



TEIDOWN: USO DE MARKDOWN EXTENDIDO PARA EL MERCADO AUTOMÁTICO DE DOCUMENTOS TEI

TEIDOWN: USE OF MARKDOWN EXTENDED FOR AUTOMATIC MARKING OF TEI DOCUMENTS

Alejandro Bia Platas

Universidad Miguel Hernández de Elche

abia@umh.es

Ramón P. Neco García

Universidad Miguel Hernández de Elche

ramon.neco@umh.es

Resumen

La creación de nuevos documentos XML, desde cero o a partir de texto plano, puede ser una tarea difícil, lenta y propensa a errores, sobre todo cuando el vocabulario de marcado utilizado es rico y complejo, como es el caso del TEI. Por lo general, lleva bastante tiempo lograr que el documento valide por primera vez.

Juntando el espíritu del viejo SGML y los principios del Markdown, llegamos a la idea del proyecto TEIdown, que consiste en una ampliación de la sintaxis del Markdown para crear documentos XML-TEI, y los programas de transformación correspondientes. Con este enfoque es fácil obtener un documento TEI válido en un tiempo muy corto, evitando pasar por una larga lista de errores de validación.

Palabras clave: Marcado automático. XML. TEI. TEIdown. Markdown extendido.

Abstract

Creating new XML documents, from scratch or from plain text, can be a difficult, time consuming and error prone task, especially when the markup vocabulary used is rich and complex, as is the case of the TEI. It usually takes a good amount of time to make the document validate for the first time.

By joining the spirit of good-old SGML, and the principles behind Markdown, we came to the idea of the down2TEI project, which consists of an extension to the markdown syntax meant for the creation of TEI-XML documents, and the corresponding parsers needed to perform such conversion. With this approach, it is easy to obtain a valid TEI document in a very short time, avoiding going through a long list of validation errors.

Keywords: Automatic Tagging, XML, TEI, TEI-down, Extended-Markdown.

1. INTRODUCCIÓN

La creación de documentos nuevos en XML (*Extensible Markup Language*) partiendo de cero, o a partir de texto plano, puede ser una tarea difícil, lenta y propensa a errores, especialmente cuando el vocabulario de marcas utilizado es rico y complejo, como es el caso de la norma TEI (*Text Encoding Initiative*), cuyo vocabulario posee más de 500 etiquetas diferentes. En general, lleva bastante tiempo conseguir que el nuevo documento TEI se valide por primera vez y los errores que aparecen pueden ser numerosos y difíciles de encontrar y corregir.

Hace un par de décadas el SGML (*Structured Generalized Markup Language*) permitía ciertas libertades a los codificadores de documentos, orientadas a ahorrar tiempo y esfuerzo, como por ejemplo dejar ciertas etiquetas sin cerrar, u omitir las comillas en los valores de algunos atributos. En este sentido, el SGML era más permisivo que el XML. Esto era bueno para los codificadores de documentos, pero hacía que el desarrollo de programas de análisis y procesamiento de documentos SGML fuera una tarea muy complicada, por el número y tipo de reglas de inferencia que estos programas debían incorporar. Por el contrario, el XML, al ser mucho más restrictivo y no permitir todas las libertades que brindaba el SGML, posee una sintaxis más previsible y en consecuencia más fácil de procesar, lo cual contribuyó a su rápida popularidad.

Por otro lado, en el mundo de las wikis (sitios web donde el lector puede editar y agregar contenidos directamente a través del navegador web), surgieron varias *notaciones wiki* (*wiki-languages*), con el fin de simplificar o evitar por completo el uso de etiquetas HTML

en la edición de textos para la web. Entre estas notaciones breves, el Markdown es una de las más recientes y ha sido ampliamente aceptada e incorporada por muchos proyectos importantes. Su objetivo, al igual que otras notaciones wiki, es evitar escribir etiquetas HTML; pero en el caso del Markdown se insiste además en que la legibilidad del texto se mantenga intacta.

Uniendo el espíritu del SGML y los principios del Markdown, en el presente artículo se presenta el proyecto TEI down, que consiste en una ampliación del Markdown para obtener una sintaxis abreviada que sirva para la creación rápida de documentos XML-TEI y en la creación de los programas de procesamiento correspondientes para llevar a cabo dicha conversión. Con este enfoque, resulta más sencilla la obtención de un documento TEI válido en un tiempo reducido, evitando pasar por una larga lista de errores de validación.

Este enfoque, sin embargo, tiene algunas limitaciones. Fue pensado para procesar las etiquetas más comunes, como las utilizadas para marcar textos en prosa y en verso, y las más comúnmente usadas en el *teiHeader* (el encabezamiento de metadatos de los documentos TEI). En resumen, las etiquetas más frecuentes de la DTD son *teixlile.dtd*. Para aplicaciones más especializadas, como el marcado de manuscritos, es necesario agregar más etiquetas a mano después de la conversión inicial, pero incluso en estos casos se ahorra una cantidad significativa de tiempo en la creación del documento TEI.

En el presente artículo se describirá la notación Markdown extendida, el proceso de transformación a TEI y las características de los documentos TEI resultantes, así como los beneficios y limitaciones de esta técnica.

1.1. TEI: Text Encoding Initiative

La *Text Encoding Initiative* (TEI) es un consorcio que desarrolla y mantiene un estándar para la representación de textos de forma digital. Se trata de un proyecto de investigación en humanidades digitales con una amplísima difusión y utilización en bibliotecas y colecciones de textos digitales y en la creación de corpus lingüísticos. Se basa en el lenguaje XML, una versión simplificada del SGML.

La mayoría de usuarios del formato no usan todas las etiquetas disponibles, sino que definen una *personalización*, en la que se usa un subconjunto de etiquetas que son específicas para el proyecto en cuestión. El propio TEI define un mecanismo de personalización conocido como ODD que sirve para este propósito, de tal forma que en una especificación ODD se define el modelo del contenido del documento, así como otras restricciones de uso. Un ejemplo de personalización es *TEI Lite*, en el que se define un formato de fichero XML para el intercambio de textos y que consiste básicamente en una selección

manejable de los numerosos elementos que están disponibles en las *Guidelines* completas de TEI.

2. DIFERENTES APROXIMACIONES PARA COMENZAR LA ESCRITURA DE UN DOCUMENTO TEI-XML NUEVO

En general, cuando se trata de comenzar la escritura de un documento TEI-XML nuevo, se puede realizar desde distintas aproximaciones o estrategias. En este epígrafe se describirán las más comunes usadas en la práctica con el objetivo de justificar la herramienta propuesta en nuestro proyecto (TEI-down). Estas aproximaciones generales son las siguientes:

- A partir de la salida de un proceso de reconocimiento óptico de caracteres (OCR).
- A partir de textos escritos en HTML u otros lenguajes de marcado.
- A partir de textos escritos en MS-Word u otros formatos de ofimática.
- A partir de documentos escritos en LaTeX.
- Partiendo *de cero* o de un documento *en blanco*.
- Usando una plantilla vacía (*empty-skeleton*).

Tradicionalmente se han destacado los principales inconvenientes que tienen los procesadores de texto de tipo WYSIWYG (*What You See Is What You Get* o *Lo que ves es lo que obtienes*). Estos procesadores se diseñaron inicialmente con el objetivo de que se pueda editar de una forma muy sencilla el contenido de documentos y obtener una visualización atractiva de forma simultánea a la escritura del contenido y la estructuración del documento. Así, los autores no tienen que andar preocupándose del código y de las etiquetas generadas, y pueden añadir formatos, alineaciones, tamaños o colores sin conocimiento alguno sobre el lenguaje de marcado subyacente.

Los inconvenientes de este tipo de procesadores se pueden resumir en el hecho de que este tipo de procesadores distraen a los autores de sus tareas de investigación y composición ya que deben ocuparse de las tareas relacionadas con la tipografía y con el aspecto del texto *simultáneamente* a la escritura de este. En general, la preocupación excesiva por los aspectos visuales de los procesadores WYSIWYG puede hacer que los autores ignoren o se distraigan de realizar una buena representación estructurada del documento, lo cual suele ser más significativo para el resultado final que los aspectos visuales o de formato que, al fin y al cabo, son más sencillos de modificar en cualquier momento de la edición.

Sin embargo, el uso exclusivo de lenguajes de marcado excesivamente complejos o pocos flexibles podría llegar a producir un efecto similar al mencionado para los procesadores WYSIWYG de alto nivel: puede hacer que los autores se preocupen excesivamente de las tareas de marcado. Esta es una de las motivaciones principales del proyecto que presentamos en el artículo: se define una versión simplificada de TEI, TEI down, con el objetivo principal de ahorrar tiempo y trabajo en las labores de producción del texto marcado.

2.1. Escritura de documentos TEI-XML a partir de la salida de un proceso OCR

Este método consiste en obtener un texto *plano* a partir de un procedimiento de reconocimiento óptico de caracteres. En general, sobre este texto plano que se obtiene de un OCR debe realizarse una tarea de corrección de los errores producidos por el propio sistema de reconocimiento. Aunque la cantidad de errores a corregir suele depender de la calidad del proceso de OCR y del tipo de texto que se ha procesado, la corrección de este texto suele ser una tarea tediosa.

Además, en este método, el marcado suele hacerse de forma simultánea a la corrección del propio texto. La complejidad de la tarea de marcado y de corrección simultánea es un procedimiento susceptible de cometer errores, tanto en el marcado como en la corrección del texto reconocido. La utilización de un lenguaje como el propuesto en nuestro proyecto (TEI down) puede hacer disminuir la complejidad de la tarea del marcado, haciendo más sencilla la corrección y marcado simultáneo del texto reconocido con OCR.

2.2. Escritura de documentos TEI-XML a partir de textos escritos en HTML u otros lenguajes de marcado

Otra posibilidad de escritura de documentos TEI-XML consiste en partir de documentos HTML u otros textos que estén marcados. Este método es especialmente útil cuando el HTML del que se parte es un código *limpio*, es decir, sin estilos insertados u otros elementos que pueden añadir algunos procesadores de texto.

De esta forma, se puede aprovechar el marcado HTML del texto y reusar en el documento TEI algunas de las etiquetas que aparezcan en el documento HTML original. Algunos ejemplos de estas etiquetas aprovechables son las etiquetas correspondientes a párrafos (<p>) o las correspondientes a cabeceras (<h1> a <h6>). Otro tipo de documentos que pueden usarse en este método son las publicaciones electrónicas (como el formato ePub), que también contiene etiquetas HTML o XHTML.

MARKUP LANGUAGES (in black)

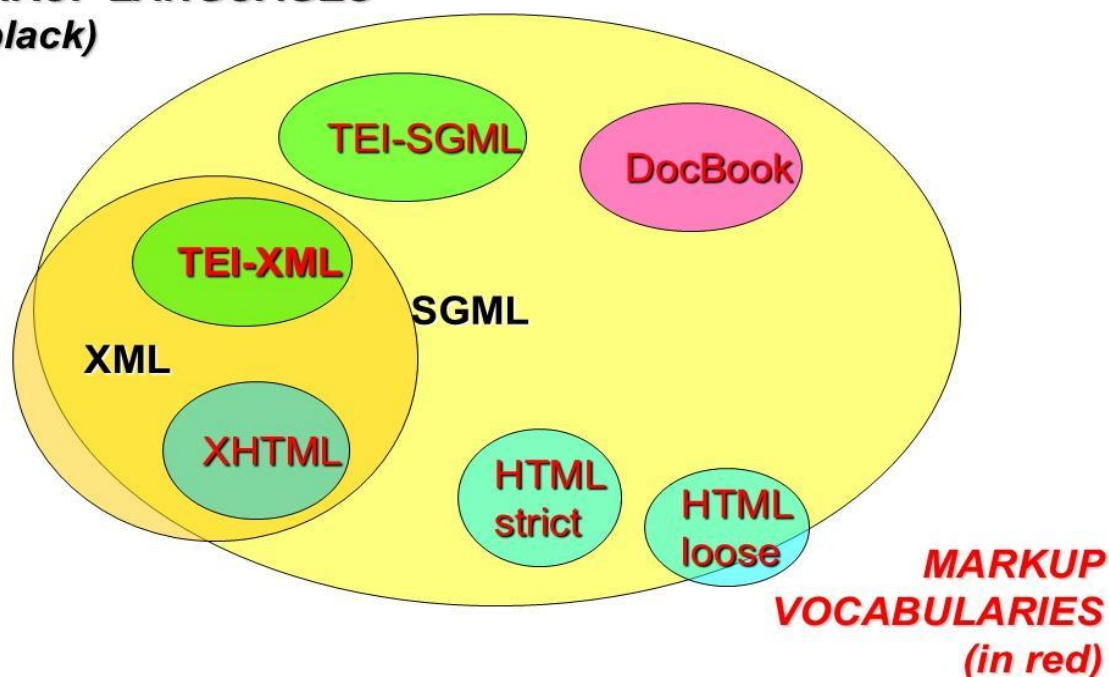


Figura 1. Lenguajes y vocabularios de marcado.

2.3. Escritura de documentos TEI-XML a partir de textos escritos en MS-Word u otros formatos de ofimática

Un documento escrito con un procesador de textos como MS-Word puede servir como base para la escritura de documentos TEI. Existen algunas opciones o herramientas incluidas en estos procesadores que pueden mejorar la productividad del marcado TEI y ahorrar tiempo. Así, por ejemplo, puede aprovecharse la herramienta de creación de macros disponible en MS-Word para encontrar y etiquetar párrafos o cabeceras con etiquetas TEI. Además, algunas características de los documentos generados con este tipo de procesadores pueden ser analizadas automáticamente con analizadores sintácticos y semánticos, para su posterior traducción o conversión automática al marcado TEI. Sin embargo, esta tarea requiere el diseño de *parsers* o traductores sintáctico-semánticos, diseño que no siempre puede ser sencillo debido a la complejidad del TEI completo.

2.4. Escritura de documentos TEI-XML a partir de documentos escritos en LaTeX

LaTeX es un sistema de composición de textos que, por sus características y posibilidades, es usado especialmente para la generación de artículos y libros científicos que incluyen expresiones matemáticas. Una de las características de LaTeX que puede ser útil para su traducción a TEI es el hecho de que los párrafos deben estar separados por dos caracteres de línea nueva, de forma similar a como se hace en Markdown. Otra característica

útil de LaTeX es la existencia de elementos de su estructura tales como las cabeceras, que pueden ser analizados y traducidos a etiquetas XML-TEI.

Sin embargo, LaTeX no está basado en XML por lo que no pueden aplicarse procesamientos XSLT. Esta característica es un inconveniente ya que implica que deben usarse analizadores sintácticos y semánticos más tradicionales. A pesar de este inconveniente, LaTeX sí puede resultar de interés para la comunidad de humanidades digitales en general ya que, por ejemplo, la notación matemática de LaTeX puede ser traducida a otros formatos como MathML.

$$\iint_{\Delta S} (\nabla \times \mathbf{A}) \cdot \mathbf{n} \, dS \quad \iiint_V \Psi \nabla^2 \phi \, dV \quad \int \dots \int f(x_1, \dots, x_k) \, dx_1 \dots dx_k$$

$$a_1 + a_2 + \dots + a_n = \sum_{i=1}^n a_i \quad L_1(x) \leq \dots \leq f(x) \dots \leq U_2(x) \leq U_1(x) \quad B \xrightarrow[H^+]{130^\circ C} C$$

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \quad \underline{e}_i = \frac{\partial x^k}{\partial y^i} \underline{E}_k \quad H(j\omega) = \begin{cases} e^{-j\omega t_0} & \text{for } |\omega| < \omega_c \\ 0 & \text{for } |\omega| > \omega_c \end{cases}$$

$$\left\langle \frac{\mathbf{p}^2}{m} \right\rangle = (\mathbf{r} \cdot \nabla V(r)) \quad \left| \frac{f(x+\delta_m) - f(x)}{\delta_m} \right| \geq \zeta \quad 1 = \sum_n |\langle u_n | \langle u_n | \quad \lim_{n \rightarrow \infty} p_n$$

$$\limsup_{n \rightarrow \infty} \sqrt[n]{c_n} \leq \alpha \quad \begin{pmatrix} a_{11} - \lambda & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} - \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \mathbf{0} \quad \overline{(\Delta n)^2} \equiv \overline{(n - \bar{n})^2}$$

$$\vec{AB} + \vec{BC} = \vec{AC} \quad (\mathbf{A} \times \mathbf{B})_\alpha = \varepsilon_{\alpha\beta\gamma} A_\beta B_\gamma$$

Figura 2. Notación matemática expresable en LaTeX o MathML.

2.5. Escritura de documentos TEI-XML a partir de un documento en blanco o usando una plantilla vacía

El método de escritura a partir de un documento en blanco o plano es el método tradicional: simplemente escribir y etiquetar. Debe tenerse en cuenta que en este proceso el tiempo empleado en el etiquetado o marcado es muy pequeño comparado con el tiempo empleado en la escritura. Sin embargo, una buena parte de la atención y del trabajo estará dedicado a este etiquetado, por lo que no debemos olvidar su importancia para la evaluación de la productividad del proceso.

Una forma de mejorar la productividad puede ser la creación de plantillas TEI vacías y básicas (construcción de un *esqueleto* TEI mínimo), que debería contener todos los elementos y atributos obligatorios, pero nada más. Este fichero TEI mínimo puede ser usado posteriormente como una plantilla para crear documentos TEI-XML nuevos.

3. PROPUESTA: UNA SOLUCIÓN DISTINTA

El estándar SGML fue diseñado para permitir el intercambio de información entre distintas plataformas, soportes físicos, lógicos, y diferentes sistemas de almacenamiento y presentación (bases de datos, edición electrónica, etc.), con independencia de su grado de complejidad. Se trata de un estándar internacional que proporciona un método para la descripción de la estructura de documentos basándose en la relación lógica de sus partes.

Este estándar SGML está en el origen del lenguaje de marcas XML (eXtensible Markup Language, “lenguaje de marcas Extensible”), que puede considerarse una versión “abreviada” de SGML que permite la definición de distintos tipos de documentos específicos, así como la creación de aplicaciones que procesen estos documentos. La característica básica en la creación de XML fue la omisión de la “complejidad” de SGML, tratando de no perder su potencia o capacidad de representación.

Una de las virtudes del uso del estándar SGML es la libertad a la hora de escribir documentos ya que es un estándar que no especifica demasiadas restricciones. Sin embargo, esta ventaja también pudo convertirse al mismo tiempo en su principal inconveniente: el procesamiento de los documentos se hace más complejo debido precisamente a las pocas restricciones que se introducen. El lenguaje de marcas XML sí introdujo más restricciones que SGML (de hecho, XML puede considerarse un subconjunto de SGML), lo cual hace que el procesamiento de los documentos XML no sea tan complejo al ser más predecibles. Esto hizo que los documentos XML sean mucho más fáciles de usar y de procesar automáticamente.

Sin embargo, debido también a su flexibilidad, SGML es más *amigable* para los codificadores o etiquetadores de documentos. Una aproximación distinta para tratar de conseguir simultáneamente la facilidad de uso por parte de los codificadores que proporciona SGML junto con la facilidad de procesamiento automático que proporciona XML, consiste en el diseño y uso de los denominados lenguajes de marcas ligeros (LML, *Lightweight Markup Languages*). Estos lenguajes, también denominados lenguajes de marcado simples o *humanos*, se pueden definir como:

Lenguajes de marcado con una sintaxis simple y no intrusiva. Están diseñados para que sean fáciles de crear usando cualquier editor de textos genéricos, así como para ser fáciles de leer en su forma de texto plano. Los LML se usan en aplicaciones en las

que puede ser necesario leer el documento plano, así como la visualización final de la salida procesada¹.

Como en muchos otros campos de las aplicaciones informáticas, existen numerosos ejemplos de LML. Algunos de estos ejemplos son los siguientes: AsciiDoc, AFT, BBCode, Creole, Crossmark, Deplate, Epytext, EtText, Haml, JsonML, MakeDoc, Markdown, Org-mode, POD, reST, RD, Setext, SiSU, SPIP, Xupl, Taxy!, Textile, txt2tags, UDO y Wikitext. En la Figura 3 se muestran las características que ofrecen algunos de estos lenguajes.

Language	HTML export tool	HTML import tool	Tables	Link titles	class attribute	id attribute
AsciiDoc	Yes	Yes	Yes	Yes	No	No
AFT	Yes	No	Yes	Yes	No	No
BBCode	No	No	Yes	No	No	No
Creole	No	No	Yes	No	No	No
deplate	Yes	No	Yes	No	Yes	Yes
GitHub Flavored Markdown	Yes	No	Yes	Yes	No	No
Jemdoc ^[1]	Yes	No	Yes	Yes	No	No
KARAS	Yes	No	Yes	Yes	Yes/No	Yes/No
Markdown	Yes	Yes	Yes/No	Yes	Yes/No	Yes/No
Markdown Extra	Yes	Yes	Yes ^[2]	Yes	Yes	Yes
MediaWiki	Yes	Yes	Yes	Yes	Yes	Yes
MultiMarkdown	Yes	No	Yes	Yes	No	No
Org-mode	Yes	Yes ^[3]	Yes	Yes	Yes	Yes
PmWiki	No	Yes	Yes	Yes	Yes	Yes

Figura 3. Características de los Lenguajes de Marcado Ligeros (LML)

¹ Definición tomada de Wikipedia el 25/05/2016.

Language ⇄	Implementations ⇄	XHTML ⇄	Con/LaTeX ⇄	PDF ⇄	DocBook ⇄	ODF ⇄	EPUB ⇄	DOC(X) ⇄
AsciiDoc	Python, Ruby, JavaScript	XHTML	LaTeX	PDF	DocBook	ODF	EPUB	No
AFT	Perl	HTML	LaTeX	No	No	No	No	RTF
BBCode	Perl, PHP, C#, Python, Ruby	(X)HTML	No	No	No	No	No	No
Creole	PHP, Python, Ruby, JavaScript [7]	Depends on implementation						
deplate	Ruby	HTML	LaTeX	No	DocBook	No	No	No
GitHub Flavored Markdown	Haskell (Pandoc)	HTML	LaTeX, ConTeXt	PDF	DocBook	ODF	EPUB	DOC
	Java, ^[8] JavaScript, ^[9] ^[10] ^[11] PHP, ^[12] ^[13] Python, ^[14] Ruby ^[15]	HTML ^[9] ^[10] ^[11] ^[13] ^[14]	No	No	No	No	No	No
Jemdoc ^[1]	Python	XHTML 1.1	No	No	No	No	No	No
KARAS	PHP, C#, JavaScript, Ruby	(X)HTML5	No	No	No	No	No	No
Markdown	Perl (originally), C, ^[16] ^[17] Python, ^[18] JavaScript, Haskell, ^[3] Ruby, ^[19] C#, Java, PHP	HTML	LaTeX, ConTeXt	PDF	DocBook	ODF	EPUB	RTF
Markdown Extra	PHP (originally), Python, Ruby	XHTML	No	No	No	No	No	No
MediaWiki	Perl, PHP, Haskell	XHTML	No	No	No	No	No	No
MultiMarkdown	C, Perl	(X)HTML	LaTeX	PDF	No	ODF	No	DOC, RTF

Figura 4. Formatos de exportación desde LMLs.

Uno de los ejemplos de LML más usados es Markdown, que está diseñado de tal forma que pueda ser traducido automáticamente a código HTML y otros muchos formatos. La motivación principal del proyecto que presentamos en el presente artículo es la adaptación de la idea del lenguaje Markdown respecto a HTML: de la misma forma que Markdown facilita la codificación en HTML, pretendemos que la herramienta propuesta (TEIdown) facilite igualmente la codificación en TEI. En el epígrafe siguiente se describen algunas características relevantes de los lenguajes Markdown y AsciiDoc que sirvieron de inspiración para TEIdown.

4. MARKDOWN Y ASCIIDOC

4.1. Markdown

Markdown es un lenguaje de marcas ligero con sintaxis de formato de texto plano tal que puede ser convertido a HTML y a otros muchos formatos usando una herramienta con el mismo nombre². Markdown se usa frecuentemente para dar formato a ficheros *readme*, para escribir mensajes en foros de discusión online y para crear texto enriquecido usando un editor de texto plano.

El lenguaje Markdown fue creado por John Gruber en el año 2004 con el objetivo de permitir a los codificadores “escribir usando un formato de texto plano fácil de leer y fácil de escribir y, opcionalmente, convertirlo en un documento XHTML (o HTML) estructuralmente válido”. Una de las ideas más interesantes de Markdown es que fue diseñado para que los documentos fuesen fácilmente legibles sin necesidad de tener la apariencia de que han sido marcados con etiquetas o con instrucciones de formateo. En la Figura 5 se muestra un ejemplo de texto marcado usando Markdown, junto con su equivalente HTML y la visualización en un navegador web. En esta figura puede observarse que el texto marcado con Markdown puede pasar perfectamente por un texto *sin marcado* ya que estas marcas son fáciles de leer (y de escribir).

² La página web del proyecto Markdown es <https://daringfireball.net/projects/markdown/>.

text using Markdown syntax	the corresponding HTML produced by a Markdown processor	the text viewed in a browser
<pre>Heading ===== Sub-heading ----- ### Another deeper heading Paragraphs are separated by a blank line. Leave 2 spaces at the end of a line to do a line break Text attributes *italic*, **bold**, `monospace`, ~~strikethrough~~ . A [link] (http://example.com). [27] Shopping list: * apples * oranges * pears Numbered list: 1. apples 2. oranges 3. pears The rain---not the reign---in Spain.</pre>	<pre><h1>Heading</h1> <h2>Sub-heading</h2> <h3>Another deeper heading</h3> <p>Paragraphs are separated by a blank line.</p> <p>Leave 2 spaces at the end of a line to do a
 line break</p> <p>Text attributes italic, bold, <code>monospace</code>, <s>strikethrough</s>.</p> <p>A link. </p> <p>Shopping list:</p> apples oranges pears <p>Numbered list:</p> apples oranges pears <p>The rain&mdash;not the reign&mdash;in Spain.</p></pre>	<p>Heading</p> <hr/> <p>Sub-heading</p> <p>Another deeper heading</p> <p>Paragraphs are separated by a blank line.</p> <p>Leave 2 spaces at the end of a line to do a line break</p> <p>Text attributes <i>italic</i>, bold, <code>monospace</code>, strikethrough.</p> <p>A link ↗.</p> <p>Shopping list:</p> <ul style="list-style-type: none"> • apples • oranges • pears <p>Numbered list:</p> <ol style="list-style-type: none"> 1. apples 2. oranges 3. pears <p>The rain—not the reign—in Spain.</p>

Figura 5. Ejemplo de texto marcado usando Markdown, junto con su equivalente HTML y la visualización en un navegador web.

Algunos sitios web como GitHub, Reddit, Diaspora, Stack Exchange, OpenStreetMap o SourceForge usan variantes de Markdown para facilitar las discusiones entre los usuarios. Aunque Gruber implementó originalmente Markdown usando el lenguaje Perl, actualmente existen numerosas implementaciones en otros lenguajes de programación. Para la realización de estas implementaciones no existe realmente un estándar claramente definido, aparte de la implementación original de Gruber. En la actualidad se está tratando de obtener un estándar por parte de la IETF (*Internet Engineering Task Force*).

4.2. AsciiDoc

AsciiDoc es un formato de documentos fácilmente legible, equivalente semánticamente a DocBook XML, que usa formatos de marcado de texto plano. Los documentos en AsciiDoc pueden crearse usando cualquier editor de textos y pueden ser leídos fácilmente, o ser convertidos o procesados al lenguaje HTML o a cualquier otro formato soportado por DocBook, tales como PDF, TeX, Ebooks, etc.

Document header

```

Main Header
=====
Optional Author Name <optional@author.email>
Optional version, optional date
:Author:      AlternativeWayToSetOptional Author Name
:Email:       <AlternativeWayToSetOptional@author.email>
:Date:        AlternativeWayToSetOptional date
:Revision:    AlternativeWayToSetOptional version

```

Figura 6. Notación del cabezal de metadatos de AsciiDoc que ha servido de inspiración para los metadatos de TEI down.

5. TRADUCCIÓN DE MARKDOWN A TEI

Partiendo de la idea básica de algunos lenguajes de marcas ligeros como los mencionados en el epígrafe anterior (Markdown y AsciiDoc), en el proyecto TEI down se trata de generar codificaciones TEI a partir de una sintaxis muy simplificada del mismo. Esta sintaxis ha sido cuidadosamente diseñada con el objetivo de que sea fácil de leer y de corregir y, sobre todo, fácil de codificar.

Para comprobar la viabilidad de la idea, se ha desarrollado un prototipo de analizador sintáctico y traductor para TEI down. En la figura 7 se muestra un esquema del proceso que se sigue en este prototipo.

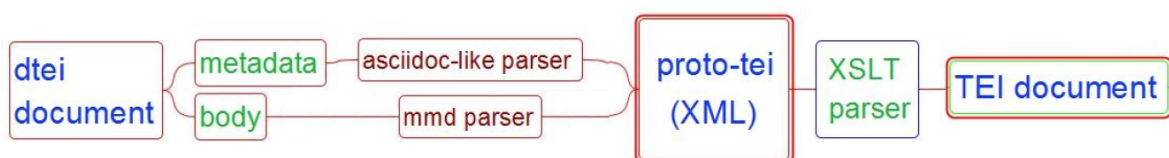


Figura 7. Diagrama de procesamiento del prototipo de TEI down.

En el prototipo se genera un documento final codificado en TEI a partir de un fichero con formato Multimarkdown extendido (mmd), con extensión *.dtei*. Multimarkdown es un lenguaje de marcado ligero que está basado en Markdown, que soporta exportación a formatos e implementa algunas características adicionales que no están disponibles en la sintaxis de Markdown.

En el prototipo se realiza una separación del texto en dos partes diferenciadas: los metadatos por un lado y el cuerpo del texto por otro, para realizar un procesamiento por separado de cada parte. Los metadatos son procesados por un analizador de AsciiDoc, mientras que el cuerpo del texto es procesado por un analizador mmd o Multimarkdown.

A partir del resultado de estos procesamientos se obtiene un fichero intermedio que todavía no está codificado en TEI, sino en *proto-TEI*. Este fichero intermedio es transformado por medio de un analizador XSLT para obtener finalmente la codificación del fichero original en TEI.

A continuación, se muestran algunos ejemplos de codificaciones usando el prototipo diseñado. En las figuras 8 a 14 se muestran algunos ejemplos de entrada de texto en formato TEIdown y la salida obtenida en formato TEI, para metadatos, prosa, poesía y teatro.

```
1
2 :title: Carta abierta
3 :author: Guiraldes, Ricardo
4 :publisher: Biblioteca Virtual Miguel de Cervantes Saavedra
5 :pubPlace: Universidad de Alicante
6 :idno: 004229
7 :availability: Copyright (c) Universidad de Alicante, Banco Santander Central Hispano. Accesible
  desde http://cervantesvirtual.com
8 :sourceDesc: Martín Fierro, segunda época, año II, núm. 14 y 15, Buenos Aires, enero 24 de 1925.
  Obra cedida por la Biblioteca de la Academia Argentina de las Letras. Digitalización realizada por
  Verónica Zumárraga.
9
10 :date: 01/12/2008
11 :name: Lorenzo García Pérez
12 :resp: Supervisor
13 :item: Supervisión del texto
```

Figura 8. Ejemplo de entrada de metadatos.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet href="teixlite2015.css" type="text/css"?>
3 <!DOCTYPE TEI SYSTEM "cervantes2015.dtd">
4 <TEI>
5 <teiHeader>
6 <fileDesc>
7 <titleStmt>
8 <title type="main">Carta abierta</title>
9 <author>Giraldes, Ricardo </author>
10 </titleStmt>
11 <publicationStmt>
12 <publisher>Biblioteca Virtual Miguel de Cervantes Saavedra</publisher>
13 <pubPlace>Universidad de Alicante</pubPlace>
14 <idno>004229</idno>
15 <availability status="free">
16 <p>Copyright (c) Universidad de Alicante, Banco Santander Central Hispano. Accesible
17 desde http://cervantesvirtual.com</p>
18 </availability>
19 <date day="21" month="11" year="2000"/>
20 </publicationStmt>
21 <sourceDesc>
22 <p>Martín Fierro, segunda época, año II, núm. 14 y 15, Buenos Aires, enero 24 de 1925.
23 Obra cedida por la Biblioteca de la Academia Argentina de las Letras. Digitalización realizada
24 por Verónica Zumárraga.</p>
25 </sourceDesc>
26 </fileDesc>
27 <profileDesc>
28 <langUsage>
29 <language ident="es">Español</language>
30 </langUsage>
31 </profileDesc>
32 <revisionDesc>
33 <change>
34 <date day="01" month="12" year="2008"/>
35 <respStmt>
36 <name>Lorenzo García Pérez</name>
37 <resp>Supervisor</resp>
38 </respStmt>
39 <item>Supervisión del texto </item>
40 </change>

```

Figura 9. Ejemplo de salida de metadatos (etiqueta *teiHeader* en TEI), generados con la entrada mostrada en la figura anterior.

```

20 # Carta abierta #
21 docAuthor: Ricardo Güiraldes
22
23 Amigos:
24
25 He leído hoy el último número de MARTÍN FIERRO. Patria chica en el papel, grande en el anhelo.
26 Vuestra juventud sube hacia mi rostro, como un aliento de pampa, cuando sobre la gramilla iluminada
27 de rocío (emoción de la madrugada, que vuelve a encontrar su mundo) me aferro al optimismo
28 ascendente de los nuevos crecimientos. El hombre se siente pequeño ante la infinita transmutación
29 que anuncia lo porvenir, pero crece con sentirse capaz de comprenderla.
30
31 En vuestra valentía se produce una vez más el eterno amanecer del espíritu. ¿No es ése el misterio
32 de la anunciación?
33
34 Vienen y vendrán los ataques. Inútil sorprenderse. Caminos sin pantanos no son caminos de hombres
35 libres y los más duros son los que más espolean el deseo de llegar. Caer no es nada. Las osamentas
36 sirven de mojón a los que después de uno sienten el vértigo del desierto. Así se conquistan
37 horizontes. Así se regala el bien habido a los timoratos.
38
39 Uno de los nuestros ha pedido piedad. Y es que dio el rostro a lo que siempre debió volver la
40 espalda. ¿Cómo la cobardía momentánea del fuerte puede pedir ayuda a la cobardía constante de los
41 débiles? Dice un refrán gaucho: "No hay que mudar caballo en medio 'el río'".
42
43 En el camino de las ideas la duda equivale a mudar de caballo.
44
45 Además, ¿qué puede esperar el que cargó sobre sus hombros con la responsabilidad de partir, de
46 aquellos que se aferraron a la inmovilidad? Solamente un reproche, una acusación de soberbia, de

```

Figura 10. Ejemplo de entrada de texto en prosa.

```

65 <text lang="es">
66 <front>
67 <titlePage>
68 <docTitle>
69 <titlePart>
70 <title type="main">Carta abierta</title>
71 </titlePart>
72 </docTitle>
73 <docAuthor>Ricardo Güiraldes</docAuthor>
74 </titlePage>
75 </front>
76 <body>
77 <div>
78 <p>Amigos:</p>
79 <p>He leído hoy el último
80 número de MARTÍN FIERRO. Patria chica en el papel, grande en el
81 anhelo. Vuestra juventud sube hacia mi rostro, como un aliento de pampa, cuando
82 sobre la gramilla iluminada de rocío (emoción de la madrugada,
83 que vuelve a encontrar su mundo) me aferro al optimismo ascendente de los
84 nuevos crecimientos. El hombre se siente pequeño ante la infinita
85 transmutación que anuncia lo porvenir, pero crece con sentirse capaz de
86 comprenderla.</p>
87 <p>En vuestra valentía se produce una
88 vez más el eterno amanecer del espíritu. ¿No es ése
89 el misterio de la anunciación?</p>
90 <p>Vienen y vendrán los ataques.
91 Inútil sorprenderse. Caminos sin pantanos no son caminos de hombres

```

Figura 11. Ejemplo de salida de texto en prosa (etiqueta *text* en TEI), generada con la entrada mostrada en la figura anterior.

```

1 # How Do I Love Thee
2 ## by Elizabeth Barrett Browning
3
4 :lg
5 How do I love thee? Let me count the ways.
6 I love thee to the depth and breadth and height
7 My soul can reach, when feeling out of sight
8 For the ends of Being and ideal Grace.
9 I love thee to the level of everyday's
10 Most quiet need, by sun and candle-light.
11 I love thee freely, as men strive for Right;
12 I love thee purely, as they turn from Praise.
13 I love thee with a passion put to use
14 In my old griefs, and with my childhood's faith.
15 I love thee with a love I seemed to lose
16 With my lost saints, --- I love thee with the breath,
17 Smiles, tears, of all my life! --- and, if God choose,
18 I shall but love thee better after death.
19

```

Figura 12. Ejemplo de entrada de poesía (lg: line group).

```

21 <text>
22   <body>
23     <head type="main">How Do I Love Thee</head>
24     <head type="sub">Elizabeth Barrett Browning</head>
25     <lg type="poem">
26       <l>How do I love thee? Let me count the ways</l>
27       <l>I love thee to the depth and breadth and height</l>
28       <l>My soul can reach, when feeling out of sight</l>
29       <l>For the ends of Being and ideal Grace.</l>
30       <l>I love thee to the level of everyday's</l>
31       <l>Most quiet need, by sun and candle-light.</l>
32       <l>I love thee freely, as men strive for Right;</l>
33       <l>I love thee purely, as they turn from Praise.</l>
34       <l>I love thee with a passion put to use</l>
35       <l>In my old griefs, and with my childhood's faith.</l>
36       <l>I love thee with a love I seemed to lose</l>
37       <l>With my lost saints, --- I love thee with the breath,</l>
38       <l>Smiles, tears, of all my life! --- and, if God choose,</l>
39       <l>I shall but love thee better after death.</l>
40     </lg>
41   </body>
42 </text>
43 </TEI>

```

Figura 13. Ejemplo de salida de poesía marcada en TEI (etiquetas *lg* y *l*).

```

:castList
PERSONAS | | ACTORES
EL BARÓN DE MONSERNIN | | SR. JOSÉ GARCÍA LUNA
DERVAL | su amigo, propietario | SR. RAMÓN LÓPEZ
EMILIA, | hermana del barón | SRA. CATALINA BRAVO

<castList>
<head index="comment" type="roles">PERSONAS</head>
<head type="actors">ACTORES</head>
<castItem>
<role>EL BARÓN DE MONSERNIN</role>
<actor>SR. JOSÉ GARCÍA LUNA</actor>
</castItem>
<castItem>
<role>DERVAL, </role>
<roleDesc>su amigo, propietario</roleDesc>
<actor>SR. RAMÓN LÓPEZ</actor>
</castItem>
<castItem>
<role>EMILIA, </role>
<roleDesc>hermana del barón</roleDesc>
<actor>SRA. CATALINA BRAVO</actor>
</castItem>
</castList>

```

Figura 14. Ejemplo de entrada (izquierda) y salida (derecha) de una lista de personajes (etiqueta *castList*).

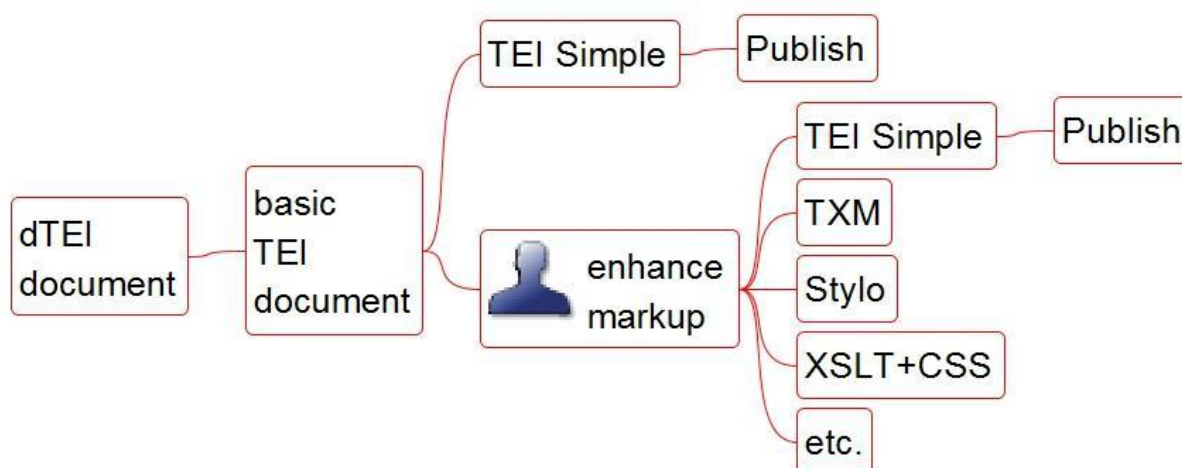


Figura 15. Implementación futura de TEI Down, cuya salida se puede post-procesar con otras herramientas existentes.

6. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo permitió demostrar que se puede utilizar una notación de marcado ligero para realizar marcados complejos como los realizados con TEI. Este tipo de marcado sirve para ahorrar mucho tiempo y esfuerzo a la hora de crear colecciones de documentos TEI. El marcado obtenido es TEI válido.

Este método sirve para realizar un marcado sencillo con elementos de prosa, poesía y algo de teatro, pero no está pensado para casos más especializados o complejos, como por ejemplo el marcado de manuscritos, ediciones críticas o análisis genético. Este marcado

especializado siempre se puede agregar a posteriori, una vez convertido el texto con marcado ligero a TEI. Incluso en estos casos, se ahorraría bastante tiempo en el marcado básico inicial.

Las pruebas presentadas aquí se realizaron con un prototipo *quick-and-dirty*, a modo de prueba de concepto. Como trabajo futuro, tenemos planeado hacer una implementación completa y robusta de este método y hacerla disponible como servicio web.

RECONOCIMIENTOS

Este trabajo ha sido desarrollado dentro del proyecto TRACESofTools (herramientas informáticas para análisis contrastivo de textos en corpus bilingües paralelos: alineación y marcado automático, y minería de datos semiestructurados y metadatos), y financiados con la ayuda FFI2012-39012-C04-02 del VI Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica del Ministerio de Economía y Competitividad de España.