

Sistemas Gestores de Contenido.

¿Qué es un CMS?

Los sistemas de gestión de contenidos (*Content Management Systems* o CMS) son software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (*Web Content Management* o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las webs.

James Robertson propone una división de la funcionalidad de los sistemas de gestión de contenidos en cuatro categorías: creación de contenido, gestión de contenido, publicación y presentación.

Creación de contenido

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG, en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio web.

Hay otras herramientas como la edición de los documentos en XML, utilización de aplicaciones ofimáticas con las que se integra el CMS, importación de documentos existentes y editores que permiten añadir marcas, habitualmente HTML, para indicar el formato y estructura de un documento.

Un CMS puede incorporar una o varias de estas herramientas, pero siempre tendría que proporcionar un editor WYSIWYG por su facilidad de uso y la comodidad de acceso desde cualquier ordenador con un navegador y acceso a Internet.

Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente.

Gestión de contenido

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la web, etc.

La estructura de la web se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta

estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. Eso es imprescindible para facilitar el ciclo de trabajo (*workflow*) con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del ciclo de trabajo.

Publicación

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas.

Presentación

Un CMS puede gestionar automáticamente la accesibilidad del web, con soporte de normas internacionales de accesibilidad como WAI, y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del visitante.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro del web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otro web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado.

Necesidad de un CMS

En el apartado anterior se han presentado bastantes motivos para ver la utilidad de un sistema que gestione un entorno web, pero se podría pensar que no es necesario para una web relativamente pequeña o cuando no se necesitan tantas funcionalidades. Eso

sólo podría ser cierto para un web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de web.

Muchos usuarios particulares utilizan CMS gratuitos para elaborar y gestionar sus webs personales, obteniendo webs dinámicos llenos de funcionalidades. El resultado que obtienen es superior al de algunas empresas que se limitan a tener páginas estáticas que no aportan ningún valor añadido.

Éstos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

Inclusión de nuevas funcionalidades en el web. Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.

Mantenimiento de gran cantidad de páginas. En una web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.

Reutilización de objetos o componentes. Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.

Páginas interactivas. Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la web.

Cambios del aspecto de la web. Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (*Cascading Style Sheets* u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.

Consistencia de la web. La consistencia en un web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de las de la misma web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.

Control de acceso. Controlar el acceso a un web no consiste simplemente al permitir la entrada a el web, sino que comporta gestionar los diferentes permisos a cada área del la web aplicados a grupos o individuos.

CMS comerciales y de código abierto

Se puede hacer una primera división de los CMS según el tipo de licencia escogido. Por una parte están los CMS comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso. Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente.

La disponibilidad del código fuente posibilita que se hagan personalizaciones del producto, correcciones de errores y desarrollo de nuevas funciones. Este hecho es una garantía de que el producto podrá evolucionar incluso después de la desaparición del grupo o empresa creadora.

Algunas empresas también dan acceso al código, pero sólo con la adquisición de una licencia especial o después de su desaparición. Generalmente las modificaciones sólo pueden hacerlas los mismos desarrolladores, y siempre según sus prioridades.

Los CMS de código abierto son mucho más flexibles en este sentido, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los CMS de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales.

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría de usuarios: su coste. Habitualmente todo el software de código abierto es de acceso libre, es decir, sin ningún coste en licencias. Sólo en casos aislados se hacen distinciones entre empresas y entidades sin ánimo de lucro o particulares. En comparación, los productos comerciales pueden llegar a tener un coste que sólo una gran empresa puede asumir.

En cuanto al soporte, los CMS comerciales acostumbran a dar soporte profesional, con un coste elevado en muchos casos, mientras que los de código abierto se basan más en las comunidades de usuarios que comparten información y solución a los problemas. Las formas de soporte se pueden mezclar, y así encontramos CMS de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuarios. En el caso comercial también sucede, pero el coste de las licencias hace que el gran público se decante por otras opciones y por lo tanto las comunidades de soporte son más pequeñas.

Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos o mal redactada. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

En el mercado hay CMS de calidad tanto comerciales como de código abierto. Muchos CMS de código abierto están poco elaborados (aunque en plena evolución), pero también lo encontramos entre los comerciales. En definitiva, un buen CMS de código abierto es mucho más económico que su homólogo comercial, con la ventaja de disponer de todo el código fuente y de una extensa comunidad de usuarios.

Por todos estos motivos, y como apuesta por la filosofía del software libre, en este proyecto sólo estudiaré algunos CMS de código abierto.

Historia de los CMS

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a medida.

Ya el año 1994 Illustra Information Technology utilizaba una base de datos de objetos como repositorio de los contenidos de una web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no cuajó entre el público y la parte de la empresa enfocada a la Web fue comprada por AOL, mientras que Informix adquirió la parte de bases de datos.

RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos el año 1994. No fue hasta a finales del año siguiente que presentaron su CMS basado en una base de datos.

Entre los CMS de código abierto uno de los primeros fue Typo 3, que empezó su desarrollo el año 1997, en palabras de su autor, Kasper Skårhøj, "antes de que el término gestión de contenidos fuera conocido sobradamente".

PHPNuke, la herramienta que popularizó el uso de estos sistemas para las comunidades de usuarios en Internet, se empezó a desarrollar el año 2000. La primera versión supuso tres semanas de trabajo al creador, rescribiendo el código de otra herramienta, Thatware.

Presente y futuro de los CMS

En la actualidad, aparte de la ampliación de las funcionalidades de los CMS, uno de los campos más interesantes es la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al cambiar de sistema y aportan calidad y estabilidad.

Algunos de estos estándares son CSS, que permite la creación de hojas de estilo; XML, un lenguaje de marcas que permite estructurar un documento; XHTML, que es un subconjunto del anterior orientado a la presentación de documentos vía web; WAI, que asegura la accesibilidad del sistema; y RSS, para syndicar contenidos de tipo noticia.

También las aplicaciones que rodean los CMS acostumbran a ser estándar (de facto), como los servidores web Apache y ISS; los lenguajes PHP, Perl y Python; y las bases de datos MySQL y PostgreSQL. La disponibilidad para los principales sistemas operativos de estas aplicaciones y módulos, permite que los CMS puedan funcionar en diversas plataformas sin muchas modificaciones.

Sobre el futuro de los CMS, Robertson apunta que:

- Los CMS se convertirán en un artículo de consumo, cuando los productos se hayan establecido y más soluciones lleguen al mercado. Eso provocará una disminución de los precios en los productos comerciales y una mayor consistencia en las funcionalidades que ofrecen.
- En este entorno, muchas empresas que implementan webs tendrán que cerrar.
- Muchos proyectos fracasarán por no ajustarse a los estándares y no entender conceptos como usabilidad, arquitectura de la información, gestión del conocimiento y contenido.
- El campo de los gestores de contenido madurará hasta conseguir un alto grado de consistencia y profesionalismo.
- Se adoptarán estándares en el almacenaje, estructuración y gestión del contenido.
- Se producirá una fusión entre gestión de contenidos, gestión de documentos y gestión de registros.

También se puede añadir la incorporación de sistemas de e-learning y gestión del conocimiento, y en los entornos de intranet corporativa, la posibilidad de acceder a otras fuentes de datos como por ejemplo sistemas de soporte de decisiones (*Decision Support Systems* o DSS). El campo de los CMS de código abierto tendría que seguir un desarrollo similar.

Criterios de selección

Antes de empezar el proceso de selección de un CMS concreto, hay que tener claros los objetivos de la web, teniendo en cuenta al público destinatario, y estableciendo una serie de requerimientos que tendría que poder satisfacer el CMS.

La siguiente lista está basada en las funciones principales de los CMS expuestas anteriormente, las indicaciones de Robertson, J. (2002) y una recopilación de los requerimientos básicos de una web.

- ✓ **Código abierto.** Por los motivos mencionados anteriormente, el CMS tendría que ser de código fuente abierto (o libre).
- ✓ **Arquitectura técnica.** Tiene que ser fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos de contenido, presentación y estructura que permita la modificación de uno de ellos sin afectar a los otros. Es recomendable, pues, que se utilicen hojas de estilo (CSS) y patrones de páginas.
- ✓ **Grado de desarrollo.** Madurez de la aplicación y disponibilidad de módulos que le añaden funcionalidades.
- ✓ **Soporte.** La herramienta tiene que tener soporte tanto por parte de los creadores como por otros desarrolladores. De esta manera se puede asegurar de que en el futuro habrá mejoras de la herramienta y que se podrá encontrar respuesta a los posibles problemas.
- ✓ **Posición en el mercado y opiniones.** Una herramienta poco conocida puede ser muy buena, pero hay que asegurar de que tiene un cierto futuro. También son importantes las opiniones de los usuarios y de los expertos.
- ✓ **Usabilidad.** La herramienta tiene que ser fácil de utilizar y aprender. Los usuarios no siempre serán técnicos, por lo tanto hace falta asegurar que podrán utilizar la herramienta sin muchos esfuerzos y sacarle el máximo rendimiento.

- ✓ **Accesibilidad.** Para asegurar la accesibilidad de una web, el CMS tendría que cumplir un estándar de accesibilidad. El más extendido es WAI (*Web Accessibility Initiative*) del World Wide Web Consortium.
- ✓ **Velocidad de descarga.** Teniendo en cuenta que no todos los usuarios disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente o dar la opción.
- ✓ **Funcionalidades.** No se espera que todas las herramientas ofrezcan todas las funcionalidades, ni que éstas sean las únicas que tendrá finalmente la web.

Entre otras:

- Editor de texto WYSIWYG a través del navegador.
- Herramienta de búsqueda.
- Comunicación entre los usuarios (foros, correo electrónico, chat).
- Noticias.
- Artículos.
- Ciclo de trabajo (*workflow*) con diferentes perfiles de usuarios y grupos de trabajo.
- Fechas de publicación y caducidad.
- Webs personales.
- Carga y descarga de documentos y material multimedia.
- Avisos de actualización de páginas o mensajes en los foros, y envío automático de avisos por correo electrónico.
- Envío de páginas por correo electrónico.
- Páginas en versión imprimible.
- Personalización según el usuario.
- Disponibilidad o posibilidad de traducción a varios idiomas.
- Soporte de múltiples formatos (HTML, Word, Excel, Acrobat, etc.).
- Soporte de múltiples navegadores (Internet Explorer, Mozilla, etc.).
- Soporte de sindicación (RSS, NewsML, etc.).
- Estadísticas de uso e informes.
- Control de páginas caducadas y enlaces rotos.

Listado de CMS basados en java

Como ya hemos mencionado en el apartado anterior, el lenguaje por el que hemos optado para desarrollar este proyecto a sido java. Por lo tanto vamos a describir las características principales de los gestores de contenido más importantes basados en este lenguaje.

- **InfoGlue** es un avanzado, potente y robusto gestor de contenido. Es apropiado para una amplia gama de aplicaciones y organizaciones. Entre sus usos típicos se incluyen sitios webs, portales, intranets y todo tipo de webs en general. La plataforma está realizada bajo la licencia GPL y puede funcionar en casi todas las plataformas y bases de datos conocidas.
- **Magnolia** es el primer CMS basado en Java y desarrollable en J2EE que ha sido creado desde cero para soportar el próximo estándar Java Content Repositories (JCR). Según sus desarrolladores, Magnolia ha sido creado para hacer fácil la gestión de contenidos, tanto en su uso, como en su instalación e implementación. Magnolia puede ser instalada en cualquier plataforma y usa Java, XML, JSP, JSL y una librería de tags personalizables.
- **Ion** es un esfuerzo para proporcionar un muy simple pero poderoso sistema gestor de contenido basado en las más recientes tecnologías. Ion es 100% compatible con todos los estándares J2EE, usa XML para definir estructuras de contenido, XSL para mostrar el contenido de numerosas formas (HTML, XML, PDF, TXT, ...) y jsp con JSTL como sistema de plantillas.
- **OpenCms** es el gestor de contenido de código abierto más profesional que hay en el mercado. Ayuda a crear y dirigir complejos sitios webs sin dificultades y sin conocimiento previo de HTML. Tiene un editor integrado WYSIWYG, y una interfaz similar a las ya conocidas en las herramientas office, las cuales ayudan al usuario a crear los contenidos mediante un sofisticado sistema de plantillas diseñadas para hacer cumplir la estructura propia de nuestra web. Como buen Open Source es completamente libre y exento de coste de licencia.
- **Jahia** está a mitad de camino entre un gestor de contenido y un framework para construir portales. Está integrada en un solo paquete que agrupa todo lo necesario para realizar webs completas.
- **JBoss Nukes** es un completo y nuevo gestor de contenido con un avanzado gestor de usuarios y grupos, reglas de seguridad y componentes capaces de conectarse entre ellos para realizar una función común. Estos componentes pueden ser desarrollados en tiempo de ejecución mientras que la aplicación está funcionando. Está completamente escrito en JBoss.
- **MMBase** es un sistema gestor de contenido web con un potente componente para sistemas multimedia. Tiene un extendido uso en Holanda y es usado en la mayoría de difusiones, publicaciones, instituciones educativas holandesas, y en gobiernos nacionales y locales. Está escrito completamente en java, es también un open source y todos los estándares en los que se basa son libres dentro de lo posible.

- **Apache Lenya** es un sistema de gestión de contenido libre escrito en Java, manejando el formato de representación XML. Apache Lenya está basado en Apache Cocoon. Entre sus características están control de versiones, programación de actividades, capacidades de búsqueda, herramientas de workflow, y editores WYSIWYG manejados desde el navegador.
- **Slide** tiene como modulo principal un repositorio de contenido, el cual puede ser visto como un framework para manejar contenido de bajo nivel. Conceptualmente, proporciona una organización jerárquica de contenido binario que se almacena arbitraria y heterogéneamente en una base de datos. Integra seguridad, bloqueos, versiones, así como otros muchos servicios. Puede integrar y dirigir bases de datos sin repositorios externos, requiriendo solo pequeñas capas de abstracción que son escritas en cada repositorio. Los usos de Slider pueden ir desde la creación de intranets al uso de este como servidor de ficheros. Slider también ofrece un modulo de acceso WebDAV (implementado como un Servlet). El WebDAV es un IETF estándar que usan compañías como Microsoft, IBM, Adobe y muchas otras. Esto hace que Slide sea una ideal elección para la realización de webs.
- **Daisy** es un detallado gestor de contenido, consistente en un repositorio autónomo accesible a través de HTML/XML y/o desde una aplicación (remota) java de alto nivel y de una considerable navegador y editor DaisyWiki incluido en Apache Cocoon. Daisy se encuentra bajo la licencia Apache License 2.0.

Referencias

- Robertson, J., *How to evaluate a content management system*. 2002.
- Rhyno, A. *The Ten Commandments of Content Management*. 2003
- <http://www.uoc.edu/mosaic/articulos/cms1204.html>
- <http://www.desarrolloweb.com/>
- <http://www.javahispano.org>
- <http://barrapunto.com>
- <http://www.opensourcecms.com>
- www.cms-spain.com
- www.cmswatch.com
- www.wikipedia.org