3. INTRODUCCIÓN A XML

3. 1. Marcado y etiquetado. Primeros conceptos de XML

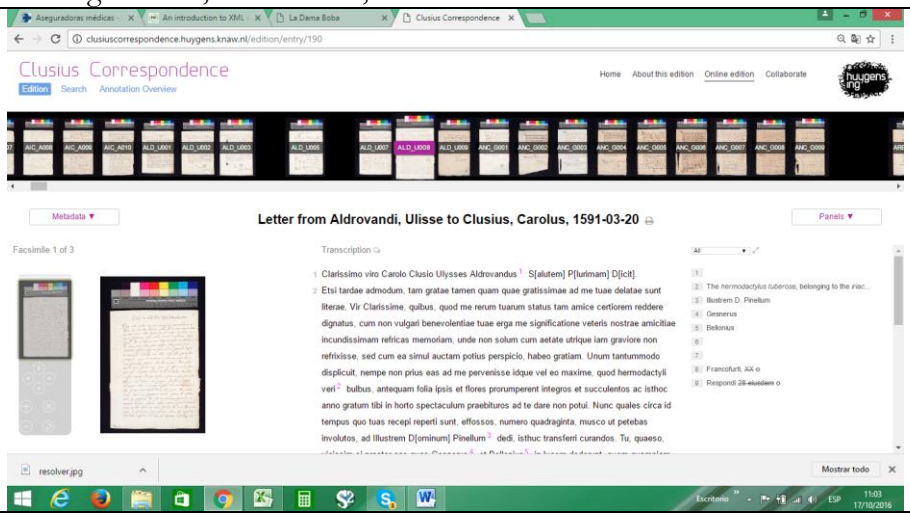
Las humanidades giran en torno a textos que se encuentran en libros, manuscritos y documentos de archivo. En las últimas décadas, sin embargo, están proliferando otras manifestaciones culturales digitales como sonidos, imágenes, blogs, tweets, etc. Las humanidades digitales abordan las tecnologías digitales y las técnicas para manipular tales manifestaciones de un modo integrado. El lenguaje de marcado o etiquetado es una de las tecnologías claves que subyace en esta integración.

Los textos tienen distintas dimensiones: un documento tiene una presencia física con aspectos visuales que pueden ser transferidos más o menos automáticamente desde una instancia física a otra; un texto tiene propiedades lingüísticas y estructurales que pueden ser transcritas, traducidas y transmitidas con alguna intervención humana; un texto transmite información sobre el mundo real, información que puede ser comprendida (o no), anotada o incluso usada para generar nuevos textos; un texto viene usualmente asociado a metadatos que documentan qué es, de dónde viene, dónde y cuándo fue publicado, etc. Un buen etiquetado debe operar en todas estas dimensiones.

Un texto digital puede ser un simple sustituto que representa la apariencia de un documento existente:

Facsímil de carta de Ulisse Aldrovandi a Carlos Clusio, 1591-03 20

O puede ser una representación de su contenido lingüístico y estructura, con anotaciones adicionales sobre su significado, su contexto, etc.



Clusius Correspondence - http://clusiuscorrespondence.huygens.knaw.nl/edition/entry/190

¿Cuál es la parte esencial de un texto? ¿La forma de las letras y su presentación? ¿El original del que deriva esta copia? ¿Las historias que leemos en él? ¿Las intenciones de su autor? Un documento es algo que existe en el mundo, que se puede digitalizar. Un texto es una abstracción, creada por y para una comunidad de lectores, que podemos etiquetar.

Un texto es más que una secuencia de caracteres o símbolos codificados, y también es más que una secuencia de formas lingüísticas. Tiene una estructura y una función comunicativa. Tiene además múltiples posibles lecturas y su significado puede ser enriquecido por medio de la anotación. El lenguaje de marcado hace que todas estas posibilidades se vuelvan explícitas. Y solo lo que sea explícito puede ser buscado, encontrado, mostrado y analizado posteriormente de una manera fidedigna.

El lenguaje de marcado sirve:

- Para hacer explícito a una máquina lo que es implícito para una persona.[1]
- Para añadir valor aportando múltiples anotaciones.
- Para facilitar el reuso del mismo material en diferentes formatos, en diferentes contextos, por diferentes usuarios. De esta manera, no tenemos que estar limitados a la vista de un solo editor o receptor.

El estándar[2] XML (Extensible Mark-up Language), una versión simplificada de SGML, cumple todos estos requisitos. Es uno de los lenguajes más utilizados en el mundo de la informática, por su simpleza, flexibilidad y adaptación para asegurar la interoperabilidad con una gran cantidad de aplicaciones, plataformas y lenguajes informáticos. XML, por ejemplo, está en la base de cualquier documento Word.

La unidad básica del marcado en XML es la "marca" o "etiqueta" ("tag", en inglés), denominada técnicamente elemento. La misión principal de XML es el marcado electrónico de textos, de cualquier tipología, para que sean procesados por una simple máquina. El lenguaje utilizado para la marcación de datos es simple y legible por los humanos ("human-readable"), frente a los códigos de barras o datos binarios (que pueden ser leídos por un ordenador). Como hemos dicho, XML es un lengua de marcado descriptivo o semántico que consiste en aislar y describir fragmentos de texto a través de etiquetas para indicar una función semántica o estructural, sin especificar el formato, la presentación o el orden que tendrán en su aspecto final. Son otros lenguajes de marcado los que se encargan de la presentación, como HTML o CSS. XML, por tanto, no hace nada por sí mismo, simplemente es un lenguaje de marcado, no un lenguaje de presentación, ni de programación, ni una base de datos.

El marcado o etiquetado es una actividad eminentemente académica. La aplicación del etiquetado a un documento no es un proceso automático. A la hora de decidir qué etiqueta aplicar y cómo esta representa el original, se emprende las tareas propias de un editor. No existe un marcado neutral, pues siempre implica una interpretación. El etiquetado puede ayudar a responder cuestiones de investigación y decidir qué etiquetas son necesarias para poder responder estas cuestiones es una actividad de investigación en sí misma. Una buena codificación textual no es tan fácil y rápida como se puede pensar. Se necesita un detallado análisis del documento antes de codificar para que el etiquetado resultante sea útil. El lenguaje de marcado, en definitiva, debe ser capaz de especificar todos los

[1] "The web of human-readable document is being merged with a web of machine-understandable data. The potential of the mixture of humans and machines working together and communicating through the web could be inmense"; cf. Tim Berners-Lee, *The World Wide Web: a very short personal history* (1998).

[2] Un estándar informático es un conjunto de prácticas consensuadas que permiten el intercambio de información entre los usuarios de internet. Al igual que las lenguas modernas, que tienen un "estándar" o sistema y un conjunto de reglas gramaticales, la web también tiene diferentes lenguajes regidos por una serie de reglas y recomendaciones. Hay muchos tipos de estándares, pero los más conocidos son los del W3C (World Wide Web Consortium), que se ocupa de potenciar y optimizar la web creada por Tim Berners-Lee en 1994. Algunos de los estándares más conocidos, además de XML, son HTML, HTML5, XHTML, CSS y DOM.

caracteres encontrados, hacer explícitas las estructuras percibidas, representar esa estructura de una forma lineal procesable y proporcionar una variedad de metadatos o anotaciones.

XML juega hoy en día un papel central en el intercambio de una gran variedad de datos en la Web y otras aplicaciones, y es el más usado para estructurar datos que pretendan durar en el tiempo y ser interoperables con otras aplicaciones y plataformas. Es compatible con la mayoría de los lenguajes Web (tipos de documentos, de programación, de presentación, de metadatos, etc.) y es utilizado por una gran comunidad a nivel mundial que proporciona soporte y herramientas para trabajar con él.

XML se parece mucho a HTML, pero hay algunas diferencias:

- XML es extensible, lo que significa que puede ser extendido y adaptado para responder a necesidades diferentes. Incluso no existe una lista fija de etiquetas, que se pueden crear en función del proyecto. TEI (*Text Encoding Initiative*) trabaja con sus propias etiquetas.
- XML tiene que estar bien formado, de lo contrario será rechazado por cualquier procesador. Un documento XML tiene, además, que ser validado a través de un esquema específico, como veremos más adelante.
- XML se centra en la descripción semántica de los fragmentos de texto, mientras que HTML se ocupa principalmente de la presentación. Por ejemplo, tenemos un texto donde aparecen dos tipos de palabras en cursiva, unas son los títulos de las obras y otras palabras extranjeras. HTML solo marcaría la idéntica forma de presentar ambos tipos de palabras palabra; XML se centraría en el contenido de las palabras, distinguiendo entre títulos y palabras extranjeras: <palabraExtranjera>palabra</palabraExtranjera>; <título>palabra</título>. La forma en que las palabras comprendidas por estas etiquetas serán presentadas depende de un paso posterior por medio de una hoja de estilo (XSLT). Es decir, XML separa el contenido de la presentación.

3. 2. Partes, reglas y estructura de un documento XML

Un documento XML contiene al menos un elemento representado por una etiqueta de inicio, algún contenido opcional y una etiqueta de cierre. El contenido de un elemento puede ser una cadena de caracteres Unicode, o uno o más elementos. Un elemento puede también tener atributos (@when, en los ejemplos de más abajo), cada uno de los cuales está formado por un nombre y un valor. Un documento XML debe estar bien formado y debe ser válido.

```
<?xml version="1.0" encoding="UTF-8"?>
<listPerson>
  <person>
    <persName>Tito Livio </persName>
    <birthdate when="59 a.C.">59 a.C. </birthdate>
    <birthplace>Padua</birthplace>
    <deathdate when="17">17 d. C.</deathdate>
    <deathplace>Padua</deathplace>
  </person>
  <person>
    <persName>Cornelio Tacito</persName>
    <birthdate when="55">ca. 55</birthdate>
    <birthplace>Gallia Narbonense</birthplace>
    <deathdate when="120">ca. 120</deathdate>
    <deathplace>Italia</deathplace>
```

```
  </person>
  <person>
    <persName>Gayo Suetonio Tranquilo</persName>
    <birthdate when="70">ca. 70</birthdate>
    <birthplace>Hippo Regius, Argelia</birthplace>
    <deathdate when="122">ca. 122</deathdate>
    <deathplace>Italia</deathplace>
  </person>
</listPerson>
```

Partes de un documento XML:

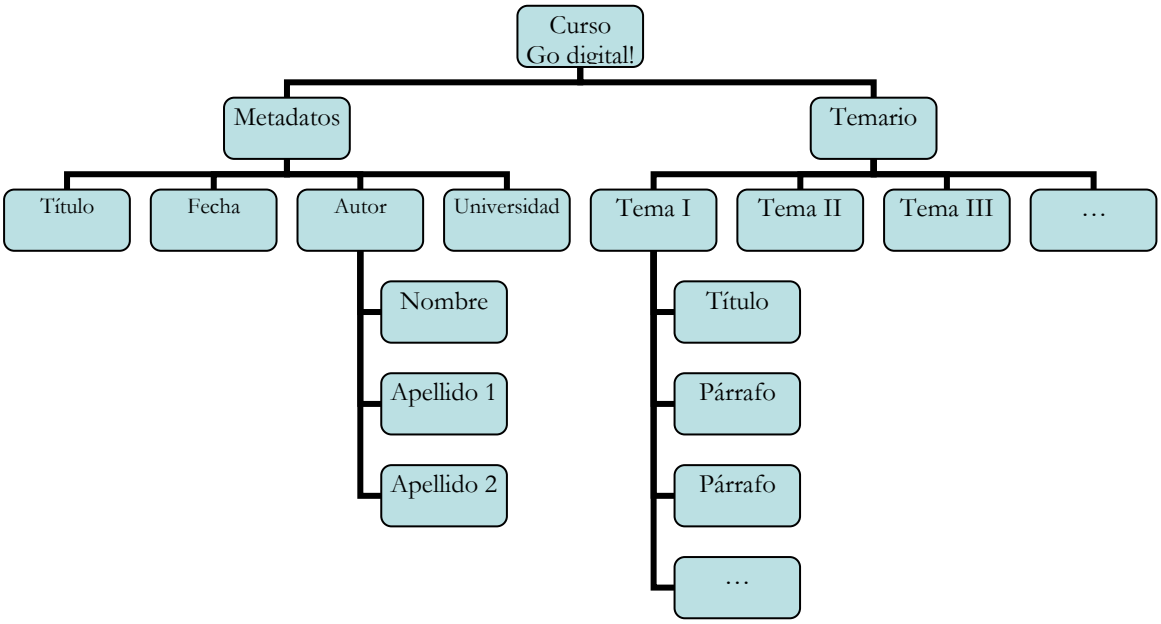
- La declaración XML. Este elemento no puede ir precedido por ningún comentario, ni espacio en blanco ni ningún otro elemento. Sus objetivos son declarar que estamos ante un documento XML, declarar qué versión del estándar XML sigue y declarar que caracteres de codificación sigue el documento La recomendada por defecto es 'UTF-8' (Unicode).
- En algunos casos un documento XML puede depender de un modelo abstracto que puede tener diferentes formas. Nos referimos a los esquemas, que establece la serie de elementos, sus nombres y atributos que estos podrán contener, y de qué manera podrán anidar los unos en los otros. Un esquema permite, por tanto, comprobar (por medio del "parser") o que cada "capítulo debe comenzar con un encabezado", o qué "recetas deben incluir una lista de ingredientes" o que "los valores para el atributo @when son todos fechas válidas". Los esquemas son, en general, ficheros diferentes: un caso frecuente es el modelo de la DTD (Document Type Declaration), ampliamente usado en XML, aunque en desuso en TEI, donde se prefieren los esquemas RelaxNG (Regular Language XML Next Generation). Así pues, podemos encontrarnos en el prólogo del documento con una declaración del tipo de documento que tiene esta forma: <!DOCTYPE nombre del elemento raíz SYSTEM "nombre.dtd">
- Declaración de Namespace o espacio de nombre. Un espacio de nombre (Namespace) es una etiqueta, declarada con sintaxis URI (Uniform Resource Identifier), usada para identificar un grupo de nombres de elementos XML y distinguirlos de otros. Un documento XML puede incluir elementos de muchos espacios de nombre diferentes. Todos los documentos TEI muestran por defecto que sus elementos pertenecen al espacio de nombre TEI comenzando con la siguiente declaración <TEI xmlns="http://www.tei-c.org/ns/1.0">. El espacio de nombre XML se usa también por TEI para los atributos globales @xml:id y @xml.lang. Otros espacios de nombre pueden aparecer también en un documento TEI: por ejemplo, MathML: <TEI xmlns= "http://www.tei-c.org/ns/1.0" xmlns:math= "http://www.mathml.org">
- El elemento raíz del documento. Todo documento XML tiene un solo elemento raíz ("root" en inglés), que no depende de ningún otro elemento y que contiene todos los otros.
- Otros elementos y contenido.
- Atributos y valor.
- Comentarios y entidades. Los documentos XML pueden tener comentarios para que los autores puedan dejar sus notas, normalmente referidas al proceso de codificación. No están destinados a ser procesados por la máquina.

  <!-- ¡esto es un comentario! -->

- Entidades de referencia ("entity references"). Pueden ser de diversos tipos; citamos aquí entidades de caracteres especiales que no deben ser interpretados como código. Los signos que deben ser codificados de manera especial son los siguientes:
  < para indicar "menor que" o el paréntesis angular de la marca de apertura <
  > para indicar "mayor que" o el paréntesis angular de cierre de la marca >
  & para indicar &
  " para indicar las comillas dobles ""
  ' para indicar el apóstrofe o las comillas simples.

Las reglas de un documento XML:

- Un documento XML representa una especie de árbol. Tiene un elemento simple raíz y muchos nodos.
- Cada nodo puede ser un subárbol, un solo elemento (posiblemente con algunos atributos) o una cadena de caracteres.
- Cada elemento tiene un nombre o identificador general. Un elemento es una unidad de información semántica compuesta por una marca de apertura (<) y una de cierre (/>). Puede contener texto, otro elemento o estar vacío (<elemento/>). El nombre de su identificador puede solo contener letras y cifras, sin espacios ni acentos; solo se puede usar algunos signos como – (guión), _ (guión bajo) y . (punto); el nombre no puede empezar ni por un número, ni por un signo, ni por "xml".
- Los elementos XML y los atributos son sensibles a las mayúsculas y las minúsculas. Los atributos añaden características adicionales al propio elemento. Su nombre va siempre dentro de la etiqueta de apertura y va precedido de un espacio, seguido del símbolo = y de comillas, dentro de las cuales aparece el valor:
  <fecha cuando= "2016-11-05">5 de noviembre de 2016</fecha>



El elemento raíz es "curso" del que descienden todos los otros elementos. Estos elementos pueden constituirse en nodos, que pueden ser: a) un subárbol, por ejemplo <metadatos> con sus cuatro elementos descendientes <título>, <fecha>, <profesor> y <universidad>; b) un simple elemento

<nombre>; o la secuencia de caracteres que se encuentra como contenido en el interior de los documentos.

Para representar un árbol XML recordemos que hay que tener en cuenta lo siguiente:

- El documento comienza con una instrucción especial de procesamiento.
- Los elementos están marcados por etiquetas de inicio y de fin.
- Los caracteres < y & deben evitarse si queremos utilizarlos como tal
- Los comentarios están delimitados por <!-- y -->.
- Las parejas Attribute name = value son suministradas en la etiqueta de inicio y pueden ser dadas en cualquier orden, separadas por espacios
- Las Entity references (por ejemplo < para < y Ç para Ç) están delimitadas por & y ;.

3. 3. Conceptos de "bien formado" y "válido"

Un documento XML debe estar bien formado y ser válido. Para que esté bien formado debe cumplir los siguientes requisitos: que haya un simple nodo raíz que contenga todo el documento XML; que cada subárbol esté propiamente anidado en el nodo raíz; que los nombres de los elementos y atributos XML sean sensibles a las mayúsculas y minúsculas; que se usen marcas de inicio y de fin; que los valores de los atributos estén siempre especificados. Además de bien formado, el documento XML debe ser válido, lo que significa que obedece las reglas del esquema especificado, tal como los de TEI.

Un documento XML es válido si se ajusta a algunas reglas estructurales adicionales que conforman el esquema. Recordemos que el esquema permite especificar qué elementos pueden aparecer como el elemento raíz de un documento, qué elementos y qué atributos pueden aparecer y dónde; y los nombres, tipos de datos y valores por defecto de los atributos.

Para concluir este capítulo resolvamos la siguiente sopa de letras:

SGML Standard Generalized Markup Language
HTML Hypertext Markup Language
W3C World Wide Web Consortium
XML eXtensible Markup Language
DTD Document Type Definition (or Declaration)
CSS Cascading Style Sheet
Xpath XML Path Language
XSLT eXtensible Stylesheet Language - Transformations
XQuery XML Querying
RELAXNG Regular Expression Language for XML (New Generation)
TEI Text Encoding Initiative

Bibliografía

Introduction to Tei: Training Workshop - http://tei.it.ox.ac.uk/Talks/2014-11-warsaw/
Manual en español XML - http://www.desarrolloweb.com/manuales/manual-introduccion-xml.html
Tutorial en español XML - <http://zvon.org/xxl/XMLTutorial/General_spa/book.html>
W3schools.com- Tutorial XML - <*http://www.w3schools.com/xml/default.asp*>
W3schools.com- Tutorial XPath - <http://www.w3schools.com/xml/xpath_nodes.asp>