

Universidad de Mendoza
Facultad de Ingeniería
Licenciatura en Sistemas

“GESTOR DE CONTENIDOS MVC”

**CREACIÓN DE UN GESTOR MVC ORIENTADO
A COMPONENTES Y MÓDULOS DRAG&DROP
CON SISTEMA DE CACHE**

Autor: Borja Abad López

Legajo: 5544

Profesor titular: Ing. Osvaldo Marianetti

Año: 2013

AGRADECIMIENTOS

La Universidad de Mendoza, fue la tercera universidad a la que accedí en mi vida. En las dos ocasiones anteriores, por circunstancias personales, no pude completar mi formación. Con 25 años decidí una vez más intentarlo, y como dicen, a la tercera va la vencida.

Ahora ya con 30 años y a punto de recibirme, quiero animar a toda la gente que por circunstancias no haya podido completar sus estudios, a que lo hagan, merece la pena el esfuerzo.

Agradecimientos a la Dra. Ing. Cristina Párraga quien me animó a inscribirme, el primer día que accedí a preguntar a la secretaría de la Facultad de Ingeniería.

Agradecimiento obligado a mis padres por todo su apoyo y por transmitirme la importancia de la formación. También a toda la gente que me ha apoyado. A profesores, compañeros y amigos que lo han hecho posible. Gracias.

ÍNDICE DE CONTENIDOS

[INTRODUCCIÓN](#)

[RESUMEN](#)

[Metodología](#)

[Avance establecido](#)

[CAPÍTULO I](#)

[PLANTEAMIENTO DEL PROBLEMA](#)

[Preguntas](#)

[Límites](#)

[Ventajas](#)

[Viabilidad](#)

[Datos concretos](#)

[CAPÍTULO II](#)

[ANTECEDENTES](#)

[BASES TEÓRICAS](#)

[Funcionalidades básicas](#)

[Clasificaciones](#)

[Frameworks y CMS's](#)

[Persistencia](#)

[Caché](#)

[Plantillas o themes](#)

[CAPÍTULO III](#)

[METODOLOGÍA APLICADA](#)

[Hipótesis](#)

[Tipo de estudio](#)

[CAPÍTULO IV](#)

[CREACIÓN DEL CMS](#)

[Introducción](#)

INTRODUCCIÓN

Todo empezó por una idea, el patrón MVC. He conocido decenas de códigos OpenSource, y siempre me ha llamado mucho la atención la manera en que estructuran sus archivos, como llaman a los métodos, como organizan sus variables, etc. Si algo detestaba siempre era el afrontar un nuevo proyecto de código desconocido, y el árduo camino que era necesario recorrer hasta desentrañar sus misterios y hacerse con el código.

Al conocer el patrón *modelo vista controlador* entendí que cualquier proyecto, por diferente que fuera, siempre que lo implementara, podría seguir siendo fácilmente comprensible, más allá del lenguaje de programación empleado.

La elección de esta temática en mi trabajo final, no es casual, viene de atrás. Hace años que trabajo en una empresa cuyo principal producto es un Gestor de contenidos (CMS), Editmaker, que desarrollan desde hace 15 años.

Esto me ha ayudado a comprender los problemas y necesidades en la creación de software a largo plazo, que evoluciona, que crece y se complica.

En mi formación universitaria también he aprendido la necesidad de normalizar y regir las comunicaciones que genera el desarrollo de software en equipo. El proceso unificado de la creación de software inspirado en la organización de Microsoft, las metodologías ágiles orientadas a la productividad, comunicación y trabajo en equipo, y, la necesidad de usar patrones, técnicas de abstracción y otras arquitecturas que de alguna manera permitan serializar o estandarizar la creación de un software concreto, han sido valores que me han generado la necesidad de integrar, o al menos intentarlo, todos estos conocimientos en este trabajo.

La creación de un CMS, implica aparatosidad. Son aplicaciones pesadas bastante complejas. Clonar decenas de instancias (sitios webs) en un solo servidor, va acompañado de seguro de una gran saturación en el rendimiento del mismo.

La necesidad de crear un framework ligero me hizo descartar todos los CMS que existen en el mercado (llevan años evolucionando y son muy completos), y tratar de empezar uno desde abajo, de cero, teniendo en cuenta desde su concepción la importancia de que sea ligero.

Un factor en común entre los existentes en el mercado, es que son PHP, usan MVC y en su mayoría son orientados a componentes.

Me ha ayudado mucho seguir los cursos de Jaisiel Delance, un dominicano dedicado a la enseñanza online de tecnologías que tiene un canal de youtube, cuenta de twitter, perfil de facebook y página web propia, dlancedu.com. Experto en el uso de CMS opensource, ha rescatado lo básico y común a todos, haciendo una base realmente ligera y potente que valoro enormemente y he adoptado como punto de partida.

Entiendo que la creación de un proyecto de este tipo debería contar con el trabajo de un equipo, así como meses de planeación y desarrollo. Todo depende de la ambición de la iniciativa. Si bien soy consciente de este límite, en esta tesis, me gustaría al menos poder sentar las bases de una arquitectura válida para un sistema con los requerimientos planteados, que sea escalable, ligero, preparado para el trabajo en equipo, etc.

RESUMEN

El trabajo final será una bitácora de experiencias, caminos, aprendizajes y conclusiones surgidas en la creación de la arquitectura fundamental de un CMS.

Se trata de desarrollar una aplicación con su *backend* para administrar el *frontend*. La estructura del CMS permitirá el desarrollo en equipos por ser orientado a componentes.

Por un lado, el back, ha de ser robusto y no importa si pesado, confiable, escalable, etc. El tema de orientarlo a componentes, significa que una vez establecido el *core* de la aplicación cualquier desarrollador pueda agregar componentes si está bien definida la *interfaz*.

Por otro lado, el *front*, ha de ser simple, y sobre todo muy liviano, ya que es el que sufre la mayor carga en el servidor. También será orientado a módulos, los cuales agregados, son los que conforman las páginas.

Se profundizará en el concepto página, se desarrollará un sistema *Drag&Drop* para que desde el back se puedan crear las páginas arrastrando módulos dinámicos o estáticos sobre ellas.

Se hará manejo de los buffers del servidor web para implementar un sistema de caché bajo demanda apoyada en el sistema de archivos, con los outputs procesados como archivos planos, tratando así de demostrar que un sistema como el descrito puede ser de mayor rendimiento sin aumentar notablemente la complejidad del mismo y ni que la performance del usuario se vea afectada.

Metodología

Por trabajar solo y no en equipo, he podido ahorrar mucho tiempo en la comunicación con diagramas e información del sistema, puesta en común, etc. En grandes organizaciones, el uso de herramientas de modelado *UML*, o la producción bajo el concepto *RUP*, se vuelve imprescindible, ya que en la práctica a veces el programador no tiene ni idea de para qué programa.

Se ha hecho uso del lenguaje de programación *PHP5*, apoyado en la *BBDD Mysql* para afrontar las funcionalidades y objetivos planteados en esta tesis. Ésta ha sido mi principal herramienta, y aunque he estado más cerca del *ExtremeProgramming* que del *RUP*, he incluido varios diagramas de análisis y diseño.

El desarrollo ha sido espejado en *GitHub* como central de versionado.

Avance establecido

Se parte del patrón MVC base de Jaisiel Delance. Se define el proyecto como la adición de dos aplicaciones, una el Front y otra el Back.

Empezamos por el Back, agregamos una capa de abstracción al sistema con el recurso Componentes. Se definen algunos básicos y se genera una interfaz para su creación.

En el Front, se introduce en el concepto de página, y por tanto el de módulo. Se usa una librería drag&drop para la creación de páginas en el back. Se adapta al sistema y se le agrega la posibilidad de incluir módulos dinámicos. Por último, y en base al concepto página, se desarrolla un sistema de caché bajo demanda.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

La mayoría de las pequeñas y medianas empresa productoras de software suelen producir sistemas y proyectos de pequeño y mediano alcance.

En el mercado existen soluciones de software para casi todo. En concreto para la gestión de contenidos hay algunos muy extendidos, hasta tal punto que muchos de los sitios web por los que solemos navegar esconden detrás un *CMS*. Existen algunos especializados en el sector editorial, otros en sector audiovisual, también hay *CMS*'s para *bloggers*, o para anuncios de clasificados, etc.

No hay duda de que los *CMS*'s son una tendencia al alza. En el caso de los *opensource*, con las nuevas versiones cada vez son más modulares, se pueden agregar o eliminar funcionalidades, se forman comunidades de gente que colabora, agregando fragmentos o tomando otros.

Hoy en día cualquier *PYME* sabe que necesita organizar su información en torno a sistemas informáticos fiables, y además con el crecimiento de la web, también quieren poder gestionar esa información *online*.

Entiendo, que estos *CMS* existentes en el mercado, son por lejos unos gigantes, en cuanto a volumen de código empleado en su desarrollo, horas/hombre invertidas, comunidad de apoyo y sobre todo sitios reales que los usan.

Estos CMS poseen un gran nivel de personalización, se ajustan a casi todo. Además, tiene una interfaz muy bien definida que permite agregar módulos, componentes o *plugins*, programados a medida según las necesidades.

Desde el punto de vista del desarrollador, usar estas herramientas para la construcción de sistemas en la web es sin duda algo muy positivo y ventajoso.

Desde un punto de vista empresarial de los **productores de software**, la perspectiva cambia. Adoptar este tipo de herramientas como la base de sus desarrollos a medio y largo plazo puede convertirse en un lastre, son muchos los riesgos y **problemas** que podrían encontrar:

1. Los CMS han evolucionado tanto que son realmente completos, y por tanto **complejos**. Es posible que para un determinado proyecto web, se ajusten bien, pero para sistemas más grandes posiblemente sea necesario modificar ciertos comportamientos, que con un software de tal magnitud serían muy altos los recursos a invertir necesarios para su modificación.
2. El core de la aplicación que pudieran desarrollar, estaría atado a terceros, y esto en productos de software es una incongruencia muy limitante. Tendrían que actualizar sus sistemas según el calendario de otra organización, y la obsolescencia que pudiera imponer el tercero en su producto, afectaría al que lo consume, el productor de software. En definitiva, su **versionado** sería **esclavo**, o quedarían atados a una versión obsoleta.
3. Son aplicaciones que consumen muchos recursos de los servidores que las alojan, por lo tanto la implementación en serie de éstas supondría **altos niveles de consumo**, aunque éstas no contuvieran un modelo de negocio complejo o con un número alto de requerimientos.

4. **Son focos de ataques informáticos.** Al ser código libre, con estudio y esfuerzo, cualquiera puede conocer sus misterios, es por ello que diferentes individuos u organizaciones con diferentes propósitos consiguen hackear sistemáticamente estos sistemas, obligando a la organización productora del software a sacar parches de seguridad y nuevas versiones. No todas las nuevas versiones incluyen mejoras, sino también bug's resueltos.

Preguntas

¿Qué software debe usar como base un productor de software de sistemas web en sus proyectos?

¿Es correcto hacer un sistema a medida en lugar de tomar uno del mercado?

¿Un único sistema puede ser la base de casi todos los proyectos web de un productor de software?

Las respuestas a estas preguntas en ningún caso podrán ser categóricas. Todo dependerá de quien la responda y de cuáles sean sus experiencias y necesidades. En los anexos, se podrán consultar las respuestas de algunos miembros de la comunidad de desarrollo a estas preguntas.

Objetivos

En este trabajo, intentaré probar si un software hecho a medida hecho desde abajo, puede tener ventajas arquitectónicas como las tienen los CMS's más destacados del mercado.

Presentar una estructura mayormente válida para la creación de un software propio que sirva como base a una empresa productora de software a modo de gestor de contenidos.

Generar una interfaz de carga de páginas visual en el backend diferente a las del mercado, tratándola de hacer más natural e intuitiva. Configurando la página y el contenido que puede alojar, en lugar de que contenido va en qué página.

Implementar un sistema de caché bajo demanda. Y que no sea caché por expiración de tiempo.

Usar componentes en el back, para encapsular funcionalidades.

Usar módulos en el front, para pintar las páginas.

Mantener esquema MVC en la aplicación.

Definir interfaces para la creación de módulos y componentes.

Límites

El número de funcionalidades de la aplicación estará muy limitado, la cantidad de componentes o módulos implementados será pequeña, así como funciones de la aplicación.

El programa informático resultado de este estudio, no será muy complejo por cuestiones de recursos, pero trabajado, con el tiempo, puede llegar a ser una base de software o marco de trabajo para pequeños y medianos productores de software.

Ventajas

Cuando trato de equiparar las ventajas de un software de producción 'local' con la de los grandes del mercado, me refiero a las principales ventajas de su arquitectura:

1. Suelen ser orientados a **componentes**, encapsulando funcionalidades, y con posibilidad de agregarlos o eliminarlos al gusto. Y permite el trabajo en equipos.
2. Cuentan con sistemas de **caché**, todos ellos caché de expiración por tiempo.
3. Son altamente **escalables** y configurables.
4. Implementan el patrón **MVC** en su concepción.
5. Usan gestores de estilos como **themes o plantillas**.

Viabilidad

Creo que si se consiguen reunir las principales ventajas de su arquitectura comunes a los CMS's existentes, en un software de producción propia, que pueda servir como base para casi todos los proyectos que pueda tener un pequeño y mediano productor de software, se habrá conseguido una herramienta poderosa y maleable para hacer crecer su organización.

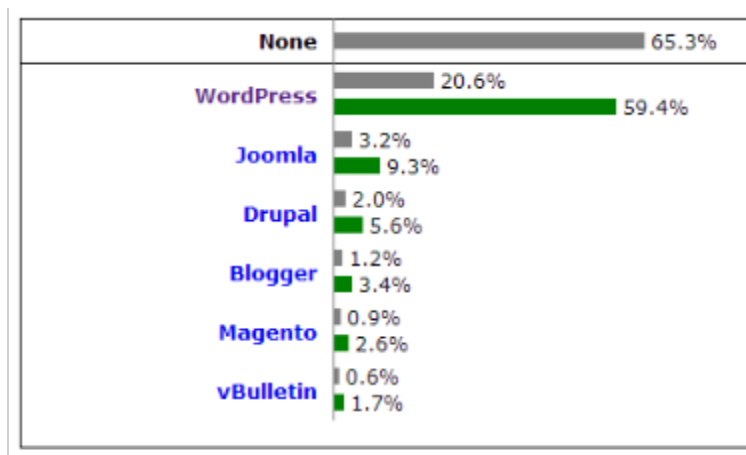
Datos concretos

Penetración de los CMS's

Según la *W3techs*¹ el 65.3% de entre los 10 millones de sitios analizados no usa ningún gestor de contenidos.

Por tanto, si decimos que los CMS conforman en torno al 35% de todas las web mundiales, inmediatamente observamos la relevancia de estas herramientas.

En el siguiente gráfico vemos como Wordpress, Joomla y Drupal lideran el mercado, hasta tal punto que la relevancia a nivel mundial solo de Wordpress es apabullante. La medición verde se refiere al total dentro de los gestores de contenidos, pero la medición gris corresponde al total de las webs del mundo, así es pues decir que Wordpress acapara más de un 20% de las webs del mundo es mucho decir.



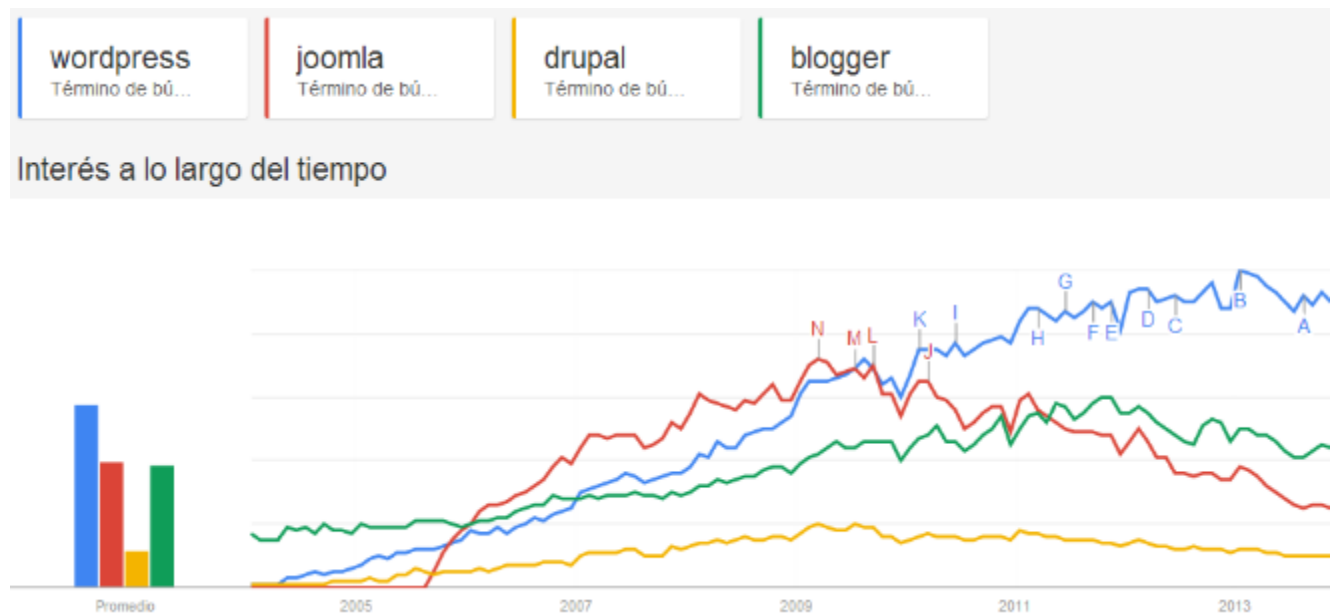
Fuente: http://w3techs.com/technologies/overview/content_management/all

Listado de los 6 CMS con más cuota de mercado.

¹ El sitio w3techs.com provee información acerca del uso de varios tipos de tecnología en la web. Aunque es una empresa privada (como casi todas las que mueven internet), promete información veraz y dice no tener ningún tipo de afiliación con los proveedores tecnológicos que aparecen en sus informes. Investigan la tecnología de los sitios web, no páginas concretas. Para hacer el estudio han incluido las 10 millones de páginas más vistas. A diferencia de Alexa (una compañía adquirida por el gigante Amazon, que lidera la medición de tráfico, visitas y estadísticas web a nivel mundial), W3techs no considera, a los subdominios como sitios diferentes, sino que los considera como el mismo sitio.

Evolución del interés en los CMSs

Otros datos que pueden ser analizados, demuestran una vez más la magnitud e inclusión de estos programas en la sociedad actual. Son los datos provenientes de la evolución en **términos de búsqueda** de palabras claves relacionadas con el sector, se usará *Google Trend*² para llegar a estas conclusiones mostradas en el siguiente gráfico.:



A pesar de que Joomla muestra una tendencia a la baja, vemos que todos tienen valores de crecimiento positivos desde el comienzo de la gráfica en el año 2005. Y es que los CMS son un pilar de la web moderna.

Problemas de seguridad en los CMS's

Haciendo una búsqueda por la cadena "hack joomla" en *google.es* se obtienen más de tres millones de resultados. Aparecen cientos de post, de gente que ha encontrado vulnerabilidades de seguridad, y explica como puedes reproducirlas.

² Google Trend es una herramienta de Google Inc. que provee información sobre las diferentes tendencias y su evolución de los diferentes términos de búsqueda, usados en su buscador, gracias al cual obtienen la información. Entendemos a Google como el buscador de referencia para nuestro estudio.

Haciendo la búsqueda con la cadena “parche seguridad joomla” obtengo casi 30.000 resultados. Se pueden ver cientos de enlaces a páginas que ofrecen parches de seguridad para las diferentes versiones.

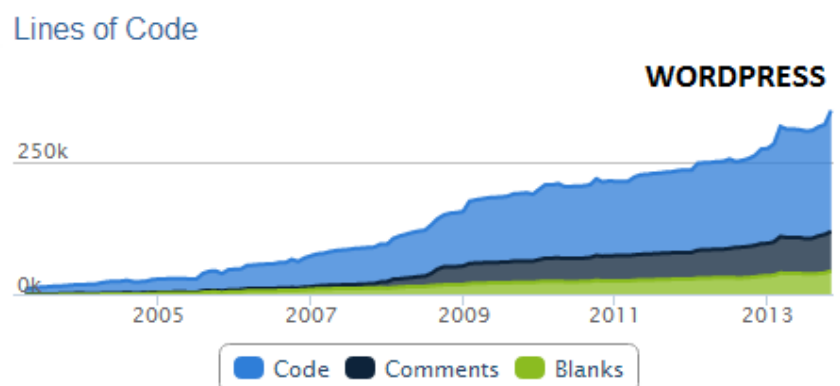
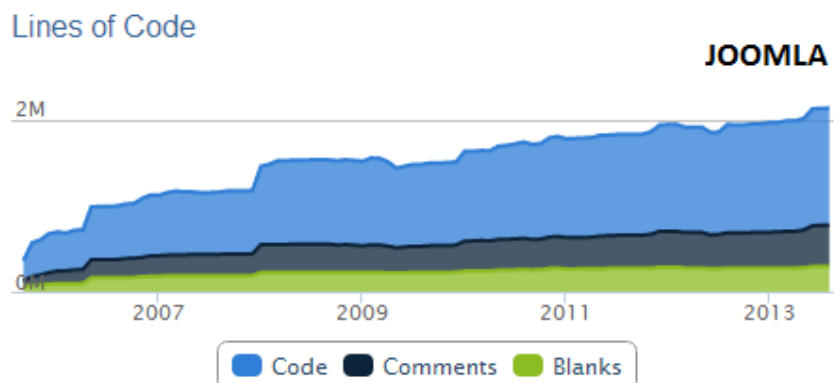
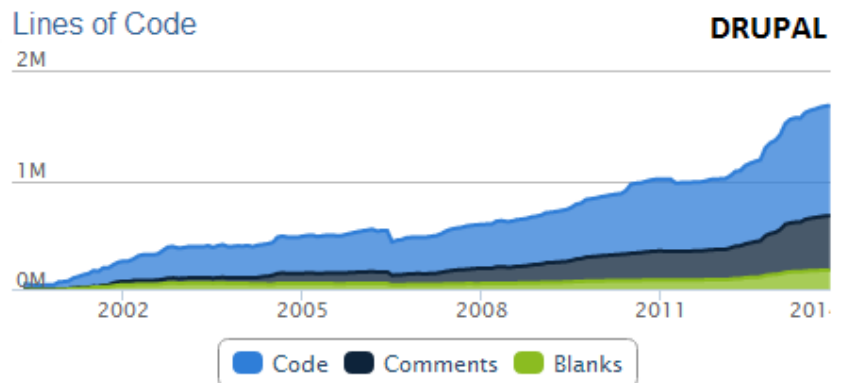
Rendimiento de los CMS's

No hay duda de que los CMS's han crecido exageradamente en código, a veces pueden volverse monstruosos para según que proyectos.

Usando la herramienta provista por Ohloh.net³ podemos llegar a conclusiones muy interesantes.

Se puede observar como todos los CMS's han aumentado sustancialmente el número de líneas en los últimos años, razón más para entender que son aplicaciones cada vez más completas, y por tanto pesadas.

Y esto es sin contar a la comunidad que aporta y



³ Ohloh.net es un sitio web que permite comparar proyectos Open Source. Usa trackers para rastrear los repositorios públicos donde éstos se alojan para así obtener información como el número de contribuciones a cada proyecto, líneas de código, etc.

colabora, estos gráficos se han realizado sólo teniendo en cuenta a la empresa que produce el core del CMS.

Caché

El principal problema que he detectado en los CMS's del mercado, es que todos usan una caché basada en el tiempo. En sitios de noticias donde rápidamente actualizan la información, y en muchas ocasiones los textos se editan en caliente, la caché puede ser un problema, que provoque que los usuarios que visitan la web vean contenido desactualizado.

Es cierto que los CMS's permiten vaciar la caché, pero es un proceso manual que depende del entrenamiento del usuario, y por tanto muchas veces genera problemas.

CAPÍTULO II

ANTECEDENTES

Un sistema de gestión de contenidos (Content Managment Systems) es un software que se utilizar para el almacenamiento, edición y publicación de información, normalmente son aplicaciones web, aunque pudiera usarse en una *intranet* igualmente.

Permite que multitud de personas puedan acceder a comunicarse a través de Internet, sin necesidad de que tengan conocimientos técnicos, como lenguajes de programación.

Los CMS's aparecen a finales de los 90 con la popularización de Internet, cubriendo las nuevas necesidades que se iban generando. Los grandes editores del papel, empezaban a sacar su versión web, y rápidamente se dieron cuenta que tendrían que desarrollar sistemas que les permitiesen actualizar continuamente sus contenidos.

El boom llegó con Phpnuke⁴, cada vez más gente tenía conexión a internet, salían los planes de hosting baratos y todos querían comunicar y aparecer en la red.

Que fueran los editores de diarios y revistas los precursores de los primeros CMS's, no impide que, su uso se haya extendido a todos los sectores, siendo hoy día estas herramientas un pilar del Internet actual.

Actualmente en el mundo hacen uso de ellos todo tipo de empresas, individuos, universidades, organismos oficiales y ONG's.

⁴ Phpnuke fue el más popular de los primeros CMS's, también el primero que generó una gran comunidad. Más tarde llegarían otros como Mambo del cual surgió Joomla.



Logos de CMS's populares.

Existen cientos de tipos, muchos gratuitos (los más populares), y también muchos privados. Existen CMS's específicos para cada sector, CMS's estandarizados, CMS's a medida. Los hay con diferentes tecnologías, para diferentes usos.

BASES TEÓRICAS

Un sistema de gestión de contenidos (Content Management Systems) es un software que se utiliza para el almacenamiento, edición y publicación de información, normalmente son aplicaciones web, aunque pudiera usarse en una intranet igualmente.

Funcionalidades básicas

Basándonos en el esquema planteado por James Robertson⁵ podemos decir que cualquier CMS debería poder subdividirse en estas cuatro funcionalidades básicas:

1. **Creación:** esto resumiría todos los formularios de carga de información y los procesos que los hacen funcionar. Ya sea para la inclusión de una noticia, video, galería, etc.
2. **Gestión:** toda la información creada debe poder gestionarse, es decir, editarse, eliminarse, modificarse. Además tiene que ser posible modificar algunos elementos estructurales, textos o páginas que aparecen en la web.
3. **Publicación:** debe ser posible publicar o despublicar elementos. Algunos CMS's lo hacen por fechas, por ediciones, u otros factores. La información que no se publica se archiva y está disponible cuando se requiera.
4. **Presentación:** esto hace referencia a todo el marco y estructura de la web, donde se aloja el contenido. Se refiere a los menús, a los enlaces, a las publicidades, etc. Puede ser que un CMS's concreto tenga una versión para móviles y otra de escritorio.

⁵ James Robertson es el fundador y Director Ejecutivo de *Step Two Designs*, una empresa Australiana que tiene como producto un CMS especializado en *intranets*. Se define como especialistas de los gestores de contenido, es fundador miembro de la asociación profesional de gestores de contenidos (*CM Professionals association*) y ha ayudado a muchas organizaciones a elegir el CMS adecuado a sus necesidades.
<http://www.steptwo.com.au>

Clasificaciones

Si es por **lenguaje** utilizado, la mayoría están escritos en *Php*, aunque hay muchos en *Asp*, *Java*, *.net*, *Ruby*, *Perl*, *python*.

Por su **licencia** podríamos resumirlos en *OpenSource* y *privados*.

Por su **uso** tenemos CMS's pensados para *blog's*, foros, publicaciones, tiendas, redes sociales, etc.

Frameworks y CMS's

Para empezar, es muy importante tener claras las diferencias entre estos dos conceptos.

Un *Framework* proporciona un marco de trabajo sobre el que construir tu aplicación. Tal como una caja de herramientas sirve al mecánico, los *frameworks* ayudan a los programadores en la creación de aplicaciones, otorgándoles en su entorno de desarrollo librerías, funciones y otras soluciones.

Los frameworks te dan cosas hechas, pero no dan una base de la web. Hoy en día, hay *frameworks* que se acercan muchos a algunos CMS's, esto es causado por una evolución de los mismos, aunque conceptualmente sean muy distintos.

Un CMS's es más que un *framework* y seguramente su construcción está realizada con uno a medida. Se podría decir que, un CMS es un framework aplicado.

Existen determinados CMS's, que pueden ser usados como frameworks porque tienen interfaces bien definidas que permiten a los programadores realizar sus aplicaciones como extensiones de estos CMS.

Persistencia

Con la evolución de Internet, han surgido diferentes sitios webs que ofrecen *sitebuilders*, y de manera muy rápida y sencilla permiten generar webs sobre plantillas con estilos para contenidos estáticos. Podría decirse excepcionalmente que son CMS, aunque no usen BBDD y aunque sea muy simple la gestión que hacen del contenido.

Habitualmente, los CMS siempre han ido asociados a una base de datos para poder persistir la información, además de clasificarla y tener libre acceso a ella. Los motores de BBDD son tantos como ofrece el mercado, y según el tipo de CMS que los emplea.

Además, en determinadas ocasiones para persistir o compartir determinadas configuraciones, pueden hacer uso de archivos de configuración. El manejo de imágenes y otro tipo de recurso visual normalmente se aloja en el directorio web.

Caché

La principal razón de la caché es ahorrar procesamiento al servidor. Hay que entender, que cada vez que una página es llamada, multitud de procesos se desencadenan, la ejecución de los script del servidor, con sus variables, funciones, llamadas, etc; la llamada a los recursos físicos del sistema de archivos; los accesos a la BBDD, alocaiones de memoria, etc.

El principio de la caché es bien sencillo. Toma el resultado que devuelve el conjunto de procesos que hicieron ejecutar ese requerimiento, y lo guarda en un archivo de texto plano en el sistema de archivos. En la próxima llamada que tiene al mismo requerimiento, reutiliza el resultado devuelto en la llamada anterior, ahorrándose así, todo este procesamiento.

Plantillas o themes

Si algo en común tienen todos los CMS's son las plantillas o *themes*.

Una plantilla hace referencia al continente donde se va a materializar el contenido y en conjunto conforman la interfaz o respuesta que se genera al usuario tras su requerimiento. Una plantilla permite alojar el contenido que pueda generar el CMS.

A las plantillas está asociada la estructura. Así por ejemplo, cambiando una plantilla estaríamos cambiando la organización visual del contenido, así como además los estilos del mismo.

Los *themes* en cambio, son más básicos, no afectan a la estructura sino que sólo a los estilos. En el caso de los *themes*, la organización visual del contenido permanece constante.

CAPÍTULO III

METODOLOGÍA APLICADA

Hipótesis

Sería correcto que un productor de software de sistemas web pudiera elaborar un software a medida, en lugar de tomar uno del mercado. Un único sistema web que fuera la base de casi todos sus proyectos. Un sistema que de una manera visual permitiera organizar su contenido, usando un sistema de caché que no produzca errores de actualización y con un sistema de componentes que permita el trabajo en equipo.

Tipo de estudio

El tipo de estudio es propositivo. Trataré de formular una propuesta concreta al problema planteado, en la realización de una arquitectura básica del CMS. El estudio se presenta en detalle en el CAPÍTULO IV.

CAPÍTULO IV

CREACIÓN DEL CMS

Introducción

Ser

WYSIWYG

Drag&drop. La idea de tener combinar la potencia de los grandes CMS con la de los grandes editores HTML, es sin duda un gran reto. WYSIWYG por sus siglas en inglés lo que ves es lo que tienes, *what you see is what you get*, es mucho más que un acrónimo, supone un objetivo de performance para el usuario muy importante. Permite que los usuarios se adapten de manera intuitiva y segura con el sistema.

Hay programas WYSIWYG para hacer páginas web's, algunos muy conocidos como el *Dreamweaver de Adobe*. También los hay a modo de editores modulares, que sirven de complementos a los desarrolladores para mejorar la entrada de datos del usuario.

Al emplear el término WYSIWYG en este proyecto, trataré de abarcarlo desde un punto de vista integral, refiriéndome tanto a las entradas de datos aislados del usuario (editores modulares), como al de la maquetación de los contenido dinámicos, estáticos y estructurales, que mediante una facilidad arrastre y suelte, *drag&drop*, permitirá diagramar las páginas web de una manera rápida e intuitiva.

Desde un punto de vista social, creo que podría acercar un poco más este tipo de herramientas a la gente, si se sigue trabajando en este sentido.

Estandarización

El sistema puede ser implementado en servidores Unix, MacOS y Windows. Puede correr sobre Apache, IIS y cualquier otro servidor web que sea intérprete de php.

Deben tenerse en cuenta las reescrituras de *URL*, ya que para conseguir url's amigables la funcionalidad la proporciona el servidor web, que atiende el requerimiento. En el caso de Apache, será necesario configurar un *.htaccess* y en el caso de IIS, un *web.config*.

La aplicación se presenta con ambas configuraciones, ya que son compatibles y coexistentes.

CAPÍTULO V

ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

FALTANTES

Sería necesario aplicar el patrón singleton en la creación de instancias de los módulos. Devolviendo el anterior estado, así se mejoraría el rendimiento cuando haya muchos módulos del mismo componente.

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFÍA

Título: Sistemas de gestión de contenidos en la gestión del conocimiento

Autores: Jesús Salinas, Bárbara de Benito, Victoria Marín, Juan Moreno, María Eugenia Morales

Edita: Universitat Illes Balears

<http://gte.uib.es/pape/gte/sites/gte.uib.es.pape.gte/files/Herramientas%20y%20sistemas%20de%20gestion%20del%20conocimiento%20para%20el%20desarrollo%20de%20metodologias.pdf>

<http://bid.ub.edu/14monto2.htm>

Introducción a los Sistemas de Gestión de Contenidos: CMS

<http://www.slideshare.net/santillan/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms>

Sistemas de gestión de contenidos

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

Gestión de contenidos

Título: Artículos sobre los gestores de contenidos

Autor: James Robertson

<http://www.steptwo.com.au/category/papers/content-management>

Imagen y curiosidad

Imagen obtenida de una página que muestra actualidad acerca del mundo de los CMS's, Superwebsitebuilders.com en un artículo acerca de unos premios de *Cmscritic.com* que se celebran para distintos CMS's. Premios que van desde el mejor CMS *OpenSource*, hasta el mejor CMS empresarial, *sitebuilders*, foros o los más reducidos

En 2013 la gente eligió como ganador a Wordpress en la categoría de los *OpenSource*.

Premios:

<http://www.cmscritic.com/critics-choice-cms-awards/>

<http://superbwebsitebuilders.com/critics-choice-cms-award/>

Imagen:

<http://superbwebsitebuilders.com/wp-content/uploads/2013/07/logos-CMS.jpg>

Auge de los sitios web basados en CMS sin base de datos

http://www.interdixit.com/esp/noticia.php?pag=auge_los_sitios_web_basados_cms_sin_bases_datos

Metáfora de escritorio

http://es.wikipedia.org/wiki/Met%C3%A1fora_de_escritorio

GLOSARIO

AXENOS

Estudio externo comparativo de Joomla y Wordpress

URL del estudio completo:

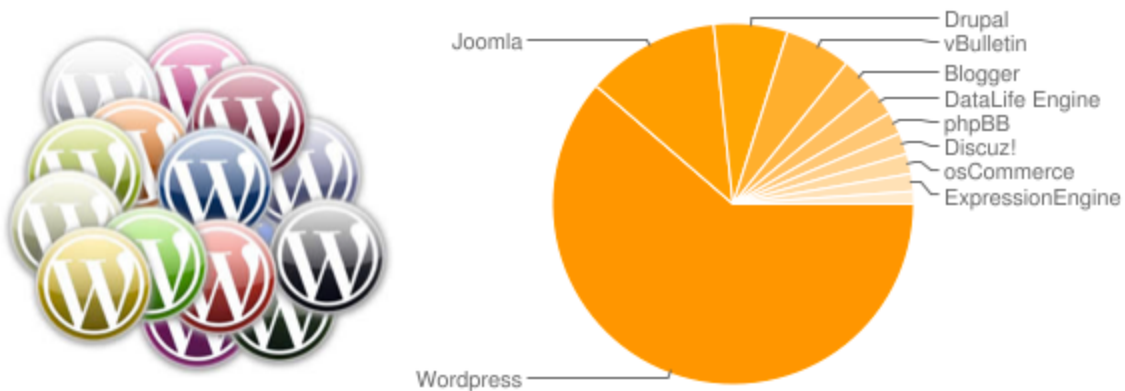
<http://redgiantdesign.co.za/rg-design-blog/wordpress-vs-joomla-2013-infographic.html>

URL de la infografía comparativa:

http://redgiantdesign.co.za/images/easyblog_images/42/WordPress-vs-Joomla.png

Realizado por el cofundador del prestigioso estudio de diseño *Red Giant Design*, ofrece información muy interesante sobre estos dos gigantes, acerca de sus datos, su evolución, etc.

En una serie de artículos va desentrañando los misterios, analizando su estructura y funcionalidades, y realizando diferentes comparativas.



Estudio interno tiempos de respuesta CMS´s existentes

Para realizar las pruebas se ha sido el mismo hardware, usando una página tipo muy básica, con una concurrencia de peticiones al servidor de 10, en un total de 1000 requerimientos al servidor.

La misma prueba se ha hecho sobre los siguientes CMS´s:

Joomla

```
C:\xampp\apache\bin> ab -n 1000 -c 10 http://localhost/um_joomla/
```

Tiempo total: 138.29 s

Nro. requerimientos por segundo: **7.23**

Joomla usando sistema integrado de caché

```
C:\xampp\apache\bin> ab -n 1000 -c 10 http://localhost/um_joomla/
```

Tiempo total: 122.29 s

Nro. requerimientos por segundo: **8.19**

Wordpress

```
C:\xampp\apache\bin> ab -n 1000 -c 10 http://localhost/um_wordpress/
```

Tiempo total: 124.77 s

Nro. requerimientos por segundo: **8.01**

Proveedores de estadísticas sobre el uso de los CMS

URL del proveedor estadístico:

<http://w3techs.com/>

URL de los datos sobre los CMS:




http://w3techs.com/technologies/overview/content_management/all

Quiénes son W3techs:

<http://w3techs.com/about>

Búsquedas en Internet

<https://www.google.es/search?q=hack+joomla>

hack joomla

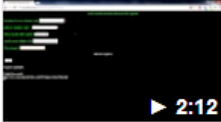
Web Imágenes Maps Shopping Videos Más ▾ Herramientas de búsqueda

Cerca de 3.520.000 resultados (0,37 segundos)

[HACKEAR PAGINAS JOOMLA \[Facilísimo\] - Portalnet.CL](#)
[www.portalnet.cl/...de.../194199-hackear-paginas-joomla-facilismo.html](#) ▾
04/05/2009 - 10 publicaciones - 6 autores
Ayudando a que la seccion sea realmente **hacking** Tuto echo por mi con mi exp Nota: Solo funciona con paginas con **joomla** 1.5 y realmente ...
Visitaste esta página el 20/11/13.

[Hackeando Joomla! - Hacking Ético](#)
[hacking-etico.com/2013/03/11/hackeandojoomla/](#) ▾
11/03/2013 - Hackeando **Joomla!** Artículo EDUCATIVO para comprometer un portal nuestro basado en **Joomla** 1.5.5. Se demuestra la fragilidad de este.

[¿Como hackear Joomla \(v. 1.6 / 1.7 / 2.5.0 / 2.5.2\)? - Taringa!](#)
[www.taringa.net/.../Como-hackear-Joomla-v-1-6-1-7-2-5-0-2-5-2.html](#) ▾
La vulnerabilidad en **Joomla**, relacionada con las versiones posteriores, que son la 1.6, la 1.7, la 2.5.0 y la 2.5.2 son las que presentan una vulnerab...
Visitaste esta página el 20/11/13.

[Hacking Joomla Website Tutorial \(JCE\) - YouTube](#)
 [www.youtube.com/watch?v=Yhok0g9UISY](#) ▾
29/08/2013 - Subido por x10LeBlanc
USE PROTECTION (CyberGhost VPN)*** Instructions: 1.Download all files and place them in JCE folder on ...

[Hacked Joomla! v. \[1.6.x\]\[1.7.x\]\[2.5.0-2.5.2\] - Escalación de Privilegios](#)
[calebbucker.blogspot.com/2012/.../hacked-joomla-v-16x17x250-252.ht...](#) ▾
09/07/2012 - Esta vulnerabilidad de **joomla** nos permite escalar privilegios durante el registro de un usuario nuevo, para la versiones 1.6.x/1.7.x no se han ...

[Security Checklist/You have been hacked or defaced - Joomla ...](#)
[docs.joomla.org/.../You_have_been_hacked_or_def...](#) ▾ Traducir esta página

<https://www.google.es/search?q=parche+seguridad+joomla>



[Web](#) [Imágenes](#) [Maps](#) [Shopping](#) [Videos](#) [Más ▼](#) [Herramientas de](#)

Cerca de 28.100 resultados (0,35 segundos)

[Aplicando el Parche de Seguridad para Joomla 1.5.26 - Web Empresa](#)



www.webempresa.com/.../1126-aplicando-el-parche-de-seguridad... ▼
De Alejo Luis - en 84 círculos de Google+
03/08/2013 - **Parche de Seguridad** para **Joomla** 1.5.26 Tal como hemos publicado en este blog recientemente, todas las versiones de **Joomla** anteriores a ...

[Si aún usas Joomla 1.5 lee esto URGENTEMENTE - Parche de ...](#)



www.gnumla.com > [Joomla!](#) > [Noticias](#) ▼
De Isidro Baquero - en 596 círculos de Google+
01/08/2013 - **Parche de seguridad** CRÍTICO para cualquier versión de **Joomla** 1.5.

[Necesito instalar un parche de Seguridad - Foros Joomla! Spanish ...](#)

www.joomlaspanish.org > ... > [Joomla! 1.5](#) > [Foro general \(1.5\)](#) ▼
19/07/2009 - 2 publicaciones - 2 autores
Hola necesito que me enseñen dónde y como instalar un **parche de seguridad** y a la vez que los usuarios registrados no puedan modificar ni ...

[Parche de Seguridad para Joomla 1.5.26](#)

www.joomlamantenimiento.com/.../109-parche-de-seguridad-para-joomla... ▼
10/09/2013 - El pasado mes de agosto se detecto una vulnerabilidad crítica en la versión obsoleta de **Joomla** 1.5.26 que permite subir archivos al servidor ...

[Aplicando el Parche de Seguridad para Joomla 1.5.26 - YouTube](#)



www.youtube.com/watch?v=ShwTgKm0R_g ▼
04/08/2013 - Subido por webempresa
Aplicando el **Parche de Seguridad** para **Joomla** 1.5.26. El Equipo de Producción de **Joomla** ha liberado un ...

[Novedades Joomla 2.5 - Actualización con 2 clics parches de ...](#)



www.youtube.com/watch?v=27TUZdUhlBg ▼
11/04/2012 - Subido por Emer Figueroa
Visita <http://CursoJoomlaEnLinea.com> Las novedades de **joomla** 2.5

Cuestionario 1

Juan Capristan

Desarrollador, freelance y empresario

CEO Creative Cocos

<http://creativecocos.com/>

¿Qué software debe usar como base un productor de software de sistemas web en sus proyectos?

Hay tantas opciones válidas como gustos. ¿Qué tipo de brocha debe usar un pintor? ¿qué juego de herramientas debe usar un mecánico?. Intentando centrar un poco la respuesta en mi caso particular: desarrollo la mayor parte del tiempo en un equipo Windows, aplicaciones Rails, PHP y ASP, así que mis herramientas básicas son:

- IIS, Apache y Webmin como servidores locales
- MySQL, PostgreSQL y Riak como servidores de BD
- Notepad++ y RubyMine como editor de código e IDE.
- Todos los navegadores disponibles, junto con plugins de desarrollo web para cada uno.
- Diferentes validadores, minimizadores de código, consolas, etc.

¿Es correcto hacer un sistema web a medida en lugar de tomar uno del mercado?

Depende del caso, de lo que se pretenda conseguir, del cliente, del presupuesto... ¿Es correcto construir una casa a medida en vez de comprar una prefabricada?. No es una pregunta que tenga una respuesta correcta genérica. En general es más cómodo reutilizar, adaptar, configurar, sin embargo no siempre es posible, ni deseable. Por otro lado, en general creo que es preferible una aplicación a medida bien fabricada que una aplicación genérica bien fabricada. Finalmente, hay distintos niveles de reutilización de código. Que

una aplicación se haya fabricado a medida no significa que hayamos tenido que reinventar la rueda.

¿Un único sistema web puede ser la base de casi todos los proyectos de un productor de software?

Sí. Sobre todo lo digo pensando en cualquiera de los frameworks de desarrollo web existentes, como Ruby on Rails o Yii.

Cuestionario 2

Carlos Peralta

Licenciado en informática

Team Leader Doopress

Administrador de Sistemas

<http://doopress.com/>

¿Qué software debe usar como base un productor de software de sistemas web en sus proyectos?

Entiendo se le denominan “stack” o pilas de sistemas (software). Sugiero la pila LAMP por su amplio uso y cantidad de sistemas realizadas con la misma: Linux, Apache, MySQL y PHP (la última P también puede ser Phython o Perl; pero PHP es el más popular)

¿Es correcto hacer un sistema web a medida en lugar de tomar uno del mercado?

Si el software del mercado es de código libre y tiene soporte por la comunidad; sin duda mejor seleccionar algo ya creado. Hay que analizar en este caso la popularidad del sistema para ver hasta cuando o la cantidad de personas que le darán soporte a mediano, largo plazo. Para el caso de los sistemas pagos creo que el factor más relevante es el costo. Por último el sistema a medida puede ser y a veces es la única forma de satisfacer los requerimientos; pero hay que tener en cuenta el mantenimiento de dicho sistema a mediano y largo plazo; por las personas o empresas que desarrollaron el mismo.

¿Un único sistema web puede ser la base de casi todos los proyectos web de un productor de software?

La diferencia entre “sistema Web” y “proyecto Web” no creo sea muy relevante o clara. De todas formas si se refiere a las tecnologías el hecho es que una pila (como LAMP) no siempre es la base de un proyecto Web. Solo puede haber Apache y PHP; y atrás otro SO, otro motor de base de datos, lenguajes compilados a nivel de SO (Ej. C++), etc.

Cuestionario 3

David Ramos

Desarrollador freelance

<http://davidramosweb.com/>

¿Qué software debe usar como base un productor de software de sistemas web en sus proyectos?

Dependiendo de la envergadura del proyecto y las necesidades del mismo, podemos optar por distintos tipos de frameworks para ayudarnos en su desarrollo.

Si nuestro proyecto no requiere algo muy específico podríamos usar diferentes CMS (joomla, drupal...) como base del mismo y de esta manera podríamos ahorrarnos bastante trabajo, ya que a parte de tener parte del código desarrollado, como los gestores de artículos, usuarios ACL... nos permitiría utilizar módulos, componentes y plugins desarrollados por terceros evitando tener que realizar parte de la programación. A demás estos CMS poseen una gran comunidad de desarrolladores y documentación a la cual podemos acudir en cualquier momento.

En cambio, si nuestro proyecto es de mayor envergadura podríamos recurrir a frameworks como Zend o Symfony los cuales nos permiten tener nuestro código estructurado y organizado.

¿Es correcto hacer un sistema a medida en lugar de tomar uno del mercado?

Desarrollar un sistema a medida tiene sus ventajas y desventajas.

En primer lugar el desarrollo íntegro de un sistema a medida puede resultar muy costoso, dependiendo de la envergadura del sistema, además de que se debe de desarrollar con un equipo de programadores experimentados, llevando unas pautas de trabajo muy bien

estructuradas, con una planificación y organización estricta y ofreciendo una escalabilidad del sistema.

La ventaja de desarrollar un sistema a medida, es que se puede optimizar mucho, ya que evitamos pasar por clases, funciones... que se incluyen en los sistemas del mercado orientados para un uso más extendido.

También evitamos ser víctimas de agujeros de seguridad de los sistemas del mercado los cuales una vez detectados pueden ser atacados rápidamente ya que poseen una gran comunidad que los utiliza, pero atención, por este mismo motivo debemos ser más cautelosos con la seguridad de nuestro código ya que no obtenemos beneficio de una comunidad que advierte y corrige estos problemas de seguridad.

Por el contrario no nos beneficiarnos de un código y documentación que a sido desarrollado por una comunidad de expertos y la cual esta en continuo desarrollo.

¿Un único sistema web puede ser la base de casi todos los proyectos web de un productor de software?

Esto depende mucho de los proyectos del productor de software. Generalmente podría decirse que sí, ya que normalmente un productor de software utilizará un sistema como base para sus proyectos ya que este le ahorrará parte del desarrollo de los mismo, y podrá ofrecer precios más competitivos a sus clientes.

Existirán proyectos específicos en los cuales no utilizara un sistema base, ya sea porque este no cumpla con las necesidades del proyecto o por exigencias de sus propios clientes. En todo caso un productor de software acumulara a lo largo del tiempo “scripts” de código que reutilizará en sus futuros proyectos.