

# Desarrollos en elaboración de documentación técnica: lenguajes de marcado ligero

## Developements in the preparation of technical documentation: Lightweight markup languages

### Jesús Tramullas

<https://orcid.org/0000-0002-5374-9993>

Universidad de Zaragoza  
Departamento de Ciencias de la Documentación  
C/ Pedro Cerbuna, 12. 50009 Zaragoza, España  
<http://tramullas.com>  
[tramullas@unizar.es](mailto:tramullas@unizar.es)

**Tramullas, Jesús** (2019). "Desarrollos en elaboración de documentación técnica: los lenguajes de marcado ligero". *Anuario ThinkEPI*, v. 13, e13f03.

<https://doi.org/10.3145/thinkepi.2019.e13f03>

Publicado en *IweTel* el 18 de febrero de 2019



**Resumen:** Los lenguajes de marcado ligero se han consolidado como la *lingua franca* para la elaboración de documentación técnica en los entornos digitales. Se revisan sus fundamentos e historia, así como los principales lenguajes y los programas utilizados para la generación de documentos.

**Palabras clave:** Lenguajes de marcado ligero; *Markdown*; *ReStructuredText*; Documentación técnica.

**Abstract:** Lightweight markup languages have been consolidated as the *lingua franca* for the preparation of technical documents. Their foundations and history are revised, as well as the main languages and documentation generators.

**Keywords:** Lightweight markup language; *Markdown*; *ReStructuredText*; Technical documentation.

## 1. Introducción

Las últimas versiones de *WordPress* han popularizado entre los editores de contenidos el uso de un lenguaje de marcado conocido como *Markdown*, al que se identifica como un "lenguaje de marcado ligero" (*lightweight markup language*). Y no es la única plataforma de este tipo que puede hacer uso de *Markdown*. Dado el volumen de sedes web que se desarrollan sobre este sistema de gestión de contenidos, tarde o temprano puede extenderse su utilización en la edición y publicación de contenidos en otras plataformas.

*Markdown* no es el único lenguaje de marcado ligero existente. El *CamelCase* que todavía puede usarse en wikis es otro ejemplo de ello. *BBCode* puede usarse en foros de opinión desde hace dos décadas, y fue especialmente diseñado a tal fin. Gran parte de la documentación que acompaña a diferentes herramientas de software, y que está disponible en internet, ha sido etiquetada y procesada utilizando *reStructuredText*. Cabe plantearse entonces si se está produciendo una multiplicación de lenguajes de marcado ligero, y el impacto que ello puede tener en la gestión de contenidos.

En primer lugar, es necesario distinguir entre:

- lenguajes de marcado o etiquetado: son aquellos que hacen uso de etiquetas estructuradas para identificar los elementos componentes de un documento. Derivados del ya venerable SGML, lenguajes como HTML, XML o TEI (y los múltiples derivados de éstos) son ejemplos de lenguajes de etiquetado. Los elementos están “marcados” en su inicio y en su final por etiquetas, las cuales adoptan como elementos definitorios los símbolos “<” y “>”.
- lenguajes de marcado ligero: no usan este tipo de etiquetado para identificar los elementos, en su lugar utilizan símbolos tipográficos o estructuras sencillas de símbolos y texto, como “:”, “#”, “\*”, “[ ]” y similares (Di-Castro, 2011).

Otra de las diferencias entre ellos es que, si bien los lenguajes de etiquetado hacen casi imposible la lectura comprensiva del contenido textual sobre el que se aplican, los lenguajes de marcado ligero pretenden que el contenido original siga siendo fácilmente comprensible, sin que las marcas usadas sean impedimento para ello.

---

**“Los lenguajes de marcado ligero utilizan símbolos tipográficos o estructuras sencillas de símbolos y texto, como “:”, “#”, “\*”, “[ ]” y similares”**

---

## 2. Una breve historia

Las leyendas de internet colocan el nacimiento de los lenguajes de marcado ligero en la creación de *Markdown* por parte de Gruber (2004) para su uso en *MovableType*, entonces un exitoso software para blogs. En su primera formulación, *Markdown* ya incorporaba los dos elementos que se han convertido en clásicos en los entornos de aplicación de los lenguajes de marcado ligero:

- un conjunto de elementos de marcado y su sintaxis;
- un procesador de software que convierte el texto marcado en documentos HTML o XHTML (o en otros formatos de salida).

*Markdown* adoptaba los principios establecidos previamente con su *html2text* por Swartz (2002). También en 2004 se publicaba *Textile*, en el ámbito del sistema de gestión de contenidos *TextPattern*. *Textile Markup Language* (*Textile Language Development Team*, 2003-2019). *AsciiDoc*, creado por Rakkham (2013), se presentó en 2002. Sin embargo, *reStructuredText* (en ocasiones abreviado como *reSt*) puede presentarse como anterior a ellos. Fruto del desarrollo y la reescritura de lenguajes previos como *StructuredText* y *StructuredTextNG*, procedentes del entorno del gestor de contenidos *Zope*, *reStructuredText* se fusionó con el proyecto del procesador *Docstring Processing System*, y ambos pasaron a formar *Docutils*, un generador de documentación textual programado en *Python*, presentado en 2002 (Goodger, 2012).

Las especificaciones de estos lenguajes se publicaron bajo licencias de software libre, lo que ha hecho posible su desarrollo abierto y evolución a lo largo de casi dos décadas. Los lenguajes de programación más extendidos para el desarrollo web, como *Perl*, *PHP*, *Python* o *Ruby*, han creado y publicado librerías y procesadores que hacen posible la generación de documentos en diferentes formatos (HTML, EPUB, PDF, etc.), tomando como base textos sobre los que se aplican lenguajes de marcado ligero. Estas funciones también se han introducido en lenguajes tradicionales como *Haskell*, *C*, *Java*... No sólo las posibilidades se han multiplicado, además, han aparecido procesadores más elaborados que hacen uso de librerías más avanzadas y de hojas de estilo para generar documentación técnica, acortando los ciclos de edición, publicación, revisión y actualización.

El potencial del desarrollo abierto para estas aplicaciones ha hecho posible la apa-

Basic Syntax	
These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.	
Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	**bold text**
Italic	*italicized text*
Blockquote	> blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

Figura 1. Sintaxis básica de *Markdown*

rión de una segunda generación de aplicaciones a las que se ha denominado “generadores de documentación”.

En 2006 John MacFarlane, de *Berkeley*, publica *Pandoc* (Mullen, 2012), una herramienta capaz de procesar y transformar ficheros de y en diferentes formatos, incluyendo los lenguajes de marcado ligero más comunes, especialmente orientada a la edición y publicación automática (Phillips, 2017).

En 2008 se lanzó la primera versión de *Sphinx* (Brandl, 2007-2019), una aplicación sobre *Python*, a la que se denominó “generador de documentación”, y que ampliaba el uso de *reStructuredText* y de *docutils* para convertirse en la base de documentación técnica de *Python* y del *kernel* de Linux, por ejemplo.

En 2014 apareció *MkDocs* (Christie, 2014), también programado en *Python*, que usa *Markdown* como lenguaje de marcado ligero, y que está especialmente orientado a la generación de documentación para publicar en sitios web estáticos, como la que se ofrece en *Read the docs* (Holscher, 2016), que también trabaja con *reStructuredText*.

### 3. El flujo de trabajo de los lenguajes de marcado ligero

El flujo de trabajo básico para elaborar documentación usando un lenguaje de marcado ligero es sencillo:

- el proceso de edición sólo requiere de un editor de texto, aunque hay editores ya preparados para el trabajo con lenguajes de marcado ligero;
- todo documento en el cual haya sido utilizado un lenguaje de marcas necesita imperativamente que se utilice una herramienta de software capaz de leer el texto y las marcas, interpretarlo y procesarlo de acuerdo a unas reglas. Los principales generadores de documentación utilizados son *Sphinx* y *MkDocs*;
- a los procesadores o generadores se le provee con el fichero (o ficheros) de texto marcado, el formato de salida, y la hoja de estilo o plantilla a utilizar;
- los procesadores generan el documento o documentos resultantes, informando con mayor o menor detalle, según se configure, de los resultados del proceso y de los posibles errores;
- el documento resultante está preparado para su publicación y distribución en internet, bien como un sitio web estático, bien en otro formato legible por máquina.

### 4. La variedad de los lenguajes de marcado ligero

La especificación y publicación de los lenguajes de marcado ligero bajo licencias de software libre ha favorecido la aparición de variantes de los lenguajes de marcado ligero. Si se exceptúa *reStructuredText*, que mantiene una especificación común, el resto se ha visto enriquecido con el aporte de versiones adecuadas a entornos específicos. Las más numerosas son las variantes derivadas de *Markdown*:

- el paquete estadístico *R* usa su propia versión, *R Markdown*;
- *GitHub* recomienda *GitHub Flavored Markdown*;
- hay una propuesta de estándar, *CommonMark*;

**“Las especificaciones de estos lenguajes se publicaron bajo licencias de software libre, lo que ha hecho posible su desarrollo abierto y evolución a lo largo de casi dos décadas”**

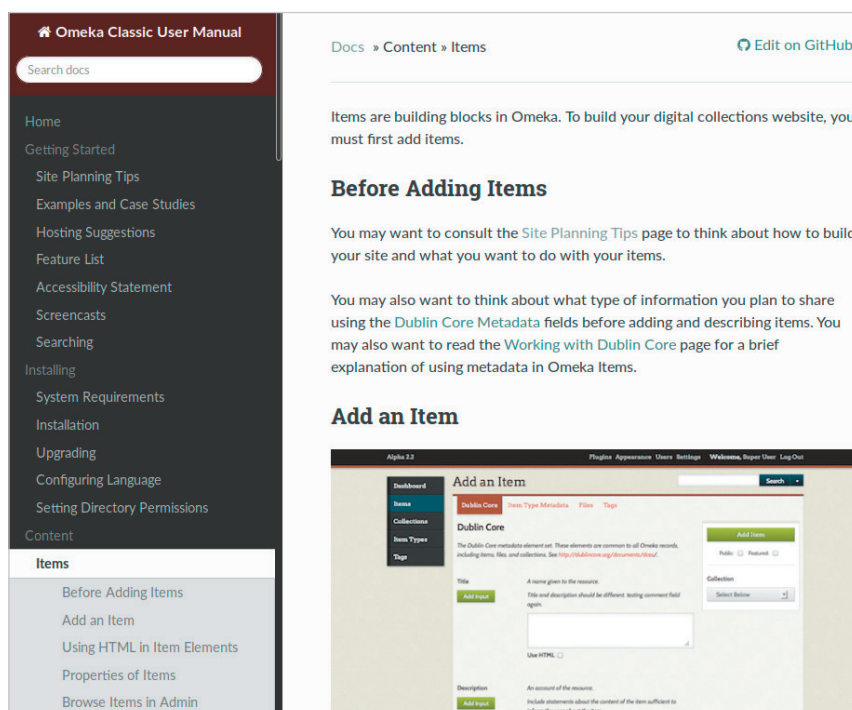


Fig. 2. El manual de Omeka ha sido generado con MkDocs.

- *Ghost's Markdown* es la propuesta del gestor de blogs del mismo nombre;
- los gestores de tareas y grupos *Taiga* y *Trello* tienen sus "gustos" de *Markdown*;
- hay un *Markdown 2*, un *Multi-MarkDown* y un *Markdown Extra*;
- hasta se ha propuesto un *ScholarlyMarkdown* orientado a la edición y publicación de textos académicos y científicos.

A todo lo anterior cabría añadir la capacidad de combinar estos lenguajes de marcado ligero con otros lenguajes de etiquetado, de manera que sea posible generar etiquetados más complejos partiendo precisamente del marcado ligero, como se ha propuesto, por ejemplo, con TEI (**Bias-Plata; Neco-García, 2017**).

Las variantes recogidas en el párrafo anterior, sin ser exhaustivas, sirven como muestra de la capacidad de los lenguajes de marcado ligero para adaptarse a múltiples contextos y situaciones. Las extensiones al mismo hacen posible ampliar sus prestaciones y hacer frente a nuevas necesidades, mediante el cambio y actualización de librerías y procesadores. Al no tratarse tampoco de estándares sujetos a aprobación por organismos, sino por comunidades especializadas, la velocidad de respuesta a nuevos requerimientos es elevada. La simplicidad, la variedad de herramientas y la capacidad de adaptación le dan una notable ventaja competitiva frente a lenguajes de etiquetado dependientes de procesos de estandarización. La creciente expansión de la edición de contenidos en *Markdown* es prueba de ello.

---

**“Cabe plantear si, realmente, los lenguajes de marcado ligero van a tener un impacto en la gestión de información y de contenidos en entorno digital. La respuesta, indudablemente es que sí, y que ya lo están teniendo en varios campos”**

---

## 5. Cambios en la edición y publicación de documentos.

Cabe entonces plantear si, realmente, los lenguajes de marcado ligero van a tener un impacto en la gestión de información y de contenidos en entorno digital. La respuesta, indudablemente es que sí, y que ya lo están teniendo en varios campos.

En primer lugar, la simplicidad de las marcas, la facilidad de uso, y la ligereza del código que generan, superan a las aproximaciones basadas en HTML y editores *wysiwyg*. La creciente disponibilidad de plataformas que generan sitios web estáticos con un mínimo uso de recursos, con mayor velocidad de despliegue, con programación basada en *JavaScript*, y más baratos de crear y mantener, está favoreciendo que los sistemas de gestión de contenidos tradicionales sean sustituidos en entornos en los que la publicación masiva no sea una necesidad diaria.

Un segundo campo en el cual tienen un impacto decisivo es en la edición, publicación y mantenimiento de documentación técnica (**García-Tovar, 2016**). Las funciones disponibles para documentar el código fuente de las aplicaciones están basadas precisamente en la aproximación basada en etiquetado ligero (**Gentle, 2017**). Si bien es un ámbito que en el campo hispanoparlante todavía está dominado por especialistas de disciplinas técnicas (como si la gestión de la información y documentación no lo fuese hace ya tiempo), la capacidad y simplicidad de los lenguajes de marcado ligero y de sus procesadores para la elaboración y despliegue de manuales y guías en entornos digitales no tiene rival en los entornos de publicación clásicos. Basta ver el ejemplo de la generación integrada de documentación entre *GitHub* o *GitLab* y *Read the Docs* para comprender que los derroteros de la documentación para usuario final han cambiado. La documentación técnica tiene una destacada relación con la documentación de usuario final, entendiendo la misma como la reformulación de sus contenidos en manuales, guías o tutoriales, orientados a satisfacer los procesos de aprendizaje o la resolución de problemas (**Fouh et al., 2014**). La elaboración de material educativo también se puede beneficiar de esta aproximación (**Voegler; Borschein; Weber, 2014**). La reducción de los ciclos de producción, publicación y revisión de estos tipos de documentos en estos entornos resulta un factor clave de éxito.

Finalmente, un tercer campo que se ve afectado por la aplicación de los lenguajes de marcado ligero es la publicación académica (**Ovadia, 2014**). Es más sencillo y rápido usar marcado ligero y luego aplicar un procesador para obtener documentos en diferentes formatos, como demuestra *Pandoc* (**Krewinkel; Winkler, 2017**), que tener que utilizar plantillas propietarias de procesador de textos.

---

**“La capacidad y simplicidad de los lenguajes de marcado ligero y de sus procesadores para la elaboración y despliegue de manuales y guías en entornos digitales no tiene rival en los entornos de publicación clásicos”**

---

Crear, editar y publicar documentos en abierto, de cualquier tipo o sobre cualquier tema, es ahora más ágil que nunca: un único documento fuente sirve para generar productos en diferentes formatos, según demanda o necesidad. Tampoco puede obviarse la creciente importancia del código en gran número de campos científicos, código que hasta tiempo relativamente reciente no podía integrarse activamente en la publicación científica. Ahora, los generadores de documentación permiten integrar ese código, y ejecutarlo conjuntamente con el documento matriz, como demuestran los *Jupyter Notebooks* (Willems, 2019). Ciencia abierta, en publicación abierta, y que se beneficia de repositorios de datos y de documentos preparados para el trabajo colaborativo.

## 6. Referencias

- Bias-Plata, Alejandro; Ñeco-García, Ramón P.** (2017). "TEI down: Uso de Markdown extendido para el marcado automático de documentos TEI". *Revista de humanidades digitales*, n. 1, pp. 57-75.  
<http://revistas.uned.es/index.php/RHD/article/view/16683/16586>
- Brandl, George** (2007-2019). *Sphinx. Python documentation generator*.  
<http://www.sphinx-doc.org>
- Christie, Tom** (2014). *MkDocs. Project documentation with Markdown*.  
<https://www.mkdocs.org>
- Di-Castro, Joe** (2011). "Comparativa de lenguajes de marcado ligero". *Joe di Castro*, 2 abril.  
<https://joedicastro.com/comparativa-de-lenguajes-de-marcado-ligero.html>
- Fouh, Eric; Karavirta, Ville; Breakiron, Daniel A.; Hamouda, Sally; Hall, Simin; Naps, Thomas; Shaffer, Clifford A.** (2014). "Design and architecture of an interactive eTextbook – The OpenDSA system". *Science of computer programming*, v. 88, pp. 22-40.  
<https://doi.org/10.1016/J.SCICO.2013.11.040>
- García-Tovar, Javier** (2016). "Sphinx como herramienta para documentar proyectos técnicos". *Técnica industrial*, n. 315, pp. 48-52.  
<http://www.tecnicaindustrial.es/TIFrontalla-7789-sphinx-herramienta-documentar-proyectos-tecnicos.aspx>
- Gentle, Anne** (2017). *Docs like code*.  
<https://www.docslikecode.com>
- Goodger, David** (2012). *An introduction to reStructuredText*.  
<http://docutils.sourceforge.net/docs/ref/rst/introduction.html#history>
- Gruber, John** (2004). "Introducing Markdown". *Daring fireball*, 15 marzo.  
[https://daringfireball.net/2004/03/introducing\\_markdown](https://daringfireball.net/2004/03/introducing_markdown)
- Holscher, Eric** (2016). "An introduction to Sphinx and Read the Docs for Technical Writers". *Surfing in Kansas*, 1 julio.  
<http://www.ericholscher.com/blog/2016/jul/1/sphinx-and-rtd-for-writers/>
- Krewinkel, Albert; Winkler, Robert** (2017). "Formatting open science: Agilely creating multiple document formats for academic manuscripts with Pandoc Scholar". *PeerJ computer science*, n. 3, e112.  
<https://doi.org/10.7717/peerj-cs.112>
- Mullen, Lincoln** (2012). "Pandoc converts all your (text) documents". *The chronicle of higher education. Blogs: ProfHacker*, 23 febrero.  
<https://www.chronicle.com/blogs/profhacker/pandoc-converts-all-your-text-documents/38700>
- Ovadia, Steven** (2014). "Markdown for librarians and academics". *Behavioral & social sciences librarian*, v. 33, n. 2, pp. 120-124.  
<https://doi.org/10.1080/01639269.2014.904696>
- Phillips, Lee** (2017). "Technical writing with Pandoc and Panflute". *Linux journal*, n. 281, pp. 90-107.  
<https://lee-phillips.org/panflute-gnuplot>
- Rackham, Stuart** (2013). *AsciiDoc. Text based document generation*.  
<http://www.methods.co.nz/asciidoc/index.html>
- Swartz, Aaron** (2002-2011). *Html2text (the asciinator)*.  
<http://www.aaronsw.com/2002/html2text>
- Textile Language Development Team** (2003-2019). *Textile markup language*.  
<https://textile-lang.com>
- Voegler, Jens; Bornschein, Jens; Weber, Gerhard** (2014). "Markdown – A simple syntax for transcription of accessible study materials". *Computers helping people with special needs*. En: *ICCHP 2014*, pp. 545-548.  
[https://doi.org/10.1007/978-3-319-08596-8\\_85](https://doi.org/10.1007/978-3-319-08596-8_85)
- Willems, Karen** (2019). "Jupyter notebook tutorial: The definitive guide". *DataCamp*, 9 enero.  
<https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>