Etiquetados para creación de contenidos en internet

Principios básicos sobre lenguajes de marcado para documentos y datos en internet

- Créditos
- 1. Introducción al etiquetado de documentos
  - 1.1. Fundamentos de los lenguajes de marcado
  - 1.2. Nociones básicas sobre lenguajes de marcado
- 2. Lenguajes de marcado
  - 2.1. HTML
  - 2.2. XML
  - 2.3. Markdown
  - 2.4. Otros lenguajes de marcado
- 3. Gestión y publicación de contenidos
  - 3.1. Los procesos de creación y publicación de contenidos
  - 3.2. Los CMS y las pilas JAMstack
  - 3.3. Herramientas: editores
  - 3.4. Herramientas: generadores de sitios web
- 4. Normativa MRCDD relacionada

Créditos

Departamento de Ciencias de la Documentación e Historia de la Ciencia
Universidad Zaragoza

Jesús Tramullas Saz

Seminario Permanente en Información y Documentación
Grado en Información y Documentación, Universidad de Zaragoza

Versión 1.0. Enero de 2023

Cualquier observación o detección de error en soporte.catedu.es

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

1. Introducción al etiquetado de documentos

Introducción a los lenguajes de marcado

1. Introducción al etiquetado de documentos

1.1. Fundamentos de los lenguajes de marcado

Los orígenes de los lenguajes de marcado hay que buscarlos en los primeros procesadores de texto y en los procesos de impresión física relacionados. Estas marcas indicaban a las impresoras cómo debían imprimir determinadas partes del texto. Con la generalización de las pantallas, **el uso de estas marcas se amplió para facilitar la presentación visual de la información**. La utilidad de las marcas en múltiples contextos no pasó desapercibida para la industria informática, y en la década de 1960 C. Goldfarb crea, a instancias de IBM, el *GML* (*Generalized Markup Language*), como lenguaje de marcas de formato estandarizado para cualquier clase de documento, independientemente de su contenido. En 1986 se convirtió en el estándar ISO 8879, *SGML* (*Standard Generalized Markup Language*). SGML es muy complejo, por lo que su uso es limitado. La aparición del *HTML*, que usaba un subconjunto limitado de SGML, a comienzos de la década de 1990 y su expansión en el web puso las bases para la aparición, durante esa década, de otros lenguajes de marcado de documentos, como *XML*, *DockBook* o *DITA*, y de un buen número de lenguajes especializados, que hacen uso del marcado.

El conocimiento y la difusión generalizada de los lenguajes de marcado se ha producido como consecuencia del desarrollo del world wide web, desde la década de 1990. Hasta ese momento, su utilización se limitaba al campo de la generación de documentación técnica especializada. Sin embargo, cuando Tim Bernes-Lee busco y desarrolló la manera de preparar documentos para distribuirlos y visualizarlos a través de internet, usando un navegador web, recurrió a los fundamentos del *SGML*, creó el primer *HTML*, y dio paso al web tal y como lo conocemos en la actualidad.

Las páginas o documentos web son, en realidad, documentos etiquetados. Su contenido es textual, y el texto, el contenido informativo y documental, se ve complementado por un **conjunto de etiquetas o marcas**. La lógica subyacente es que cualquier contenido, sea texto, datos, enlaces, etc, se puede marcar, y esa esa marca significará algo para cualquier aplicación que pueda leer ese documento. Si una aplicación, un programa, recibe un documento con etiquetas o marcas, y sabe qué significan esas marcas, entonces puede ejecutar sobre el contenido marcado todas las instrucciones que se le den.

Los lenguajes de marcado se usan en documentos que tienen un contenido informativo y documental. A diferencia de los lenguajes de programación, que sirven para escribir algoritmos (órdenes, secuencias y acciones que una máquina debe ejecutar), **los lenguajes de marcado no**

se ejecutan: simplemente, etiquetan o marcan elementos. El marcado o etiquetado de los elementos no es arbitrario: todo lenguaje tiene una estructura lógica. **Cada etiqueta o marca lo que hace es indicar un atributo de aquello que está marcando**.

Los lenguajes de marcado se utilizan para todo en internet. No sólo para las páginas web: gran parte de **la información que circula entre máquinas a través de la red adopta la forma de documentos etiquetados con lenguajes de marcas**. El **intercambio de datos entre aplicaciones se hace usando lenguajes de marcado**. En los ordenadores de escritorio se pueden encontrar muchas aplicaciones que guardan información en ficheros etiquetados, aunque el usuario o usuaria sea ignorante de ello. Los documentos con información etiquetada mediante lenguajes de marcado son omnipresentes.

Los lenguajes de marcado no son para humanos: **son para máquinas**. Aunque es común marcar elementos cuando se está creando un documento (por ejemplo, al crear una página web usando directamente etiquetas o marcas de HTML), son las máquinas y su software el destinatario de estos marcados. En el mismo caso del HTML, se marcan elementos para que el navegador sepa cómo presentarlos al usuario. Al usuario se le aísla de las etiquetas o marcas: cuando se crea un contenido en un blog, por ejemplo, el editor visual lo que está haciendo es trasladar a lenguaje de etiquetado, de manera transparente, lo que el usuario está introduciendo y maquetando.

Material complementario

- Wikipedia: lenguaje de marcado.

1. Introducción al etiquetado de documentos

1.2. Nociones básicas sobre lenguajes de marcado

La definición más común establece que un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información acerca de la estructura del texto o de su formato de presentación. Estos lenguajes pueden hacen explícita la estructura del documento que se trate, pueden indicar el contenido semántico, o pueden señalar e indicar cualquier otro tipo de información que pueda ser relevante para un uso dado.

Los lenguajes de marcas se dividen en tres grandes grupos:

- **Lenguajes de presentación**: son aquellos orientados a definir el formato o la capa de presentación del texto. Suelen ocultar las etiquetas y mostrar al usuario solamente el texto con su formato. El conocido RTF para ficheros de texto es un marcado de este tipo.
- **Lenguajes de procedimientos**: orientados también a la presentación, pero además incorporan elementos que la aplicación o programa que representa el documento debe interpretar para ejecutar acciones en función de éstos. El HTML de las páginas web es un ejemplo.
- **Lenguajes descriptivos o semánticos**: son lenguajes diseñados para representar las diferentes partes en las que se estructura un documento, y para definir su contenido. Sin embargo, y a diferencia de los anteriores, no especifican cómo deben representarse los documentos en su capa visual. Son los utilizados para facilitar el intercambio de información y datos entre aplicaciones. XML es el estándar actual para ello.

El funcionamiento de los lenguajes de marcado es simple: un elemento se destaca del resto de información mediante una marca o etiqueta:

El documento marcado se somete a un procesador, que interpreta las marcas y genera un documento final, o bien ejecuta una serie de acciones sobre el contenido etiquetado (presentación en pantalla, incorporación a una base de datos, relación con otros elementos etiquetados...). Es importante señalar que **un mismo documento puede contener al mismo tiempo diferentes lenguajes de marcado**, y que será el procesador de documentos o la aplicación que lo trate, en cada caso, la que decida que debe hacer con el contenido marcado.

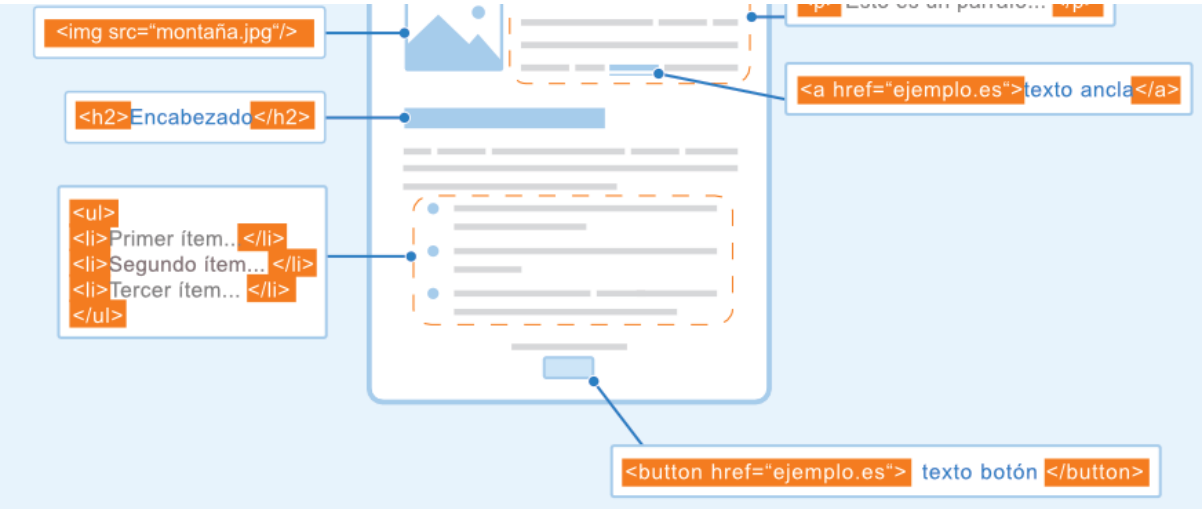

Fig. 1.

Ejemplo de marcado en HTML y su representación visual (fuente original).

Los lenguajes de marcas se usan para etiquetar, para marcar, elementos dentro de un documento. En el párrafo anterior se ha indicado que este documento se somete a un procesador. Ahora bien ¿cómo sabe el procesador cómo procesar, valga la redundancia, las marcas? Esto es posible porque todos y cada uno de **los lenguajes de marcado tienen un documento de referencia en el que se explicitan las reglas sobre cómo se estructuran los documentos, que marcas y etiquetas se usan, lo que significan, y cómo se aplican y relacionan**. Estos documentos de referencia se identifican con las siglas *DTD* (*Document Type Definition*), o *XML Schema*. Por ejemplo, hay un DTD/XML Schema para HTML, otro para XML, etc. Estos documentos de referencia se publican en internet de manera abierta, de forma que cada procesador o aplicación pueda acudir a la url correspondiente, y cargarlo para saber cómo actuar ante cada documento.

El **flujo de trabajo** que se establece es el siguiente:

1. Una persona o una máquina crea un **documento de texto, sobre cuyo contenido aplica un lenguaje de marcado.**
2. Una aplicación o un **procesador accede al contenido** de ese documento.
3. **Identifica**, generalmente en la cabecera o primeras líneas de texto, los **DTD o XML Schema que debe usar para procesar el contenido**.
4. **Utiliza los url de los DTD/XML Schema para** ir a la localización original, y **cargar su contenido** como parámetros de trabajo.
5. Una vez cargados, **procesa el documento y su contenido marcado, de acuerdo con las reglas obtenidas** del DTD/XML Schema.

El resultado de este procesamiento puede ser la visualización de una página web, la creación de un nuevo documento, la incoporación de datos a una base de datos, u otros que puedan haber sido programados en la aplicación correspondiente. Por ejemplo, la posibilidad de crear un documento HTML, o en formato EPUB, desde un documento generado por un procesador de textos, es un ejemplo del uso de lenguajes de marcado y del procesamiento que crea diferentes tipos de resultados en virtud de diferentes reglas de procesamiento, todo ello de forma transparente para el usuario final.

2. Lenguajes de marcado

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

2. 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 s**u 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.*

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**.

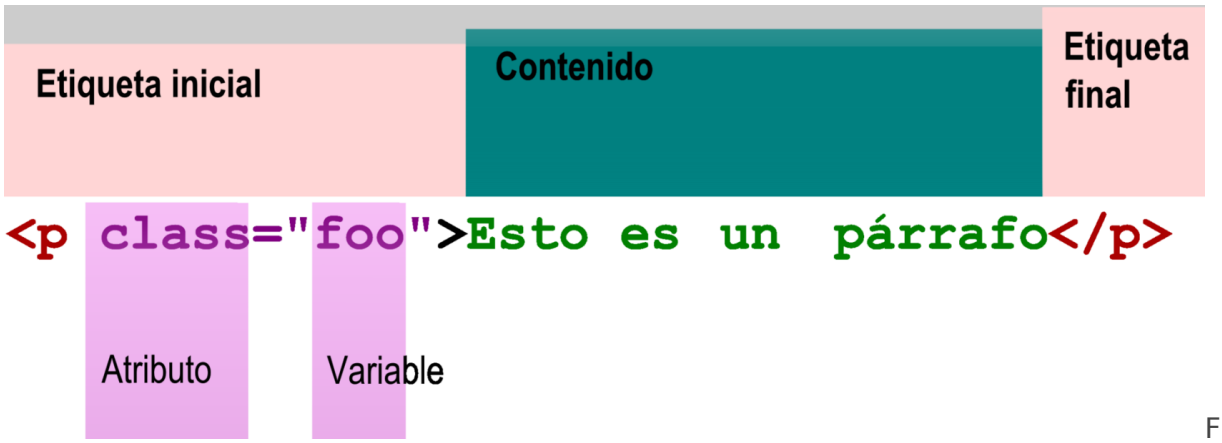

Fig. x.

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.

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

2. Lenguajes de marcado

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. S**u 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 l**os 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 Sxchema 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.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE recipe PUBLIC "-//Happy-Monkey//DTD RecipeBook//EN"
"http://www.happy-monkey.net/recipebook/recipebook.dtd">

<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).

```
<?xml version="1.0" encoding="UTF-8"?>
<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. Lenguajes de marcado

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 a**doptado 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 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**.

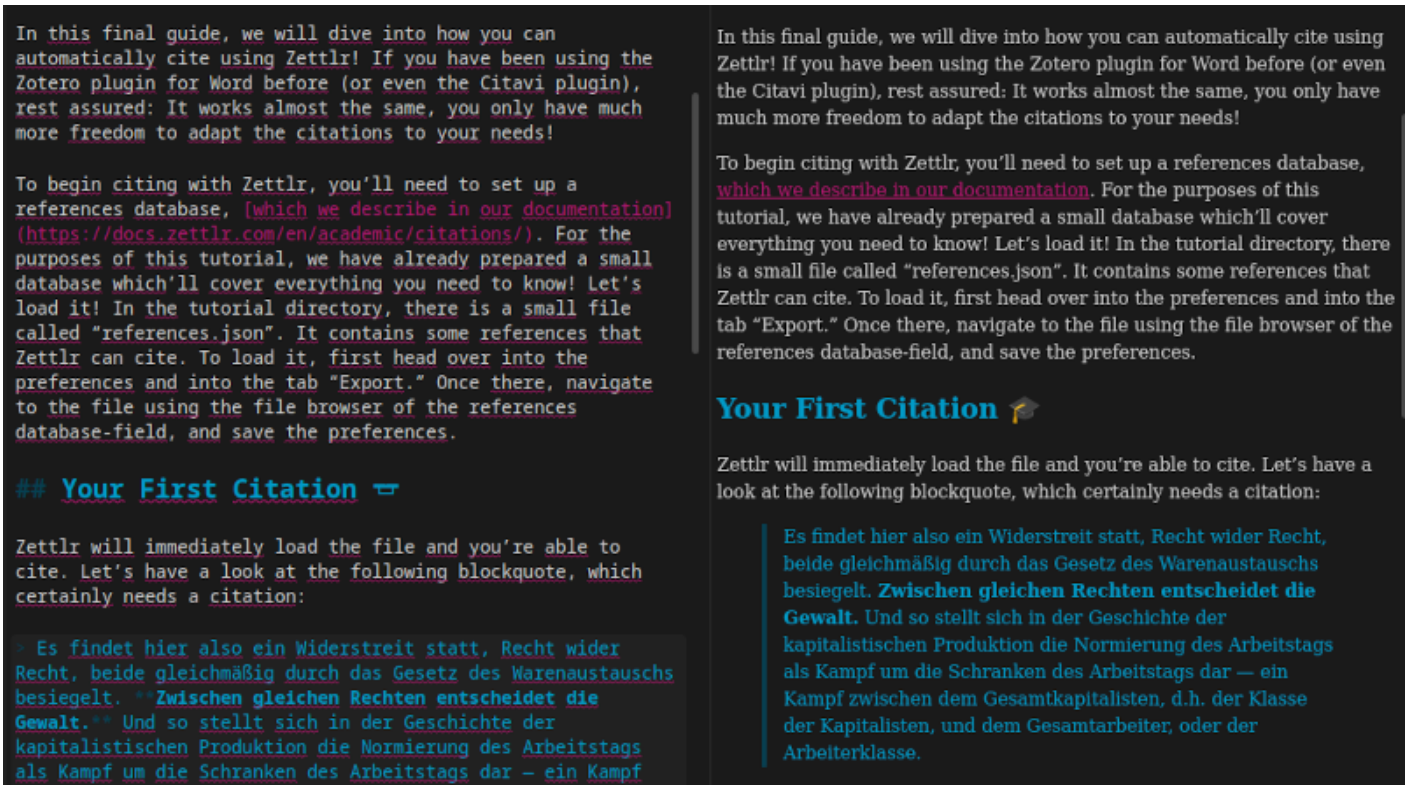

Fig. 5. Marcado en Markdown y aspecto final en HTML

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 vuelva 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 documententación 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 ocasiones 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. Lenguajes de marcado

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 informacion 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 Objetc 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** (***YAML Ain't Markup language***): se trata de un lenguaje ligero para marcado de datos. Se utliza 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)

3. Gestión y publicación de contenidos

Procesos y herramientas para crear, editar y publicar contenidos

3. Gestión y publicación de contenidos

3.1. Los procesos de creación y publicación de contenidos

El desarrollo del web ha traido consigo la popularización de los procesos de creación de contenidos. La facilidad para crear páginas web gracias al marcado en HTML, que se popularizó en la década de 1990, trajo consigo un desarrollo acelerado de tecnologías y de contenidos, creando un volumen de información libremente accesible como nunca antes. Sin embargo, la mera capacidad técnica de crear páginas en HTML pronto se reveló como insuficiente. **La evolución de los procesos pronto hizo necesaria la formalización y estudio de los procedimientos, tanto organizativos como técnicos e informacionales, que soportaban la creación de contenidos**. Ello llevó a la aparición y formalización de la **gestión de contenidos**, entendiendo como tal "*el resultado final de un proceso de creación y distribución de información, que se concreta en la organización de recursos de información, su acceso y el acceso, la presentación de diferentes procedencia y naturaleza, respaldados desde una política editorial*."

Se trata de un proceso complejo de **colección, gestión y publicación de contenido informativo en el entorno digital**, que atiende a tres procesos principales:

1. **Colección**: la creación o adquisición de contenidos, de diferentes tipos y orígenes.
2. **Gestión**: la creación, gestión y mantenimiento de repositorios (depósitos de datos y documentos).
3. **Publicación**: la extracción de componentes del repositorio y la aplicación de un conjunto de reglas para la creación de publicaciones.

Aunque en la década de 1990 las herramientas informáticas solían preceder a las formulaciones teóricas y a la mejora y afinamiento de métodos y técnicas, la gestión de contenidos se convirtió, de manera empírica, en una disciplina cuyo objetivo era el **diseño y publicación de contenidos ajustados a las necesidades informacionales de una organización o de un conjunto específico de usuarios**. En la misma se conjugaban métodos y técnicas de gestión de proyectos, análisis de usuarios, arquitectura de la información, definición de políticas editoriales y productos de información... hasta dar forma a una **estrategia de contenidos**. La estrategia se plasma en la puesta en marcha de un prodcuto de información, que debe dar respuesta a unas necesidades, y cuyos alcance y costes deben situarse dentro de un contexto bien dimensionado.

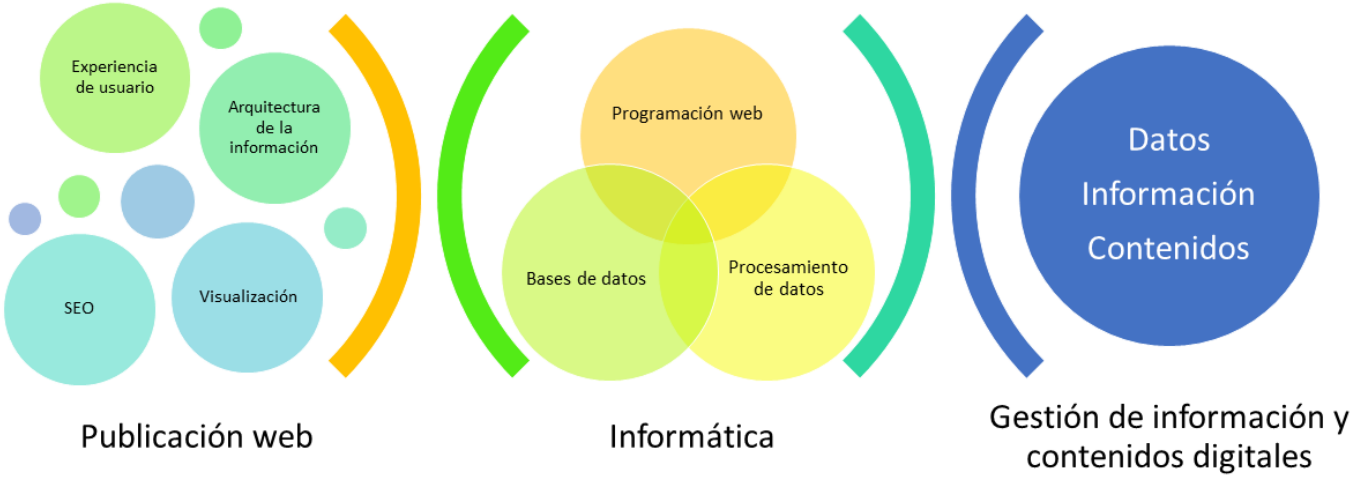

Fig. 7. Técnicas y herramientas de la gestión de contenidos (fuente original)

El desarrollo clásico de un producto de información responde a un conjunto de fases, muy similares a las fases clásicas de desarrollo de proyectos:

1. Definición de **objetivos** del proyecto.
2. Estudio de **necesidades de información** y de la **comunidad de usuarios**.
3. Definición de **objetivos del producto de información.**
4. Definición de la **política editorial.**
5. Selección de **componentes** del producto.
6. Implementación de los **flujos de trabajo.**
7. **Publicación**.
8. **Seguimiento, evaluación y mejora**.

Material complementario

- Madurga, J. Cómo crear un plan de contenidos paso a paso desde cero. Semrush. 2022.
- Sánchez Valls, A. *Guía paso a paso para la gestión de proyectos digitales*. 2019.

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

3. Gestión y publicación de contenidos

3.2. Los CMS y las pilas JAMstack

Durante la segunda mitad de la década de 1990, en plena expansión de internet y sus servicios, los **sistemas de gestión de contenidos (*Content Management Systems*, CMS)** se convirtieron en la herramienta clave para el desarrollo y la implementación de productos de información digital en la red. Hubo que esperar a los inicios de la década siguiente para que los CMS se popularizaran a todos los niveles, gracias a la publicación de gran número de herramientas, muchas de ellas bajo licencias de software libre.

La base se identifica con el acrónimo **LAMP**, que corresponde a la combinación de Linux, Apache, MySQL y PHP ó Perl. Sobre un servidor de espacio dedicado o compartido, la combinación del software para servidor web Apache, del sistema de gestión de bases de datos MySQL (o María DB), y de intérpretes de lenguajes PHP y Perl, permite instalar y operar casi cualquier tipo de servicio o de producto de información digital. Los **sistemas de gestión de contenidos** instalados sobre estas bases han respondido a una **arquitectura casi estandarizada**, en la cual pueden identificarse tres subsistemas principales, correspondientes a los bloques de procesos de gestión y administración, de colección y de publicación. El aumento de prestaciones de los CMS ha hecho más complicada su gestión. Si bien en sus primeros momentos se pretendía simplificar y democratizar los procesos de publicación en internet, el aumento de la complejidad de los sistemas, la proliferación de plataformas y canales de publicación, y el aumento de fuentes de información externas de las que es necesario integrar información ha limitado las posibilidades de los CMS que se podrían considerar "tradicionales". La falta de flexibilidad ha sido señalada como uno de los problemas de esta generación de CMS.

También resulta evidente que **existen situaciones en las cuales determinados productos de información no requieren de las prestaciones que ofrecen los sistemas de gestión de contenidos**. La codificación de documentos mediante HTML puede ser suficiente en contextos en los que no sea necesaria una actualización continua de contenido. Sin embargo, si se atiende al principio citado previamente de "crear una vez, publicar en muchos lugares", HTML no es esquema de codificación que se preste a un proceso de edición ágil.

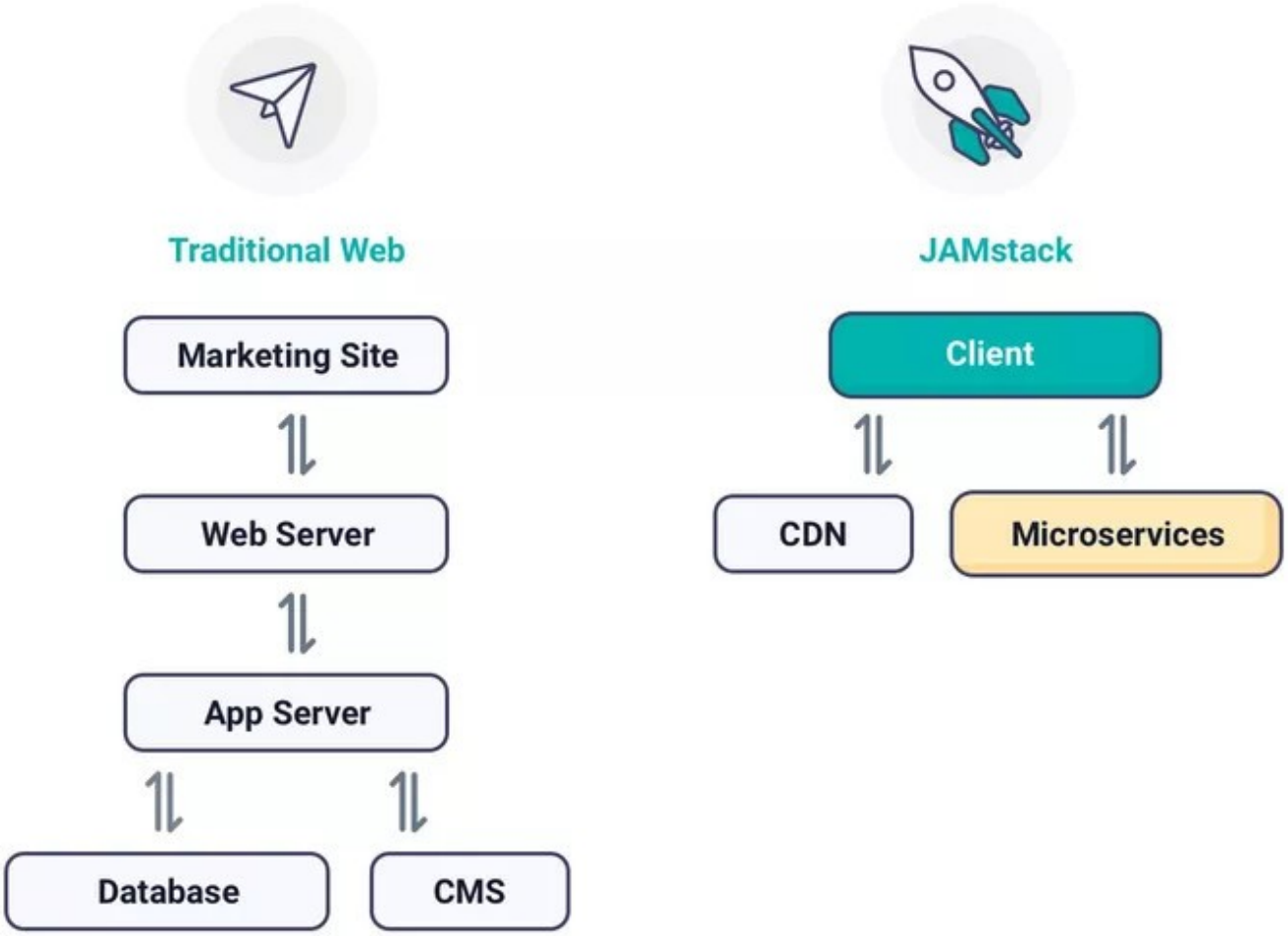

Fig. 8. Comparación de estructuras entre CMS tradicional y JAMstack (fuente original).

La necesidad de generar documentos web de forma rápida y sencilla ha llevado a la aparición de herramientas generadoras de páginas web estáticas. Por este motivo están adquiriendo una creciente importancia las denominadas JAMstack. **Las JAMstack son la combinación de código escrito en JavaScript, diferentes API o microservicios reutilizables, y documentos cuyo contenido se ha etiquetado con lenguajes de marcado ligero** (véase apartado 2.3). Derivados principalmente del campo de la elaboración de documentación de software, se basan en la utilización de **lenguajes de etiquetado ligero**, como MarkDown o reStructuredText, que luego son **procesados por generadores de documentación** como Sphinx o MkDocs. El resultado se envía a un espacio de hosting, en el cual se publica el contenido generado. La **velocidad de respuesta** de los servidores al enviar páginas HTML estáticas es mucho mayor que al interactuar con un CMS. La **gestión de las modificaciones** en estructura y contenido del conjunto de documentos, en un entorno local de producción, también es **más ágil y sencilla**, ya que es el generador el que se encarga de actualizar y regenerar la estructura de acuerdo a los cambios introducidos. Si a esto se acompaña el control de versiones y un cliente de sincronización con repositorios, se dispone de un entorno completo para la

elaboración de documentos para internet.

Material complementario

- Fernández Alonso, A. ¿Qué es un CMS? Sistema de gestión de contenido *webempresa* . 2022.
- Acosta, J. Jamstack: Qué es y ventajas que ofrece. *Openwebinars*. 2022.

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

3. Gestión y publicación de contenidos

3.3. Herramientas: editores

El etiquetado de documentos para internet puede llevarse a cabo con cualquier editor de texto plano, sin mayores complicaciones. Sin embargo, la pauta de trabajo recomendable, y recomendada, es utilizar para ello alguno de los múltiples editores especialmente diseñados a tal fin, que cuentan con funcionalidades para ayudar al marcado, como el destacado de color para etiquetas u otros elementos, o la previsualización del resultado final. Al alcance del usuario se encuentran disponibles editores para Markdown, para XML, para HTML... o editores multiformato, pensados para poder atender las necesidades de diferentes marcados y codificaciones con una misma herramienta.

- **VSCodium**: es un editor rmuy recomendable, que está preparado para dar soporte a diferentes lenguajes de programación y de marcado. Sus funcionalidades pueden ampliarse a través de extensiones, y está preparado para trabajar con ficheros en XMl, HTML, Markdown y YAML, pudiendo incorporar JSON y ReStructuredText . Es mutiplataforma.
- **Kate:** es un editor también preparado para dar soporte a más de 300 lenguajes, incluyendo XML, HTML, Markdown, ReStructuredText, JSON y YAML. También puede incorporar extensions para ampliar prestaciones. Al igual que VSCodium, tienen instaladores para GNU/Linux, OSX y Windows.
- **Code Browser**: otro editor para lenguajes de programación, incorpora soporte para XML y HTML. Disponible para GNU/Linux y Windows.
- **XML Copy Editor**: editor especializado para XML y sus derivados. Tiene versiones para GNU/Linux, OS X y Windows.
- **Notepad++**: es un editor para sólo para máquinas con Windows, con soporte para HTML, XML, YAML y JSON.
- **BlueGriffon**: editor para ficheros con marcado en HTML. Da soporte a HTMl y a XHTML. Ofrece versiones para GNU/LInux (Debian), OS X y Windows. La versión de softwre libre ofrece funcionalidades limitadas para CSS.

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

3. Gestión y publicación de contenidos

3.4. Herramientas: generadores de sitios web

El complemento de los lenguajes de marcado ligero son los generadores de documentación. Estas herramientas toman como material de trabajo los ficheros etiquetados en Markdown o en ReStructuredText (véase apartado 2.3), leen la configuración, aplican una hoja de estilos, y **generan los documentos resultantes normalmente en HTML**. En los últimos años han parecido un buen número de generadores de documentación, la mayoría bajo licencias de software libre, algunos de los cuales se han popularizado rápidamente. Esto ha permitido que los usuarios puedan dejar de lado los blogs soportados por Wordpress, por ejemplo, y optar por webs de páginas estáticas generadas de esta forma, lo que supone un ahorro considerable de espacio y de velocidad de respuesta, además de simplificación de gestión de datos, al no necesitar un sistema de gestión de bases de datos instalado en el espacio web del que disponen.

- **Jekyll**: es un generador que toma como material de trabajo fichero etiquetados en Markdown. Se utiliza bastante para generar estructuras de contenido similares a los blogs. Necesita instalar el lenguaje de programación *Ruby* en el ordenador del usuario. Se puede instalar en GNU/Linux, OS X y Windows.
- **Hugo**: otro popular generador, que partiendo de ficheros marcados con Markdown, es capaz de generar estructuras de páginas web estáticas. Está programado en *Go.* Ofrece instaladores para varios sistemas operativos.
- **Sphinx**: un generador específico para trabajar con *ReStructuredText*, ya que se usa para la documentación de aplicación en *Python*. Ofrece prestaciones avanzadas de organización de páginas e información. Requiere instalar *Python* y *Docutils*. También es capaz de generar documentos en PDF y en EPUB.
- **MkDocs**: es une generador rápido y sencillo de utlizar, que toma como material de partida ficheros en Markdown. La configuración se escribe en un fichero en YAML. Ofrece un buen número de estilos de presentación visual entre los que elegir. Incluye un servidor web local en el que visualizar, en tiempo real, los cambios que se vayan haciendo sobre los ficheros fuente. Requiere instalar Python en la máquina del usuario.

Es importante destacar que **las JAMstack desarrollan todo su potencial cuando los documentos marcado están despositados en repositorios de código**, ya que se mantiene un control de versiones, lo que significa que se automatiza el histórico de cambios, y es posible

recuperar y cambiar a versiones anteriores en cualquier momento. Además, al estar separada la capa de presentación, en plantillas y hojas de estilo, los cambios de ésta, o del contenido que muestra, no afectan una a la otra, ni tampoco a las funcionalidades que se hayan integrado con JavaScript. Finalmente, hay que recordar que el código JavaScript se ejecuta en el navegador del usuario, lo que resulta más rápido, seguro y eficiente.

El **flujo de trabajo** ideal es simple:

1. Los ficheros con el contenido etiquetado y el código se almacenan en un repositorio, con funcionalidades de edición y modificación de los mismos. Esto quiere decir que las fuentes se encontrarán en un repositorio Git, como *GitHub* o *Gitlab*.
2. Cuando se produce un cambio en los ficheros, se lanza un proceso de construcción del sitio, que da como resultado un conjunto de páginas web etiquetadas en HTML, y que integran contenido, datos, capa de presentación... trabajo que desarrollan los generadores de sitios (como Jekyll, Hugo, etc.).
3. El resultado se publica automáticamente en una CDN (*Content Delivery Network*), asegurando un rápido despliegue y la disponibilidad física de las páginas, eliminando los tiempos de espera típicos de un CMS sobre base de datos tradicional.

Material complementario

- Site Generators. A List of Static Site Generators for Jamstack Sites. Jamstack. 2023.

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

4. Normativa MRCDD relacionada

En un marco competencial y de gran disponibilidad de contenidos digitales abiertos, la labor del profesorado cada día más se especializa en la **curación de contenidos educativos digitales,** es decir en la adecuada selección de fuentes, adaptación del contenido de ellas al contexto educativo, y posterior catalogación y compartición de los mismos utilizando criterios de seguridad y respeto a las licencias de uso. En este contexto, son necesarias por parte del profesorado unas habilidades técnicas que le permitan utilizar las herramientas digitales que ayudan en el etiquetado y posterior catalogación de sus contenidos educativos digitales.

La Resolución de 4 de mayo de 2022, de la Dirección General de Evaluación y Cooperación Territorial, por la que se publica el Acuerdo de la Conferencia Sectorial de Educación, sobre la actualización del marco de referencia de la competencia digital docente establece las siguientes Competencias Digitales Docentes:

- Dentro del Área 2 correspondiente a Contenidos Digitales, la competencia **2.2 Creación y modificación de contenidos digitales** *que se describe como la capacidad de modificar y adaptar los contenidos educativos digitales, respetando las condiciones de uso establecidas por cada licencia,* y que incluye la utilización de herramientas de autor para modificar o crear contenidos educativos digitales e introducir los **metadatos para su catalogación.**

- Dentro de ese mismo Área 2, la competencia **2.3 Protección, gestión y compartición de contenidos digitales** que se describe como *catalogar los contenidos educativos digitales y ponerlos a disposición de la comunidad educativa y del colectivo profesional utilizando entornos seguros,* y que requiere el uso de distintos **sistemas de clasificación** tanto normalizados como de etiquetado libre.

Todo ello hace que los contenidos tratados en este curso se consideren de vital importancia en la formación del profesorado de cualquier etapa educativa y materia impartida como creador de contenidos educativos digitales, si bien no están específicamente destinados a su transmisión al alumnado, salvo quizás en algunas materias específicas del bachillerato.

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

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
GOBIERNO DE ARAGON