

2. Lenguajes de marcado

Introducción a HTML, XML, lenguajes de marcado ligero...

- [2.1. HTML](#)
- [2.2. XML](#)
- [2.3. Markdown](#)
- [2.4. Otros lenguajes de marcado](#)

2.1. HTML

Entre 1989 y 1990 Tim Berners-Lee, en el CERN, desarrolló un lenguaje de marcado, basado en el SGML, que le permitiese enlazar y visualizar, a través de una red de ordenadores, documentos de diferente tipo, con contenidos textuales y gráficos. Había nacido el **HTML (HyperText Markup Language)**. Cuando esa arquitectura de la información se puso en internet, se creó el world wide web. El XHTML surge en el año 2000, con el objetivo de que HTML cumpliera como lenguaje XML válido ([véase apartado 2.2](#)), y convertirlo en un lenguaje de marcas descriptivo, pero no tuvo éxito. En la actualidad **el HTML ya se encuentra en su versión 5**, y existe un consorcio internacional encargado de su gestión y desarrollo. Hay que tener en cuenta que en internet conviven paginas en diferentes versiones de HTML, y que todas son presentadas por los navegadores, pero que hay diferencias en el uso y significado de las etiquetas o marcas entre las diferentes versiones, y especial desde la aparición de HTML 5. **Esta última versión ha incorporado un buen número de etiquetas que se centran a la descripción de la estructura y semántica del documento.**

HTML es un lenguaje de marcado que usa un conjunto muy básico de marcas o etiquetas, un mínimo casi imprescindible, y **su objetivo principal (hasta el momento) es indicar a los navegadores web como presentar la información en un pantalla de visualización**. Una de sus características importantes, que ha ido siendo heredada en el resto de lenguajes de marcado aparecidos posteriormente, es su capacidad para crear **estructuras hipertextuales de información mediante el principio de asociación de ideas**, lo que lleva a cabo mediante marcado de enlaces, o url.

Un navegador no recibe una página web: recibe un documento textual con marcado HTML, que contiene el contenido informativo-documental, los enlaces de los ficheros que debe combinar (imágenes, sonido, video...) , y con todo ello y las reglas del HTML genera una visualización en pantalla. El navegador procesa todo el conjunto, según las reglas del HTML, y ofrece al usuario el resultado final. *Para comparar ambas capas, basta con colocar el cursor sobre cualquier lugar de la página visualizada, botón derecho, menú emergente, opción "ver código fuente" o similar.*



CC BY-NC-ND



COMPARTIR

DESCARGAR

Arcos decorativos del mihrab de la mezquita de la Aljafería, trasladados al Museo de Zaragoza y actualmente reubicados en el Palacio (Zaragoza). Hacia 1920

Dos Arcos Islámicos. Uno polilobulado y otro de herradura. Aljafería.

[Leer más](#)



Fig. 2.

Visualización de una [página web de Europeana](#). Puede usar el menú emergente para ver el código fuente de la página.

Un documento en HTML contiene elementos. Estos elementos están formados por las etiquetas o marcas que identifican a una entidad, y por esa entidad. **Las etiquetas o marcas pueden mostrar, en ocasiones, atributos y variables.** Una característica clásica de los lenguajes de marcado derivados del SGML, como el HTML, es que **cada elemento tiene etiqueta de apertura y etiqueta de cierre.**



Fig. x.

Estructura de un elemento HTML ([fuente original](#)).

La creación de documentos en HTML es muy simple, dado que la sencillez del formato (texto en ASCII), y lo intuitivo del conjunto de marcas usadas lo hacen muy sencillo de aprender y de editar. Las marcas, en su mayoría, indican atributos relacionados con aspectos de presentación o de visualización del contenido, siendo muy pocas las etiquetas que pueden identificarse como estructurales. Aún así, los atributos de las marcas en HTML son inexistentes, limitando lo que podía hacerse en la presentación de información. Por esta razón en 1996 se presentara la especificación **CSS** (*Cascade Style Sheets, hojas de estilos en cascada*), que define un lenguaje de diseño gráfico que hace posible ampliar las características y atributos de los elementos en la presentación o visualización de los mismos, y que ofrece prestaciones de las que carece el HTML. La aplicación de estas hojas de estilo hace posible presentar adecuadamente el contenido en diferentes pantallas, o en una versión para impresión, o integrarlo en interfaces diferentes al de una página web.

<https://www.youtube.com/embed/mNbnV3aN3KA>

Videotutorial: Víctor Robles Web, *Aprende HTML en 15 minutos* (2021).

Material complementario

- [Mozilla Developer Network: Referencia de elementos HTML.](#)
- [Manz: Lenguaje HTML5.](#)
- [Etiquetas HTML.](#)



2.2. XML

Los primeros años del web demostraron que **el marcado con HTML no era adecuado para estructurar datos**, ni facilitaba el intercambio de información entre máquinas y aplicaciones, un intercambio de datos que dada vez era más necesario. Para dar una solución a estos problemas, el [W3C](#) (*World Wide Web Consortium*) desarrolla y publica en 1998 el estándar internacional **XML (*Extensible Markup Language*)**, que es independiente de cualquier empresa o administración. Se trataba de un lenguaje de marcado puramente estructural, que no incluía ninguna información relativa a la capa de presentación de la información. **Su objetivo principal es facilitar el intercambio de información y datos estructurados entre aplicaciones.**

Esto quiere decir que se trata de un **metalenguaje**, es decir, un lenguaje de marcado pensado y diseñado para poder ser utilizado en cualquier contexto, siempre y cuando se respetan las lógica y reglas del mismo. Está basado en SGML, del que se ha dicho que es un subconjunto, pero con una reglas menos complicadas y rígidas, buscando favorecer su uso. Al ser un metalenguaje, **permite crear nuevos lenguajes de marcado**, los cuales pueden **definir nuevos tipos de documentos**, así como las **etiquetas o marcas que se usan para identificar y describir los elementos de esos tipos de documentos**. Al definir las etiquetas o marcas, también se puede establece la **sintaxis** de los elementos, en qué orden y de que manera se puede utilizar y combinar, o las dependencias o jerarquías entre ellos, si es necesario.

Las características fundamentales de XML son:

- La **estructura y el diseño son independientes**.
- Utiliza un **esquema** para definir de forma exacta las etiquetas o marcas y los atributos.
- Permite **definir** y nombrar **etiquetas** propias.
- Permite **asignar atributos** a las etiquetas.

<https://www.youtube.com/embed/AZihBEg8VBk>

Videotutorial: OpenWebinars, *¿Qué es XML y para qué se usa?* (2014).

Un documento XML es un fichero de texto plano, que está formado por el conjunto de información y el marcado correspondiente de sus elementos. Este fichero incluirá en su **cabecera** la **declaración** de que se trata de un **documento XML**, y las indicaciones de los url necesarios para que las aplicaciones que los usen sepan dónde encontrar **los DTD o XML Schema pertinentes para su procesamiento** ([véase apartado 1.2](#)). Después de esto, el documento mostrará los **datos estructurados y marcados con las etiquetas** correspondientes, de acuerdo a cómo hayan sido definidas en el DTD o XML Schema de referencia. **Si todo está formulado y marcado correctamente, se dice que el documento XML está bien formado**, y entonces es cuando las aplicaciones proceden a procesar su contenido.

```

<recipe>

<title>Peanutbutter On A Spoon</title>

<ingredientlist>
  <ingredient>Peanutbutter</ingredient>
</ingredientlist>

<preparation>Stick a spoon in a jar of peanutbutter, scoop
and pull out a big glob of peanutbutter.</preparation>

</recipe>

```

fig. 3. Una receta estructurada y

marcada en XML, con su declaración, DTD y elementos ([fuente original](#)).

En XML las etiquetas o marcas indican el significado de los datos; no hacen referencia a su presentación o formato final. Esto es así porque **en el diseño del XML se ha separado el contenido de la presentación**. Si se quiere presentar información tomada de un documento marcado en XML, es necesario **procesar el contenido del documento con hojas de estilo externas**. Esto permite usar los mismos datos y prepararlos para diferentes medios y/o presentaciones, así como usarlos en múltiples aplicaciones. Los datos se mantienen y fluyen marcados en XML, lo que hace posible que este lenguaje de marcado se convierta en un **estándar para el intercambio de datos en la Web** (aunque no el único, [véase JSON en el apartado 2.4](#)).

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
    <dc:title>Homage to Catalonia,</dc:title>
    <dc:creator>Orwell, George, 1903-1950.</dc:creator>
    <dc:type>text</dc:type>
    <dc:publisher>London, Secker and Warburg</dc:publisher>
    <dc:date>[1938]</dc:date>
    <dc:language>eng</dc:language>
  </rdf:Description>
</rdf:RDF>

```

Fig. 4. Combinación de

marcados: documento en XML con su cabecera de declaraciones, que contiene datos en RDF del [estándar de metadatos Dublin Core](#) ([fuente original](#)).

En el uso diario, en muchas páginas web pueden encontrarse combinaciones de HTML con XML y lenguajes derivados del mismo. Como estas combinaciones vienen expresadas en el marcado de las páginas web, no se aprecian en la capa de presentación, por lo que pasan desapercibidas para el usuario final. Si se piensa sólo en XML, son numerosos los gestores de bases de datos que ya son capaces de importar o exportar datos desde ficheros en XML, y, por ejemplo, los ficheros de procesadores de textos almacenan la información en documentos XML dentro de ficheros compactados. El formato [EPUB](#) para libros electrónicos está formado por conjuntos de documentos en XML y derivados como XHTML y SVG.

Material complementario

- Morales, R. [Qué es XML](#). *ticARTE*, 2014.
- Sánchez. J. [Fundamentos de XML](#). 2016.
- desarrolloweb.com [Introducción a XML](#). sin fecha.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



2.3. Markdown

Una variante en auge de los lenguajes de marcado son los denominados **lenguajes de marcado ligero (*Lightweight Markup Language*)**. Se diferencian porque los lenguajes de marcado de la familia de XML usan etiquetas estructuradas para identificar los elementos componentes de un documento (véase apartado 2.x), mientras que el marcado ligero se **utilizan símbolos tipográficos o estructuras sencillas de símbolos y texto, como “:”, “#”, “*”, “[]” y similares**. Además, los lenguajes de marcado ligero **pretenden que el contenido original sea fácilmente comprensible**, sin que las marcas usadas sean impedimento para la **legibilidad**. En consecuencia, se trata de **lenguajes orientados a la producción de documentos**, no al intercambio de datos.

El nacimiento de los lenguajes de marcado ligero se sitúa generalmente en 2004, con la aparición de **Markdown**, de J. Gruber, al que siguieron *Textile*, *AsciiDOC*, *reStructuredText*... **El objetivo de estos marcados ligeros era simplificar y acelerar la producción de documentación digital**, de ahí que su uso principal haya sido en entornos de programación. Sin embargo, la facilidad de uso y su sencilla curva de aprendizaje ha hecho que se hayan **adoptado ampliamente como base para las aplicaciones de anotación, y como lenguaje de marcado para la creación de documentos maestros**, de los cuales generar, posteriormente, versiones en diferentes formatos digitales (procesadores de texto, HTML, PDF, EPUB...). También se han extendido a los entornos de gestión de contenidos para la creación de webs, de tal forma que, por ejemplo, *Wordpress*, entre otros, acepta el uso de Markdown para la creación de entradas y páginas.

https://www.youtube.com/embed/y6XdzBNC0_0

Videotutorial: Javier Cristóbal Gutiérrez. *Aprende Markdown RÁPIDO! - Sintaxis básica en menos de 5 MIN.* 2018.

La edición de documentos en Markdown puede hacerse con un simple editor de ficheros de texto plano, aunque es más recomendable usar alguno de los editores específicos que ofrecen soporte específico para Markdown ([véase apartado 3.3](#)). Los documentos etiquetados con marcado ligero, no sólo en Markdown, necesitan de una aplicación que haga la transformación a un formato legible. Esta tarea es llevada a cabo por los llamados generadores de documentación. El **flujo de trabajo básico para elaborar documentación usando un lenguaje de marcado ligero como Markdown** es sencillo:

1. En primer lugar, el proceso de edición sólo requiere de un **editor de texto**, aunque es preferible emplear editores específicos para los lenguajes de marcado ligero.

2. Todo documento necesita que se utilice una herramienta de software capaz de leer el texto y las marcas, interpretarlo y procesarlo de acuerdo a unas reglas, los **generadores de documentación** ([véase apartado 3.4](#)).
3. A los procesadores o generadores se les provee con el **fichero (o ficheros) de texto marcado, el fichero de configuración, el formato de salida, y la hoja de estilo o plantilla** a utilizar.
4. **Los generadores crean el documento o documentos resultantes**, con su estructura organizativa si así se ha indicado, informando con mayor o menor detalle, según la configuración, de los resultados del proceso y de los posibles errores.
5. El documento resultante está preparado para su **publicación y distribución** en internet, bien como un sitio **web** estático, bien en **otros formatos legibles por máquina**.

<p>In this final guide, we will dive into how you can automatically cite using Zettlr! If you have been using the Zotero plugin for Word before (or even the Citavi plugin), rest assured: It works almost the same, you only have much more freedom to adapt the citations to your needs!</p> <p>To begin citing with Zettlr, you'll need to set up a references database, (which we describe in our documentation) (https://docs.zettlr.com/en/academic/citations/). For the purposes of this tutorial, we have already prepared a small database which'll cover everything you need to know! Let's load it! In the tutorial directory, there is a small file called "references.json". It contains some references that Zettlr can cite. To load it, first head over into the preferences and into the tab "Export." Once there, navigate to the file using the file browser of the references database-field, and save the preferences.</p> <h2>## Your First Citation 📄</h2> <p>Zettlr will immediately load the file and you're able to cite. Let's have a look at the following blockquote, which certainly needs a citation:</p> <p>Es findet hier also ein Widerstreit statt, Recht wider Recht, beide gleichmäßig durch das Gesetz des Warenaustauschs besiegelt. Zwischen gleichen Rechten entscheidet die Gewalt. Und so stellt sich in der Geschichte der kapitalistischen Produktion die Normierung des Arbeitstags als Kampf um die Schranken des Arbeitstags dar – ein Kampf</p>	<p>In this final guide, we will dive into how you can automatically cite using Zettlr! If you have been using the Zotero plugin for Word before (or even the Citavi plugin), rest assured: It works almost the same, you only have much more freedom to adapt the citations to your needs!</p> <p>To begin citing with Zettlr, you'll need to set up a references database, which we describe in our documentation. For the purposes of this tutorial, we have already prepared a small database which'll cover everything you need to know! Let's load it! In the tutorial directory, there is a small file called "references.json". It contains some references that Zettlr can cite. To load it, first head over into the preferences and into the tab "Export." Once there, navigate to the file using the file browser of the references database-field, and save the preferences.</p> <h2>Your First Citation 📄</h2> <p>Zettlr will immediately load the file and you're able to cite. Let's have a look at the following blockquote, which certainly needs a citation:</p> <p>Es findet hier also ein Widerstreit statt, Recht wider Recht, beide gleichmäßig durch das Gesetz des Warenaustauschs besiegelt. Zwischen gleichen Rechten entscheidet die Gewalt. Und so stellt sich in der Geschichte der kapitalistischen Produktion die Normierung des Arbeitstags als Kampf um die Schranken des Arbeitstags dar – ein Kampf zwischen dem Gesamtkapitalisten, d.h. der Klasse der Kapitalisten, und dem Gesamtarbeiter, oder der Arbeiterklasse.</p>
---	---

Fig. 5. Marcado en Markdown y aspecto final en HTML

Esto incide en la **rapidez de elaboración y actualización**: en caso de introducir cambios, basta con retocar el documento maestro en Markdown, y ordenar al generador de vuelta a generar, valga la redundancia, nuevas versiones de los documentos resultantes, tarea que se lleva a cabo en segundos. El potencial de Markdown ha hecho que se use como lenguaje de documentación en repositorios como *Github*, o para ampliar las prestaciones de paquete estadístico *R*. Sin embargo esta expansión y desarrollo también ha propiciado que haya varios "sabores" de Markdown, con cambios en lagunas etiquetas o marcas, lo que en ocasiones pueden ocasionar problemas en la generación de los documentos finales.

Material complementario

- Cristóbal J. markdown.es. 2016
- Lázaro, E. [Tutorial Markdown](#). 2023.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



2.4. Otros lenguajes de marcado

Los lenguajes vistos en los apartados anteriores no son los únicos lenguajes de marcado que pueden encontrarse y utilizarse en internet. Se encuentran disponibles otros marcados que es necesario conocer, y que pueden ayudar a comprender la estructura y los flujos de información y de datos que circulan por internet.

- **RDF (Resource Description Framework)**: es una especificación que se usa para modelos de metadatos. Está orientado a su uso por máquinas, y estructura la información por tripletas. Aunque en origen no es un marcado XML, se ha reformulado una versión que cumple con los requerimientos de XML. Se utiliza en aplicaciones del web semántico.
- **reStructuredText**: es un lenguaje de marcado ligero utilizado sobre todo en el ámbito de la programación en *Python*, ya que forma parte del paquete *Docutils*, para la generación de documentación. El generador de documentación con el que suele ir asociado es *Sphinx*.
- **JSON (JavaScript Object Notation)**: es un marcado sencillo para intercambiar datos en ficheros de texto. Se usa en entornos donde no es necesario un control de los datos tan estricto como el que hace XML. Su nivel de seguridad es bajo. Se pueden encontrar aplicaciones web que hacen uso combinado de JSON y de XML. Se han desarrollado especializaciones, como *GeoJSON*, para estructuras de datos geográficos.



Fig. 6. Comparación de marcado de datos entre XML y JSON ([fuente original](#)).

- **YAML (YAMI Ain't Markup language)**: se trata de un lenguaje ligero para marcado de datos. Se utiliza ampliamente para datos en diferentes lenguajes de programación, y en ficheros de configuración. Por ejemplo, los generadores de documentación usan YAML para especificar las estructuras de organización de documentos y otros parámetros.

- **MathML:** es un lenguaje de marcado, basado en XML, que se usa para expresar notaciones matemáticas, y permite el intercambio de información entre aplicaciones matemáticas. Es un ejemplo de los diferentes lenguajes por campos específicos que se han desarrollado siguiendo el modelo y requerimientos del XML.

Material complementario

- [*Introducción a MathML, el lenguaje de diseño para las matemáticas.*](#)
programación.net. (sin fecha)