3. Gestión y publicación de contenidos

Procesos y herramientas para crear, editar y publicar 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

Copyright 2026

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

Copyright 2026

cuyos alcance y costes deben situarse dentro de un contexto bien dimensionado.

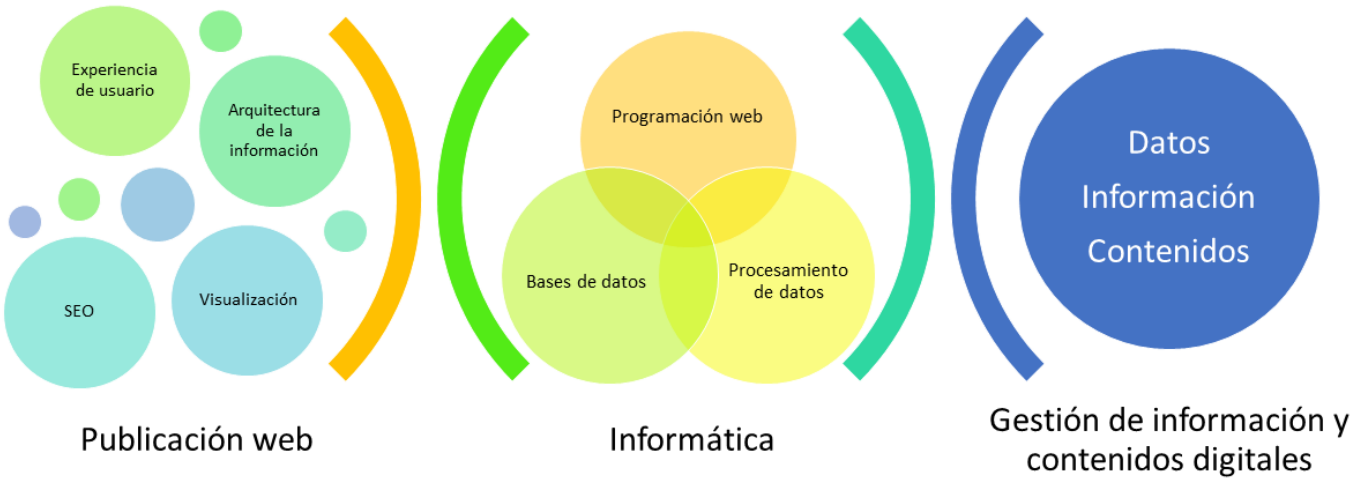


Fig. 7. Técnicas y herramientas de la gestión de contenidos (fuente original)

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

Copyright 2026



Copyright 2026

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.

Copyright 2026

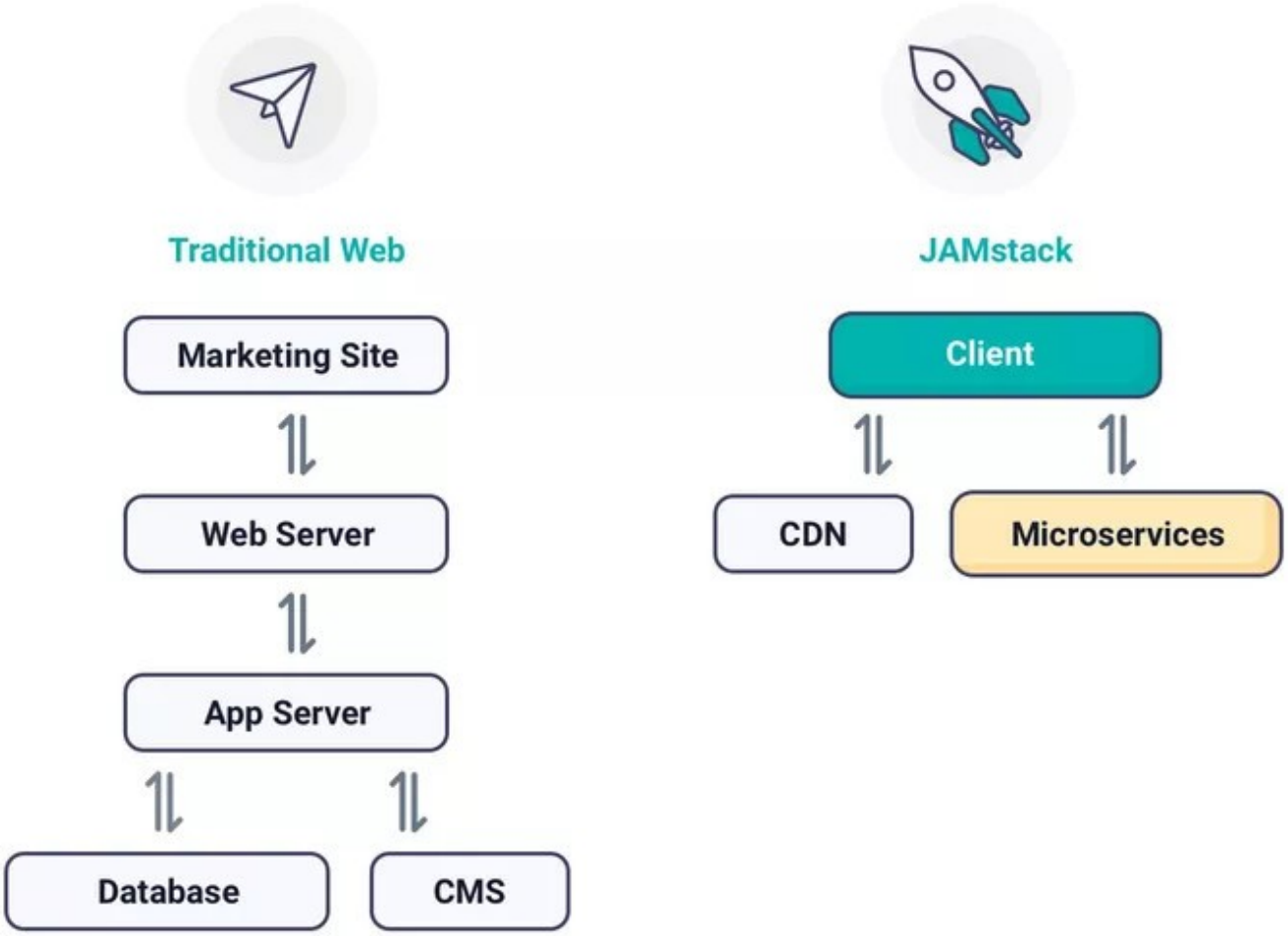


Fig. 8. Comparación de estructuras entre CMS tradicional y JAMstack (fuente original).

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



Copyright 2026

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



Copyright 2026

Copyright 2026

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



Copyright 2026