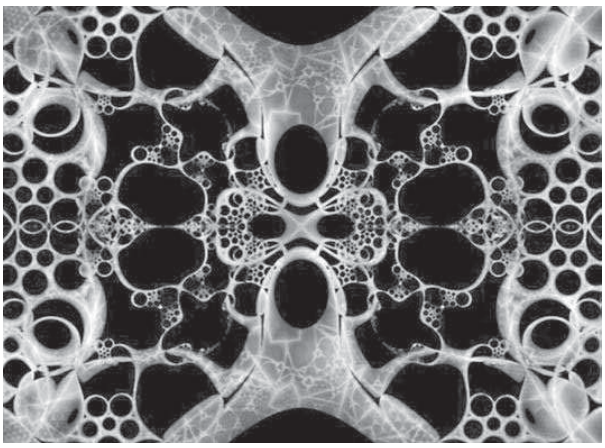
Susanna Allés Torrent / Edición digital y algunas tecnologías aliadas



Fractal.

Introducción

En los últimos años han proliferado múltiples iniciativas de ediciones digitales y esta parece ser una tendencia en auge en los proyectos de Ciencias Sociales y Humanidades. El cambio de soporte ha implicado una reflexión sobre la metodología del tratamiento del texto y la presentación del material, pues, a diferencia de la crítica textual tradicional, la edición digital carece todavía de unos principios definitivos y de unas prácticas compartidas. Aun así, algunas técnicas y lenguajes informáticos han venido al rescate de los investigadores y a formar parte de las primeras etapas del proceso ecdótico digital.

Al hablar de edición digital debemos trazar una clara línea divisoria entre «digital» y «digitalizado». En la actualidad una gran cantidad de textos se encuentran accesibles en línea, pero no debe confundirse una edición científica con una edición digitalizada (como son los casos de proyectos como *Google books*, HathiTrust o Internet Archive). En términos académicos y científicos una edición concebida en formato digital implica un trabajo que debe conjugar la práctica de la crítica textual tradicional con las posibilidades del soporte digital (Pierazzo, 2014).

De hecho, los objetivos de una edición científica, ya sea en formato papel o electrónico, continúan siendo los mismos: preservación del material, acceso, divulgación y análisis e interpretación (Buzzetti y McGann, 2006). Pero mientras que para el formato papel existen metodologías claras y presentaciones concretas, no sucede así con el formato digital. Todavía faltan modelos teóricos que podrían incluir múltiples presentaciones, imágenes, audio, vídeo, anotación e iniciativas de *crowdsourcing*.

La edición digital como concepto va más allá de la simple página web que, por lo demás, conlleva la obsolescencia de ciertos lenguajes de marcado como el HTML; el reto reside en vehicular una nueva y variada aproximación al texto y un análisis optimizado de su contenido. Además, una edición digital implica familiarizarse con los diferentes mecanismos de representación digital del texto y, en concreto, es necesario considerar cómo los materiales pueden ser codificados y organizados para un análisis formal. Por ello, para entender qué posibilidades nos ofrece la edición digital, es necesario conocer el proceso de marcado y los mecanismos internos del proceso de publicación.

A la hora de realizar una edición digital es ya una práctica consolidada el uso de algunas tecnologías, que trataremos aquí. De hecho, en las directrices para editores de ediciones científicas, la Modern Language Association (MLA) ha establecido algunos puntos que ayudan a valorar la calidad de las prácticas editoriales en el campo digital. Entre ellas aparece el uso de los estándares ISO. Así, la adopción de lenguajes de marcado como el XML, o en menor medida el SGML, es altamente recomendable, junto con el acatamiento de unas guías concretas, como el de la Text Encoding Initiative (TEI). Otro de los estándares en los que incide la MLA es el de las instrucciones de transformación a través del uso de lenguajes como el XSLT. Se mencionan, en fin, otros formatos estándares relativos a imágenes (JPG, PNG) o a audio (MPEG, MP3), siempre compatibles en cualquier sistema.

Parece, pues, que dentro del horizonte de la edición digital existen ya algunas pistas que seguir. Concentrémonos, pues, en dos de estos estándares que actualmente gozan de un amplio uso: por un lado, la iniciativa de codificación de textos en XML, conocida como TEI, y, por el otro, las hojas de estilo en XSLT, que permiten la transformación del documento XML-TEI en otros formatos de salida, como (X)HTML, destinado a la visualización web.

Text Encoding Initiative o Iniciativa de codificación de textos

En la comunidad de críticos textuales es ya una práctica difundida y casi consolidada el uso de estándares web XML y de las directrices de la TEI. Este sistema ha demostrado su fiabilidad y su idoneidad para el marcado informático de textos en Humanidades, pues permite captar la estructura lógica, las particularidades físicas y el contenido textual, además de incluir y conectarse a otros elementos como pueden ser bases de datos, imágenes, vídeo o sonido. Permite, además, enriquecer el texto con múltiples capas de información y explotar al máximo las posibilidades de análisis del texto. Su uso, en definitiva, facilita la reutilización del texto en distintos formatos, en contextos diversos, por múltiples usuarios y en diferentes plataformas.

Sus inicios se remontan al año 1987 cuando se expresó la necesidad de establecer unas directrices y unas pautas comunes para la codificación informática de textos en formato electrónico. La primera versión oficial (correspondiente a la TEI P3) no salió a la luz hasta el año 1994. Esta primera versión se construyó con SGML (*Standard Generalized Markup Language*, norma ISO 8879:1986), que resultó demasiado complejo y pesado. Por ello, en el año 2002 se sustituyó por el lenguaje XML y se publicó la

versión TEI P4. Actualmente contamos ya con la versión P5 2.6.0, consultable en su página web oficial.

TEI se expresa en lenguaje XML, que es un estándar ISO, y sus siglas corresponden en inglés a *eXtensible Markup Language*. El uso de estándares es actualmente un requisito indispensable, pues su formato abierto supone un marco de trabajo independiente de cualquier plataforma y asegura la fiabilidad, la sostenibilidad y la interoperabilidad entre diferentes proyectos y distintos sistemas (Smith, 2004). El XML puede combinarse y ser interoperable con formatos diferentes, con lenguajes informáticos de metadatos, de modelización, de presentación, de programación e incluso entre protocolos diferentes.

Este lenguaje es hoy el más usado para estructurar informaciones que pretendan durar en el tiempo y dialogar con otras aplicaciones o plataformas. Se trata de un lenguaje simple, flexible y apto para la legibilidad humana. Además, soporta diferentes sistemas de escritura al utilizar el estándar Unicode, pues su formato de caracteres por defecto es el UTF-8.

Otra de las características del XML es su estructura arbórea, es decir, la estructura de un documento XML contiene siempre un elemento raíz que se ramifica y dentro del cual se anidan el resto de elementos, a la manera de un árbol genealógico. Esta característica es quizás una de las más arduas de comprender desde el punto de vista del humanista, pues pocas veces los textos humanísticos pueden ser desglosados a partir de una estructura similar. Otro concepto básico es el hecho de que XML separa el contenido de su presentación, pues el interés recae en los datos, en la información y en la estructura en la que estos son representados, independientemente de cómo deban presentarse al lector o al usuario *a posteriori*.

En su origen, las siglas TEI correspondían a *Text Encoding for Interchange,* pero actualmente se conocen como *Text Encoding Initiative*. Sus objetivos son, por un lado, promover la creación, el intercambio y la integración de los datos textuales informatizados; por el otro, tratar textos de cualquier tipología, género y disciplina, especialmente en Humanidades, Ciencias Sociales y Lingüística, en cualquier lengua y de cualquier período cronológico. En fin, se dirige tanto a un público novel, sin conocimientos previos en Informática, interesado en codificar un material textual, como también a un público especializado, capaz de buscar nuevas soluciones técnicas, explotar al máximo la naturaleza extensible de TEI y colaborar en la creación de una infraestructura digital sólida.

Gracias a su organización y a la constitución, en el año 2000, del TEI Consortium se asegura el desarrollo y el mantenimiento actualizado de las *Guidelines* o manual de uso y de buenas prácticas. La labor de divulgación llevada a cabo tanto por el Consorcio, que vela por su desarrollo, como por la comunidad de usuarios TEI, siempre con un espíritu de acceso abierto, convierte este sistema de marcado de textos en una de las opciones más sensatas.

En España, el uso de TEI se remonta a algunas décadas aunque no está todavía extendido. Lo cierto es que hay muestras de un interés creciente y son ya varios los proyectos científicos que han adoptado este marco, como demuestra la publicación de la *Guía para editar textos CHARTA según el estándar TEI* (Isasi y Spence, 2014) o las diferentes iniciativas de formación y divulgación del Laboratorio de Innovación en Humanidades Digitales de la UNED. Los motivos de su uso minoritario deben buscarse, quizás, en la existencia solo parcial de las *Guidelines* en traducción castellana. Además, se tiende todavía a delegar el trabajo de codificación a los informáticos, un enfoque no del todo acertado, pues la edición digital debería siempre depender en cada una de las etapas del trabajo filológico. Tampoco existen, por otro lado, empresas especializadas en edición digital y en proyectos científicos digitales que ofrezcan una buena consultoría y servicios adecuados. Hay poca oferta de formación inicial y avanzada, lo que impide enseñar a doctorandos e investigadores, y también faltan herramientas de producción ergonómicas que ahorren el trabajo directo con el código informático a aquellos investigadores que prefieran obviarlo.

El sistema TEI ofrece su código en abierto y, al ser XML, es independiente de cualquier plataforma o programa propietario, aunque hoy en día se halla una clara preferencia por el uso de determinados programas como <oXygen/> (SyncRO Soft) porque ofrece funcionalidades específicas para el manejo de TEI.

No entraremos en los detalles técnicos de la infraestructura TEI, pero sí insistiremos en su naturaleza modular y personalizable. Todo proyecto de edición nace del análisis minucioso del texto, su estructura y sus características tanto formales como semánticas. A partir de este primer análisis, se diseña un modelo abstracto de datos que se aplicará de manera concreta en la codificación del texto. Para ello, es necesario crear lo que se conoce como DTD (*Data Type Definition*), un esquema XML (W3C) o, el más utilizado, un esquema Relax NG (norma ISO/IEC 19757-2), también escrito en XML. Este esquema será el encargado de validar el documento XML-TEI a lo largo del proceso de codificación, como puede verse en la Figura 1.

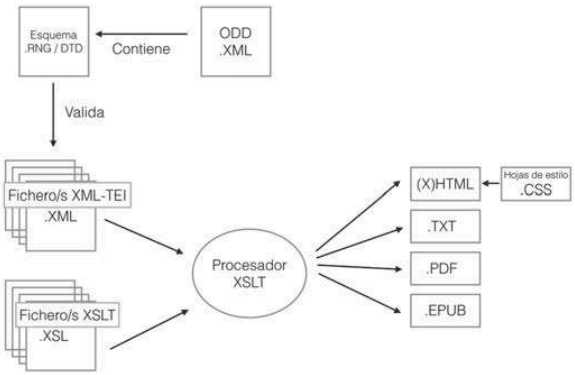


Figura 1. Proceso de codificación y publicación de documentos XML-TEI

Para la creación de este esquema, disponemos de una herramienta, llamada TEI ROMA, creada por el propio TEI Consortium que genera, a través de una interfaz web, el esquema individualizado. En este momento se eligen los módulos en función de las necesidades del editor. Las guías directrices ofrecen un marco de trabajo flexible basado en módulos: hay un total de 21, entre los cuales 4 son obligatorios (*tei*, *core*, *header*, *textstructure*), mientras que los otros son opcionales (*analysis, certainty, corpus, dictionaries, drama, figures, gaiji, iso-fs, linking, msdescription, namesdates, nets, spoken, tagdocs, textcrit, transcr, verse*). TEI cubre una amplia gama de la fenomenología textual: divisiones estructurales del texto, informaciones semánticas, elementos tipográficos puntuales, personas, lugares, fechas, informaciones gramaticales, descripción de fuentes primarias tales como manuscritos o ilustraciones, etc. Así, si un proyecto debe trabajar con textos líri-

cos, incluirá el módulo *verse;* si codifica textos dramáticos, el módulo *drame;* si trabaja con aparato crítico el módulo *textcrit*, y así sucesivamente. Además, podrá elegir entre más de 500 etiquetas incluidas en los diferentes módulos, añadiendo o eliminando las que considere oportunas. Todo documento TEI, pues, necesita un esquema que establezca una sintaxis precisa. Cada proyecto debe crear un esquema a su medida, adoptando los módulos que considere convenientes, eliminando y añadiendo elementos y atributos, y concretando valores. Así la labor de codificación será mucho más fácil y consistente.

Gracias a esta aplicación también se puede crear el fichero ODD, acrónimo correspondiente a *One Document Does it All*, esencial para el proyecto. Se trata de un fichero XML que contiene todas las características del esquema creado y registra las modificaciones que se van realizando. Este fichero permitirá generar cuantas veces deseemos un nuevo esquema y es de vital importancia para compartir la información sobre el modelo de datos personalizado.

La práctica de codificación TEI sugiere establecer una clara diferencia entre, por un lado, lo que corresponde a la estructura lógica textual (como serían capítulos, parágrafos, etc.) y las unidades semánticas, y por el otro, lo que atañe toda cuestión tipográfica o de presentación gráfica. Es, pues, necesario para el procesamiento informático hacer explícita toda la información que se halla en el texto. Todas las partes que se quieran procesar deberán ser marcadas una a una con las etiquetas apropiadas. Existen, obviamente, sistemas más o menos automáticos de marcar y realizar transformaciones del mismo documento, pero una verificación manual es muy aconsejable. En este sentido deberíamos insistir en la importancia de la labor de codificación. En cierta medida, cuando editamos un texto estamos llevando a cabo una codificación particular del texto y, de la misma manera, codificar el texto con marcas o etiquetas implica editarlo. Smith expresó esta idea de manera sucinta y acertada: «*editing is a kind of encoding and encoding is a kind of editing*» (Smith, 2004). Cuando editamos un texto, lo interpretamos, y toda edición, sea cual sea su soporte, es un acto interpretativo susceptible de ser considerado más o menos adecuado según la naturaleza del texto y los fines del proyecto. La codificación informática, pues, debería concebirse como una parte más del proceso de transcripción y publicación en formato digital. Un buen marcado, de hecho, refleja la comprensión del texto por parte del que transcribe y edita, al mismo tiempo que trae a la luz de manera explícita todas las características estructurales implícitas y virtuales del texto (Buzzeti y McGann, 2006).

Las guías directrices establecidas por la TEI son unas pautas recomendadas para la codificación de los textos; no hay una sola manera «correcta» de marcar los textos, sino una serie de buenas prácticas que se aconseja respetar. La libertad a la hora de establecer el esquema de marcado conlleva a su vez una gran diversidad de esquemas de marcado. Por ello, se considera una práctica muy recomendable poner a disposición del público una documentación detallada sobre el tipo de codificación que se ha utilizado, como el ya mencionado proyecto CHARTA.

De todos modos, no podemos obviar algunos inconvenientes del proceso. En primer lugar, no siempre un texto tiene una estructura arbórea y pretender encajarlo en una forma tal supone simplificarlo. En segundo lugar, determinados rasgos textuales a veces no hallan una respuesta satisfactoria o simplemente no están previstos por las guías directrices de TEI. En fin, el uso de TEI implica el conocimiento de lenguajes informáticos, no solo de XML, sino también de otros relacionados para la transformación y su presentación. Y todavía hoy muchos investigadores no están dispuestos a invertir su tiempo en esta labor.

XSLT: lenguaje de transformación

XSLT, acrónimo correspondiente a *eXtensible Stylesheet Language Transformations*, es un estándar desarrollado por el World Wide Web Consortium (W3C) en noviembre 1999. Este lenguaje de programación está escrito en XML, ahora normalmente usado en su versión 2.0 (enero 2007).

Forma parte de la familia de los estándares XSL (*eXtensible Stylesheet Language*), donde también encontramos el XSL:FO (*Formatting Objects*) que permite, por ejemplo, transformar los documentos XML en formatos de lectura como PDF. De hecho, todos los documentos de esta familia tienen como extensión .XSL (y no .XSLT o .XSLFO).

El último elemento del acrónimo, pues, corresponde a la palabra «transformación». Se trata de un lenguaje de programación que permite la transformación documentos XML (*input*) y generar otros de salida (*output*) en formato diferente. A este se relegan todas las transformaciones de carácter estructural y de codificación, mientras que la presentación y cuestiones de diseño web (*layout*), como la distribución en la página HTML, cuestiones tipográficas, colores, etc. se relegan a las hojas de estilo CSS (*Cascading Style Sheets*).

Se trata de un lenguaje declarativo basado en una serie de instrucciones no ordenadas, llamadas *templates* («plantilla», «formulario»); cada una de estas reglas especifica un resultado concreto, a partir de un segmento concreto del código XML, en el fichero resultante. En la actualidad existen algunos tutoriales sobre XSLT concebidos especialmente en contextos de Humanidades, pero la mayoría van dirigidos a un público diferente y desde la perspectiva del informático. Por último, vale la pena señalar que trabajar con XSLT implica a su vez saber los principios básicos de XML, y es muy recomendable conocer también HTML y CSS.

Principios de funcionamiento técnico

Para crear una transformación en XSLT necesitamos contar con tres elementos (Figura 1). En primer lugar, uno o varios documentos XML-TEI bien formados y válidos, con su modelo asociado de DTD o de esquema. A continuación, un programa u hoja de estilo XSLT con todas las instrucciones de conversión requeridas. Y, finalmente, un procesador XSLT, es decir, un programa que lea la hoja XSLT y el documento XML y que opere la transformación solicitada siguiendo las instrucciones del programa.

Veamos, por encima, cuál es el funcionamiento interno de esta transformación. Por un lado, tenemos el documento (.XML), por el otro, la hoja de estilo (.XSL); la elaboración de una hoja XSLT dará como resultado un nuevo documento, del formato elegido, sin modificar nuestro fichero principal. Recordemos que los documentos XML tienen como característica principal la de poseer una estructura arbórea. Pues bien, el lenguaje XSLT, al estar escrito en

XML, funciona consecuentemente recorriendo el documento XML desde el elemento raíz, a través de todos sus descendientes, de izquierda a derecha. Así, XSLT recorre o, en términos informáticos, *parsea* («analiza las partes», «descompone», «decodifica», del inglés *parse* o *parsing*) el documento fuente y crea en memoria una representación arbórea de este. A continuación, busca en el programa XSLT las reglas que se aplican a la raíz del documento XML, ejecuta la instrucción y crea el segmento del árbol resultante en el nuevo documento. El programa entonces regresa al árbol del documento principal y lo recorre de nuevo, buscando si hay alguna regla XSLT que deba aplicar. Si el procesador encuentra dos reglas aplicables a un mismo nodo, se regirá por sus reglas internas de prioridad; en cambio, si no encuentra regla alguna, aplicará las que tiene por defecto.

Se obtiene como resultado un nuevo fichero creado por el procesador, con todas las reglas aplicadas y del formato que se haya especificado. Es realmente importante comprender el funcionamiento arborescente pues de ello depende la lógica de la transformación.

El lenguaje utilizado para expresar los elementos del documento XML y su ubicación es el lenguaje XPath, otro estándar creado y desarrollado por el W3C y expresado en XML; este permite seleccionar cualquier nodo de la estructura arbórea del documento XML para su procesado informático posterior. Se trata de un lenguaje que es utilizado e interpretado por otros lenguajes basados en XML; además de XSLT, lo utilizan XQuery, Schematron, XLink o XPointer. En definitiva, la expresión de los ejes en XPath es indispensable, dado que esta sintaxis es la que nos permite individualizar cualquier punto de nuestro documento a través de patrones (o *paths*) y recuperar ciertos atributos y valores del documento base XML que queramos filtrar.

Casuística o modalidades de aplicación

Los casos en los que podemos utilizar XSLT son múltiples, aunque existen al menos tres contextos en los que su uso es realmente útil:

- Para ejecutar una transformación de un documento XML-TEI en un (X) HTML con el fin de visualizarlo en la web; otras transformaciones también muy utilizadas son las encaminadas a obtener otros formatos de salida como un texto plano o un PDF.
- Aliadocon XPath sirve para interrogar un documento XML en búsquedas más o menos complejas, de manera que nos ayuden a explotar, a analizar y a verificar un documento XML.
- Para hacer un copia idéntica del documento base XML-TEI (*identity transform*) o transformar solo pequeñas porciones del código original.

En fin, XSLT es capaz de convertir las informaciones estructuradas en XML, según un modelo A, por ejemplo un documento TEI o un modelo propietario, y transformarlo, con las reglas oportunas, en otro modelo B, por ejemplo, una gramática de *wiki* Semántica, o un archivo EAD (*Encoding Archival Description*) u otro lenguaje de metadatos como Dublin Core. El objetivo es permitir el intercambio de informaciones entre aplicaciones utilizando los modelos de datos diferentes y asegurar la interoperabilidad.

Las posibilidades que ofrece el lenguaje XSLT son muy poderosas. El más conocido es el caso de una edición crítica digital que, colacionados todos los manuscritos y codificadas informáticamente todas las variantes, ofrezca la posibilidad de reconstruir cada uno de los manuscritos y, a su vez, proporcione un texto crítico con aparato de variantes y notas de cualquier tipo. Es aquí donde el trabajo del filólogo abre nuevos horizontes en la práctica de la crítica textual.

Amplio uso de XSLT para la transformación y la visualización de documentos XML-TEI

En el entorno TEI, el lenguaje XSLT es realmente muy usado y las iniciativas que lo integran son de diversa naturaleza. Quizás la más relevante sea el conjunto de hojas de estilo prediseñadas por el mismo TEI Consortium. Estas hojas XSLT para documentos XML-TEI se mantienen como código abierto en GitHub, van acompañadas de una buena documentación en línea y son actualizadas a cada nueva salida de las directrices de la TEI. Disponen, además, de una *wiki* TEI, donde se tratan los puntos más relevantes sobre el uso y reutilización de XSLT.

Asimismo, han surgido diversas plataformas que generan transformaciones con la tecnología XSLT, convirtiendo de este modo los documentos de un formato a otro. Así, por ejemplo, en aplicaciones como OxGarage Conversion se pueden subir documentos en diversos formatos (.DOC, .TXT, .ODT, .RTF, etc.), presentaciones u hojas de cálculo y convertirlos en otros tantos (XHTML, PDF, TEI, EPUB, LaTex, etc.). Desde la École Nationale des Chartes en París también se han creado aplicaciones que transforman en línea los documentos e incluso hacen estadísticas de los documentos XML-TEI. Además, existen plataformas de publicación que también utilizan XSLT para la transformación y presentación de sus documentos, como el TEI Boilerplate. Algunas incluso ofrecen las hojas de estilo y permiten modificar el código según las exigencias de su proyecto.

En fin, la mayoría de proyectos científicos, al afrontar una edición en XML-TEI apuestan por una transformación a través de hojas de estilo XSLT. Muy frecuentemente a esta infraestructura, a la que acompañan una presentación en HTML y una o varias hojas de estilo CSS, viene a añadirse algún que otro lenguaje de programación, como JavaScript, que permite montar páginas dinámicas y obtener resultados satisfactorios.

Para concluir, querría retomar la idea sobre la necesidad del uso de estándares, porque este parece ser el camino consensuado por la comunidad académica. Para llevar a cabo un proyecto de edición digital deben entenderse los mecanismos de lenguajes de codificación, como XML, y hacerlo desde un marco de trabajo adecuado a las necesidades editoriales, tal como sucede con la iniciativa TEI. Además, comprender los mecanismos de transformación y publicación implica acercarnos a algunos lenguajes como el XSLT. Se trata, en definitiva, de unas tecnologías que corresponden a las dos caras de una misma moneda: el conocimiento y la comprensión de sus mecanismos internos se complementan ayudando a concebir la edición digital y su proceso de creación de una manera óptima.

S. A.— COLUMBIA UNIVERSITY