint>

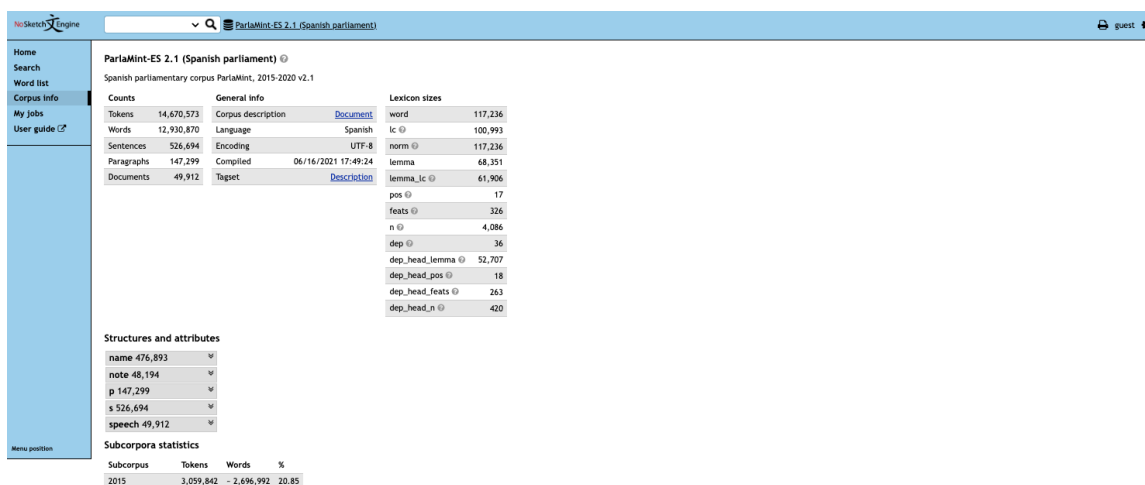
² <https://github.com/clarin-eric/ParlaMint/blob/main/Schema/ParlaMint-TEI.rng>

producidas en circunstancias controladas y reguladas, son ricas en valiosos metadatos sociodemográficos. Además, están disponibles —usualmente— online pero sin estar sujetos a protecciones de copyright, en vistas de mejorar el funcionamiento de los sistemas democráticos (párr. 2).

Hasta el momento, los corpus de la primera edición del proyecto son accesibles a través del repositorio de [CLARIN.SI](http://clarin.si)³ y, por medio de plataformas como NoSketch Engine (clarin.si/noske), que permiten su explotación a partir de una interfaz similar a la de [Sketch Engine](https://www.sketchengine.eu/)⁴, pero de código abierto. A través de ellas, entre otras cosas, es posible: encontrar información básica sobre el corpus (Figura 1); realizar investigaciones sobre elementos léxicos a través de listas de palabras o sus frecuencias; indagar sobre aspectos semánticos a través del análisis de concordancias; o explorar cuestiones sociolingüísticas —donde los metadatos resultan indispensables—.

Figura 1

Información básica de un corpus en NoSketchEngine



The screenshot shows the NoSketch Engine interface for the 'ParlaMint-ES 2.1 (Spanish parliament)' corpus. The interface is divided into several sections: Home, Search, Word list, Corpus info, My Jobs, and User guide. The 'Corpus info' section is active, displaying various statistics and details about the corpus.

Counts		General info		Lexicon sizes	
Tokens	14,670,573	Corpus description	Document	word	117,236
Words	12,930,870	Language	Spanish	lc	100,993
Sentences	526,694	Encoding	UTF-8	norm	117,236
Paragraphs	147,299	Compiled	06/16/2021 17:49:24	lemma	48,351
Documents	49,912	Target	Description	lemma_lc	61,906
				pos	17
				feats	326
				n	4,086
				dep	36
				dep_head_lemma	52,707
				dep_head_pos	18
				dep_head_feats	263
				dep_head_n	420

Structures and attributes			
name	476,893		
note	48,194		
p	147,299		
s	526,694		
speech	49,912		

Subcorpora statistics			
Subcorpus	Tokens	Words	%
2015	3,059,842	2,696,992	20.85

Nota. La imagen muestra la información del corpus de los discursos parlamentarios del Congreso de Diputados de España.

³ <http://www.clarin.si/info/about/>

⁴ <https://www.sketchengine.eu/>

El menú de búsqueda permite filtrar el corpus a partir de los siguientes campos: subcorpus, fecha, término o sesión y a partir de los datos del interviniente como su rol en la sesión, su afiliación, edad o género. Un correcto etiquetado siguiendo las Guías Directrices TEI posibilita un buen funcionamiento de estos filtros.

1.3.2. La Iniciativa de Codificación de Texto

La TEI es un consorcio que desarrolla y mantiene colectivamente un estándar para la representación de textos en formato digital, siendo las Guías Directrices su núcleo central (Allés Torrent, 2019). En ellas, se define un lenguaje de marcado que permite etiquetar características textuales estructurales (títulos, capítulos o párrafos), semánticas (personas, lugares) o gramaticales⁵.

El esquema de codificación TEI consta de una serie de módulos, cada uno de los cuales declara determinados elementos XML y sus atributos (The TEI Guidelines, 2022a). A diferencia de una base de datos relacional, la estructura de estos elementos es caracterizada como un árbol, que contiene un elemento raíz y elementos hijos. Así, las relaciones entre los elementos son jerárquicas: un capítulo es un elemento padre de un párrafo pero un elemento hijo de un tomo, por ejemplo.

Algunas de sus aplicaciones prácticas han sido ediciones digitales, bibliotecas o archivos virtuales, manuscritos, diccionarios y corpus (Allés Torrent, 2019). Bajo las Directrices TEI, los corpus son considerados como textos compuestos más que unitarios, que forman parte de un objeto más grande. Esta relación se define a través de los elementos <teiCorpus> y <TEI>. El primero contiene toda la colección de textos, mientras que cada texto individual pertenece al segundo. Las guías también recomiendan la inclusión de información contextual, como datos personales de los participantes de una interacción o datos de publicación de los textos (The TEI Guidelines, 2022c).

⁵ A lo largo del trabajo se usan de manera intercambiable los términos “etiqueta” y “marca”, por un lado, y “anotación”, “marcación” y “codificación”, por otro. En general, la “etiqueta” o “marca” refieren a un elemento, que puede tener atributos. “Anotación”, “marcado” o “codificación” nombran la actividad de asignar etiquetas al texto.

Cuando se trata de corpus de gran tamaño, las guías aconsejan personalizar los esquemas para el proyecto en cuestión. A su vez, indican como buena práctica la separación de elementos entre aquellos que son requeridos, recomendados, opcionales o prohibidos. Estas recomendaciones responden al “elevado coste de la identificación y codificación de muchos rasgos textuales, y la dificultad de garantizar una práctica coherente en corpus muy grandes” (The TEI Guidelines, 2022c). El etiquetado correcto de un corpus es, entonces, una tarea compleja.

Dentro del “mosaico de iniciativas” de las humanidades digitales, en este trabajo se responde al desafío que presenta el etiquetado de un corpus de gran tamaño. Esta respuesta consiste en los lineamientos de una metodología cuyo eje central es un lenguaje de programación. El trabajo se divide en cinco capítulos. En el capítulo siguiente se describe en más detalle el problema. En el tercer capítulo, se presentan algunas de las soluciones disponibles y se analizan sus aspectos positivos y sus dificultades. En el cuarto, se desarrolla el marco metodológico de la nueva propuesta. En el quinto, se aplica la metodología a un ejemplo: el corpus de los discursos del Parlamento de Cataluña entre los años 2015 y 2020. Finalmente, en el último capítulo se presentan las conclusiones y se reflexiona sobre el lugar de la programación en las humanidades digitales.

2. Planteamiento del Problema

2.1. Definición del Problema

En el marco del proyecto ParlaMint, el objeto de estudio de este trabajo es el corpus de los discursos del Parlamento de Cataluña entre los años 2015 y 2020, compuesto por más de 200 documentos de Microsoft Word de extensión DOCX, que fueron facilitados por la misma institución. Este corpus debe ser codificado siguiendo las guías TEI y los procesos de validación del proyecto.

Ahora bien, el tamaño del corpus parece ser un impedimento para una codificación de calidad y —frente a la falta de una metodología estándar— no es claro cómo alcanzar un equilibrio. Este problema es una instancia de una cuestión más general a la que Schöch (2013) se refiere como la posibilidad de combinar —en los estudios humanísticos— los enfoques de Smart y Big Data (párr. 31). Por big data —en general— se entiende a grandes volúmenes de datos, desorganizados, crudos y de actualización permanente. Los smart data —para el autor— son datos estructurados o semiestructurados; son explícitos y enriquecidos, porque además de los datos brutos, contienen marcas, anotaciones y metadatos (párr. 11). Las ediciones digitales producidas siguiendo las Directrices TEI son un ejemplo paradigmático de smart data.

A pesar de su atractivo, la desventaja de los smart data es que “no son escalables, no pueden ser automatizados, porque en la mayoría de los casos debe llevarse a cabo la anotación manual por parte de los investigadores, y eso toma mucho tiempo” (Allés Torrent, 2019, p. 21). Así, la pregunta fundamental que se busca responder en este trabajo es:

- ¿Cómo puede el humanista digital transformar un gran conjunto de discursos parlamentarios —big data— en un corpus codificado siguiendo las Directrices TEI —smart data—?

Para resolver este problema, Schöch propone automatizar el etiquetado. Para el autor, la automatización consiste en un proceso heurístico donde se descubren estructuras y relaciones implícitas en los datos⁶. Algunas de estas propuestas se analizarán en el capítulo siguiente. Sin embargo, la automatización presenta un desafío adicional en relación al uso de las herramientas tecnológicas en los proyectos de las humanidades digitales. Este uso, como ya señaló Drucker (2011), no puede implicar la renuncia a los compromisos epistemológicos básicos de las humanidades. Entonces, ¿cómo automatizar el proceso manteniendo la dimensión interpretativa humanística?

2.2. Objetivos

Para responder a estas preguntas, en el trabajo se persiguen los siguientes objetivos:

- Proponer una metodología para la codificación automática de los discursos del Parlamento de Cataluña —entre los años 2015 y 2020— a través de un programa en Python.
 - Codificar los discursos del Parlamento de Cataluña —entre los años 2015 y 2020— siguiendo las Directrices TEI y los procesos de validación del Proyecto ParlaMint.
 - Ofrecer una herramienta de libre acceso —en el formato de un cuaderno de Google Colab en Python— diseñado para codificar los discursos del Parlamento de Cataluña, pero con la flexibilidad de ser utilizado para otros proyectos.
 - Demostrar la importancia de utilizar un lenguaje de programación —Python— para la codificación según las Directrices TEI de grandes volúmenes de texto.

⁶ El autor propone una combinación entre automatización y crowdsourcing. El crowdsourcing consiste en distribuir tareas pequeñas a una gran cantidad de voluntarios que pueden, tanto realizar anotaciones como comprobar las resultantes de un proceso de automatización (Schöch, 2013, párr. 32 y 33). Sin embargo, en los últimos años algunos autores han comenzado a cuestionar al crowdsourcing en función de argumentos éticos (Keralis, 2018). No será considerada en este trabajo.

2.3. Justificación

La relevancia de contar con corpus codificados para ser luego objetos de investigación por académicos —o incluso, por el público en general— ha sido extensamente presentada por otros autores (Erjavec et al., 2022; Truan y Romary, 2022). Es necesario, además, que esta codificación sea precisa, ya que de esto dependerá la corrección de las investigaciones posteriores; reproducible, de manera de facilitar las revisiones por pares; automatizable, para evitar un exceso de trabajo manual por parte del humanista; adaptable, si hay cambios en el esquema; aumentable, con la ocurrencia de nuevas sesiones parlamentarias.

Por otro lado, a pesar de que el pensamiento computacional es considerado una competencia básica y está incluida dentro de las enseñanzas mínimas de la educación infantil, no parece ocurrir lo mismo en el caso de la educación superior⁷. En este sentido, se espera que mostrar las posibilidades que las competencias en programación ofrecen a los humanistas, promueva un rol protagónico de éstos en el desarrollo de nuevos métodos y herramientas.

⁷ Real Decreto 95/2022, de 1 de febrero, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Infantil. *Boletín Oficial del Estado*, 28, de 2 de febrero de 2022. <https://www.boe.es/eli/es/rd/2022/02/01/95>

3. Estado de la Cuestión

3.1. Metodologías Utilizadas para el Etiquetado en TEI de Corpus ParlaMint

A pesar de las pautas propuestas por el Proyecto ParlaMint, las metodologías utilizadas para la construcción de los corpus por cada participante —documentadas por Erjavec et al. (2022) e ilustradas en la Tabla 1— fueron diversas.

Tabla 1

Herramientas utilizadas para la conversión al formato ParlaMint de los corpus individuales

Participante	Formato de los documentos fuente	Herramienta utilizada
Bélgica	HTML (exportados de MS Word)	XSLT, Python
Bulgaria	HTML	
Croacia	JSON	
República Checa	Publicación anterior.	
Dinamarca	XML	
Francia	XML	Python, Perl, XSLT
Hungría		Códigos dedicados
Islandia	HTML	Python
Italia	HTML (Akoma Ntoso XML)	Java
Latvia	HTML	JSON, Python
Lituania		Python
Países Bajos	XML	XSLT
Polonia	Publicación anterior.	Python
Eslovenia	Publicación anterior.	XSLT
España	XML	Perl, Python y Bash, XSLT
Turquía	HTML	Python
Reino Unido	XML, RDF	XSLT

Nota. Elaboración propia a partir de (Erjavec et al., 2022, párr. 15-69)

Esta diversidad metodológica responde —entre otras cosas— a las diferencias en los formatos de los texto fuente: HTML, JSON, y XML. Si bien en el Github del proyecto se

comparten códigos de muestra en Python para la anotación lingüística de los corpus —aunque se observa en la Tabla 1 que muchos participantes optaron por este lenguaje— no se presentan modelos de código para la construcción inicial del corpus. También llama la atención que no se encuentren ficheros DOCX entre los formatos de los textos fuente. A pesar de que no se pudo acceder a los documentos originales de otras sesiones parlamentarias —lo que imposibilita una comparación entre las fuentes y los métodos elegidos para procesarlas— se reconoce que contar con el documento original en DOCX del transcriptor tiene la ventaja de recuperar su marcado original.

Un antecedente muy cercano a este trabajo es la primera propuesta de codificación de los discursos del Parlamento de Cataluña que Antiba (2021) realizó durante su participación en el proyecto ParlaMint y que documenta en su Trabajo de Fin de Máster. La metodología que siguió Antiba incluyó los siguientes pasos:

1. La conversión de los documentos originales —en formato DOCX— a XML, a través del convertidor en línea recomendado por el proyecto.
2. La codificación inicial, mediante la herramienta de “Búsqueda y Sustitución” del programa Oxygen XML Editor 23.1.⁸
3. Limpieza del texto, que implica “sacar las marcas de estilo provenientes del texto en formato docx de las transcripciones que dificultan la transformación del archivo hacia el formato de esquema propuesto por ParlaMint” (p. 23).
4. Correcciones necesarias para la transformación a través de una hoja de estilo (XSL):
 - a. Identificación de intervinientes no nombrados por su nombre sino por su cargo. En muchos casos, el nombre aparece a continuación de la descripción —“La consellera de Salut (Alba Vergés i Bosch)”—, pero en otros no —“El president”—. Adicionalmente, la manera en que son nombrados los cargos pueden cambiar en función de la sesión —Peré Aragonés en ocasiones

⁸ Por ejemplo, “la sustitución de todos los valores “D3/ Text normal” y “D3/ Intervinent” en “Text” y “Speaker” correspondientemente” (p. 24)

aparece como “Vicepresident del Govern i Conseller d’Hisenda i Economia”, otras como “Vicepresident” o “Conseller d’Hisenda”.

- b. Remoción de saltos de línea provocados por la conversión del DOCX a XML.
 - c. Corrección de nombres de intervinientes mal escritos.
 - d. Corrección de intervenciones marcadas como intervinientes y viceversa.
 - e. Inclusión del idioma del discurso en función del hablante.
5. Creación de una plantilla XSL que adecue el formato al propuesto por el Proyecto ParlaMint. Esta hoja de estilo “tiene la estructura de <teiHeader> al principio y se elige cada elemento Speaker como inicio elemento padre o <u> en el XML resultante, a la vez que se crea un contador de repeticiones por cada elemento <u>” (p. 27). Además, se realiza una búsqueda de coincidencia exacta entre el elemento Speaker y los metadatos. Una vez creado los elementos <u>, cada elemento Texto se convierte en hijo de <u> y se le asigna el nombre <seg> (págs. 22-23).

La recopilación de los metadatos —datos personales de los intervinientes, como fecha de nacimiento y afiliaciones— la realizó de manera manual a través de páginas web y quedaron a disposición en un fichero EXCEL.

Como líneas para trabajos futuros, el autor señala: primero, la identificación de los intervinientes faltantes y, segundo, la marcación de las notas o comentarios del taquígrafo que quedaron como parte de los discursos debido a un error en la conversión (p. 32). En relación a la identificación de los intervinientes nombrados por sus cargos, señala como dificultad el trabajo manual implicado: “ejemplos como éste y llevados a todos los parlamentarios que ostentan un cargo dentro del gobierno crea un problema dificultoso de automatizar o lento a la hora de corregir con Búsqueda y Sustitución en los archivos” (p. 25).

En trabajos recientes de etiquetado a partir de las guías TEI, se advierte la misma tensión entre precisión y cantidad. Truan y Romary (2022) reconocen que el pequeño tamaño de los

corpus contruidos en su investigación —de aproximadamente 137 000 tokens para el corpus francés a 417 000 para el corpus alemán— permitió una anotación fina que puede ser más difícil de implementar a mayor escala (p. 60). Por otro lado, Arano (2020) —en relación a la edición digital de un “capbreu”— señala que “proponer una codificación TEI teniendo como perspectiva un futuro procesamiento automatizado ayudaría a la gestión más eficaz, a medio o largo plazo de los datos” (p. 67).

3.2. Propuestas de Automatización del Etiquetado en TEI

Fuera de la utilización de hojas de estilo que, por lo observado anteriormente, no resultan lo suficientemente flexibles para realizar una anotación confiable y exhaustiva de un corpus, otros autores han presentado algunas alternativas de automatización.

Entre las propuestas dentro del área de las Humanidades Digitales, se destaca la investigación —en curso— de Janes et al. (2021). Las autoras (con formación en ciencias, historia y literatura medieval) y el autor (filólogo), indagan sobre la posibilidad de transformar imágenes en documentos XML-TEI. Su trabajo puede enmarcarse dentro de una tradición más amplia que estudia la codificación automática a partir del reconocimiento de texto en imágenes (OCR).

Dentro de las ciencias de la computación, merecen especial atención los trabajos de Khemakhem. Entre ellos, GROBID, una biblioteca de aprendizaje automático para extraer, analizar y reestructurar documentos en bruto —como PDF— en documentos estructurados y codificados en XML/TEI (Khemakhem, s.f.). El estudio que lo lleva a construirla se encuentra en su tesis doctoral. Allí, el autor propone el análisis —*parseo*— de diccionarios impresos mediante modelos léxicos y la generación de resultados codificados (en TEI y LMF).

Khemakhem menciona dos estrategias utilizadas para analizar la estructura de documentos impresos: una, basada en reglas y, la otra, en el aprendizaje automático. La primera, que el autor ubica temporalmente con el nacimiento de los sistemas expertos, consiste en el

establecimiento de reglas determinísticas que son definidas mediante la observación de patrones en los datos (p. 21). Estas reglas tienen una estructura condicional básica “si... entonces...”. En algunos casos, se hace uso de la información de la tipografía y de las marcas textuales. Como ventajas de esta estrategia, Khemakhem señala que, primero, las reglas pueden ser rápidamente implementadas y, segundo, ofrecen una manera de simular a gran escala el proceso de decisión de un experto. Sin embargo, según el autor, resulta un abordaje limitado para el análisis sintáctico de descripciones léxicas profundas y extensas. En este sentido, afirma que cualquier regla definida por los humanos es demasiado subjetiva para cubrir todos los patrones ocultos en el cuerpo de tales entradas léxicas (p. 22).

La segunda estrategia que analiza el autor es la de los modelos probabilísticos. A partir de las capacidades del aprendizaje automático, estos modelos permiten ir más allá de la simple búsqueda de patrones de los expertos humanos (pp. 22-23). Son una opción cuando las reglas de experto son imposibles de definir. Así, Khemakhem aborda la cuestión de la codificación automática como la tarea de asignar una etiqueta de un conjunto de posibles etiquetas a cada palabra de una secuencia de texto. Una tarea habitual en el campo del procesamiento del lenguaje natural que es abordada a través de modelos en grafo (p. 23). La suposición clave en la modelización gráfica es, señala el autor, que una distribución sobre muchas variables puede representarse a menudo como un producto de funciones locales que depende de un subconjunto más pequeño de variables (p. 23). Esta es la estrategia que Khemakhem aborda en su tesis pero que, para alguien que no se especializa en estudios computacionales o que no tiene una formación en ciencias, resulta difícil seguir.

Es de especial interés que en la introducción de su tesis, Khemakhem señala como interrogante fundamental “to what extent could a collaboration be possible between the computational and the humanist?” [¿hasta qué punto podría ser posible una colaboración entre lo *computacional* y el *humanista*?] (énfasis en el original, p. 3). Su trabajo implicó la participación de humanistas encargados del etiquetado en TEI y de la definición de los

esquemas y proporcionó, además, una herramienta tecnológica para ser explotada dentro del área de las humanidades.

A partir de estas investigaciones, si la exigencia es obtener un proceso automático y escalable del etiquetado en TEI, el humanista puede obtener la ayuda de un experto en programación y encomendar la tarea de automatización. En este sentido, es cierta la recomendación con la que O'Sullivan et al. (2015) finalizan el análisis de sus encuestas: “you do not ‘have’ to code, as long as you can work —effectively— with someone who does” [“no tienes que ‘programar’, siempre que puedas trabajar —eficazmente— con alguien que lo haga”] (p. 147).

Sin embargo, en este trabajo se mostrará que existe un término medio entre las hojas de estilo —habituales entre los humanistas— y los modelos probabilísticos —habituales entre los expertos en computación—.

4. Marco Metodológico

A pesar de que no existe una metodología precisa para la codificación automática de los corpus de los discursos parlamentarios, sí están disponibles recursos metodológicos relevantes para su desarrollo. En primer lugar, la comunidad TEI pone a disposición innumerables ejemplos, esquemas e instructivos. En este trabajo, se utilizará la guía de preguntas y respuestas que presenta Romary (2009). En segundo lugar, al estar enmarcado dentro de un proyecto de codificación más grande, como lo es ParlaMint, el etiquetado deberá seguir las recomendaciones propuestas. En tercer lugar, como este trabajo presentará la alternativa de utilizar un lenguaje de programación para realizar el etiquetado, resulta imperioso contar con directrices en relación a la construcción de programas y al análisis de los datos.

4.1. Guía para Comenzar un Proyecto en TEI

Elaboradas para los recién llegados a TEI, Romary (2009) propone una serie de preguntas que todo investigador debe hacerse antes de comenzar un proyecto.

1. ¿Para qué se necesita el etiquetado? En este sentido, la planificación será distinta, señala el autor, si se trata de un proyecto de archivística, académico, de disseminación o de una edición digital. Con respecto a la dimensión académica, sostiene que resulta necesario determinar qué aspectos del proyecto de codificación serán específicos para la investigación prevista.
2. ¿Qué necesito realmente (realmente) etiquetar? La respuesta a esta pregunta puede involucrar distintos niveles: la macroestructura del documento (la organización del texto en divisiones y su estructura interna compuesta por elementos diversos como párrafos, figuras, ejemplos, etc.); la documentación (qué fuentes de información se incluirán); anotación de la superficie (cuáles son los elementos que es necesario identificar en el texto y etiquetar —nombres, lugares, expresiones temporales—)

3. ¿Cuál es el material disponible? Las prioridades y actividades variarán en función de si la fuente es una edición impresa, la retro conversión de una fuente digital, una fuente nacida digitalmente (“born digital” en inglés) o un manuscrito (p. 5).

Por otro lado, el autor también enumera las siguientes herramientas para comenzar a trabajar: un buen editor XML para controlar que el archivo esté siendo creado siguiendo el esquema TEI; acceso a Roma, un programa para definir la variante TEI y generar el esquema correspondiente; y acceso a la documentación de la TEI (p. 14).

4.2. Validación de un Corpus ParlaMint

Además de estar bien formado —es decir, ser gramaticalmente correcto en el lenguaje XML—, un documento XML/TEI debe estar validado con respecto al esquema de codificación elegido. Para garantizar la interoperabilidad entre los corpus de distintos países, el proyecto ParlaMint lanzó una serie de recomendaciones —Parla-CLARIN— y un conjunto de códigos de validación para comprobar la corrección de los documentos que se incorporan. Esta validación se lleva a cabo a través del directorio en Github del proyecto, lugar en el que también se encuentran las recomendaciones.

Estas recomendaciones adoptan un abordaje descriptivo, en la que se conservan, en la medida de lo posible, las distinciones de los datos originales en la codificación de destino (Erjavec y Pančur, 2022). Sin embargo, se intenta limitar las opciones disponibles de etiquetado a las aplicables a los corpus de las interacciones parlamentarias. En la documentación, resulta explícito que sólo serán válidos los documentos anotados a partir de ellas (Erjavec y Pančur, 2022).

La máxima principal es capturar la mayor cantidad posible de texto y marcas de la fuente (Erjavec y Pančur, 2022). Los requerimientos generales se han resumido en la Tabla 2.

Tabla 2*Requerimientos generales de PARLA-CLARIN*

Nombre del requerimiento	Descripción
Codificación	UTF-8
Documentación del proceso	Elemento <editorialDecl>
Idiomas	@xml:lang para cada elemento de texto o su antecesor
Identificadores	@xml:id con valor único para todo elemento que pueda ser referenciado.
Información temporal	@when; @from y @to; @notBefore, @notAfter
Inclusión de archivos	Mecanismo XInclude, elemento <include>

Nota. Elaboración propia a partir de Erjavec y Pančur (2022).

Cada país participante debe proporcionar —y validar— tres tipos de archivos XML/TEI: el archivo “corpus” que contiene los metadatos y las referencias a los demás archivos, las transcripciones etiquetadas y la anotación lingüística. No se detallan en este trabajo los requerimientos para la anotación del corpus ni de la anotación lingüística, por quedar fuera del alcance de la tarea propuesta.

Las transcripciones de los discursos parlamentarios —el núcleo de las recomendaciones PARLA-CLARIN— siguen el capítulo de las guías TEI sobre “Transcripciones del Habla” (Text Encoding Initiative, 2022b). En la Tabla 3 se propone un resumen de los principales aspectos a tener en cuenta. Resulta de interés que, a pesar de las limitaciones, encontramos libertad editorial en varias cuestiones centrales como la anotación de los comentarios de los transcriptores y las divisiones del texto.

4.3. Fundamentos de la Programación

El último aspecto a considerar es cómo abordar la programación del etiquetado, de manera de obtener un proceso semiautomático. Dentro de los recursos acerca de la programación en Python para humanistas, podemos destacar a “The Programming Historian” (TPH).

Tabla 3*Elementos esenciales en las recomendaciones PARLA-CLARIN*

Elemento	Jerarquía	Etiqueta	Atributos	Observaciones
Discursos	Elemento	<u>	@who (la persona que habla) @ana (el rol que tiene)	El atributo @ana puede tomar dos valores: "member" o "chair". ^a
	Sub Elemento	<seg>	@xml:lang	
Comentarios del transcriptor	Elemento	<note>	@type	Se recomienda ubicar estos elementos lo más alto en la jerarquía posible, a menos que se trate de interrupciones.
		<vocal>	@who	
		<kinesic>	@who	
		<incident>	@who	
Pausas	Elemento	<gap>	@reason	El atributo "@reason" puede tomar los valores "inaudible" o "editorial"
	Sub Elemento	<desc>	@xml:lang	
Referencias, preguntas y respuestas	Elemento	<u>	@who (la persona que habla) @toWhom (a quien es dirigida la acción) @ana	El atributo "@ana" puede tomar los valores "question" o "answer".
Resultados de la votación	Elemento	<measure>	@corresp @quantity	Si son mencionados como comentario.
	Elemento	<listEvent>	@type	Al comienzo del texto, dentro del elemento <body>
Divisiones del texto	Elemento	<div>	@type	
	Sub Elemento	<head>		

Nota. Elaboración propia a partir de Erjavec y Pančur (2022).

TPH es un proyecto voluntario, fundado en 2008 por William J. Turkel y Alan MacEachern, que ofrece tutoriales para especialistas en humanidades y ciencias sociales interesados en aplicar herramientas informáticas a sus investigaciones. Entre los recursos dedicados a Python, se encuentra el instructivo "From HTML to List of Words (part 1)" (Del HTML a una Lista de Palabras). Allí, se recomienda escribir un algoritmo primero en lenguaje natural

(inglés o español, por ejemplo) indicando paso a paso las acciones que deben realizarse (Turkel y Crymble, 2012).

Además, los mismos autores, en el tutorial “Code Reuse and Modularity in Python” explican cómo un código escrito se puede reutilizar a través de lo que se conoce como la definición de funciones y módulos (Turkel y Crymble, 2012). Las funciones son rutinas lo suficientemente generales como para volver a ser utilizadas dentro del mismo programa y, similarmente, los módulos son códigos que han sido guardados previamente y quedan a disposición para ser utilizados en otros programas (Turkel y Crymble, 2012).

Python es también utilizado para el análisis de datos. Ahmed y Mukhiya exponen cuatro etapas principales del análisis exploratorio de los datos: la definición del problema, donde se definen los objetivos del análisis, las preguntas y principales variables; la preparación de los datos, donde se definen las fuentes y los tipos de datos, y se llevan a cabo procesos de limpieza (eliminando datos o transformándolos); el análisis de los datos, donde se utilizan conceptos de estadística descriptiva para resumir los datos y encontrar correlaciones escondidas; y el desarrollo y representación de los resultados, a través de visualizaciones gráficas o resúmenes.

Los autores recomiendan la librería “pandas” para trabajar con los datos y explican una serie de técnicas como el subconjunto o la indexación de datos y la creación de análisis visuales. El principal objeto de esta librería es el DataFrame, un conjunto de “datos tabulares bidimensionales, de tamaño variable y potencialmente heterogéneos” (Pandas, s.f.). Este objeto está en el centro de la propuesta metodológica del presente trabajo, que se desarrolla en la siguiente sección.

5. Codificar en TEI un Corpus de Interacciones Parlamentarias con Python: Los Discursos del Parlamento de Cataluña (2015-2020)

5.1. Presentación de la Propuesta Metodológica

Respondiendo a los interrogantes básicos que Romary (2009) propone considerar antes de comenzar un proyecto de etiquetado en TEI, se advierte que la codificación del corpus de los discursos del Parlamento de Cataluña se realiza dentro de una dimensión académica —promocionada por el proyecto ParlaMint—, donde resultan de interés los fenómenos lingüísticos, sociodemográficos y políticos. Por este motivo, especial atención se presta al etiquetado de los hablantes (de lo que depende una correcta asociación con los metadatos), de los comentarios del transcriptor (ricos en la descripción de fenómenos propios de las interacciones de este estilo) y de los idiomas (para una correcta anotación lingüística).

Según las recomendaciones PARLA-CLARIN, el etiquetado debe: identificar cada documento, discurso y párrafos con identificadores únicos; marcar intervinientes y sus discursos; segmentar los discursos en párrafos; marcar el idioma del discurso o segmento de texto; marcar comentarios del transcriptor; y dividir el texto en secciones. En su estructura básica, el documento XML resultante debe contener dos grandes secciones: el encabezado (<teiHeader>) y el cuerpo (<body>). En este trabajo, se exponen las acciones llevadas a cabo para etiquetar el cuerpo, que contiene la interacción parlamentaria.

El corpus comprende el período entre los años 2015 y 2020, y consiste en 211 documentos de extensión DOCX, accesibles a través de un repositorio en Github creado para este proyecto⁹. Se trata, entonces, de una fuente “born digital”, con el añadido de contar con un marcado inicial por parte del equipo de taquígrafos.

Cada documento está nombrado en base a la fecha, el número de sesión y de reunión. El sistema de trabajo de este Parlamento es a través de debates y votaciones, organizados en

⁹ https://github.com/marilinaapisani/docx2tei_ParlaMint

sesiones. Una sesión se define a partir del tiempo de trabajo parlamentario destinado a agotar un orden del día, mientras que una reunión es la parte de la sesión mantenida en un mismo día natural (Parlament De Catalunya, s.f.). En la Figura 2 se observa la estructura típica de estos ficheros.

Figura 2

Estructura típica de las transcripciones de los discursos del Parlamento de Cataluña

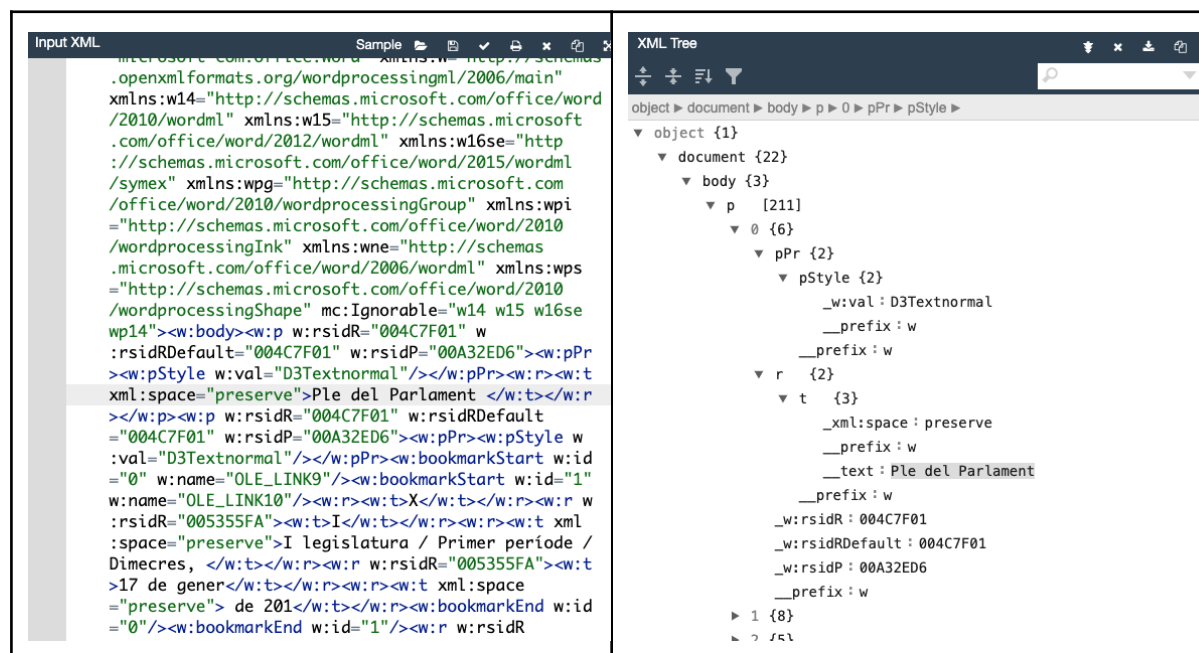
<p>Ple del Parlament / sessió núm. 1 / 17 de gener de 2018</p> <p>Ple del Parlament XII legislatura / Primer període / Dimecres, 17 de gener de 2018</p> <p>Ple del Parlament</p> <p>Presidència d'Edat del Sr. Ernest Maragall i Mira Presidència del M. H. Sr. Roger Torrent i Ramió</p> <p>Sessió núm. 1, de constitució</p> <p>SESSIÓ NÚM. 1</p> <p>La sessió s'obre a les onze del matí i dos minuts. Presideix el president de la Mesa d'Edat, acompanyat dels secretaris de la Mesa d'Edat, la qual és assistida pel secretari general i el lletrat major.</p> <p>ORDRE DEL DIA DE LA CONVOCATÒRIA</p> <p>Punt únic: Constitució del Ple del Parlament i elecció de la Mesa del Parlament (tram. 396-00001/12 i 398-00001/12).</p> <p>El secretari general (Xavier Muro i Bas)</p> <p>Bon dia a tothom, il·lustres senyores diputades i senyores diputats, autoritats que ens acompanyen, senyores i senyors, sigueu benvinguts al Parlament de Catalunya en aquesta sessió constitutiva de la seva dotzena legislatura.</p>	<p>Ple del Parlament / sessió núm. 1 / 17 de gener de 2018</p> <p>Ple del Parlament XII legislatura / Primer període / Dimecres, 17 de gener de 2018</p> <p>Ple del Parlament</p> <p>Presidència d'Edat del Sr. Ernest Maragall i Mira Presidència del M. H. Sr. Roger Torrent i Ramió</p> <p>Sessió núm. 1, de constitució</p> <p>SESSIÓ NÚM. 1</p> <p>La sessió s'obre a les onze del matí i dos minuts. Presideix el president de la Mesa d'Edat, acompanyat dels secretaris de la Mesa d'Edat, la qual és assistida pel secretari general i el lletrat major.</p> <p>ORDRE DEL DIA DE LA CONVOCATÒRIA</p> <p>Punt únic: Constitució del Ple del Parlament i elecció de la Mesa del Parlament (tram. 396-00001/12 i 398-00001/12).</p> <p>El secretari general (Xavier Muro i Bas)</p> <p>Bon dia a tothom, il·lustres senyores diputades i senyores diputats, autoritats que ens acompanyen, senyores i senyors, sigueu benvinguts al Parlament de Catalunya en aquesta sessió constitutiva de la seva dotzena legislatura.</p>
---	---

Nota. A la izquierda, el documento original. A la derecha, se resaltan con distintos colores las secciones del documento.

Desde 2007, todo documento DOCX de Microsoft Word se guarda en el formato XML. Si bien un editor de XML como Oxygen permite abrir este fichero —generalmente nombrado “document.xml”— un *parseador* en línea permite ordenar su estructura jerárquica de manera más legible —ver Figura 3—.

Figura 3

Visualizaciones de un documento XML



Nota. A la izquierda, un documento XML sin indentación. A la derecha, el mismo documento tras ser analizado —*parseado*— por el formateador en línea “JSON formatter”.

En la Figura 3, se observan las marcas originales del taquígrafo mediante estilos de formato del texto (“D3Textnormal”). A través de Python, es posible automatizar la tarea de cambiar una etiqueta por otra —similar al abordaje de Antiba (2021) con la función “Buscar y Reemplazar”—. Por ejemplo, Sweigart (2015) explica cómo extraer y procesar información de archivos PDF, hojas de cálculo y documentos de Microsoft Word. Sin embargo, esta tarea se muestra insuficiente con la aparición de excepciones, errores en los documentos fuente y la variedad de posibilidades en la marca del texto (Antiba, 2021).

¿Cómo es posible, entonces, transformar un gran conjunto de discursos parlamentarios en un corpus codificado siguiendo las Directrices TEI? Inspirada en el concepto de lectura distante, la alternativa que se presenta en este trabajo es comenzar con la construcción de una matriz de datos mediante la librería pandas en Python. Como se espera mostrar, esta nueva configuración permite visualizar la estructura subyacente de los documentos y —a través de un análisis exploratorio que incluye árboles, tablas y gráficos— posibilita una

codificación más precisa. Así, en esta propuesta, el humanista diseña y programa un modelo similar a los sistemas de expertos basándose en un análisis exhaustivo de los datos que se hace posible gracias a las visualizaciones mencionadas.

5.2. Herramientas Utilizadas

Con el objetivo de ofrecer una herramienta reproducible y de acceso libre, el programa para automatizar la codificación se pone a disposición en un cuaderno de [Colab](#)¹⁰. En la elaboración de este programa se utilizaron una serie de librerías en Python, siendo las principales:

- xpython-docx, que permite leer el fichero xml contenido en el docx;
- xml.etree.ElementTree, que permite leer y construir árboles en xml;
- pandas, que permite la construcción y transformación de matrices de datos;
- potly, que permite graficar una matriz de datos.

Asimismo, para la verificación de los ficheros finales, se utilizó el editor de XML Oxygen, el repositorio de Github del Proyecto ParlaMint¹¹ y un repositorio personal¹².

La propuesta metodológica comprende cuatro grandes pasos. En el primero, se procesan los documentos y se crea la matriz de datos, o en términos de la librería pandas, un DataFrame (DF). En el segundo, se preparan los datos de manera que se puedan codificar según las guías TEI mediante una serie de transformaciones sobre la matriz de datos creada. En el tercero, se realiza una primera codificación del corpus sobre el DF. En el cuarto, se crean los documentos XML a partir de la conversión del DF en árboles —es decir, en objetos de estructura jerárquica—. Finalmente, se analizan los resultados y se validan a través de Oxygen (para verificar que esté bien formado) y del esquema PARLA-CLARIN. El

¹⁰ El cuaderno de Colab es accesible a través del repositorio personal o este link: <https://bit.ly/3pRsxtS>

¹¹ <https://github.com/clarin-eric/ParlaMint>

¹² https://github.com/marilinaapisani/docx2tei_ParlaMint

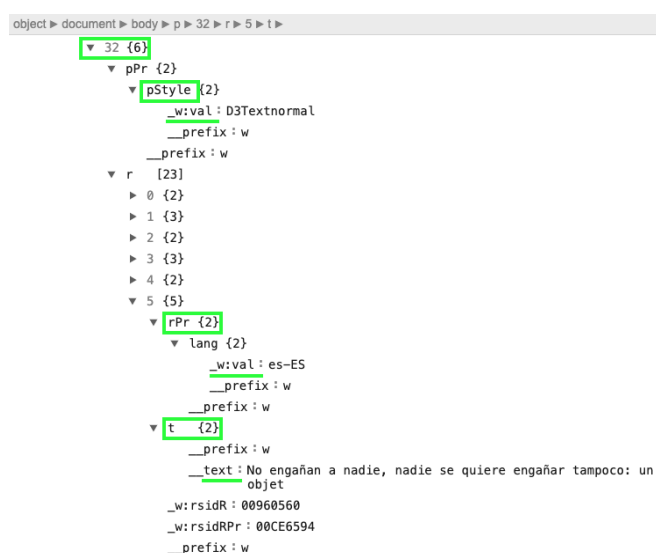
proceso es iterativo, en el sentido de que los pasos se repiten hasta alcanzar el resultado deseado.

5.3. Primer Paso: Construcción de un DataFrame

La estructura jerárquica de los ficheros XML de los discursos parlamentarios —que se observa en la Figura 4— evidencia tres elementos fundamentales: <p>, <r>, y <t>.

Figura 4

Estructura jerárquica del fichero XML de un discurso parlamentario



Nota. Se resaltan los elementos principales de la estructura.

Cada elemento <p> se corresponde con un párrafo en el documento original. Cada <p> está compuesta por el elemento <r> que contiene, a su vez, varios elementos <t>. El elemento <t> tiene el atributo “text” donde encontramos un segmento de texto —una oración o una palabra—.

Siguiendo la máxima de las recomendaciones PARLA-CLARIN —capturar la mayor cantidad posible de texto y marcas de la fuente (Erjavec y Pančur, 2022)— y partiendo del análisis previo de Antiba (2021) sobre los formatos de estilo, se recuperaron los siguientes elementos:

- `<pStyle>`: Contiene la marca del taquígrafo (D3Textnormal, D3Intervinent, etc.)
- `<rPr>`: Tiene el atributo “lang” del elemento `<r>` y permite advertir si hay cambios de idioma por parte del hablante. Contiene, además, elemento `` que marca el formato de texto negrita.
- `<rStyle>`: Contiene otros formatos de texto como cursiva o normal.

Para poder leer todos los elementos presentes en la totalidad de los documentos que componen el corpus, se propone la construcción de una matriz de datos, o DF. La función que se define para obtenerla consta de estas instrucciones:

1. Tomar como input un documento DOCX de la carpeta contenedora.
2. Leer el documento y extraer el fichero XML.
3. Convertir el fichero XML en un objeto de la librería `xml.etree.ElementTree`, que contiene su estructura jerárquica.
4. De este objeto, extraer los elementos deseados, identificados a través de sus etiquetas.
5. Almacenar estos elementos en distintas listas según sus etiquetas.
6. Construir un DF a partir de las listas.
7. Repetir las instrucciones para el siguiente documento.
8. Concatenar el nuevo DF al anterior.
9. Ofrecer como output un único DF.

En la Figura 5 se compara el documento original y el DF que se obtiene como resultado de la función anterior. En la siguiente sección se presentan las técnicas y herramientas principales para analizar —o leer— el DF.

5.4. Segundo Paso: Análisis del DataFrame

Usualmente, el análisis de un DF comienza con una exploración a través de funciones de descripción y resumen, y continúa con la construcción de subconjuntos y de visualizaciones

a través de gráficos o tablas¹³. Estas técnicas permiten no sólo tener una visión general —panorámica— del corpus, sino también identificar excepciones —hacer zoom-in—. A partir de la ejemplificación de estas técnicas, el objetivo de esta sección es reconocer los patrones en los datos que permiten definir las reglas de etiquetado correctas.

Figura 5

Comparación del documento original y del DataFrame

<p>Ple del Parlament / sessió 3.3 / 12 de novembre de 2015</p> <p>XI legislatura - primer període - sèrie P - número 4</p> <p>Sessió 3, tercera i darrera reunió, dijous 12 de novembre de 2015</p> <p>Ple del Parlament</p> <p>Presidència de la M. H. Sra. Carme Forcadell i Lluís</p> <p>SESSIÓ 3.3</p> <p>La sessió, suspesa el dia 10 de novembre, es reprèn a les deu del matí. Presideix la presidenta del Parlament, acompanyada de tots els membres de la Mesa, la qual és assistida pel secretari general i el lletrat Francesc Pau i Vall.</p> <p>Al banc del Govern seu el president de la Generalitat en funcions, acompanyat de tot el Govern en funcions.</p> <p>La presidenta</p> <p>Es reprèn la sessió.</p> <p>D'acord amb l'article 4.3 de la Llei de la presidència de la Generalitat i del Govern, els proposats dies 9 i 10 de novembre va tenir lloc la presentació davant del Ple de la cambra del programa de govern del candidat a la presidència de la Generalitat...</p> <p>(Veus de fons.) Senyor García Albiol, per què em demana la paraula?</p> <p>Xavier García Albiol</p>		<table> <tr> <th>index</th><th>text</th><th>style</th></tr> <tr> <td>0</td><td>XI legislatura - primer període - sèrie P - nú...</td><td>D3Textnorma</td></tr> <tr> <td>1</td><td>Sessió 3, tercera i darrera reunió, dijous 12 ...</td><td>D3Textnorma</td></tr> <tr> <td>2</td><td></td><td>D3Textnorma</td></tr> <tr> <td>3</td><td>Ple del Parlament</td><td>Crgar</td></tr> <tr> <td>4</td><td>Presidència de la M. H. Sra. Carme Forcadell i...</td><td>CPresidenci</td></tr> <tr> <td>5</td><td></td><td>D3Textnorma</td></tr> <tr> <td>6</td><td>SESSIÓ 3.3</td><td>D2Davantal-Sessió</td></tr> <tr> <td>7</td><td>La sessió, suspesa el dia 10 de novembre, es r...</td><td>D2Davanta</td></tr> <tr> <td>8</td><td>Al banc del Govern seu el president de la Gene...</td><td>D2Davanta</td></tr> <tr> <td>9</td><td></td><td>D3Textnorma</td></tr> <tr> <td>10</td><td>La presidenta</td><td>D3IntervinentObertura</td></tr> <tr> <td>11</td><td>Es reprèn la sessió.</td><td>D3Textnorma</td></tr> <tr> <td>12</td><td>D'acord amb l'article 4.3 de la Llei de la pre...</td><td>D3Textnorma</td></tr> <tr> <td>13</td><td>(Veus de fons.) Senyor García Albiol, per què ...</td><td>D3Textnorma</td></tr> <tr> <td>14</td><td>Xavier García Albiol</td><td>D3Intervinen</td></tr> </table>	index	text	style	0	XI legislatura - primer període - sèrie P - nú...	D3Textnorma	1	Sessió 3, tercera i darrera reunió, dijous 12 ...	D3Textnorma	2		D3Textnorma	3	Ple del Parlament	Crgar	4	Presidència de la M. H. Sra. Carme Forcadell i...	CPresidenci	5		D3Textnorma	6	SESSIÓ 3.3	D2Davantal-Sessió	7	La sessió, suspesa el dia 10 de novembre, es r...	D2Davanta	8	Al banc del Govern seu el president de la Gene...	D2Davanta	9		D3Textnorma	10	La presidenta	D3IntervinentObertura	11	Es reprèn la sessió.	D3Textnorma	12	D'acord amb l'article 4.3 de la Llei de la pre...	D3Textnorma	13	(Veus de fons.) Senyor García Albiol, per què ...	D3Textnorma	14	Xavier García Albiol	D3Intervinen
index	text	style																																																
0	XI legislatura - primer període - sèrie P - nú...	D3Textnorma																																																
1	Sessió 3, tercera i darrera reunió, dijous 12 ...	D3Textnorma																																																
2		D3Textnorma																																																
3	Ple del Parlament	Crgar																																																
4	Presidència de la M. H. Sra. Carme Forcadell i...	CPresidenci																																																
5		D3Textnorma																																																
6	SESSIÓ 3.3	D2Davantal-Sessió																																																
7	La sessió, suspesa el dia 10 de novembre, es r...	D2Davanta																																																
8	Al banc del Govern seu el president de la Gene...	D2Davanta																																																
9		D3Textnorma																																																
10	La presidenta	D3IntervinentObertura																																																
11	Es reprèn la sessió.	D3Textnorma																																																
12	D'acord amb l'article 4.3 de la Llei de la pre...	D3Textnorma																																																
13	(Veus de fons.) Senyor García Albiol, per què ...	D3Textnorma																																																
14	Xavier García Albiol	D3Intervinen																																																

Nota. A la izquierda, un extracto del documento original. A la derecha, su correspondiente matriz de datos. Cada línea o fila corresponde a un segmento de texto del documento, y las columnas contienen etiquetas (“style”) o atributos de ese segmento (“text”).

5.4.1. Descripción y resumen

Las funciones de descripción permiten obtener información general, por ejemplo, la cantidad de documentos procesados, los tipos de datos que se han recogido y las fechas de los documentos. En el corpus de los discursos del Parlamento de Cataluña, el elemento que se revela como central es la marca de estilo, que permite identificar intervinientes, comentarios,

¹³ Se sigue, con cierta libertad, las técnicas propuestas en (Ahmed y Mukhiya, 2020).

títulos y el texto del discurso. Este dato fue guardado en una columna del DF (“style”), y es posible conocer los valores que toma con una fórmula, como la que se muestra en la Figura 6. A simple vista, se observa que hay distintas marcas para el mismo tipo de elemento. Por ejemplo, los intervinientes se han marcado con la etiqueta “D3Intervinent” y “D3IntervinentObertura”, mientras que las notas del transcriptor —de las que se hablará más adelante— se refieren a interrupciones (D3Acotacicva), a la aparición de información sobre la sesión (D2Davantal) o al orden del día (D2Ordredia). También del listado se advierte que hay textos sin estilo —donde el campo se encuentra vacío— o con un estilo sin nombre —“Estilo1”—.

Figura 6

Fórmula para obtener un listado de valores de una columna en Python

```
[8] sample_dfs['style'].unique().tolist()

['D3Textnormal',
 'Crgan',
 'CSessi',
 'CPresidncia',
 'D2Davantal-Sessio',
 'D2Davantal',
 'D3IntervinentObertura',
 'D3Ttolnegreta',
 'D3TtolTram',
 'D3Intervinent',
 'D3Acotacicva',
 'D3Acotacihorria',
 'D2Ordredia-Ttol',
 'D2Ordredia',
 'D3Ttolrodona',
 '',
 'D2Davantalambespai',
 'Estilo1']
```

Nota. La fórmula muestra los valores únicos de la columna “style”.

Cuando la marca de estilo es incorrecta, existen otros elementos que permiten inferir la etiqueta que corresponde, como el formato del texto (los valores recogidos en los elementos <rPr> y rStyle del XML), su longitud o si se encuentra entre paréntesis¹⁴. Por ejemplo, para calcular la longitud del texto —cuántos caracteres tiene— se puede utilizar una fórmula. De

¹⁴ De los 211 documentos, 2 presentaron marcas de estilo completamente distintas y quedaron por fuera de esta versión del etiquetado. Aunque las funciones incluyen algunas excepciones para procesarlos, se recomienda cambiar las marcas de estilo en el texto fuente. Este fenómeno resalta la importancia que el marcado original tiene en el proceso y cómo la posibilidad de su aprovechamiento está condicionada por la corrección de la tarea del equipo de taquígrafos.

esta manera, se puede analizar si existe alguna relación entre este valor y la etiqueta de estilo asignada por el transcriptor.

De los primeros análisis visuales sobre los documentos, se conjetura que los segmentos de texto donde se nombra a un hablante no son muy extensos. Esta hipótesis se puede comprobar comparando la diferencia entre la mediana y la máxima de la longitud del texto cuando la etiqueta es “D3Intervinent”. Una diferencia muy alta entre estos dos valores —como se advierte en la Figura 7— apunta a la existencia de valores atípicos.

Figura 7

Fórmula para obtener una descripción estadística básica de una columna

```
[ ] sample_dfs.loc[(sample_dfs['style'].isin(interv_style)), 'len'].describe()

count    39556.000000
mean      20.513727
std       23.785312
min        0.000000
25%       12.000000
50%       19.000000
75%       23.000000
max      1248.000000
Name: len, dtype: float64
```

Nota. La fórmula aporta cálculos estadísticos básicos (media, mediana, mínimo, máximo, etc.) sobre la columna “len” del DataFrame, que contiene valores numéricos (la cantidad de caracteres de un texto), cuando el valor de “style” coincide con los identificados para un interviniente (“D3Intervinent” y “D3IntervinentObertura”).

Para ver en más detalle estos casos, y comprobar que se trata de un error en la etiqueta original (donde el texto en cuestión no es un hablante sino parte del discurso) se puede recurrir a algún tipo de visualización.

5.4.2. Visualizaciones

En este trabajo se utilizaron tres maneras de visualizar una matriz de datos. La primera es la más simple: una vista que puede ser completa o parcial (las primeras o las últimas líneas). Colab incluye una herramienta que permite “navegar” el DataFrame cuando resulta

muy grande para mostrarlo en pantalla. La imagen que se mostró anteriormente en la Figura 7 es un caso de este tipo de visualización.

La segunda forma es a través de un subconjunto, que se obtiene luego de una acción de filtrado. Por ejemplo, en la Figura 8 se observan los primeros cinco casos cuando la etiqueta de estilo es “D3Intervinent” y la cantidad de caracteres del texto supera los 100. A partir de esta vista se infiere que se trata de un error en la marca, ya que el texto no es el nombre de un interviniente sino parte del discurso.

Figura 8

Fórmula para obtener un subconjunto del DataFrame

```
[ ] sample_dfs.loc[(sample_dfs['style'].isin(interv_style)) &
                  (sample_dfs['len'] > 100), ['text', 'file']].head(5)
```

	text	file
8253	Gràcies, consellera. A continuació passem a la...	ParlaMint-ES-CT_2019-03-06
11228	Moltes gràcies. Hem assistit durant aquest Ple...	ParlaMint-ES-CT_2019-04-04
11229	I també resulta bastant curiós que utilitzin e...	ParlaMint-ES-CT_2019-04-04
11230	Han hagut de sortir molts milers de ciutadans ...	ParlaMint-ES-CT_2019-04-04
11231	Sotmetre's a una qüestió de confiança..., no c...	ParlaMint-ES-CT_2019-04-04

Nota. La función “head” especifica hasta qué número de filas se van a mostrar.

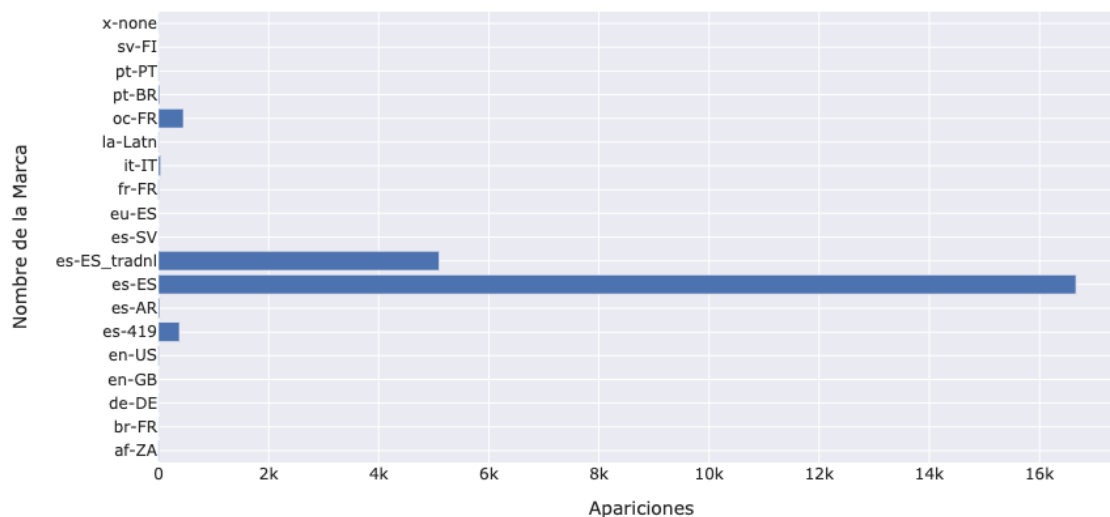
La tercera forma es a través de un gráfico. Una librería como potly, permite realizar gráficos de barras, histogramas, nubes de palabras. Por ejemplo, en la Figura 9 se observa un gráfico de barras que muestra la cantidad de texto según la marca de idioma. Este gráfico no solo anticipa los distintos idiomas identificados en los documentos, también indica su frecuencia.

Mediante estas técnicas de análisis —documentadas en detalle en el cuaderno de Colab— se identificaron errores en las marcas de los documentos originales (por ejemplo, un interviniente marcado como texto normal o viceversa); se reconocieron comentarios del transcriptor no marcados (notas que, en ocasiones, se encuentran dentro de un párrafo del discurso); se advirtieron problemas en la identificación de los intervinientes (determinando

las distintas maneras en las que son nombrados); y se anticipó la variedad de idiomas presente en el corpus.

Figura 9

Frecuencia de las marcas de idioma originales en el corpus



Nota. El gráfico muestra las distintas etiquetas utilizadas para marcar que un segmento de texto pertenece a un idioma distinto al catalán.

A partir del reconocimiento de estas características, en el capítulo siguiente se presentan las técnicas que permiten un etiquetado “distante” del corpus.

5.5. Tercer Paso: el Etiquetado Distante

El objetivo del apartado anterior fue visualizar los patrones en el DF que permiten inferir reglas de codificación. En esta sección se presentan las funciones que hacen posible al etiquetado en TEI. Para ello, se definen una serie de instrucciones que implican la transformación del DF y su enriquecimiento: el momento en el que el humanista toma las decisiones de cómo marcar el texto, desde lo conceptual y lo técnico.

5.5.1. Técnicas principales

Siguiendo un esquema similar al de las reglas de los sistemas expertos, se definen sentencias condicionales del tipo “Si se cumplen ciertas condiciones, entonces se realizará cierta acción”. A diferencia de una típica función de “Buscar y Reemplazar”, estas sentencias pueden admitir mayor complejidad tanto en el antecedente como en el consecuente del condicional.

La manera de programar estas sentencias en Python sobre un DF es a través de las funciones de filtrado y de asignación de nuevos valores. Por ejemplo, en la Figura 10 se observa una de estas reglas que asigna la marca “D3Intervinent” a todo texto que haya sido marcado como texto normal pero que tenga una longitud menor a 100 caracteres y que el formato de fuente sea negrita.

Figura 10

Fórmula para asignar un nuevo valor a un subconjunto del DataFrame

```
df.loc[(df['style'] == 'D3Textnormal') &
      (df['len'] < intervinent_len) &
      (df['bold'] == 'bold') , "style"] = 'D3Intervinent'
```

Es posible considerar también los elementos que están antes o después de un valor en consideración. De esta manera, se puede determinar —por ejemplo— la extensión de un discurso. Un discurso comienza con la marca de un interviniente y continúa hasta la aparición de otro. Si lo que le sigue a un interviniente es un segmento de texto, ese texto es parte del discurso hasta que se identifique la aparición de otro interviniente.

Cuando las reglas son muy complejas —o los fenómenos muy variados como en el caso de los comentarios del transcriptor— es recomendable trabajar con una tabla de referencia. Esta tabla indica, en el caso más simple, que para determinado valor, corresponde otro valor. Los datos de esta tabla se pueden luego cruzar con el DF para obtener el valor buscado.

5.5.2. Ejemplos de Aplicación

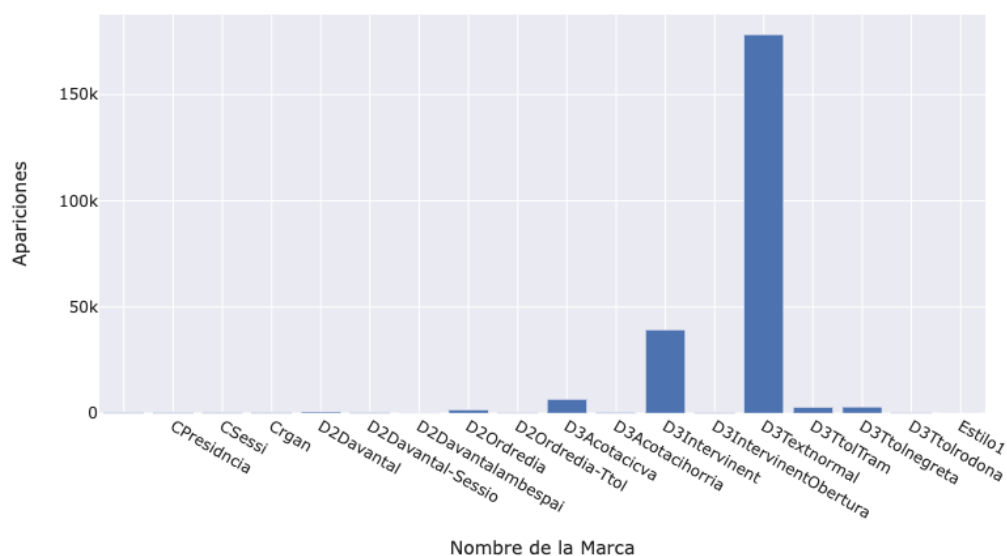
En este apartado se presentan cuatro grandes desafíos en la codificación del corpus de las interacciones del Parlamento de Cataluña y las técnicas abordadas para sortearlos: el etiquetado de los comentarios del transcriptor, la identificación del idioma de cada discurso y párrafo, la identificación de los intervinientes y la división del documento.

5.5.2.1. Comentarios del Transcriptor

En el corpus se encuentran distintos tipos de comentarios del transcriptor: títulos, notas e interrupciones. De especial interés resultan las interrupciones que, generalmente, llevan la marca “D3Acotaciva”. Como puede observarse en la Figura 11, la mayoría de los comentarios pertenecen a este grupo.

Figura 11

Frecuencia de las marcas de estilo originales en el corpus



Las interrupciones a veces figuran como un párrafo aparte y en otras ocasiones están incluidas dentro de un mismo párrafo —Figura 12—. En este último caso, dentro del esquema PARLA-CLARIN, existen dos opciones: marcar la interrupción como un elemento dentro del segmento o dividir el segmento y marcar la interrupción como un elemento del

discurso. La propuesta en este trabajo es optar por la segunda opción, ya que se considera que una interrupción provoca una ruptura en el discurso y la distinción entre éste y el párrafo se vuelve difusa. Esta acción se llevó a cabo mediante una transformación del DF, en la que se generaron nuevas filas para cada interrupción marcada dentro de un párrafo.

A partir de estas acciones, la cantidad de segmentos de texto marcados con la etiqueta “D3Acotacicva” triplicó su número —de 6 475 marcas de estilo identificadas originalmente en el corpus, se obtuvieron 19 831—.

Figura 12

Ejemplos de los comentarios del transcriptor en el corpus

Ple del Parlament / sessions 2 i 3.1 / 9 de novembre de 2015	Ple del Parlament / sessions 2 i 3.1 / 9 de novembre de 2015
<p>Manifestació de condol per les víctimes del desbordament del riu Sió a Agramunt</p> <p>En primer lloc, abans d'iniciar l'ordre del dia d'avui, si m'ho permeten, els proposo de fer un minut de silenci en record de les quatre víctimes que varen morir el passat dia 3 de novembre a causa del desbordament del riu Sió a Agramunt. Així, doncs, guardem un minut de silenci en la seva memòria.</p> <p><i>(La cambra serveix un minut de silenci.)</i></p> <p>Em plau saludar, en nom de la cambra, l'expresident Benach, exconsellers, exdiputats i autoritats que avui ens acompanyen. A continuació, vull donar la benvinguda a la il·lustre senyora Hortènsia Grau i Juan i a l'il·lustre senyor Fernando Sánchez Costa, que avui s'incorporen com a nous diputats de l'onzena legislatura. Ambdós són diputats que coneixen molt bé el Parlament, atès que ja van ser diputats a l'anterior legislatura.</p>	<p>Espanya i a Europa, i que no hi ha una altra via possible. No ho diem nosaltres, ho diu Alex Salmond. No es pot parlar d'apoderar la ciutadania i a continuació negar el referèndum. És cert que ens ho prohibeixen des del Partit Popular i el búnquer espanyol; però és cert que aquells que donen aquest procés per amortitzat i volen passar pantalla obliden que sense referèndum la ciutadania de Catalunya no s'haurà pronunciat legítimament i democràticament.</p> <p>I, per tant, els torno a plantejar: no sé qui anirà al lloc de qui, però crec que tots hauríem d'anar a l'única via possible, que és la de donar la veu a la ciutadania de Catalunya amb un referèndum lliure, en què poguéssim expressar quina és la forma en què vol viure i construir el seu futur políticament, i que parlar en nom d'ells és tot menys apoderar-los. <i>(Véus de fons.)</i> Tranquils, tranquils, que tot just hem començat.</p> <p>La presidenta</p> <p>Respectin l'ús de la paraula.</p>

Nota. Se muestran dos ejemplos. A la izquierda, un comentario entre párrafos. A la derecha, un comentario dentro de un párrafo.

Ahora bien, esta marca indica la ocurrencia de una interrupción pero no explica el motivo de la misma. Analizando el texto escrito por el transcriptor, es posible categorizar las interrupciones. El módulo TEI dedicado a las Transcripciones del Habla recomienda la utilización de los siguientes elementos:

- <pause>, para marcar las pausas que pueden ocurrir entre discursos o dentro de un mismo discurso.
- <vocal>, para marcar fenómenos vocales pero no necesariamente léxicos (las risas o los sonidos de desacuerdo).

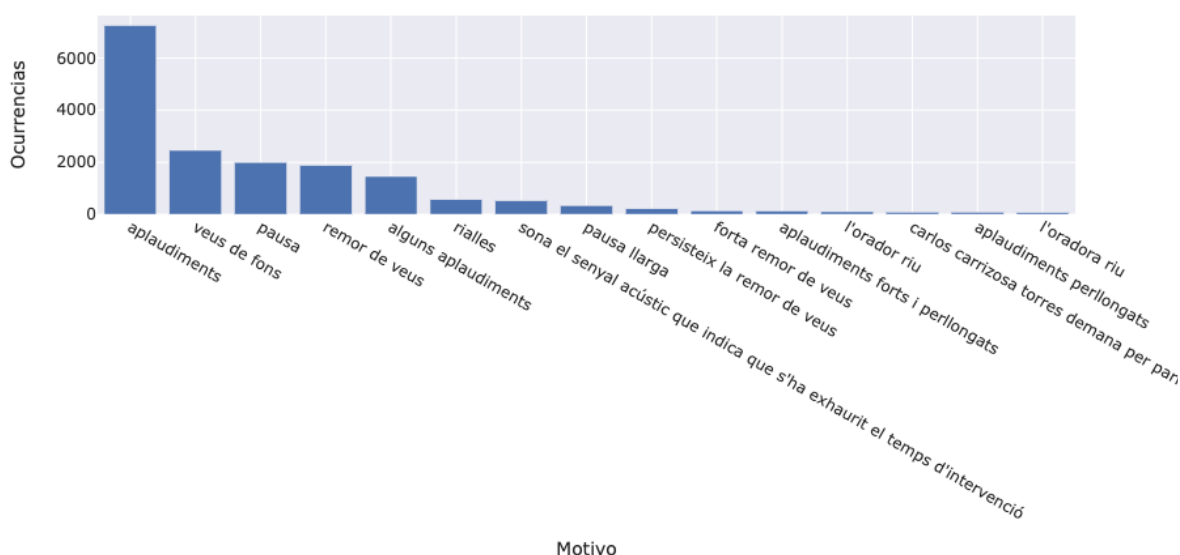
- <kinesic>, para marcar un fenómeno comunicativo no necesariamente vocalizado (gestos, aplausos).
- <incident>, para marcar cualquier fenómeno que no sea necesariamente vocalizado o comunicativo (sonidos incidentales) (Text Encoding Initiative, 2022b).

También, el esquema PARLA-CLARIN sugiere el elemento <gap> para marcar partes del discurso omitidas por razones técnicas o editoriales.

De la Figura 13 se advierte que, en el corpus, las interrupciones más frecuentes ocurren porque hay aplausos, voces de fondo, pausas o risas.

Figura 13

Motivos de interrupción más frecuentes en el corpus



Nota. El gráfico muestra los 15 comentarios del transcriptor más frecuentes marcados con la etiqueta de estilo “D3Acotativa”.

A pesar de contar con esta información, algunos fenómenos encontrados en el corpus escapan al alcance de las recomendaciones listadas arriba. En primer lugar, hay sonidos comunicativos que no son vocales ni corporales, por ejemplo, el sonido que anuncia que a un participante se le terminó el tiempo. Se propone la nota <incident> para marcar este fenómeno. En segundo lugar, muchos de los comentarios del transcriptor contienen más de

un tipo de nota —por ejemplo, “riales i aplaudiments”—. No es claro qué estrategia es la indicada en estas situaciones. Una posibilidad es marcar el conjunto como <note>, otra es tomar el tipo de la primera nota (<vocal> en el ejemplo) y otra posibilidad es separar los comentarios y marcar cada uno con la etiqueta correspondiente¹⁵. En tercer lugar, el esquema PARLA-CLARIN no incluye el elemento “<pause>”. Como es uno de los motivos más frecuentes en el corpus estudiado, se sugirió a los miembros del proyecto ParlaMint la adición de este elemento¹⁶.

Dada la gran variedad de descripciones, la técnica más precisa para etiquetar estos fenómenos es utilizar una tabla de referencia. Para construirla, primero, se elabora la tabla con todas las alternativas —es decir, con todos los comentarios distintos del transcriptor—. Esta tabla se puede exportar en formato CSV y asignar la etiqueta deseada de manera manual en cualquier editor de hojas de cálculo. De esta manera, se reduce la cantidad de trabajo —no se actúa sobre cada comentario, ya que éstos están agrupados en casos únicos— sin perder precisión en la marca¹⁷.

A través de la tipología recomendada (vocal, kinesic, incident y gap) se marcaron 15 945 interrupciones en el corpus.

5.5.2.2. *Idiomas*

Los discursos del parlamento de Cataluña son mayormente pronunciados en catalán y en español. En algunos casos, ocurre lo que se conoce como “code switching”, esto es, cuando el hablante va alternando entre varios idiomas en un mismo acto de habla.

Una identificación acertada del idioma permitirá luego analizar estos fenómenos y una anotación lingüística del corpus correcta. Con esto en vista, se utilizaron distintas

¹⁵ Si bien la última propuesta resulta en un etiquetado más fino, por cuestiones de tiempo, en este trabajo se opta por la segunda.

¹⁶ Será incorporado como un tipo del elemento “<incident>”.

¹⁷ Al momento de escribir el trabajo, esta tabla se encuentra en construcción y se adoptó en su lugar la definición de reglas.

herramientas para identificar el idioma de cada segmento de texto (párrafos del discurso) y de cada discurso como unidad.

Para los segmentos de texto, se utilizó un algoritmo de reconocimiento de idiomas, que —según las pruebas realizadas— funciona con confiabilidad en textos de más de 200 caracteres¹⁸. Como en los documentos originales solo aparece la marca del idioma cuando es distinto al catalán, esta función permite inferir con mayor seguridad los segmentos de texto que no fueron marcados.

Para los discursos, se construyó una tabla de referencia creada a partir del mismo corpus. Para cada discurso, esta tabla contiene la cantidad de segmentos que lo componen. A partir de esta información, se toma como idioma del discurso el idioma que tiene más ocurrencias¹⁹.

En algunas ocasiones, cuando los segmentos contienen muy poco texto, se optó por asignarle el idioma con el que el interviniente pronuncia sus discursos más frecuentemente. Para obtener este dato, se construyó otra tabla de referencia, tomando en cuenta todos los intervinientes y la cantidad de discursos que pronunció en los distintos idiomas.

A pesar de estos esfuerzos, se advierte que existen cambios de idioma en el mismo segmento de texto. En general, esto ocurre cuando el hablante realiza una cita en otro idioma. La alternativa para etiquetar correctamente estos segmentos es similar a la llevada a cabo para los comentarios, dividir el segmento cuando ocurre una cita²⁰.

5.5.2.3. *Intervinientes*

Según las recomendaciones PARLA-CLARIN, cada discurso —es decir, cada elemento <u>— debe tener como atributo el identificador del hablante (@who). De esta manera, es posible asociar al discurso con datos como la fecha de nacimiento del hablante, su género o

¹⁸ El reconocedor de idiomas que se utilizó es cld3.

¹⁹ Este criterio es similar al tomado por otros participantes del proyecto ParlaMint, como Bélgica.

²⁰ Al momento de escribir este trabajo, tal abordaje no fue posible por cuestiones de tiempo. En su lugar, se definieron reglas más generales.

afiliación política. Tanto los identificadores como los metadatos de los intervinientes se encuentran en un fichero EXCEL confeccionado para este proyecto (Antiba, 2021). En este punto, la tarea implicó identificar al hablante en el documento e incorporar su identificador ya asignado en ese fichero.

En la mayoría de los casos, los intervinientes son nombrados por su nombre al inicio de su discurso. Sin embargo, a lo largo de los documentos el nombre puede sufrir variaciones tipográficas. Estos casos se resolvieron aplicando sobre el DF una función que permite realizar un cruce sensible a estas variaciones, buscando la coincidencia más exacta²¹.

En ocasiones, los hablantes son nombrados a través de sus cargos. Muchas veces, esta descripción incluye el nombre de la persona entre paréntesis. A veces, esta correspondencia se da únicamente en la primera intervención que realiza el hablante. La complejidad de este fenómeno implicó que en la versión inicial del etiquetado de este corpus estos intervinientes no fueran identificados (Antiba, 2021). Para revertir esta situación se utilizaron, nuevamente, tablas de referencia.

Como resultado, se identificaron 255 intervinientes distintos en el corpus. De no haber realizado estas acciones, sólo se habrían reconocido 216²². En todos los casos, se optó por incluir el segmento de texto —que contiene ya sea el nombre o el cargo— como una nota del transcriptor de tipo “speaker”.

5.5.2.4. Divisiones

Teniendo en cuenta que en algunos documentos ocurre más de una sesión en un día natural, se optó por utilizar el elemento <div> para dividir el documento en sesiones. De esta manera, cada división corresponde, dentro del documento, a una sesión. Una estrategia que

²¹ Una opción para documentar esas variaciones es la etiqueta <choice>, aunque no fue utilizada en esta codificación del corpus ya que no se encontraba dentro del esquema PARLA-CLARIN. La herramienta utilizada es la librería fuzzywuzzy.

²² Al momento de escribir el trabajo aún quedan por identificar algunas intervenciones cuando el hablante es nombrado a través de su cargo, pero sin que exista la concordancia —a lo largo del documento— entre cargo y nombre. El listado está disponible en el cuaderno de Colab, al igual que las indicaciones para completarlo.

no se adoptó en este trabajo —por motivos de tiempo— es dividir el documento en base a los títulos. La dificultad de este abordaje radica en que los títulos aparecen dentro de un discurso, por lo que son necesarias técnicas para determinar la extensión de cada división —similares a las utilizadas para determinar la extensión de un discurso—. Estos títulos han quedado marcados como comentarios del transcriptor en el lugar en el que aparecen en el texto original.

5.6. Cuarto Paso: Del DataFrame al Árbol XML/TEI

En los apartados anteriores se construyó un DF con todos los discursos parlamentarios disponibles y —a partir de las técnicas presentadas— se transformaron los datos para cumplir con las recomendaciones PARLA-CLARIN y posibilitar un etiquetado preciso. Todas estas acciones se realizaron sobre una matriz de datos. El último paso es, entonces, obtener de ésta la estructura jerárquica propia del marcado en XML: el árbol.

Los datos necesarios para llevar a cabo la anotación en TEI se encuentran en las distintas columnas del DF: el texto, el nombre de la etiqueta, el nivel al que corresponde —si es a nivel de segmento o de discurso— y sus atributos. Para construir el árbol se propone una función que sigue las siguientes instrucciones:

1. Crear el elemento raíz <text>.
2. Crear un sub elemento <body>.
3. Leer cada fila del DF.
4. Para cada fila, analizar si se trata de un elemento <div>, <u>, <seg> o <note>
5. Crear el elemento correspondiente y sus atributos.
6. Adjuntar el elemento al padre correspondiente.
7. Escribir el árbol en un nuevo fichero XML y grabarlo en la carpeta correspondiente.
8. Repetir las instrucciones anteriores para todos los documentos.

Como resultado de esta función, para cada documento DOCX se obtiene un fichero XML/TEI bien formado y validado por el esquema PARLA-CLARIN. En la Figura 14 se

observa una muestra del etiquetado obtenido: las divisiones en discursos y segmentos, títulos y comentarios²³.

Figura 14

Muestra del resultado final del proceso de etiquetado

La presidenta

A continuació té la paraula l'il·lustre senyor Joan Coscubiela, en nom del Grup Parlamentari de Catalunya Sí que es Pot...

Proposta de resolució sobre la prioritització d'un pla de rescat ciutadà i l'inici d'un procés constituent

250-00002/11

...per a presentar la proposta de resolució.

Joan Coscubiela Conesa

Bon dia, senyora presidenta; gràcies. Bon dia, senyores i senyors diputats. Vull que les meves primeres paraules siguin per denunciar i rebutjar els nous assassinats, els nous casos de violència masclista produïts ahir. Aquest cop no han estat dones catalanes, però també són les nostres dones. *(Aplaudiments.)*

Si la darrera legislatura va acabar malament, aquesta no pot començar pitjor: un president en funcions que ha de donar explicacions polítiques i el seu partit, Convergència Democràtica de Catalunya, responen judicialment d'imputacions greus de finançament irregular; una mesa del Parlament que actua de manera absolutista i una presidenta que s'estrena violentant el Reglament que va prometre respectar, i, per acabar-ho d'adobar, tres grups parlamentaris que porten la solució d'un problema polític al Tribunal Constitucional. Els ingredients perfectes per a una política partidista de búnquers, de blocs, que busquen retroalimentar-se mútuament en els conflictes, mentre obliden els problemes quotidians de la gent. Junts pel Sí i la CUP tenen tot el dret del món a voler discutir la seva resolució, tot el dret a escenificar i teatralitzar les seves negociacions per invertir el senyor Mas de president, però no tenen cap dret a tenir paralitzats el Parlament i el Govern, mentre al carrer es viuen situacions dramàtiques.

```
181 <note type="speaker">La presidenta</note>
182 <u ana="chair" who="ForcadellCarme" xml:id="ParlaMint-ES-CT_2015-11-09-0301.5" xml:lang="ca">
183 <seg xml:id="ParlaMint-ES-CT_2015-11-09-0301.5.0" xml:lang="ca">A continuació té la paraula
184 l'il·lustre senyor Joan Coscubiela, en nom del Grup Parlamentari de Catalunya Sí que es Pot...</seg>
185 <note type="speaker">Joan Coscubiela Conesa</note>
186 <u ana="member" who="CoscubielaJoan" xml:id="ParlaMint-ES-CT_2015-11-09-0301.6" xml:lang="ca">
187 <seg xml:id="ParlaMint-ES-CT_2015-11-09-0301.6.0" xml:lang="ca">Bon dia, senyora presidenta;
188 gràcies. Bon dia, senyores i senyors diputats. Vull que les meves primeres paraules siguin per denunciar i
189 rebutjar els nous assassinats, els nous casos de violència masclista produïts ahir. Aquest cop no han estat
190 dones catalanes, però també són les nostres dones.</seg>
191 <kinesic type="applause">
192 <desc>(Aplaudiments.)</desc>
193 </kinesic>
194 <seg xml:id="ParlaMint-ES-CT_2015-11-09-0301.6.1" xml:lang="ca">Si la darrera legislatura va
195 acabar malament, aquesta no pot començar pitjor: un president en funcions que ha de donar explicacions
196 polítiques i el seu partit, Convergència Democràtica de Catalunya, responen judicialment d'imputacions
197 greus de finançament irregular; una mesa del Parlament que actua de manera absolutista i una presidenta que
198 s'estrena violentant el Reglament que va prometre respectar, i, per acabar-ho d'adobar, tres grups
199 parlamentaris que porten la solució d'un problema polític al Tribunal Constitucional. Els ingredients
200 perfectes per a una política partidista de búnquers, de blocs, que busquen retroalimentar-se mútuament en
201 els conflictes, mentre obliden els problemes quotidians de la gent. Junts pel Sí i la CUP tenen tot el dret
202 del món a voler discutir la seva resolució, tot el dret a escenificar i teatralitzar les seves negociacions
203 per invertir el senyor Mas de president, però no tenen cap dret a tenir paralitzats el Parlament i el
204 Govern, mentre al carrer es viuen situacions dramàtiques.</seg>
205 <seg xml:id="ParlaMint-ES-CT_2015-11-09-0301.6.2" xml:lang="ca">La ciutadania espera de
206 nosaltres que centrem tots els nostres esforços a encarrar una situació d'emergència social que viuen
207 àmplies capes de la població. Jo sé que vostès ho saben, però crec que val la pena recordar-ho avui:
208 660.000 persones aturades a Catalunya, 275.000 de les quals porten més de dos anys aturades; 381.000
209 d'aquestes persones aturades no cobren cap tipus de prestació d'atur; 116.000 joves sense feina, i les
210 dones condemnades a contractes a temps parcial amb menys d'una estructura de societat laboral... la taxa
```

Nota. A la izquierda, el documento original. A la derecha, el resultado de aplicarle las funciones programadas.

Una vez definidas las cuatro funciones que analizan y transforman los datos —a partir de 12 reglas de experto— el etiquetado del corpus de 11 680 879 palabras, con 28 299 etiquetas adicionales —de comentarios e interrupciones— se realiza en cuestión de minutos. La atención está puesta, entonces, en el diseño y configuración del etiquetado, no en la repetición de la tarea. Si bien el cuaderno que se propone como ejemplo de esta metodología muestra un código terminado, el proceso —como ya se anticipó— es iterativo. Los resultados se analizan y las funciones se personalizan hasta que el documento resultante está bien formado y es válido.

²³ La muestra de los documentos finales —5 documentos— se encuentra en el repositorio de Github, bajo este link: https://github.com/mariliniapisani/docx2tei_ParlaMint/tree/main/sample_output

El programa no está escrito en un lenguaje avanzado de Python y se espera que un usuario versado en TEI pueda seguir las funciones descritas. Personalizadas para este proyecto, las funciones presentan muchas excepciones y detalles. Difícilmente puedan ser utilizadas para otro corpus sin sufrir modificaciones. Sin embargo, estas modificaciones son posibles dada la gran comunidad en línea que existe en torno a este lenguaje.

En definitiva, la apuesta de este trabajo no es por un código preciso y eficiente. Tampoco es la apuesta exhortar a los humanistas a reemplazar sus habituales métodos de etiquetado en TEI. La apuesta es, simplemente, mostrar que saber utilizar lenguajes de programación comporta valor para tareas centrales en las humanidades digitales y, para aquellos interesados, proponer un camino hacia un etiquetado automatizable a través de árboles, gráficos y matrices de datos.

6. Conclusiones

Continuando con su defensa de la programación dentro de las humanidades digitales, Ramsay y Rockwell (2012) la comparan a la escritura: “to ask whether coding is a scholarly act is like asking whether writing is a scholarly act. Writing is the technology—or better, the methodology—that lies between model and result in humanistic discourse” [Preguntar si la programación es un acto académico es como preguntar si la escritura es un acto académico. La escritura es la tecnología —o mejor, la metodología— que se encuentra entre el modelo y el resultado en el discurso humanístico” (p. 82). Los autores aciertan en su analogía, pero en este trabajo se ha mostrado que la programación no es solo escritura: es también —y quizás más importante aún para el humanista— lectura.

La propuesta de etiquetado en TEI desarrollada en este trabajo para el corpus de los discursos del Parlamento de Cataluña ofrece una respuesta al problema más amplio que presentaba Schöch, el de cómo enriquecer el big data —o, lo que es lo mismo, cómo aumentar el smart data—. El programa en Python propuesto es una alternativa automatizable al etiquetado de corpus en TEI. Los procesos que incluye han sido diseñados a partir de un análisis general —a través de visualizaciones—, de la inspección de casos particulares y de la iteración de las funciones. Esta combinación permite realizar un etiquetado de calidad sobre un gran volumen de datos.

El valor añadido de la programación en Python para el etiquetado en TEI de un corpus no está únicamente en la posibilidad de automatizar un proceso. Esto puede realizarse con hojas de estilo o, incluso, métodos probabilísticos. Por un lado, las hojas de estilo —aunque ordenan estructuralmente y añaden información— no resultan suficientes para el etiquetado de un gran volumen de datos. En primer lugar, no permiten visibilizar las excepciones, dejando fuera de consideración a los casos particulares. En segundo lugar, sus posibilidades para transformar y enriquecer los datos son limitadas. Por otro lado, los métodos probabilísticos del aprendizaje automático son útiles cuando los patrones en los

datos no son conocidos —o conocibles— por el investigador. Si las reglas son conocidas —aunque no todos los datos las cumplan por errores en los documentos fuente— los sistemas basados en reglas de experto son los más adecuados. Además, su configuración y comprensión requiere del humanista un grado de conocimientos matemáticos y computacionales que no son centrales a su formación. Cuando el humanista aplica herramientas foráneas de forma acrítica a sus estudios, los resultados pueden ser problemáticos. Como señala Drucker “the digital humanities can no longer afford to take its tools and methods from disciplines whose fundamental epistemological assumptions are at odds with humanistic method” [Las humanidades digitales ya no pueden permitirse tomar sus herramientas y métodos de disciplinas cuyos supuestos epistemológicos fundamentales están en desacuerdo con el método humanístico] (Drucker, 2011, párr. 6).

¿Por qué adoptar un lenguaje de programación como Python? A diferencia de otras tecnologías, los lenguajes de programación ofrecen mayor flexibilidad. Si bien se utilizan librerías que se toman “prestadas” de otras áreas —científicas y no científicas— saber programar implica no estar restringido a un esquema definido por otros. Herramientas muy útiles como Voyant, por ejemplo, permiten una lectura distante pero bajo ciertos parámetros fijos, algunos otros configurables. Programando —si se conoce bien el lenguaje— casi todo es configurable.

En este sentido, si en los proyectos de humanidades digitales, los sistemas de reglas de experto son programados por los humanistas —quienes mejor conocen el objeto de estudio en cuestión— el resultado será un etiquetado masivo y preciso. Pero estas reglas no pueden ser definidas sin una lectura previa de los textos. Cuando el proyecto implica una gran cantidad de datos, esta lectura resulta muy tediosa o casi imposible.

Por lo tanto, lo que diferencia la alternativa explicada en este trabajo de otros métodos de etiquetado es la posibilidad de —al mismo tiempo— visualizar la estructura subyacente de los textos y analizar tanto los casos frecuentes como las excepciones. Una estructura que

contaba con la ventaja de ya tener un etiquetado previo, facilitando la configuración de las reglas.

Este etiquetado previo —aportado por los estilos de formato del equipo de taquígrafos— fue corregido y adaptado al formato TEI. Pero el análisis también permitió enriquecer esta codificación inicial, incorporando etiquetas que ampliaron la descripción del fenómeno. Un ejemplo de esto es la aplicación de algoritmos más complejos para reconocer el idioma a partir de un segmento de texto o realizar cruces de datos en función de su similitud (en lugar de buscar una coincidencia exacta). Otro ejemplo son los comentarios del transcriptor, que habían sido excluidos de la primera edición de etiquetado del corpus. En esta nueva edición, estos comentarios fueron marcados como notas (para los títulos y comentarios generales) e interrupciones (para los aplausos, murmullos y risas).

Además de la automatización, la propuesta presenta otras ventajas. Entre ellas, la posibilidad de reutilizar el código para el mismo proyecto —en caso de que existan, por ejemplo, modificaciones en el esquema o ampliaciones del corpus—. La propuesta de etiquetado de este trabajo no es cerrada. Aún quedan fenómenos por clasificar, algunos identificados y otros, por identificar. Después de todo, los esquemas de codificación —y en general, los sistemas de clasificación— no son fijos. Pero también el programa y la metodología pueden utilizarse en otros proyectos de etiquetado, quizás muy distintos a ParlaMint. Si bien las funciones no son lo suficientemente generales como para admitir datos muy diferentes, ofrecen un modelo general del proceso: la lectura de los documentos fuentes, la construcción de una matriz de datos, su lectura y la definición de reglas de etiquetado. El programa es, en este sentido, adaptable.

Adicionalmente, el código escrito permite la trazabilidad del proceso, esto es, la posibilidad de verificar en cada paso la acción efectuada. Esto permite la reproducibilidad de los resultados y el examen de pares. Aspectos que, en los estudios humanísticos, suelen

permanecer ocultos, y donde la tarea interpretativa sólo se expone —aunque parcialmente— a través de la publicación de un artículo.

¿Es Python amigable y fácil de aprender? Esta no sea, seguramente, la pregunta adecuada para responder aquí. A pesar de la gran comunidad en línea que sirve como soporte y acompañamiento, todo lenguaje de programación tiene una curva de aprendizaje. De todos modos, la reflexión central en este trabajo versa sobre el valor que las competencias en programación pueden aportar a los estudios humanísticos y si, entonces, vale la pena incluirlas en el estudio y trabajo académico.

Ahora bien, preguntar si estas competencias son útiles —transcurriendo el año 2022— parece una obviedad: están siendo incorporadas incluso como contenidos mínimos de la enseñanza inicial. Pero la educación superior —y sobre todo las disciplinas humanísticas— ofrece mayor resistencia. En algunos casos, esta resistencia se puede leer como un acto de rebelión frente a lo que se interpreta como —tomando las palabras de Rio Riande (2015)— un “acto de clausura de experiencias” (p. 33). Los debates de quién está adentro y quién afuera de las humanidades digitales son un ejemplo.

Entonces, ¿tienen los humanistas que saber programar? Sin ánimos de clausurar a los humanistas que no programan —o que no quieren hacerlo—, en este trabajo se presentó el valor que las competencias en programación tienen para una tarea tan central para las humanidades digitales como es la codificación en TEI. Reflexionando sobre el éxito de estas directrices, Romary (2009) advierte que existen “because it has been put together not so much by techies, but by scholars themselves” [porque han sido elaboradas no tanto por técnicos, sino por los propios académicos] (p. 3). Si son los propios humanistas los que programan para etiquetar, ¿abrazarán la programación de igual manera?

Apéndice

A continuación, se listan los links de interés:

- **Cuaderno de Colab:** Contiene el código en Python y una explicación detallada de algunas funciones programadas. Permite la descarga de los documentos etiquetados.

<https://bit.ly/3pRsxtS> Última actualización el 29 de agosto de 2022.

- **Repositorio en Github:** Contiene los documentos originales, las funciones programadas en Python, tablas de referencia y el repositorio con los documentos finales de muestra.

https://github.com/marilinaapisani/docx2tei_ParlaMint Última actualización el 29 de agosto de 2022.

- **Repositorio con la muestra de los documentos finales etiquetados:**

https://github.com/marilinaapisani/docx2tei_ParlaMint/tree/main/sample_output Última actualización el 29 de agosto de 2022.

Referencias

- Ahmed, U., y Mukhiya, S. K. (2020). *Hands-On Exploratory Data Analysis with Python*. Packt Publishing.
- Allés Torrent, S. (2019). *Introducción a la Text Encoding Initiative. Definición, aplicaciones prácticas y recursos* (2). TTHUB. Text Technologies Hub: Recursos sobre tecnologías del texto y edición digital. <https://tthub.io/aprende/l1-intro-a-tei/>
- Allés Torrent, S. (2019). Sobre la complejidad de los datos en Humanidades, o cómo traducir las ideas a datos. *Revista de Humanidades Digitales*, 4, 1-28.
- Antiba, I. J. (2021). *Corpus textual del Parlament de Catalunya per al projecte europeu ParlaMint: Subcorpus COVID* [Trabajo Final de Máster]. UPF.
- Arano Poggi, S. B. (2020). *Editar digitalment un capbreu: primera aproximació d'etiquetatge en TEI* [Trabajo Final de Máster]. l'Escola Superior d'Arxivística i Gestió de Documents.
https://ddd.uab.cat/pub/trerecpro/2020/264031/AranoPoggi_Silvia_TFM.pdf
- Boletín Oficial del Estado. (2022, February 1). *Real Decreto 95/2022, de 1 de febrero, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Infantil*. BOE.es. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2022-1654
- Drucker, J. (2011). Humanities Approaches to Graphical Display. *Digital Humanities Quarterly*, 5 (1). <http://digitalhumanities.org:8081/dhq/vol/5/1/000091/000091.html>
- Erjavec, T., Ogrodniczuk, M., Osenova, P., Ljubešić, N., Simov, K., Pančur, A., Rudolf, M., Kopp, M., Barkarson, S., Steingrímsson, S., Çöltekin, Ç., de Does, J., Agnoloni, T., Venturi, G., Pérez, M., de Macedo, L. D., Navarretta, C., Luxardo, G., y Fišer, D. (2022). The ParlaMint corpora of parliamentary proceedings. *Language Resources and Evaluation*. <https://doi.org/10.1007/s10579-021-09574-0>
- Erjavec, T., y Pančur, A. (2022). *Parla-CLARINA TEI Schema for Corpora of Parliamentary Proceedings*. GitHub Pages. Recuperado el 16 de Agosto de 2022 de <https://clarin-eric.github.io/parla-clarin>

- Gold, M. K. (2011, May 11). *The Transducer* » *Blog Archive* » *The Digital Humanities Situation*. The Transducer. <http://transducer.ontoligent.com/?p=717>
- Gold, M. K. (Ed.). (2012). *Debates in the Digital Humanities*. University of Minnesota Press.
- Gold, M. K., y Klein, L. F. (Eds.). (2019). *Debates in the Digital Humanities*. University of Minnesota Press.
- González-Blanco García, E. (2013). Actualidad de las Humanidades Digitales y un ejemplo de ensamblaje poético en la red: ReMetCa. *Cuadernos Hispanoamericanos*, (761), 53- 67.
- Guttag, J. (2016). *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press.
- Janes, J., Pinche, A., Jahan, C., y Gabay, S. (2021). Towards automatic TEI encoding via layout analysis. *Fantastic future 21, 3rd International Conference on Artificial Intelligence for Libraries, Archives and Museums, AI for Libraries, Archives, and Museums (ai4lam)*. <https://hal.archives-ouvertes.fr/hal-03527287/document>
- Keralis, S. D.C. (2018). Disrupting Labor in Digital Humanities; or, The Classroom Is Not Your Crowd. In D. Kim y J. Stommel (Eds.), *Disrupting the Digital Humanities* (pp. 273-294). Punctum Books.
- Khemakhem, M. (s.f.). *kermitt2/grobid: A machine learning software for extracting information from scholarly documents*. GitHub. Recuperado el 13 de Agosto de 2022 de <https://github.com/kermitt2/grobid>
- Khemakhem, M. (2020). *Standard-based Lexical Models for Automatically Structured Dictionaries* [PHD Thesis]. Computation and Language [cs.CL]. Université de Paris. <https://tel.archives-ouvertes.fr/tel-03153438/document>
- Kirschenbaum, M. G. (2010). What Is Digital Humanities and What's It Doing in English Departments? *ADE Bulletin*, (150). https://mkirschenbaum.files.wordpress.com/2011/01/kirschenbaum_ade150.pdf
- Kokensparger, B. (2018). *Guide to Programming for the Digital Humanities: Lessons for Introductory Python*. Springer International Publishing.

- Masterman, M. (1962, April). The Intellect's New Eye. *Times Literary Supplement: Freeing the mind*, 38-44.
- Moretti, F. (2005). *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso.
- O'Sullivan, J., Jakacki, D., y Galvin, M. (2015). Programming in the Digital Humanities. *Digital Scholarship in the Humanities*, 30(1), 142-147. doi:10.1093/llc/fqv042
- Pandas (s.f.). *pandas.DataFrame — pandas 1.4.3 documentation*. Recuperado el 18 de Agosto de 2022 de <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>
- Parlament de Catalunya. (s.f.). *El sistema de treball i decisió*. Parlament.cat. Recuperado el 18 de Agosto de 2022 de <https://www.parlament.cat/pcat/parlament/sistema-de-treball-i-decisio/>
- ParlaMint: Towards Comparable Parliamentary Corpora. (s.f.). CLARIN ERIC. Recuperado el 29 de Agosto de 2022 de <https://www.clarin.eu/parlamint>
- Python Software Foundation. (s.f.). *xml.etree.ElementTree — The ElementTree XML API — Python 3.10.6 documentation*. Python Docs. Recuperado el 19 de Agosto de 2022 de <https://docs.python.org/3/library/xml.etree.elementtree.html>
- Ramsay, S., & Rockwell, G. (2012). Developing Things: Notes toward an Epistemology of Building in the Digital Humanities. En M. K. Gold (Ed.), *Debates in the Digital Humanities* (pp. 75-84). University of Minnesota Press.
- Ramsay, S. (2013). Who's In and Who's Out. En D. E. Vanhoutte, D. J. Nyhan, y D. M. Terras (Eds.), *Defining Digital Humanities: A Reader* (pp. 239-241). Ashgate Publishing Limited.
- Rio Riande, G. d. (2015). ¿De qué hablamos cuando hablamos de Humanidades Digitales? In G. d. Rio Riande, G. Sriker, y L. Cantamutto (Eds.), *Las humanidades digitales desde Argentina: tecnologías, culturas, saberes: Actas de las I Jornadas de Humanidades Digitales* (pp. 31-41). FILO:UBA, Facultad de Filosofía y Letras.
- Río Riande, G. d. (2015). A Modo de Introducción: Humanidades Digitales. Mito, Actualidad y Condiciones de Posibilidad en España y América Latina. *ArtyHum*, 1, 7-18.

<https://www.artyhum.com/descargas/monograficos/MONOGR%C3%81FICO%20HD.pdf>

Río Riande, G. d., y Gonzalez Blanco, E. (2015). *Introducción a las Humanidades Digitales*. Material Didáctico Sistematizado.

<https://www.aacademica.org/gimena.delrio.riande/115.pdf>

Rojo, G. (2021). *Introducción a la lingüística de corpus en español*. Taylor y Francis Group.

Romary, L. (2009). Questions & Answers for TEI Newcomers. *Jahrbuch für Computerphilologie*. <https://hal.archives-ouvertes.fr/hal-00348372v2/document>

Schöch, C. (2013). Big? Smart? Clean? Messy? Data in the Humanities. *Journal of Digital Humanities*, 2(3).

<http://journalofdigitalhumanities.org/2-3/big-smart-clean-messy-data-in-the-humanities/>

Stommel, J., y Kim, D. (Eds.). (2018). *Disrupting the Digital Humanities*. Punctum Books.

Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press.

Text Encoding Initiative. (s.f.). *Text Encoding Initiative: TEI*. Recuperado el 18 de Agosto de 2022 de <https://tei-c.org/>

Text Encoding Initiative. (2022a). *1 The TEI Infrastructure - The TEI Guidelines*. Recuperado el 27 de Agosto de 2022 de <https://tei-c.org/release/doc/tei-p5-doc/en/html/ST.html>

Text Encoding Initiative. (2022b). *8 Transcriptions of Speech*. Recuperado el 25 de Agosto de 2022 de <https://tei-c.org/release/doc/tei-p5-doc/en/html/TS.html>

Text Encoding Initiative (2022c). *15 Language Corpora*. Recuperado el 27 de Agosto de 2022 de <https://tei-c.org/release/doc/tei-p5-doc/en/html/CC.html>

Truan, N., y Romary, L. (2022). Building, Encoding, and Annotating a Corpus of Parliamentary Debates in TEI XML: A Cross-Linguistic Account. *Journal of the Text Encoding Initiative*, (14). <https://doi.org/10.4000/jtei.4164>

Turkel, W. J., y Crymble, A. (2012a). *Code Reuse and Modularity in Python*. Programming Historian. <https://programminghistorian.org/en/lessons/code-reuse-and-modularity>

- Turkel, W. J., y Crymble, A. (2012b). *From HTML to List of Words (part 1)*. Programming Historian. <https://programminghistorian.org/en/lessons/from-html-to-list-of-words-1>
- Vanhoutte, E., Terras, M. M., y Nyhan, J. (Eds.). (2013). *Defining Digital Humanities: A Reader*. Ashgate Publishing Limited.
- Venners, B. (2003). Artima - The Making of Python. *Artima*.
<https://www.artima.com/articles/the-making-of-python>