

Universidad de las Ciencias Informáticas



**Selección, Adaptación e Implantación de Portales CMS libres u
open source. Portal NovaLNX.**

Trabajo para optar por el título de ingeniería en Informática

Autores:

Yunior Moiña Garcia

Noel Rondan Domínguez

Tutor:

Ing. Karel Verdecia

Ciudad de La Habana, Marzo de 2006.

Agradecimientos

A la Universidad de las Ciencias Informáticas por la formación brindada durante este tiempo.

A nuestros padres por el apoyo en cada momento y saber que siempre estuvieron ahí.

A la revolución y al comandante en jefe Fidel Castro Ruz.

A personas como la flaca Madelín, quien cargó con nosotros hasta el final y socios como marvin por sus aportes brindados.

A todas aquellas personas que de una forma u otra hicieron posible el desarrollo de este trabajo.

Dedicatoria

Dedico este trabajo a:

Mis padres: Sin ellos nada hubiese sido realidad.

Mis amigos: Los “mingos” y “netos”, Laynoll, Ronny, Rubén, Eduardo, Alejandro, los de Cienfuegos y los de la UCI también, mi banda, aunque no sepamos nada de música.

Anne, por soportarme tanto.

Mi familia: Mi hermana y mi sobrino, mis tios por ser únicos y mi abuela.

La flaca de oro: Madelín, por toda su ayuda y paciencia conmigo.

A todos los que confiaron y creyeron en mí y en que se haría realidad este sueño.

Yunior.

A mi familia en general por estar siempre a mi lado y apoyarme en todo.

A mis amigos todos, por compartir conmigo los buenos y los malos momentos.

A mi compañero de tesis por soportarme.

A Alain Fernández del Toro, por ser un guía en mi formación como informático.

A mis hermanos aquí en la UCI.

A mis compañeros de la UCLV.

A Fabio Omar Díaz Silva, el padre que no tuve y mi ángel guardián aquí en la UCI.

A todos ellos gracias.

Noel.I

Declaración de Autoría.

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los 6 días del mes de abril de 2006.

Yunior Moaña García

Noel Rondan Domínguez

Ing. Karel Verdecia

Resumen

Este trabajo integra los elementos necesarios para la selección, adaptación e implantación de Portales CMS libres u open source.

Se presenta a la gestión de contenidos como alternativa actual por parte de empresas a muchos de los problemas existentes en las mismas. Los sistemas manejadores de contenidos son detalladamente estudiados tomando como base la estructura básica de los mismos. Se trata con este trabajo de dar pie al uso de sistemas manejadores de contenidos en su variante de Portal Web bajo licencias libres en cualquier tipo de solución inmediata a proyectos de producción, organizaciones y comunidades, que requieran de las funcionalidades, fácil manejo y rápida adaptación que ofrecen los mismos. La formulación de métodos para la selección viene asociada a pasos y criterios a seguir de acuerdo a comparaciones dada la formulación de requerimientos.

Aquí se exponen de manera clara toda una gama de alternativas tecnológicas libres u open source presentes en la implementación de estas herramientas e importantes a tener en cuenta para la modificación e implantación de las mismas.

En este trabajo se llevan a cabo, y se documentan, los procesos de selección, adaptación e implantación de un Portal CMS para la distribución de Linux NovaLNX.

INTRODUCCIÓN.....1

CAPITULO 1: EL MANEJO DE LOS CONTENIDOS.....5

1.1 INTRODUCCIÓN.....5
1.2 CONTENIDOS.....5
1.3 JUSTIFICACIÓN DE LA UTILIZACIÓN DE LOS SISTEMAS GESTORES DE CONTENIDOS.....7
1.4 GESTIÓN DE CONTENIDOS.....10
1.4.1 EL CICLO DE VIDA DE LOS CONTENIDOS.....12
1.4.1.1 AGREGACIÓN.....13
1.4.1.2 PRODUCCIÓN.....14
1.4.1.3 ENTREGA.....15
1.4.1.4 ANÁLISIS.....16
1.5 TIPOLOGÍA DE SISTEMAS DE GESTIÓN.....17

CAPÍTULO 2. ESTRUCTURA BÁSICA DE UN PORTAL CMS.....22

2.1 INTRODUCCIÓN.....22
2.2 SUBSISTEMA DE COLECCIÓN.....22
2.3 SUBSISTEMA DE GESTIÓN.....26
2.4 SUBSISTEMA DE PUBLICACIÓN DE CONTENIDOS.....29
2.5 CMS LIBRES Y PROPIETARIOS.....30
2.5.1 ¿LIBRE U OPEN SOURCE?.....32
2.5.1.1 LICENCIAS.....34

CAPÍTULO 3: SELECCIÓN DE PORTALES CMS.....37

3.1 INTRODUCCIÓN.....37
3.2 PASOS EN UNA SELECCIÓN.....38
3.3 DESCRIPCIÓN DE LOS CRITERIOS DE SELECCIÓN.....45
3.3.1 CRITERIOS DE NEGOCIO.....46
3.3.2 CRITERIOS GENERALES.....46
3.3.3 CRITERIOS BASADOS EN EL SUBSISTEMA DE COLECCIÓN DE CONTENIDOS.....50
3.3.4 CRITERIOS BASADOS EN EL SUBSISTEMA DE GESTIÓN DE CONTENIDOS.....53
3.3.5 CRITERIOS RELACIONADOS CON EL SUBSISTEMA DE PUBLICACIÓN DE CONTENIDOS.....57
3.4 SISTEMAS DE REFERENCIA.....59

CAPÍTULO 4: TECNOLOGÍAS LIBRES EN LA MODIFICACIÓN E IMPLANTACIÓN.....61

4.1 INTRODUCCIÓN.....61
4.2 PLATAFORMA LAMP.....63
4.2.1 SISTEMA OPERATIVO GNU/LINUX.....64
4.2.2 SERVIDOR WEB APACHE.....65
4.2.3 SISTEMAS DE BASES DE DATOS.....67
4.2.3.1 MYSQL.....67

| | | |
|--|---|------------|
| 4.2.3.2 | POSTGRESQL..... | 68 |
| 4.2.3.3 | INTERBASE..... | 69 |
| 4.2.4 | LENGUAJES DE PROGRAMACIÓN..... | 69 |
| 4.2.4.1 | PHP..... | 70 |
| 4.2.4.2 | PERL..... | 71 |
| 4.2.4.3 | PYTHON..... | 72 |
| 4.2.5 | WAMP..... | 73 |
| 4.2.6 | MAMP..... | 74 |
| 4.3 | PLATAFORMA JAVA..... | 74 |
| 4.3.1 | JSPs, SERVLETS..... | 74 |
| 4.3.2 | MOTORES DE PLANTILLAS..... | 75 |
| 4.3.3 | APACHE TOMCAT..... | 76 |
| 4.3.4 | SERVIDORES DE APLICACIONES..... | 77 |
| 4.4 | ESTÁNDARES. LA W3C..... | 77 |
| 4.4.1 | WAI..... | 78 |
| 4.4.2 | JCP..... | 80 |
| 4.4.2.1 | PORTLET..... | 80 |
| 4.4.2.2 | JSR..... | 80 |
| CAPÍTULO 5. PORTAL NOVALNX..... | | 82 |
| | | |
| 5.1 | INTRODUCCIÓN..... | 82 |
| 5.2 | NECESIDAD DE NOVALNX..... | 82 |
| 5.2.1 | CONTENIDOS A GESTIONAR..... | 83 |
| 5.3 | DEFINICIÓN DE REQUERIMIENTOS..... | 84 |
| 5.4 | SELECCIÓN DE JETNUKE..... | 86 |
| 5.4.1 | CARACTERÍSTICAS DE JETNUKE..... | 88 |
| 5.5 | ADAPTACIÓN E IMPLANTACIÓN DEL PORTAL NOVALNX..... | 94 |
| CONCLUSIONES. | | 97 |
| | | |
| RECOMENDACIONES..... | | 98 |
| | | |
| BIBLIOGRAFÍA CONSULTADA..... | | 99 |
| | | |
| BIBLIOGRAFÍA..... | | 100 |

Introducción

El desarrollo de Internet como medio de comunicación y fuente de conocimientos, el rápido cambio de las nuevas tecnologías y la acelerada inserción de toda nuestra sociedad de una manera u otra en esta dinámica forma de concebir la comunicación, han traído consigo que toda una gama de empresas, organizaciones y comunidades vean como solución a determinados problemas de comunicación e información el uso de sistemas en entorno Web.

Los sistemas de gestión de contenidos (Content Management Systems o CMS) son softwares que se utilizan 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. (CUERDA, 2004)

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 considera 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. (CUERDA, 2004)

La gran heterogeneidad de las empresas, universidades o sistemas corporativos que suelen usar Windows NT o entornos Unix, con productos de varios tipos de fabricantes y manejan los diferentes tipos de contenidos que pudieran ser noticias, artículos, reportajes, cursos de formación, conferencias, informes, imágenes, música, animaciones, videos, entre otros; fortalecen, sin duda, la posición al uso de portales gestores de contenidos bajo licencias libres, por sus características de, en su mayoría, ser multiplataformas y orientados a dar soluciones rápidas y eficientes con un fácil manejo y una posterior gama de posibilidades dadas sus flexibilidades legales.

Siguiendo la estrategia de convertir a nuestra universidad en un fructífero parque tecnológico, ha nacido una gran cantidad de proyectos de producción que recorren casi la totalidad de las ramas existentes del desarrollo de software. También existen organizaciones de masas que aglutinan a la mayoría de las personas que ya son aproximadamente 10000 entre estudiantes, trabajadores y profesores. La creación de especies de comunidades oficiales o el simple interés de dar o recibir información,

prestar colaboración y disfrutar de la comunicación, exige la creación de diversos sistemas que gestionen gran cantidad y diversidad de contenidos.

El uso por parte de algunas comunidades y entidades de Sistemas Manejadores de Contenidos bajo licencias libres para dar soluciones a problemas de comunicación, búsqueda de información, control de usuarios, gestión de documentos entre otros, ha dejado como tarea que se realice un profundo estudio dada la diversidad de funcionalidades y distintos modelos de gestión para los que estos sistemas suelen estar diseñados.

Un ejemplo claro lo tenemos en el proyecto de producción que se dedica al desarrollo de la distribución de Linux "*NovaLNX*". Se adoptó por estrategia la posición de implantar un Portal Web ante la escasez de manejo de información y contenidos frente a una comunidad en ascenso y más con el uso de dicha distribución en algunos laboratorios de docencia de nuestra universidad.

A medida que el equipo de desarrollo asignado fue recopilando las necesidades que justifican esta decisión y algunos requerimientos no funcionales a tener en cuenta, se fue convergiendo al uso de un Portal CMS bajo licencia libre. Los principales problemas que se fueron sucediendo provenían del desconocimiento del tema de una adecuada gestión a los sistemas manejadores de contenido. Existían y existen además, gran diversidad de portales que cumplen, a primera vista, con las funcionalidades necesarias por lo que se requería se compararan bajo algunos criterios adecuados.

Otros factores considerados que dificultaban la toma de decisiones fueron la diversidad de plataformas y variantes de desarrollo en que estos sistemas estaban realizados. Estos factores son de extrema importancia teniendo en cuenta que serían objeto de modificación y adaptación a las necesidades y tendrían que servir de base a futuras aplicaciones y nuevas funcionalidades. La documentación y la información sobre cada arquitectura a escoger resultó estar en diferentes idiomas, por tanto dificultaba de cierta forma el trabajo en momentos de decisión.

Otro elemento importante a tener en cuenta es el tema legal. Las licencias de uso y modificación, no brindan todas las mismas flexibilidades y restricciones, convirtiéndose en otro punto de estudio.

Por tanto, se identificó como problema a resolver en este trabajo, la necesidad de un material que brinde a desarrolladores inexpertos de Sistemas Manejadores de

Contenidos los elementos necesarios para tomar decisiones a la hora de implantar y desarrollar sobre plataformas libres algún sistema de este tipo.

El trabajo está orientado principalmente a desarrolladores inexpertos o poco conocedores en temas de la gestión de contenidos. En la actualidad, existe un proceso de migración en nuestro país hacia el uso del software libre, por tanto se considera también que este producto pueda jugar un rol importante en manos de directivos y jefes de proyectos vinculados a esta migración.

El trabajo tiene como objetivos generales:

1. Elaborar un material de consulta sobre Sistemas Manejadores de Contenidos adaptado a la UCI con los elementos necesarios para tomar decisiones a la hora de implantar y desarrollar sobre plataformas libres.
2. Modificación e implantación de un Portal Web que responda a la gestión de contenidos de la distribución NovaLNX.

De estos objetivos se generan otros específicos que serían:

1. Realizar un estudio de la gestión de contenidos en portales Web bajo licencias libres.
2. Adaptar a las condiciones de la UCI y de la distribución NovaLNX los resultados de dicho estudio.
3. Documentar la selección de una herramienta de gestión de contenidos para NovaLNX, así como los cambios realizados una vez hecha la adaptación.

Para dar cumplimiento a los objetivos se plantean un grupo de tareas:

1. Estudiar las variantes de gestión de contenidos así como todos los procesos por los que debe transitar de acuerdo a los requerimientos y tipos de contenidos.
2. Analizar las características, estructuras y funcionalidades de los Portales CMS.
3. Investigar sobre los tipos de licencias de uso y modificación, libres y open source bajo las cuales estos sistemas pueden encontrarse.
4. Formular criterios de selección acordes a determinadas necesidades y atendiendo a las estructuras básicas de estos sistemas.
5. Analizar las plataformas y herramientas de desarrollo para Portales CMS teniendo en cuenta como resultados sus principales funcionalidades y

- rendimientos, conformando un análisis de las principales ventajas y desventajas de cada decisión.
6. Seleccionar, modificar, implantar y poner en marcha un Sistema de este tipo para la distribución de Linux “NovaLNX” como resultado de toda una investigación.

El trabajo lo conforman 5 capítulos de contenido además de algunos anexos:

Capítulo 1 El manejo de los Contenidos: En el primer capítulo se introduce a los Sistemas Gestores o Manejadores de Contenidos. Los términos asociados a estas prácticas, la justificación de la necesidad de estos sistemas, el ciclo de vida de los contenidos y la tipología actual son los principales temas tratados.

Capítulo 2 Estructura básica de un Portal CMS: En este capítulo se especifica detalladamente una estructura básica de un Portal CMS. Se logra mediante la separación de sus partes por subsistemas de acuerdo al ciclo de los contenidos antes expuesto. También se estudian las alternativas libres y open source basadas en licencias promovidas con este fin.

Capítulo 3 Selección de Portales CMS: Como parte de todo un proceso de selección de Portales CMS libres u open source, se presentan en este capítulo los pasos a seguir, así como una guía basada en criterios y aspectos detallados por categorías. Los sistemas de referencias también se presentan en esta parte para complementar lo que sería una especie de estado del arte en técnicas modernas de selección.

Capítulo 4 Tecnologías libres para la Modificación e Implantación: Este capítulo presenta las principales tecnologías libres y open source para la modificación e implantación de Portales CMS una vez adquiridos. No se especifican técnicas ni metodologías, solo un cúmulo de plataformas y alternativas tecnológicas.

Capítulo 5 Portal NovaLNX: Se especifica todo el proceso en la práctica de la selección, modificación e implantación mediante el caso particular del Portal CMS NovaLNX.

Capítulo 1: El manejo de los contenidos

1.1 Introducción.

Los servicios de información y documentación accesibles a través de Internet, más concretamente mediante servidores Web, están aumentando de una forma exponencial. La lógica evolución de la Web desde hace más de 10 años ha ido produciendo la sustitución de páginas y documentos estáticos por documentos generados dinámicamente, merced a la interacción del usuario con la lógica de procesos y flujos de trabajo definida por los creadores del servicio y a la disponibilidad de cada vez mayores repositorios de información.(TRAMULLAS, 2005)

Evidentemente, se ha ido pasando progresivamente de un concepto de publicación de páginas Web, bastante simple en su origen, a esquemas más complejos y diferenciados, fundamentados en procedimientos y técnicas basados en la gestión de información. La cada vez mayor complejidad de los servicios y de los sistemas que los soportan, ha hecho necesaria la formulación de un corpus teórico y práctico en el que se combinen las técnicas clásicas de gestión de información en las organizaciones con las características propias del medio ambiente digital. (TRAMULLAS, 2005)

Fundamentalmente, las nuevas técnicas del manejo de información se han convertido en valuarte estratégico para cualquier organización, teniendo en cuenta sus necesidades. Cada día se desarrolla más en estos aspectos, por lo que tomar desde un principio las decisiones correctas de acuerdo a la información, sería lo más recomendado, convirtiéndose estas acciones en fundamentales.

Este capítulo está enfocado a sentar las bases de lo referente a los contenidos, el ciclo de vida de los mismos, y la necesidad que justifica el uso de sistemas que los manejen o gestionen. La gestión de Contenidos se presenta en este capítulo brindando un estado del arte en cuanto a tipologías de estos sistemas.

1.2 Contenidos.

Las computadoras fueron desde un principio construidas para procesar datos. Al igual que los datos, los contenidos son también información, solo que estos mantienen su contexto humano y sus significados. Lo ideal fuese que las computadoras se encargaran por sí solas del manejo de contenidos, pero si al obtenerse la información, esta es abundante y variada, entonces se tornaría difícil que sucediera esto. El

compromiso entonces quedaría resuelto en idear cómo envolver la información en metadatos.

Los datos y el contenido son diferentes, ciertamente, pero esa diferencia no significa que no obran recíprocamente. En efecto, las innumerables transiciones a partir de uno al otro ocurren cada día. (BOIKO, 2001)

Los datos son los pequeños trozos de información que las personas recogen, coleccionan y ensamblan para posteriormente guardarlos en bases de datos. Se pudiera usar una definición para la palabra información y sería tomando esta palabra para significar todas las formas comunes de registrar la comunicación, incluyendo:

1. Textos (libros, noticias, artículos).
2. Sonidos (música, conversaciones, lecturas).
3. Imágenes (fotografías, ilustraciones).
4. Archivos digitales (presentaciones).
5. Animaciones (videos, multimedias).

La información que pasa casualmente alrededor del mundo no es contenido, llega a serlo después de que alguien la obtenga. Si se dijera que el contenido es la información puesta en uso, entonces hay que preguntar primero el tipo de uso y el propósito de formar esta información. Sin estas respuestas se diría que un gran número de proyectos de gestión de contenidos van andando a ciegas.

Partiendo de la siguiente cadena lógica "Dato-Información-Conocimiento", se puede decir que el término "Contenido" se encuentra implícito formando parte del ciclo. Las definiciones de este término se pueden ver de diferentes maneras tomando como material de estudio las principales referencias sobre Gestión de Contenidos (BOIKO, 2001; BROWNING y LOWNDES, 2001).

El contenido (...) es un compromiso entre la utilidad de datos y la riqueza de la información. El contenido es la información rica que usted envuelve en datos simples. Los datos que rodean la información (metadatos) son una versión simplificada del contexto y del significado de la información. (BOIKO, 2001)

Los contenidos se pueden encontrar en la red de forma gratuita o comercializada según su importancia, trascendencia, actualizaciones, entre otros aspectos. Son concebidos generalmente para informar, formar y entretener. Su inserción dentro de la cadena

antes expuesta, resalta el papel primordial de la misma a la hora de convertir los datos en información e intentar llegar a los individuos para convertirlos en conocimientos.

Para muchas organizaciones y empresas los contenidos son un activo más y requieren de seguridad, mantenimiento y actualización para mantenerse funcionando. Para otras sencillamente se trata de materiales que encuentran su utilización como apoyo y enriquecimiento a los objetivos de la entidad. Sea cual sea su forma de aparecer, es importante que lleguen a los usuarios finales de forma eficiente y con el máximo de prestaciones, así como de la forma más rápida posible.

1.3 Justificación de la utilización de los Sistemas Gestores de Contenidos.

Desde la introducción de las primeras redes informáticas, se viene compartiendo a nivel básico información en forma de ficheros o de bases de datos centralizadas que evitan, de alguna forma, la duplicidad de dicha información y contribuyen al ahorro de tiempo. Con la posibilidad actual de la unión de distantes sedes, hay un palpable incremento en estas prestaciones, así como un aumento de la velocidad de acceso a un inimaginable número de fuentes. Se pudieran definir entonces, dada la situación actual, los niveles de disponibilidad de información tanto externa como interna como muy elevados.

Existen otros elementos a tener en cuenta y uno de ellos por ejemplo sería lo relacionado con las curvas de aprendizaje. Cada fuente de información tiene un estilo, una forma de uso y una organización diferente. En muchos casos la información es de fácil accesibilidad y de rápido alcance, pero en otros esto no es así al no encontrarse a primera vista y necesitar de mucha navegación hasta encontrarla. Con las facilidades de algunas herramientas como los buscadores, el verdadero problema no viene en la pérdida de tiempo de localizar la información, sino en entender el funcionamiento de la propia fuente y localizar los elementos de la información que buscamos dentro de la misma fuente. Al tratarse de una sola fuente o de una acostumbrada, los niveles de dificultad se hacen pequeños, pero actualmente dada la diversidad de fuentes y a veces la urgencia de necesidad de determinados datos, se elevan dichos niveles hasta algo considerable e inabordable.

Cualquier organización que, de alguna forma, practique la creación de publicaciones, está gestionando contenidos. Una sola persona, también puede estar gestionando contenidos si de alguna manera intenta organizar archivos para no perderlos de vista y más tarde compartirlos mediante publicaciones. Un sistema de gestión de contenidos comienza a ser necesario para ayudar a organizar y automatizar los procesos manuales

y/o informales de colección, gestión y publicación. Estos procesos se pueden determinar como complejos o imposibles de mantener de forma manual si:

- *Existen numerosas operaciones sobre los contenidos:* Por lo general a un sistema que tenga gran cantidad de contenidos, se le suele categorizar de complejo. Sin embargo la complejidad viene asociada a la cantidad de cambios que se haga sobre esos contenidos, de ahí que un sistema con cientos de componentes es verdaderamente simple si estos son estáticos. Los cambios que finalmente definen la complejidad, se pueden ver de dos formas:
 1. Procesamiento de contenidos: Cantidad de contenidos que se procesan en cierta cantidad de tiempo.
 2. Cambios de diseños: Frecuencia de cambio de diseños de un sistema. Es muy usual que gráficamente se adapten diferentes opciones de diseños para las publicaciones.

Las operaciones pueden ser agregar, eliminar o modificar contenidos.

- *Se cuenta con gran cantidad de contenidos:* Generalmente los conceptos de manejo de contenidos vienen asociados al cúmulo de contenidos, pero también estos conceptos se pueden utilizar en proyectos de pequeña envergadura. Todavía en muchos lugares, incluyendo nuestro país, se sigue el esquema de una persona que administre un Portal Web o un sistema de publicación. Al hacerse grande un sistema, tan grande como para que todas las tareas no quepan en la cabeza de ese Webmaster, se empieza a necesitar la presencia de otras personas o de un sistema que cargue con el peso de las operaciones fundamentales. Un sistema se puede categorizar como grande cuando:
 1. Tiene muchos Contenidos: Actualmente hay sistemas Web que suelen alcanzar varios cientos y hasta miles de páginas, sin embargo influye más en su tamaño la cantidad de contenidos que el número de páginas. En ese aspecto se debe ser cuidadoso y saber medir la necesidad de un sistema manejador de contenidos por la cantidad de contenidos y no de páginas.
 2. Diversidad de Contenidos: Un sistema con un elevado número de contenidos pero todos del mismo tipo, no complica mucho el manejo. Sin embargo, la necesidad surge al tener que tratar con una diversidad de

envergadura donde se haga necesaria una colección y categorización de los mismos.

- *Hay gran cantidad de publicaciones a crear:* Sin importar el tamaño de la organización, sea grande o pequeña, esta puede hacer varias publicaciones (artículos, noticias, reportajes, etc.) en dependencia de sus objetivos. Los factores a tener en cuenta en presencia de publicaciones son:
 1. Tipos de publicaciones.
 2. Personalización de las publicaciones: La necesidad de un sistema gestor de contenidos, se observa también cuando una organización tiene que publicar bajo ciertos criterios de personalización una gran cantidad o tipos de publicaciones.
- *Se cuenta con gran cantidad de contribuidores:* Frente a una organización donde las informaciones vengan de diversas fuentes o contribuidores, se hace necesario crear reglas de contribución. Por lo general el número de contribuidores siempre va en ascenso y es de vital importancia un sistema que entre todas las funcionalidades posibles, se encargue de guiar a cada nueva persona que de alguna forma intercambie información con la entidad.

Para esto problemas los sistemas gestores de contenidos se han hecho eco de estándares, ejemplo de ello son los sistemas de "Preguntas Frecuentes".

En el libro Content Management Bible (BOIKO, 2001), el autor propone una fórmula para medir el grado de complejidad de una organización al necesitar un CMS. Los factores de esta fórmula en efecto son: el número de autores o contribuidores, el número de fuentes y su complejidad, el número de componentes o contenidos, los tipos de componentes, las operaciones sobre los contenidos por semana de trabajo, el total de publicaciones que se crean, las personalizaciones de las publicaciones de los contenidos y por último el número de rediseños que se realizan por año de trabajo.

Veamos un ejemplo. Una organización X que dispone de una o varias oficinas de trabajo, distribuidas en una o varias sedes, consta también de una red informática basada en protocolos estándar TCP/IP y una Intranet. Las estaciones de trabajo están conectadas entre sí y disponen o no de Internet. El trabajo hasta ahora se basaba en compartir ficheros y el acceso a bases de datos centralizadas, también se disponía de servicios como el correo electrónico y algunas páginas Web por parte de algunas sedes

proveyendo el acceso a determinada información. El número de usuarios se puede medir en centenares y hasta pudiera llegar a miles.

Las soluciones implantadas para X fueron pensadas en la centralización y el acceso común a los recursos, pero actualmente se hace insostenible al empezar a destacarse ciertas características dentro de la organización como las restricciones de acceso a archivos, el acceso a heterogéneas fuentes de información por parte de los usuarios, los diferentes formatos de documentos sujetos a cambios constantes y el mantenimiento de informaciones basado en creación, actualización y eliminación. En fin, se necesitaría un sistema que fuese algo más que pura recopilación estática de información, se necesitaría de un sistema informático que pudiera encargarse de los contenidos, que facilitase, entre otras cosas, la concesión y revocación de usuarios sobre determinados recursos, la facilidad de estadísticas para el buen manejo de contenidos y la posibilidad de reducir los niveles de dificultad de búsqueda de determinada información para determinado usuario.

1.4 Gestión de Contenidos.

La gestión de contenidos debe definirse desde la perspectiva de su objetivo y actividades. (...) La gestión de contenidos está orientada a gestionar objetos que actúan como componentes de documentos virtuales, en el contexto de lo que llama segmentación. (TRAMULLAS, 2005)

La enorme cantidad de información existente en las instituciones, las organizaciones y las empresas en general; ha dado pie a que grandes compañías, proyectos productivos, grupos de trabajo e individuos, vean la necesidad del desarrollo de herramientas que estandaricen y faciliten determinados procesos que faciliten la creación, el mantenimiento, la gestión y la publicación de dicha información, de tal forma que esta llegue a los usuarios deseados con una confiabilidad y rapidez adecuada.

La toma de soluciones eficientes y eficaces sin dudas se verá reflejada si se accede al contenido correcto en tiempo real y con prestaciones adicionales como que la información se encuentre en varios lenguajes.

Independientemente del tipo de sistema que soporte una gestión de contenidos, los principales requisitos identificados dentro de la gama de software que entran en esa clasificación son:

1. La facilidad de uso de las herramientas. Ejemplo de esto son los editores de texto online bajo la filosofía WYSIWYG (What You See Is What You Get).
2. Completo control de los contenidos, identificando un ciclo de vida dadas sus características, que puede ir desde su creación u obtención, hasta su destrucción o archivado, pasando por su control, su publicación y su mantenimiento.
3. Reutilización de contenidos y estructuras del sistema para evitar las duplicaciones innecesarias.
4. Integración y automatización de diferentes conceptos que rodean los contenidos como la sindicación, traducción y agregación, así como una correcta integración con diversas plataformas, sistemas operativos y servidores de aplicaciones.
5. Publicación manual o automática de las revisiones o actualizaciones realizadas.
6. Definición de Flujos de Trabajo o Workflows en caso de requerirse, que incluyen actividades de validación, publicación entre otras, según determinados roles de usuarios.
7. Determinación de roles de responsabilidad de acuerdo con tareas a realizar con cada contenido. Permitiendo una distribución de personal según su profesionalidad y dándole participación a individuos que no dominen distintas técnicas si la tarea no lo precisa.

El proceso más sencillo de publicación de información, pudiera definirse al diseñarse un marco que integre un conjunto de objetos para obtener como resultado una publicación digital de un documento cualquiera. Este tipo de proceso respondería a un sistema relativamente sencillo desde el punto de vista del tipo de información a publicar y la cantidad de objetos que estructuran dicha información final. Los procesos tienden a complicarse cuando se requiere de información especializada para determinadas comunidades, así como cuando se sitúan en sistemas o contextos que deban crear información digital más compleja.

Independientemente de los componentes que integran un sistema de gestión de contenidos, se destacan en su mayoría 3 subsistemas claves que interactúan entre ellos. Estos son:

1. *Subsistema de colección de información:* Se encarga generalmente de capturar y/o crear la información. Puede brindar soporte a procesos de conversión de

- documentos en diversos formatos y agregación desde fuentes externas y de diferentes estructuras, pero su principal funcionalidad viene dada por dar soporte a procesos de creación de contenidos, procesos de flujos de trabajo y sindicación.
2. *Subsistema de Gestión de Contenidos*: Define parámetros para un buen funcionamiento, controla flujos de trabajo de otros subsistemas, controla y gestiona los grupos de usuarios y los repositorios de información, entre otras funcionalidades.
 3. *Subsistema de Publicación de información*: Se encarga, mediante posibilidades de personalización y generalmente de manera automática, de producir las publicaciones o productos de información digital. Debe cumplir con requerimientos de múltiples plataformas y clientes.

Los diferentes trabajos expertos en gestión de contenidos, difieren un poco en cuanto a cantidad de subsistemas y funcionalidades de los mismos. Por ejemplo, (BROWNING y LOWNDES, 2001) señalan como subsistemas claves: Autoría/Creación, Flujo de trabajo, Almacenamiento/Repositorio y Publicación. Por otra parte, (NAKANO, 2002) señala cuatro subsistemas correspondientes a Creación/Edición de contenidos, Repositorio, Flujo de trabajo y Gestión de operaciones.

En la práctica se observa que son tantas las opciones de gestión y el número de herramientas que dan solución, que existe una variedad que sin alejarse mucho de los conceptos, ofrecen prestaciones viables y reconocidas.

1.4.1 El Ciclo de vida de los Contenidos.

Un desarrollo tecnológicamente sobresaliente no basta para cumplir las expectativas de un CMS, es necesario tener en cuenta aspectos como la facilidad de gestión por parte de usuarios con poco o ningún conocimiento del tema. El software debe ser capaz de guiar mediante interfaces al usuario a que haga una gestión sin saber siquiera que realiza dicha tarea. Los contenidos deben de pasar por diferentes fases que determinan como resultado un ciclo de vida. Estas fases son:

1. Agregación.
2. Producción.
3. Entrega.

4. Análisis.

1.4.1.1 Agregación.

Las principales tareas de esta fase son la captura y adquisición de contenidos. Estos pueden venir formando parte de archivos en diferentes formatos, algunos de estos formatos suelen ser:

- Documentos de textos (PDF, Word, etc.)
- Hojas de cálculo (Excel)
- Registros de bases de datos (JDBC, ODBC, etc.)
- Noticias
- Imágenes (BMP, JPEG, GIF, etc.)
- Animaciones (GIF, FLASH, etc.)
- Ficheros de audio (MP3, WAV, etc.)
- Ficheros de video (AVI, MPEG, etc.)
- Salida de ejecución de aplicaciones
- Resultados de consultas a bases de datos
- Otros.

El modo de recoger los contenidos puede variar y de hecho existen muchas formas que tienen como resultado su almacenado de diversos modos también. No obstante, existen etapas que caracterizan la agregación en sí y que, de una forma u otra, se ven presentes en distintos productos. Estas son:

1. *Creación*: Se puede definir como el proceso básico en la llamada introducción de contenidos al sistema. Al menos se realiza de dos formas: mediante un navegador Web, la cual permite la creación a distancia por parte de encargados o usuarios, y la usada por desarrolladores y diseñadores, mediante aplicaciones instaladas en el equipo local, permitiendo una más rápida edición, estructuración y desarrollo de los contenidos.

2. *Agregación*: Se puede encontrar de forma manual o automática. Muchas veces no se dispone de personal para la edición y por tanto debe de hacerse la agregación mediante métodos. Las agregaciones generalmente deben de poder servirse estática o dinámicamente, mostrando, por ejemplo, estados de solicitudes. Existen agentes de agregación o robots de recogidas que implementan mecanismos de análisis de las fuentes de datos mediante sistemas de patrones de expresiones regulares.
3. *Categorización*: El sistema debe permitir clasificar los contenidos mediante estructuras de información jerárquicas para representar los diferentes conceptos de dominios de elementos. Esto puede hacerse de forma manual o mediante sistemas automáticos de taxonomización.

1.4.1.2 Producción.

En el ciclo de vida de los contenidos, estos se someten a una fase donde el sistema pone sus funcionalidades en cuestión de su gestión. También se pueden hacer estas actividades de forma manual. En la producción se tiene que tener en cuenta lo siguiente:

1. *Gestión*: En esta fase de gran importancia, se implementan tareas que rodean el sistema de workflow. Dicho sistema debe dar posibilidad a varios tipos de procesos. Los procesos pueden ser automáticos o invocados remotamente, dando la posibilidad a integrar motores de workflow ya existentes en la organización. También pueden ser manuales o de usuarios, cumplimentándolos el personal que forme parte del workflow definido anteriormente. Los usuarios tendrían asignados los roles de revisión, edición, aprobación, traducción, publicación, entre otros, que permitan los procesos de validación y aprobación. Las tareas se podrían gestionar desde diferentes interfaces que como mínimo muestren las tareas pendientes y la posibilidad de notificación. Toda operación debe ser registrada, dando posibilidad a futuros mecanismos de auditorías. También la realización de copias de seguridad sobre contenidos modificados y/o actualizados podría ser de ayuda para la adecuada gestión.
2. *Objetivos*: La efectividad deseada se logra mediante la definición de los objetivos trazados para cada contenido. Esta fase de definición debe llevar el tiempo que sea necesario por su importancia. Generalmente los contenidos se designan a distintas áreas de aparición según su carácter, que puede ser público

a asignado por roles, de ahí la importancia de que no aparezcan en lugares no adecuados del sistema. La clave sería preguntarse, “*quienes deben recibir según que contenidos*”.

3. *Puesta de contenidos*: La automatización de la puesta de los contenidos por ejemplo en fechas planificadas, reduce la intervención manual para la copia o transferencia de dichos contenidos.

1.4.1.3 Entrega.

En la entrega, lo más sobresaliente es llevar a cabo acciones fundamentales de un CMS como la observación de las interacciones así como una adecuada y escalable entrega de los contenidos. En sistemas pequeños o con pocas prestaciones, esta fase prácticamente se omite, por lo que se pierde control sobre la interacción con el usuario. Más bien ocurre en sistemas de mera publicación de información editorial, donde no se pretende mucho la interacción con el usuario, ni el desarrollo de aplicaciones Web que presten servicios.

- *Ensamblado*: En el ensamblado de los contenidos, se debe disponer de todas las fuentes para de esa forma lograr un adecuado ensamblaje de la página, el portlet o el componente Web. El ensamblaje debe realizarse en tiempo real, logrando introducir contenidos dinámicamente y personalizados. La importancia de esta fase radica en poder conceptualizar y manipular mejor el contenido que estará separado de la presentación.
- *Personalización*: Esta fase es muy rica en variedad, cada organización estudia por su parte la personalización convirtiéndose muchas veces en un proceso donde interviene la colaboración de mucho personal. Cómo es lógico, la personalización va más dirigida al entorno público que a los servicios Web privados. Las categorías de la personalización son:
 1. *De adaptación al sistema*: Se personaliza teniendo en cuenta aspectos como tipo de navegador, idiomas predefinidos, tipo de dispositivo, etc. Se lleva a cabo aunque el usuario no esté aun registrado en el sistema.
 2. *Implícita*: Sin requerirse aún el registro del usuario, se usan servicios básicos como mostrar los links más utilizados.

3. *Explícita*: Se lleva a cabo mediante campañas de información dirigida a determinados usuarios.
 4. *Páginas personales*: Cada usuario previamente registrado en el sistema, decide el tipo y la forma de información a mostrar. Generalmente la apariencia e imagen constitucional no pueden ser modificadas.
- *Observación*: El rastreo a los servicios brindados para conocer el uso que los usuarios finales hacen de ellos, es muy importante. El nivel debe ser bien profundo hasta saber el gusto de los usuarios y así sacar conclusiones acerca de los contenidos. Solo herramientas bien implementadas podrían realizar estos rastreos, ya que los servidores Web no llegan a comprender los términos de contenidos para brindar estas funcionalidades.
 - *Sindicación*: En muchas ocasiones, los CMS deben de ser capaces de no solo entregar contenidos a usuarios humanos, sino que también deben de posibilitar la entrega de contenidos a otras aplicaciones Web y softwares de terceras personas. Esto se realiza mediante mecanismos para intercambio de información y puede planificarse con anterioridad para una fecha exacta, entre otras prestaciones.

1.4.1.4 Análisis.

Esta fase, es la última del ciclo de vida de los contenidos y se ve realizada una vez que se llevan a cabo análisis sobre los mismos. Generalmente se realizan mediante técnicas de análisis matemáticos y algorítmicos. Sus objetivos finales suelen estar sobre la mejora de los servicios. Es importante tener en cuenta sin embargo lo siguiente:

1. *Captura*: Para la captura de la información que servirá para el análisis, se deberá ser abarcador. Es necesario tener en cuenta módulos de rastreo, ficheros de logs en los servidores, sesiones entre diferentes servidores Web, datos del usuario, etc.
2. *Descubrimiento*: Bajo la pregunta que le da el objetivo a esta fase, *¿cuántos usuarios hacen qué?*, se descubren las preferencias de los usuarios en caso de ser un objetivo el tener una mayor interacción. Generalmente se usan técnicas de minería de datos y análisis de informes, llevando a conformar un análisis sobre las afinidades de los contenidos.

3. *Optimización*: Una vez que se tengan ciertas informaciones recogidas de las anteriores fases de captura y descubrimiento, se puede decidir hacer ciertas modificaciones al sistema. Los resultados pueden llegar a decidir cambiar varios aspectos entre ellos la categorización, modificar la navegación, los tipos de contenidos a manejar y las posiciones a mostrarlos, entre otros. Todo esto gracias a la flexibilidad que ofrecen los CMS de rápida optimización.

1.5 Tipología de Sistemas de Gestión.

Actualmente la Gestión de Contenidos enriquece las Tecnologías de la Información y las Comunicaciones. Cada día se puede ver un aumento en la cantidad de herramientas que implementan las metodologías también en ascenso. Los objetivos alcanzados con cada herramienta o aplicación se elevan, alcanzando niveles muy altos que hacen que vaya en ascenso el número de organizaciones que apuestan por sus prestaciones. Los diferentes grupos de usuarios que se ven beneficiados, también exigen que grandes compañías, grupos de trabajo o individuos, vean su aporte en el desarrollar sistemas de gestión de contenidos. La gestión de contenidos puede verse generalmente (no total ni únicamente) en sistemas de tipos:

- **Portales Web.**

Conocidos también como Web Content Management System (WCMS), existen al alcance de cualquier persona un gran número de portales que integran sistemas de gestión de contenidos. En su mayoría se conciben para que gracias a sus funcionalidades, fácil administración y diferentes mecanismos de control, lleguen a un gran número de usuarios finales. Generalmente siguen un estándar en cuanto a su estructura al estar diseñados en módulos que finalmente son administrados por una amigable interfaz de administración. Será parte de este trabajo abordar su estructura en próximos capítulos. (Ver Capítulo 2)

- **Sistemas LMS y LCMS.**

Siguiendo por lo general entornos de aulas clásicas de docencia, los sistemas de aulas virtuales, conocidos también por "Learning Management System" (LMS), suelen implementar mecanismos de aprendizaje y evaluaciones online. Generalmente se basan en herramientas como chats, foros y clases interactivas mediante profesores online y en tiempo real (Ver figura 1). No se puede decir que manejen en la totalidad de conceptos los contenidos para sus objetivos. Sus beneficios vienen dados en el orden

de constar con formas tradicionales de enseñanza. Su estructura de módulos administrativos, permiten matricular alumnos, configurar cursos online, asignar cursos a alumnos, registrar profesores, llevar al tanto informes de calificaciones, entre otras funcionalidades. Los métodos de enseñanza pueden ser de forma síncrona o asíncrona.

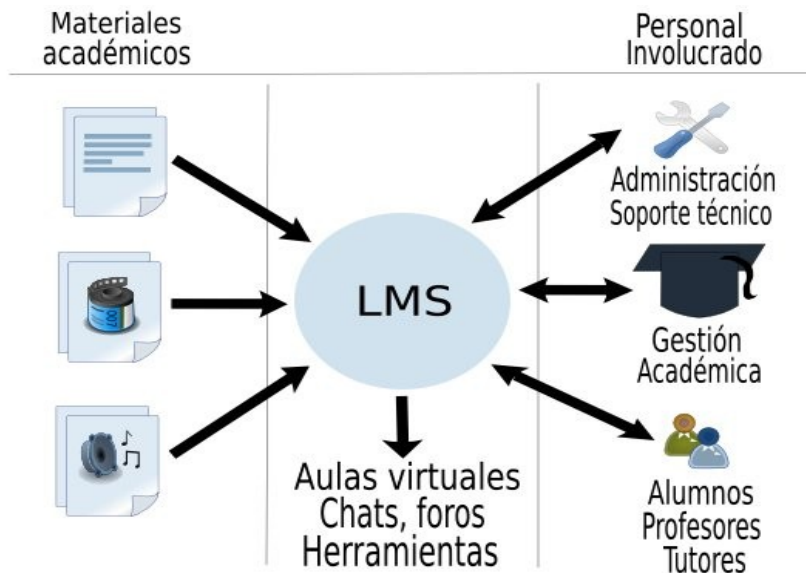


Fig 1: Estructura básica de un LMS

Por otra parte tenemos los "Learning Content Management System" (LCMS) Por lo general gestionan y administran recursos meramente educativos y no todo tipo de recursos. Se pueden definir como sistemas basados en Web que son utilizados para crear, publicar, administrar y almacenar recursos educativos y cursos en línea. En la figura 2 se representa la estructura básica de un LCMS.

Los principales usuarios suelen ser los diseñadores que utilizan los contenidos para estructurar los cursos, los profesores que utilizan los contenidos para complementar su material de clase e incluso los alumnos en algún momento pueden acceder al sistema para desarrollar sus tareas o completar sus conocimientos.

En un LCMS se tienen contenedores o repositorios para almacenar los recursos, que

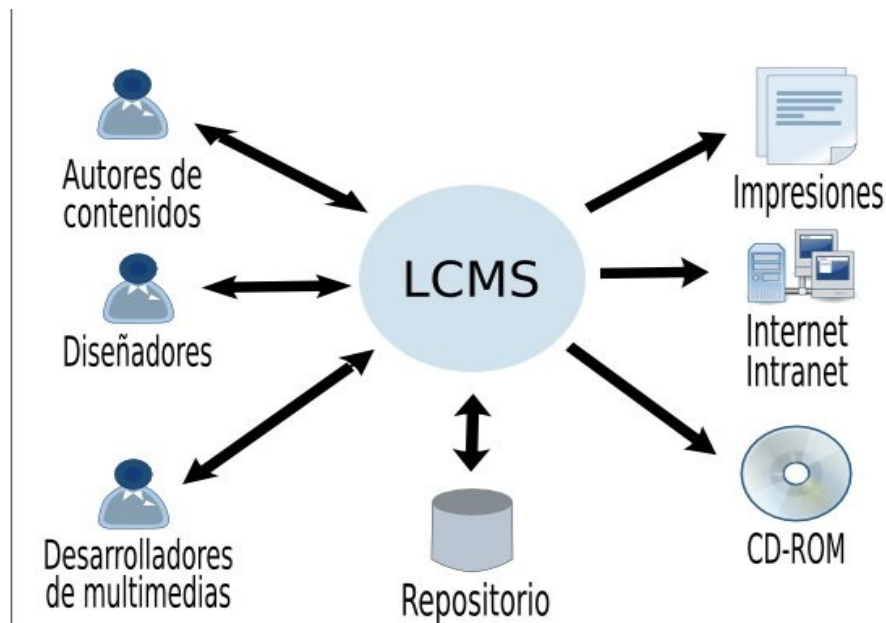


Fig 2: Estructura básica de un LCMS

pueden ser utilizados de manera independiente o directamente asociados a la creación de cursos dentro del mismo sistema. Es decir que el repositorio puede estar disponible para que los profesores organicen los cursos o también pueden estar abiertos para que cualquier usuario recupere recursos no vinculados a ningún curso en particular, pero que les pueden ser de utilidad para reforzar lo aprendido sobre algún tema. El proceso de trabajo dentro de un LCMS requiere de control en cada fase del contenido, esto conlleva un proceso editorial para controlar la calidad de los contenidos creados, así como para permitir y organizar su publicación.

Ambos sistemas pueden integrarse en un completo sistema de enseñanza y así lograr un máximo de prestaciones.

- **Plataformas para la Gestión de Contenidos.**

Se trata de plataformas ideadas para desarrollar e implementar varias aplicaciones que den solución a un gran número de necesidades específicas sobre estos conceptos de gestión. Como es lógico, demandan de un elevado nivel de conocimientos en cuanto a lenguajes de programación como Java, PHP, Python y Perl, entre otros. Sus soluciones se tratan de ajustar a diferentes problemáticas, logrando cada vez más funcionalidades al constar en algunos casos con muchos desarrolladores que contribuyen a la implementación de nuevos módulos Ver ejemplo de estas plataformas en la figura 3.

Presentan varias características como entornos básicos para la gestión y entornos con varias interfaces y herramientas para desarrollar flujos de trabajos y otras prestaciones básicas. Entre las soluciones que suelen brindar están las de diseñar e implementar Portales Colaborativos, Intranets, páginas personales con prestaciones como blogs, etc.

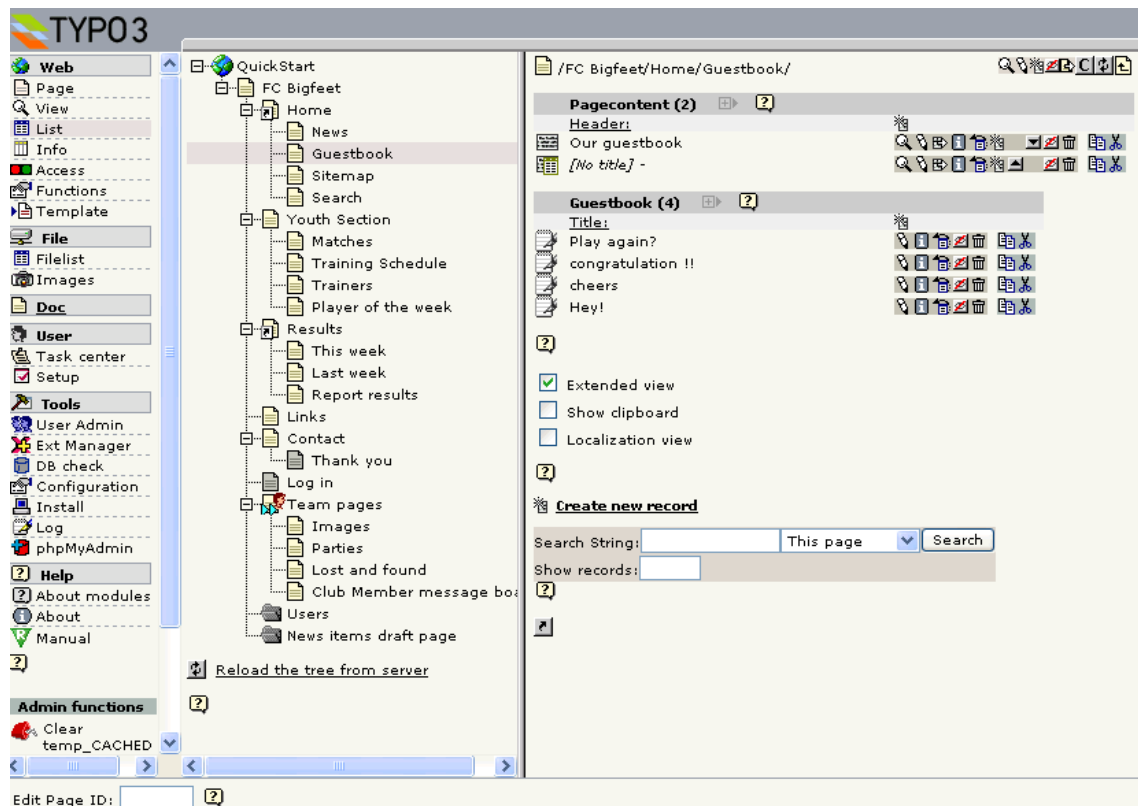


Fig 3: Ejemplo de plataforma para la gestión de contenido. Typo3.

- **Publicaciones Digitales.**

Los sistemas de prensa, entre muchas otras empresas y organizaciones, se ven muy beneficiadas con el uso de manejadores de contenidos para sus publicaciones digitales. Una publicación digital se diseña mayormente para editoriales como periódicos y revistas que tengan la necesidad de su expansión al mundo digital mediante la Web. Por tanto, la gestión en sí de los contenidos vienen dados en el orden de la manipulación de los mismos bajo el objetivo final de lograr la publicación.

Los contenidos se sujetan a procesos de creación, edición y publicación de los mismos. Estos sistemas tienen en su mayoría capacidad para soportar recursos de información en varios formatos. Generalmente los Sistemas de Publicaciones Digitales no están

orientados a manejar todos los contenidos de una editorial específica, sino solo los que intervengan en sus funciones de publicación. Entre las prestaciones que suelen verse en estos sistemas, está el soporte de las publicaciones en varios idiomas.

- **Foros de discusión.**

Los foros son sistemas que pueden verse asociados a otros sistemas como portales, o pueden encontrarse también independientes. Por lo general van dirigidos a la comunicación de usuarios en determinados temas. Son ricos en capacidades de personalización y facilidad de uso, logrando con ello el mayor uso cada día por parte de internautas. En su mayoría, estos sistemas pueden insertarse dentro de portales previamente diseñados con tan solo algunos ajustes.

En lo adelante este trabajo obviará las topologías de CMS y se referirá solamente a los Portales Web. Los próximos capítulos tratarán los Portales Web con el nombre básico de CMS, ya que es así con esas siglas como más se les conoce.

Capítulo 2. Estructura básica de un Portal CMS.

2.1 Introducción.

Los Portales Manejadores de Contenidos (WCMS mayormente conocidos como CMS), siguen por lo general una estructura básica independientemente de las alternativas tecnológicas que integran o si son comerciales o no. La misma viene siguiendo lógicamente el ciclo de vida de los contenidos expuesto en el capítulo anterior.

En este capítulo se pretende abordar lo referente a la estructura básica de estas peculiares herramientas. También de forma básica se tratará de delimitar entre libres u open source y comerciales, brindando un estudio del uso de los mismos bajo determinadas licencias.

2.2 Subsistema de colección.

Todos los procesos que anteceden a la final publicación de la información, tienen un comienzo bajo la responsabilidad del Subsistema de Colección de Contenidos. Su principal función es la de capturar los contenidos de sus fuentes y mediante procesos finalmente concretar un sistema de componentes de contenidos bien organizado. Dichos procesos pueden observarse interactuando (Ver figura 4), dichas interacciones no son obligatoriamente en la secuencia que se muestra.

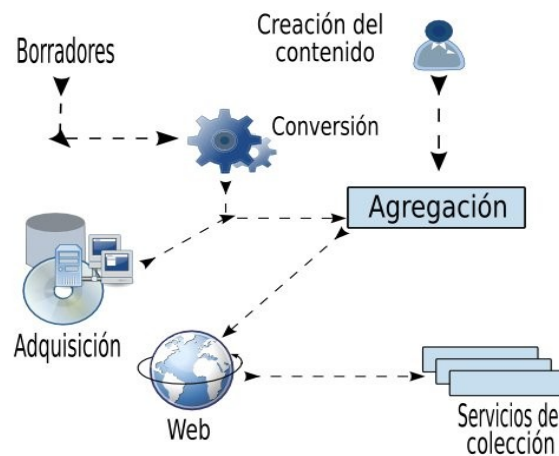


Fig 4: Subsistema de Colección de Contenidos.

Los procesos, según (BOIKO, 2001), generalmente abarcan lo siguiente:

1. Creación del contenido.
2. Adquisición.

3. Conversión.
4. Agregación.
5. Servicios de Colección.

- **Creación del Contenido.**

La creación lógicamente se refiere al hecho de crear los contenidos mediante el trabajo de un autor. El autor no se refiere a cualquier persona, sino a la que cumpla sus funciones como creador en dicha entidad. Si alguien hace esa función sin pertenecer al colectivo de trabajo de dicha entidad, entonces el contenido fue adquirido y no creado.

El proceso de crear depende mucho de la creatividad de los autores y no tanto de las funcionalidades del sistema. El CMS puede ayudar a la efectividad de este proceso mediante:

1. Comodidad del ambiente para la creación.
2. Esclarecimiento de los propósitos y los temas de los contenidos a crear.
3. Proporcionando completamiento de informaciones estándares (Nombre del autor, Fecha de confección, etc.).
4. Plantillas (El autor pudiera crear en un ambiente separado por lugares intuitivos como una zona para el resumen, otra para el cuerpo de la información, etc.).
5. Control de versiones, flujos de trabajos y estado del mismo a medida que se vaya creando.

El autor basa sus funciones generalmente en crear y revisar su trabajo hasta que se encuentre listo para su uso. En este proceso el contenido puede estar en el sistema desde el principio y así beneficiarse de algunas de sus funciones o puede ser introducido una vez que se haya concluido la creación.

- **Adquisición del contenido.**

Básicamente se puede entender que este proceso realiza funciones de adquirir informaciones manual o automáticamente no creadas por el sistema.

Por lo general todo CMS consigue parte de su información del exterior (Ver Figura 5).

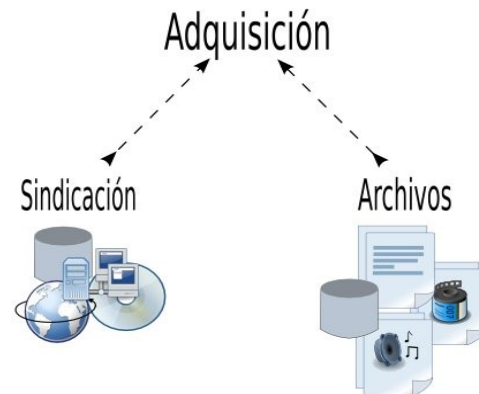


Fig 5: Adquisición de Contenidos.

La información puede venir de las siguientes fuentes:

1. *Archivos*: Los archivos pueden ser cualquier tipo de información creada y almacenada con anterioridad. Incluso en formatos no digitales como fotografías en papel o textos impresos, que luego sean convertidos digitalmente. Estos archivos por lo general requieren de procesos de transformación para lograr su usabilidad al no estar muy a menudo en formato binario ni estar preparados para la segmentación en metadatos.
2. *Sindicación*: Actualmente es frecuente encontrar formas de distribuir y hasta comercializar la información en forma de componentes reutilizables. Por lo general se encuentra en formato binario (XML por ejemplo) y dividida en componentes como el título, la fecha y un cuerpo. Muchos sitios basan sus actividades en el empaquetado y la distribución de la información mediante la sindicación.

La diferencia entre crear y adquirir la información viene asociada a la calidad y la cantidad de la misma, si es creada en el propio sistema y mediante autores propios, entonces tendrá mejor calidad y menor cantidad que si fue adquirida y viceversa.

- **Conversión del contenido.**

La información creada o adquirida debe de cumplir con los estándares de estructuración definidos para su sistema, por tanto se hace necesario que se realicen actividades de conversión de los contenidos (Ver Figura 6).



Fig 6: Conversión de Contenidos.

El proceso de conversión suele definirse en 3 etapas, aunque muchas veces marchan juntas y no son distinguibles. Estas son:

1. *Desecho de información circundante*: Se eliminan contenidos innecesarios como cabezas y pies de páginas.
2. *Cambio de formato*: Puede hacerse cambiando de formato binario a un formato predefinido por el sistema, etc.
3. *Cambio de estructura*: Se realiza la estructura de la información como explícita o se cambia como necesaria.

El resultado de un proceso de conversión es la información conformada tan bien como sea posible con los estándares previamente definidos para el formato.

- **Agregación de contenidos.**

La información no siempre tiene que necesitar ser convertida, pero si así no fuera, entonces debe ser agregada. Básicamente la agregación se trata de procesos para que la información quede definitivamente estructurada. Estos procesos pueden ser:

1. *Proceso editorial*: La información es tratada hasta que quede en función de su uso y en la consistencia adecuada. En el caso de sistemas profesionales de publicación, son usados frameworks editoriales, los cuales siguen reglas de corrección, de comunicación y de consistencia.

2. *Proceso de segmentación*: La información es, por decirlo de alguna forma, despedazada. Es en este momento donde se conforman los componentes de contenidos.
3. *Proceso de metadatos*: Los componentes de contenidos son marcados para su posterior almacenamiento y para facilitar la búsqueda automática de los mismos.

Como es lógico, no siempre serán necesarios todos estos procesos y esto se puede apreciar claramente en sistemas pequeños o en sistemas que adquieran información debidamente estructurada. Tampoco tiene que obligatoriamente delegarse los procesos en personas (en caso de que sean manuales) por separadas, se pudieran hacer por una misma persona de acuerdo a la complejidad y la carga de trabajo.

- **Servicios de Colección.**

Los procesos de colección son ayudados por servicios de colección. Estos servicios principalmente mantienen el contenido en los repositorios establecidos. Las principales tareas que realiza para esto vienen dadas por guardar contenidos creados por el propio sistema o adquiridos.

2.3 Subsistema de gestión.

Los componentes de contenidos son almacenados todo el tiempo necesario mediante el subsistema de gestión. Este subsistema no solo contiene los repositorios, sino facilidades de administración y flujos de trabajos asociados a los contenidos. Entre las utilidades más destacadas se encuentran por ejemplo las de brindar:

1. Detalles del estado de los contenidos, tipo y ciclo de vida de los mismos.
2. Reportes del uso de los contenidos y de posibles cuellos de botella.
3. Detalles de qué usuario accede a qué contenido.

En fin, cualquier pregunta sobre el funcionamiento de la publicación y la colección de los contenidos puede ser respondida mediante el subsistema de gestión.

- **Los repositorios.**

Son la pieza principal de un subsistema de gestión y pudieran ser bases de datos relacionales, ficheros o estructuras internas de determinado CMS para el almacenamiento de datos o de ficheros de control y configuración (Ver Figura 7).

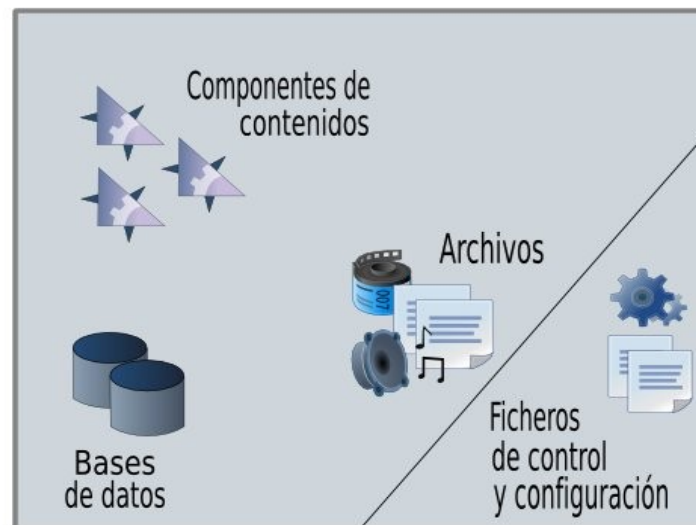


Fig 7: Ejemplo de Repositorio de Contenidos.

- **Sistemas de administración.**

Son los responsables de fijar los parámetros y la estructura de un CMS. Toma partido en todos los subsistemas de las siguientes formas:

1. *En el subsistema de colección:* En este subsistema se fijan los papeles y derechos del personal con acceso al mismo.
2. *En el subsistema de gestión:* Realizando tareas principalmente en los repositorios de mantenimiento y control de permisos a usuarios, también tareas específicas sobre los contenidos como la creación, la ejecución de revisiones y la creación de flujos de trabajos.
3. *En el subsistema de publicación:* Supervisando por lo general el estado del hardware, garantizando el funcionamiento del servidor Web y otros softwares asociados.

- **Flujos de Trabajo (Workflow).**

El workflow es una serie de tareas predefinidas que facilitan la progresión de contenidos activos a través de actividades de gestión, transformación, y publicación. (GLASSHAUS, 2003)

El sistema de flujos de trabajo es responsable de coordinar, programar y hacer cumplir finalmente horarios y tareas de cada subsistema y persona. También está presente en cada subsistema del CMS de las siguientes formas:

1. *En el Subsistema de Colección:* En este subsistema los flujos de trabajos son aplicados a las tareas de creación y colección de los contenidos. En la mayoría de los casos los flujos de trabajo siguen a los contenidos desde su creación o adquisición hasta su final publicación. Un ejemplo sería entrar todo el contenido que llegue al sistema y distribuirlo a personas para su procesamiento, conversión, hasta llegar a los redactores finales; luego sería conveniente también una exhibición de las tareas realizadas y por realizar, etc.
2. *En el subsistema de Gestión:* Cumpliendo funciones administrativas como el correcto archivado de los contenidos. También en este subsistema los flujos de trabajo se designan para verificar, repasar, cambiar la utilidad de los mismos y para establecer conexión entre otros CMS y otros sistemas que provean de información útil.
3. *En subsistemas de publicación:* Los flujos de trabajo se asocian a ciclos de publicación asegurando la eficiencia de la publicación. Generalmente se ven en sistemas que publican cada cierto período de tiempo, así revisan contenidos pendientes por publicar, reglas de personalización, etc.

Los diferentes roles que generalmente pudiesen existir en un sistema de workflow son:

1. Lectores.
2. Publicadores.
3. Solo moderación: Aprobación o no del contenido a publicar.
4. Moderación múltiple.
5. Moderación con actualización: Revisión y edición previa a la publicación.
6. Archivo de documentos: Autorización para archivar documentos del repositorio.
7. Borrado: Autorización para eliminar documentos del repositorio.
8. Responder: Funcionalidad en la que el usuario puede responder a una publicación mediante un canal de publicación. En caso de que dicho canal tenga establecido un workflow de moderación, la respuesta pasará por el usuario.

9. Traductores: Se harían versiones en diferentes idiomas de la información.

Estos roles pueden ser asignados a usuarios individuales o a grupos de usuarios.

2.4 Subsistema de publicación de contenidos.

Este subsistema tiene que ser capaz de, a grandes rasgos, crear publicaciones de información con recursos almacenados en su sistema o no (Ver Figura 8).

- **Plantillas**

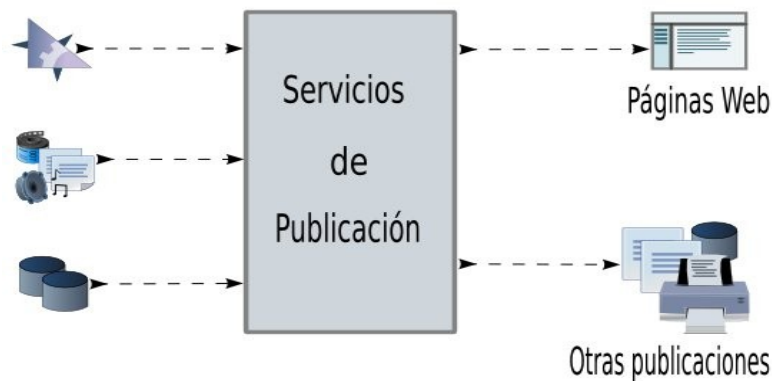


Fig 8: Subsistema de Publicación de Información.

La publicación de contenidos y el repositorio se enlazan mediante las plantillas (templates). Su tarea es guiar la creación de la publicación basada en contenidos. Generalmente incluyen los siguientes componentes:

1. Elementos estáticos.
2. Llamadas a servicios de publicación.
3. Llamadas a servicios fuera del sistema.

- **Servicios de publicación.**

Son basados en aplicaciones lógicas y servicios que un CMS proporciona para la creación de los contenidos desde los componentes almacenados en el repositorio. Estos servicios de publicación se caracterizan por las funcionalidades de:

1. *Cargar y ejecutar las plantillas:* Procesan la personalización, conversión y extracción de contenidos, asegurando las condiciones para que las plantillas creen las publicaciones.

2. *Proveer servicios de publicaciones específicas*: Incluyen servicios de impresión y conversión a diferentes formatos del contenido.
3. *Proveen conexiones con otros sistemas*: Sistemas que se necesitan para la extracción de determinados contenidos de uso.

- **Publicaciones Web.**

Las publicaciones Web básicamente son Sitios Web en Internet, una intranet o extranet que un CMS es capaz de producir. Pueden ser dinámicos, entonces se producen las páginas dinámicamente tras un clic de un usuario. Primeramente se pasa una petición al servidor Web, este acciona los servicios de publicación que siguen a muy grandes rasgos los siguientes pasos:

1. Cargar una plantilla.
2. Pasar a la plantilla cualquier parámetro que viniese junto a la petición.
3. Ejecutar la producción de una página.
4. Procesar la página al servidor Web para su publicación al usuario.

Si fuesen estáticos, el sistema los produce de una vez y los sirve como archivos en HTML.

- **Otras Publicaciones.**

Un CMS no se limita solamente a producir publicaciones Webs, los mismos procesos usados para estos fines pueden ser utilizados para publicaciones electrónicas, sindicaciones, etc.

2.5 CMS libres y propietarios.

Como se mencionaba anteriormente, en la actualidad existe un gran número de opciones que abarcan de alguna forma la gestión de contenidos. Más concretamente se puede decir que existe más variedad si se analizan todas estas opciones desde el punto de vista de las libertades que ofrecen. Ahí se encuentra que los CMS se encuentran divididos en 2 grandes grupos que los caracterizan de propietarios y de libres o de código abierto según las políticas bajo las cuales estén concebidos.

Desde el punto de vista de si cumplen o no con las necesidades de determinada entidad, se podría decir que no existe el sistema perfecto, pero si el que tras un estudio pueda concebirse o adaptarse lo mejor posible a las mismas.

Las soluciones propietarias son por lo general muy costosas, aunque también muy ventajosas en cuanto a soporte profesional. Las licencias de uso vienen dadas por la cantidad de usuarios y las empresas productoras y suministradoras de estos sistemas suelen cobrar también por consultorías, mantenimiento, etc. Por supuesto que cabe resaltar también que casi nunca o solo después de algún tiempo se tendrá acceso al código fuente del producto y que en caso de desaparición la empresa productora del software, se perderían las opciones de actualizaciones y mantenimiento técnico.

Existen proyectos libres y open source que cubren casi todas las necesidades de software existentes. También la estandarización y popularización de lenguajes como Java, Perl, PHP, Python y la plataforma Java 2 Enterprise Edition (J2EE), ha facilitado la aparición de gran cantidad y diversidad de software orientados a Internet. Sin embargo no se puede decir absolutamente que hay soluciones libres homólogas a todas las propietarias, aunque hayan muchas incluso con mayor calidad.

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 (CUERDA, 2004).

Precisamente por las características de los proyectos libres y open source es que se necesita contar con personal especializado en la tecnología que se seleccione y de esta forma garantizar tareas de pruebas al sistema y futuras adaptaciones.

Por tanto, sería justificable apostar por un producto libre u open source si:

1. Está disponible dicho producto en la tecnología que se ha decidido emplear (java, PHP, etc.).
2. El coste calculado de adquirir personas especializadas técnicamente es menor que el de comprar un producto comercial a una empresa.

Retomando a la Organización X, y tratando de hallarle solución a sus necesidades de gestión de contenidos, por la parte comercial tendríamos, entre muchas, a compañías como:

1. BEA: Ofrece Weblogic Portal.(bea.com)
2. IBM WebSphere Portal: Ofrece una estrategia unificada para portales de IBM que combina los componentes de software que necesita para crear con

- seguridad contenido personalizado y ofrecer una experiencia interactiva de portal a los empleados, proveedores y clientes.
3. Microsoft: Ofrece Content Management Server 2002. Variedad de características para la aportación y entrega de contenido, desarrollo de sitios y administración de sitios empresariales para permitir que los negocios creen, implementen y administren efectivamente sitios Web de Internet, intranet y extranet.

Tomando como ejemplo a BEA tenemos que esta compañía ofrece una sólida infraestructura de servidor de aplicaciones y portal Web. Ahora, herramientas como los foros, el cliente de correo, el chat, etc., vienen de manos de terceras empresas y su integración con la infraestructura inicial de BEA depende de servicios profesionales llevados a cabo por las mismas. En este caso los costos se centran en los productos. A estos hay que añadirles los de servicios profesionales de desarrollo de aplicaciones internas como partes de actividades, sugerencias, etc.

La opción de implantar una solución libre u open source, tiene la ventaja de centrar los costos principalmente en el soporte, ya que los productos por lo general se pueden obtener gratuitamente.

2.5.1 ¿Libre u open source?

Como se ha podido apreciar, se han mencionado varias veces en este trabajo los términos "libre" y "open source", términos que suelen asociarse y confundirse por muchas personas ligadas o no al mundo de la informática. Evidentemente no son lo mismo aunque incluso grandes medios de comunicación e información digan muchas veces lo contrario.

Al tratar este trabajo sobre tecnología eminentemente libre u open source, se hace necesario hacer saber cuando estamos en presencia de uno u otro término y así evitar equivocaciones que pudieran no ser graves en un principio pero sí a medida que el trabajo a realizar tome importancia.

El software libre tiene sus comienzos con el proyecto GNU bajo el mando de Richard M. Stallman en la década de los '80. El proyecto GNU tuvo en sus principios la idea de desarrollar un sistema operativo completamente libre, donde todos los usuarios y/o desarrolladores de software tuviesen ante todo libertad. Ante la colosal tarea de hacer un sistema operativo lo más completo posible, se fueron implementando piezas de este, y desarrollándose poco a poco toda una comunidad de usuarios y desarrolladores. En

1985 surge la FSF (Free Software Foundation) con el objetivo de desarrollar software libre y buscar financiamiento para estos proyectos. Hoy en día muchas empresas basan sus ganancias en servicios relacionados con el software libre, algunas distribuyen colecciones de software, otras proporcionan cualquier tipo de servicio técnico, ya sea atendiendo a preguntas de usuarios, subsanando bugs o incluso añadiendo nuevas funcionalidades a dichos software. También se empiezan a ver grandes empresas dedicadas a la producción neta de software libre.

Un programa es software libre para el usuario siempre que, como usuario particular, tengas:

1. La libertad de ejecutar el programa sea cual sea el propósito.
2. La libertad para modificar el programa para ajustarlo a tus necesidades. (Para que se trate de una libertad efectiva en la práctica, deberás tener acceso al código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.)
3. La libertad de redistribuir copias, ya sea de forma gratuita, ya sea a cambio del pago de un precio.
4. La libertad de distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejoras introducidas. (STALLMAN 2004)

Junto a este movimiento del Software Libre viene ligado un término de mucha importancia, el *copyleft*. La idea fundamental del copyleft es que se autoriza la ejecución del programa, su copia, modificación y distribución de versiones modificadas, siempre que no se añada ninguna clase de restricción a posteriori. De este modo, las libertades cruciales que definen el «software libre» quedan garantizadas para cualquiera que posea una copia; estas libertades se convierten en derechos inalienables. (STALLMAN, 2004)

El open source, o código abierto en español, es un término promovido por la "Open Source Initiative"(OSI). Su política es la de tener acceso al código fuente del software y a partir de ahí, en algunos casos, dar posibilidades de modificación y redistribución de copias de dicho software.

Como se observa, ambos términos difieren en cuanto a su política y sus concepciones, esto se ve en la práctica a través de las licencias que ambas fundaciones promueven como válidas para ambos casos.

2.5.1.1 Licencias.

Primeramente hay que tener en cuenta que Licencia de Software, se entiende por un permiso o autorización que concede el autor de un programa informático al usuario del mismo, para utilizar este en una forma determinada y de conformidad con unas condiciones convenidas. La licencia, que puede ser gratuita o no, precisa los derechos y límites de: uso, redistribución, modificación. Además, puede señalar el territorio de aplicación, el plazo de duración y todas las demás cláusulas que el titular del derecho de autor establezca.

Las licencias que cubren la mayoría del software actual, están diseñadas para quitarle o privarle al usuario la necesidad de modificarlo y distribuirlo. Generalmente los programas propietarios se distribuyen en binario, lo que prohíbe acceder al código fuente para cualquier modificación necesitada, por tanto un acceso al código fuente bien autorizado por una licencia y además posibilidades de modificar y redistribuir, podrían darle méritos al software y muchas veces la utilidad que realmente se desea.

Sin duda la Licencia más usada actualmente en los proyectos de Software Libre es la GNU General Public License (GNU GPL). Esta licencia se puede decir que es lo contrario a una licencia de software propietario. Es desarrollada y difundida por la FSF, y esta orientada principalmente a los términos de uso, distribución, modificación del software. Su principal propósito es el de declarar que todo software bajo esta licencia es y será siempre libre. Visto de esta forma el software siempre se verá beneficiado pero, *¿también se verán beneficiados los autores y los usuarios?* Una característica notable y frecuentemente motivo de freno para no usar esta licencia, es el hecho de que si se reutiliza en un mismo software fragmentos de código bajo GNU GPL, y fragmentos de código bajo cualquier otra licencia libre, el resultado final debe de quedar bajo la licencia GNU GPL, sin importar la cantidad o calidad de dichos fragmentos usados.

Dicha característica crea en cierta forma una división en dos grupos de licencias en cuanto a la compatibilidad con la GNU GPL. Por una parte estarían las que permiten la utilización de su código junto al de la GNU GPL y por otra parte las que dadas algunas restricciones en su uso, no permitirían dichas mezclas.

Existen otras variantes o tipos de licencias libres que con características diferentes, han tenido aceptación dentro del desarrollo de software libre. Por ejemplo, existe el Software bajo el dominio público, sobre el cual el autor de dicho software ha abandonado sus derechos de autor o simplemente lo declara así desde el comienzo. Este tipo de

software al carecer de protección alguna de copyright, pudiera ser incluido libremente en otros trabajos o proyectos libres o no libres.

Las licencias al estilo BSD, llamadas así en alusión a los sistemas operativos BSD y a que gran parte de su software están bajo estas licencias, son por lo general compatibles con la GNU GPL, y es altamente permisiva. Se dice generalmente porque la versión revisada de la licencia BSD sí es compatible, la original no, ya que incluye una cláusula sobre publicidad no permitida por la GNU GPL. La principal diferencia de estas licencias es que permiten que se cambien los acuerdos de las mismas, mientras que como se había dicho, la GNU GPL no lo permite al ser copyleft, las de tipo BSD no son copyleft. Otra característica es que no permite garantía de funcionamiento, básicamente las garantías expresas o implícitas están negadas. Son muy conocidas por su facilidad para hacer software propietario desde un programa bajo BSD, lo que muchas compañías como Sun, Microsoft y Apple entre otras, han tomado una base de código BSD y desarrollado software sin retribución alguna. Hay otras licencias como la MIT y la X, que son similares en cuanto a principales características con la BSD. Se puede afirmar entonces que estos tipos de licencias permisivas y no copyleft, se pueden usar como un sistema flexible y viable económicamente para alcanzar un estándar o para desarrollar software propietario, inclusive sin estar peleados con la filosofía del software libre.

Otra licencia muy usada es la Mozilla Public Licence (MPL). Las licencias estilo MPL promueven la colaboración evitando ligarse con la GNU GPL. No son tan permisivas como las de tipo BSD por lo que son denominadas por muchos desarrolladores y usuarios, de copyleft débil. Estas licencias se encargaron desde sus comienzos a solucionar algunos puntos no tenidos en cuenta en la GNU GPL y las BSD. Estas licencias fueron usadas por Netscape Communications Corp. para liberar Netscape Communicator 4.0 y comenzar a desarrollar el proyecto Mozilla. Ha servido de base a posteriores formatos de licencias de organizaciones como la misma OSI entre otras, creando otras incluso con mayores facilidades y libertades.

Licencias para el software en su totalidad existen muchas, aquí se han expuesto las más usadas y que con seguridad estarán presentes a la hora de desarrollar un software. Tanto la OSI, como la FSF tienen en sus sitios oficiales unos listados de licencias que aseguran compatibilidad o no con sus filosofías. Por tanto se recomienda siempre la búsqueda e investigación a fondo de las características de toda licencia bajo la cual

esté el software que se vaya a usar o distribuir, o incluso se fuese a poner a un nuevo software desarrollándolo de cero.

En fin, este trabajo va orientado a cualquier CMS que cumpla con los criterios básicos de:

1. Acceso al código fuente.
2. Sea distribuido bajo cualquiera de las licencias promovidas por la OSI o la FSF.
3. Pueda ser modificado, copiado y distribuido libremente, respetando los términos establecidos en la licencia respectiva.

Capítulo 3: Selección de Portales CMS.

3.1 Introducción.

La implantación de un Portal CMS requiere de muchos pasos y procesos. Al arribar al resultado final se tiene que haber seleccionado un software acorde con las necesidades y los requerimientos de la entidad. Los procesos de selección de dicho software tienen que ser basados en criterios y siguiendo un sistema de pasos lógicos.

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 (CUERDA, 2004).

Lógicamente hay que partir de que no existen dos entidades iguales, en el mejor de los casos existen dos con los objetivos muy parecidos. Es esencial para una selección tener bien claros los siguientes aspectos:

1. Conocer todos los contenidos que serán creados, gestionados y publicados mediante el Portal CMS.
2. Lograr identificar requerimientos con necesidades de la entidad.
3. Especificar las necesidades y funcionalidades requeridas.
4. Describir claramente o mediante ejemplos dichas necesidades.
5. Especificar necesidades futuras.

En la definición de los objetivos y estrategias de la entidad se tiene que:

1. Determinar los objetivos claros que se quieren alcanzar con la implementación de un Portal CMS.
2. Determinar el objetivo del software como Portal CMS.
3. Definir el público objetivo, la audiencia o el mercado.
4. Determinar la funcionalidad deseada.
5. Definir la estrategia de los servicios.

Para la identificación de los requerimientos, término muy utilizado en este capítulo, debe tenerse en cuenta:

1. La participación de los principales actores de la entidad: TIC, directivos generales y usuarios finales.
2. Seguir una metodología de investigación para asegurar que la lista de requerimientos sea suficiente y manejable.
3. Agrupar los requerimientos por categorías.

En el agrupamiento de requerimientos por categorías, las mismas responden a la forma de agruparlos según sus funcionalidades en cada subsistema y de forma general. Básicamente serían las actividades que debe hacer el software a seleccionar.

Este segundo capítulo está redactado en forma de guía teniendo en cuenta facilitar e integrar, al personal que va dedicado el trabajo, todos los elementos necesarios para la toma de decisiones a la hora de seleccionar un Portal CMS libre u open source.

3.2 Pasos en una selección.

Muchos procesos pueden utilizarse para la selección de un software, por lo general suelen usarse técnicas no profesionales de seleccionar el sistema que más se mencione y más propaganda tenga de que cumple a grandes rasgos con sus necesidades, pero si el deseo es el de llevar profundamente un estudio, entonces se deberían de seguir pasos que se centran alrededor de una lista de requerimientos, una lista de muchos productos a evaluar y un grupo o comité de personas encargadas de la selección. El proceso en fin seguiría los siguientes pasos:

1. Descripción a grandes rasgos de los CMS que se encuentren como candidatos.
2. Evaluar opiniones del mercado.
3. Hacer un primer corte de candidatos.
4. Establecer contacto con las empresas o los grupos de personas que realizaron los softwares candidatos.
5. Selección de finalistas.
6. Prevención del proyecto.
7. Referencias.
8. Elección final.

A continuación una explicación más detallada de dichos pasos.

- **Descripción a grandes rasgos de los CMS que se encuentren como candidatos.**

En todo comienzo de un proceso de selección siempre es útil crear una corta descripción de aquellos sistemas que a grandes rasgos cumplan con sus necesidades. Dicha descripción debe orientar al encargado de la selección sobre los puntos principales a tener en cuenta. Para crear esta descripción se toman los requerimientos y un análisis de la entidad.

Un sistema pudiera decir que se corresponde a grandes rasgos con lo que se busca si:

1. Proporciona posibilidades de crear contenidos jugando con roles de personal.
2. Se soporta de una base sólida y un gran número de contribuidores.
3. Proporciona un buen sistema de almacenamiento de contenidos, donde estos puedan ser seguidos y actualizados a cada momento.
4. Brinda flexibilidades en el sistema de publicación al presentar los contenidos mediante plantillas que permitan la personalización de los mismos.
5. Se pueden identificar herramientas que garanticen un correcto flujo de trabajo (workflow) que siga los contenidos durante todo su ciclo de vida.
6. Proporcione ambientes para la publicación, las pruebas y demás tareas por separado.
7. Pudiera ser capaz de integrarse con otros sistemas internos ya existentes en la entidad.

- **Evaluar opiniones del mercado.**

En la actualidad dada la disponibilidad de información en Internet y las opiniones de una gran parte de internautas en foros de discusión, se hace relativamente sencillo obtener datos sobre un sistema dado que goce de cierta popularidad o que haya sido evaluado anteriormente por alguna entidad, especialista o un software de análisis. Para conducir una búsqueda se debería de:

1. *Conseguir recomendaciones:* Por lo general se realiza en foros de discusión sobre temas relacionados o estableciendo correspondencia

electrónica (e-mail) con personas que puedan, dada su experiencia, ayudar en este sentido.

2. *Mantener un análisis de requisitos:* Se debe de tener siempre a mano una lista con los principales requerimientos y criterios para la selección del software en la entidad. Es bueno ir actualizándolos a medida que se vayan obteniendo posibles candidatos.
3. *Desarrollar un pequeño sistema de preguntas:* Esto ayudaría a dar un tratamiento estándar a la hora de evaluar cada sistema por separado.
4. *Aprender a identificar el software según su funcionalidad:* Existe una gama de productos que no cumplen sus necesidades, sin embargo se añaden a la lista de posibles candidatos tras dejarse llevar por lo que serían pobres funciones de gestión de contenidos y ricas personalizaciones.
5. *Hacer una lista de necesidades especiales:* Esta lista irá más allá de necesidades estándares de publicación y gestión. Pudieran tener que ver con comercialización o publicaciones inusuales de la información, integraciones con otros sistemas, etc. No siempre se tendrá la suerte de encontrar un software que brinde soluciones a estas necesidades, pero se pudiera pensar en una integración de estas funcionalidades más adelante.
6. *Documentar la búsqueda:* En muchas ocasiones sucede que determinada fuente de información brinda algo interesante y no se archiva una url mediante la cual retornar a la misma. Precisamente se debe de documentar toda búsqueda mediante documentos que recojan de alguna forma referencias a dichas fuentes de información.
7. *No demorarse mucho en esta etapa:* Ciertamente muchas veces surgen dudas de si incluir o no ciertos sistemas a los posibles candidatos y se pierde mucho tiempo en búsqueda de información adicional o más profunda. Se debería incluir estos sistemas en dicha lista en caso de inseguridad, más adelante se desecharán en caso que sea necesario.

No hay necesidad en esta etapa de ser muy estricto ni muy permisivo, si después de estos pasos, se cuenta con una lista de entre 10 y 20 sistemas candidatos, se puede decir que se va por buen camino.

- **Hacer un primer corte de candidatos.**

En el primer corte para seleccionar los software que más se adecuen a las necesidades, se recomienda haber establecido contacto con la compañía o el grupo productor de los mismos. De ser posible es muy recomendable que se prueben demos de los productos, y que se adquiera toda la documentación posible. También se recomienda investigar a fondo en foros públicos sobre posibles bugs (errores) que haya presentado el software y si fue solucionado.

En un primer corte hay que tener en cuenta que el mismo debe realizarse basado en un análisis lógico de las respuestas adquiridas a través de los criterios anteriormente formulados. Sería conveniente basarse en más de 10 criterios abordados sobre cada producto. En este corte debe de participar todo el personal que de alguna forma jugará un rol importante en el sistema que se implante.

Es recomendado en obras de referencia como (BOIKO, 2001) y (GLASSHAUS, 2003) hacer un primer corte sin muchos pasos, donde los criterios a evaluar sean bien consistentes y den peso a las decisiones. No hay que preocuparse si sobreviven de la lista poco más de 5 productos como candidatos.

- **Establecer contacto.**

Después de un primer corte donde debió reducirse el número de candidatos, se debe profundizar mucho más en los restantes software. Para esto la persona o el grupo asignado, por lo general, elige evaluar un demo de cada producto. Esto trae ventajas como: una verdadera familiarización con el software y las características que defiende el mismo. Si finalmente se decide usar demos, se recomienda lo siguiente:

1. *No requerir de preparación:* Esto se refiere a probar por vez primera la presentación estándar del producto y no la que considere adecuada para sus necesidades. Probando lo mejor de la entidad productora se pueden crear imágenes de calidad, luego entonces se pasarían a probar versiones que de cierta forma se identifiquen como posibles respuestas a sus necesidades.

2. *Entender lo más posible el producto*: La posibilidad de probar un demo, ya sea online o mediante una instalación previa, tiene como ventaja aclarar posibles interrogantes mediante la práctica. Cada software tiene características interesantes que se pueden evaluar a medida que se evalúe el mismo.
3. *Tener muy en cuenta los verdaderos requisitos*: El uso de un demo puede dar imágenes falsas sino se analiza bien. El demo está diseñado precisamente para crear expectativas en los usuarios, por tanto a la hora de la evaluación deben tenerse bien claras las características que se buscan en el producto a cada momento. Por supuesto, dicha evaluación debe tener en cuenta buscar repuestas sobre una base sólida a requerimientos futuros.
4. *Identificar autores reales*: Se recomienda identificar los verdaderos autores de cada módulo o parte del software para de ser posible mantener contacto directo con los mismos. Esto puede hacerse desde la evaluación de un demo.

- **Selección de finalistas.**

En este paso es fundamental evaluar, que tantas respuestas ha conseguido de empresas productoras. No es fundamental siempre el obtener estas respuestas, pues muchas veces se encontrarán productos realizados casi en su totalidad por colaboradores y sería muy difícil contactarlos. Ahora, la importancia del software dentro de la empresa decidirá si es muy importante tener en cuenta el contacto directo y el aval de una empresa productora respetable y líder en el mercado.

Para seleccionar los finalistas, que no deben superar los cinco softwares ni deben ser menos de dos, se recomienda constar con todo un equipo técnico que se encargue de las evaluaciones finales. El equipo técnico se recomienda que conste de:

1. *Expertos legales*: Es importante en caso de términos legales como alguna licencia un tanto confusa.
2. *Expertos en editorial*: La entidad puede tener objetivos de peso en la parte de la creación y edición de los contenidos, para eso se recomienda que se conste con un experto editorial.

3. *Programadores*: Se necesitan diversas opiniones de diferentes programadores expertos en estos sistemas, o al menos que tengan conocimientos y capacidades de asumir cambios futuros desde el punto de vista de adición de módulos o subsanación de bugs (errores).
4. *Arquitectos*: Los arquitectos no solo deben comprender totalmente la arquitectura y los modelos de despliegue del sistema, sino que su decisión es fundamental ya que tienen que ser capaces de prevenir posibles cambios arquitectónicos y saber si el software sería capaz de soportarlos.
5. *Personal especializado en las publicaciones*: Estas personas juegan un importante papel también al mostrar los puntos exactos que el software es débil o fuerte en cuanto a la publicación de los contenidos de acuerdo a los objetivos de la entidad.

En esta etapa, si se cuenta con suficiente tiempo, sería ideal llevar a cabo pequeños cambios en los demos adquiridos y comprobar más a fondo el soporte de cada software candidato, además de unir el equipo teniendo en cuenta todo un proceso que más adelante seguirá de adaptación del producto a la entidad.

Es recomendable también seguir los conceptos y las verdaderas necesidades de la entidad y para esto hacer diagramas de comparación en todos los aspectos a evaluar. Una vez que todos los expertos den su aporte, pasar a la comparación final mediante estos diagramas sería lo más sugerente.

- **Prevención del proyecto.**

Antes de pasar a la fase final, es muy importante tener una visión lo más previsible posible del futuro software que se desearía. La tarea más grande en ese momento es crear los planes visionando un producto modificado. Para esto se recomienda realizar un plan y trabajar basándose en el mismo, prestando más atención en aquellas funcionalidades que el software no brinde y que pudieran agregársele. Para esto hay que tener en cuenta las funciones más importantes que necesiten de módulos extras y crear especies de extensiones en el plan con las mismas bien detalladas, esto ayudaría a estimación de costos y ajustes de horarios de trabajo.

- **Referencias.**

Debido a la importancia que requiere tener referencias de las compañías productoras o los individuos desarrolladores de estos software, se hace necesario, tras haber hecho una evaluación de opiniones del mercado, llevar un poco más profundo este estudio teniendo en cuenta chequear lo siguiente:

1. Sitios Web Oficiales.
2. Logos de las empresas.
3. Eventos, festivales, conferencias y reportes.

Con esto se busca comparaciones entre posibles empresas en varios puntos que justifiquen calidad y mantenimiento del software. Posibles preguntas que se hacen en la búsqueda de referencias serían:

1. ¿Realmente el software es el mismo que se indicaba al obtenerlo?
2. ¿Hace el software lo que promociona?
3. ¿Qué tan cerca están las características del mismo a los requerimientos reales?
4. ¿Con qué otra herramienta esta integrado el software?

- **Elección final.**

Si los pasos antes expuestos han sido seguidos se debe tener en mayor o menor medida lo siguiente:

1. Largas descripciones de cada software.
2. Diagramas mostrando posibles integraciones.
3. Informes de posibles riesgos a correr.
4. Valoraciones de posibles costos adicionales.
5. Estudios de las flexibilidades legales.

Dicha información no por ser amplia determina la selección final, se debe tener en cuenta la voluntad del personal que emprenderá el trabajo de adaptación. Los pasos para la selección deben haber arrojado finalmente un candidato ganador entre opiniones de expertos en diferentes materias, con mucha suerte todas las opiniones apuntarán al mismo software pero casi nunca se cumple esto. En caso contrario se debería:

1. Discutir abiertamente con todo el equipo hasta que se llegue a una decisión lo más aceptada por todos.
2. Crear métodos de comparación basados en gráficos y escalas por cada factor de peso, hasta llegar a la solución más factible económicamente.
3. Extender la decisión. Si no se ha llegado a una decisión final aún, sería lo más recomendable elevar la decisión a alguien con la posibilidad y la capacidad de hacerlo. Por supuesto que para esto se tiene que proveer de los resultados y estudios hechos anteriormente para que sirvan de base en la decisión.

3.3 Descripción de los criterios de selección.

Conocidos los pasos que asegurarían, tras su seguimiento, la correcta selección de un Portal CMS que cumpla con las necesidades de la organización y los requerimientos identificados; sería conveniente conocer como conducir, mediante criterios y aspectos, las comparaciones desde el punto de vista funcional. Esta sección integra descripciones de criterios generalizados y agrupados por categorías. No se considera obligatorio que se evalúen todos estos criterios ya que eso dependería de los objetivos de la implantación o necesidades futuras.

3.3.1 Criterios de negocio.

Las compañías, empresas productoras o grupos de trabajo que producen estos software, pueden ser desde líderes en este campo, hasta totalmente nuevas o desconocidas por muchos. La imagen de un producto puede atraer y dar peso en una selección si el mismo es de una reconocida empresa. En la inmensa mayoría de los casos no se tiene que comprometer a dichas empresas, debido a muchas licencias, las mismas no se comprometen al funcionamiento de un producto tras ser adquirido de otro lugar y haber podido ser modificado por terceras personas.

Algunos de los criterios a tener en cuenta a la hora de intentar obtener datos y finalmente influir en una selección serían:

1. *Historia*: La historia de una empresa puede decir mucho sobre la misma, algunos factores a tener en cuenta serían la experiencia medida en años del trabajo conjunto de su personal, las evoluciones o cambios de objetivos en sus productos, los conceptos que tienen sobre los temas relacionados con los mismos, etc.

2. *Perfil*: Estudiando el perfil de una empresa se determina si la misma dedica sus mayores esfuerzos a temas como la gestión o se preocupa más por la implementación correcta y la innovación de técnicas. Esto influye en los objetivos reales de nuestra entidad y las necesidades del Portal CMS. Cuantos datos puedan obtenerse será mejor.
3. *Actualizaciones*: Es imprescindible obtener información sobre futuras versiones y actualizaciones periódicas del software. No solo en cuanto a fechas, sino en conocer en que consistirán las mismas y que tanto pudiera beneficiar y ahorrar trabajo.
4. *Referencias*: Se recomienda buscar referencias en Portales CMS similares a los candidatos e implantados anteriormente por otras organizaciones. Serían, más que comentarios y opiniones, puntos claves en una decisión.

3.3.2 Criterios generales.

En esta sección se detallarán criterios generales para la selección de Portales CMS, los mismos serán divididos en categorías. En próximas secciones otros criterios serán tratados teniendo en cuenta su clasificación en cuanto a los subsistemas más generales de este tipo de software.

- **Procesos.**

Con estos criterios en forma de preguntas, se busca que tanto saben los productores sobre el ciclo de implementación de su producto, así como comparar estas descripciones con lo que se ha interpretado en las pruebas del mismo. No son preguntas para hacer a los productores, sino al mismo equipo asignado a la selección mediante un estudio.

Las preguntas serían:

1. Descripción de los pasos generales implicados en la implementación del software.
2. *¿Qué factores son necesarios asegurar?* Hay que tener en cuenta si se ofrecen experiencias basadas en errores subsanados.
3. *¿Qué marco de tiempo se utilizó para el desarrollo del software?* También sería conveniente conocer tiempos de demora en la implementación o adaptación de módulos adicionales.

4. *¿Qué servicios profesionales presta la compañía productora?* Por lo general, dadas las condiciones de libres u open source, no existen posibilidades de servicios profesionales por parte de los desarrolladores.
5. *¿Qué servicios se recomiendan para nuestras necesidades específicas?*

- **Ambientes de desarrollo.**

Las herramientas que permitan el trabajo de implementación, diseño, etc, son necesarias conocerlas muy a fondo. Un Portal CMS puede ser todo lo que se desee desde el punto de vista de los objetivos que se persigan, pero se debe tener seguridad en cuanto a si se conoce como modificarlos técnicamente. Estos criterios son bastante técnicos, por lo que deben evaluarse y estudiarse por implementadores como tal. Los mismos son:

1. *Lenguaje de programación utilizado:* Tema que será tratado más a fondo en próximas secciones del trabajo.
2. *Descripción de las herramientas utilizadas.* Necesaria para conocer ventajas y desventajas de las mismas y así conocer la cobertura de modificación que se adquiere con este software.
3. *Habilidades requeridas para la modificación de componentes:* Es recomendable que se investigue que habilidades se necesitan para cada persona según su rol en una próxima adaptación y modificación del software.
4. Herramientas que soporten las funciones administrativas.
5. *Herramientas que automatizan la creación y adquisición de contenidos:* Se debe tener la posibilidad de optar por otras herramientas y tener posibilidad de una fácil adaptación e integración de las mismas.

En el próximo capítulo se aborda más profundamente sobre estos criterios.

- **Localización.**

En los últimos tiempos se vienen desarrollando Portales CMS que integran facilidades según la localización de la información. De no considerarse necesario, no debe profundizarse mucho en estos criterios ya que muchas veces sobran prestaciones de este tipo.

En caso de necesidad de localización se recomienda basar los criterios en:

1. Cómo el software soporta la localización de la información.
2. *Cómo son traducidos los contenidos y como se sincronizan los mismos con los originales*: Deben mantenerse acoplamientos entre los originales y los traducidos para casos de actualizaciones.
3. *Facilidades para la selección de imágenes u otros archivos medias*: Pudiera darse el caso de necesitar varias versiones de gráficos según la región, etc.
4. *Ayudas para la traducción*: Estas ayudas permiten comparar contenidos traducidos en busca de errores.
5. *Versiones de elementos por separado*: En estos casos es provechoso poder separar los elementos a traducir o versionar, de los que no lo necesitan.
6. Funcionalidades de identificación automática de regiones.

- **Ayuda Técnica.**

No es recomendable lanzarse a seleccionar un Portal CMS si no cuenta con personal que pueda ayudar técnicamente en caso de cualquier problema. En caso de no constar con este tipo de personal experto, se deberían evaluar en cuanto a:

1. *Posibilidad de ayuda técnica directamente desde los productores*: Por lo general es poco probable.
2. *Posibilidad de solicitar ayuda a empresas comerciales*: Muchas empresas basan su comercio en el mantenimiento de Portales CMS basados en software libre o de códigos abiertos. Si el software seleccionado es de gran referencia y uso por parte de muchas entidades, se pudiese encontrar un mayor número de terceras empresas que se dediquen al mantenimiento del mismo.
3. Idiomas disponibles de la ayuda.
4. *Facilidades de ayudas en el propio software*: Esto se refiere a las FAQs (Sistemas de Preguntas Frecuentes), listados de bugs solucionados, tutoriales para principiantes, etc.

5. *Software y Hardware que cubre la ayuda:* Muchas veces, por ejemplo, el software está concebido para su funcionamiento sobre varios sistemas operativos, pero no trae manuales de instalación sobre alguno en particular.

- **Comportamiento del software.**

Es necesario obtener respuestas generales acerca del comportamiento del software en distintos ambientes, para esto debe conocerse:

1. Sobre qué sistema operativo puede ejecutarse el software.
2. *Sistemas operativos que requieren las herramientas adicionales:* Dichas herramientas pueden ser bases de datos u otras que el autor de contenidos necesite para la creación del mismo.
3. Navegadores Web que soporta el software.
4. *Habilidades necesarias para la instalación del software:* Estas habilidades pueden investigarse mediante experiencias de terceras organizaciones o personas, o mediante la ejecución de demos.
5. *Modelos que se siguen en la distribución de ambientes de manejo y publicación de los contenidos:* Esto se refiere a la determinación de como el CMS mantiene ambientes distribuidos y organizados para que se contribuya desde cualquier lugar geográfico.
6. *Diferencias entre el comportamiento del CMS tras o fuera de un cortafuego (Firewall):* Con detalles como estos se determina si usar el Portal CMS para una intranet o para un sitio en Internet. También ayudan a prevenir posibles problemas de acceso de contribuidores externos.

3.3.3 Criterios basados en el subsistema de colección de contenidos.

Los criterios a tener en cuenta en los subsistemas de un CMS pueden ayudar a determinar por cual funcionalidad decidirse. Los mismos están basados en la estructura básica que generalmente siguen estos subsistemas como se explicaba en el primer capítulo de este trabajo.

En esta sección se abordarán los criterios a tener en cuenta según los subsistemas de colección de los software candidatos.

- **Criterios relacionados con la creación de contenidos.**

Los criterios básicos dentro de la creación de los contenidos serían:

1. *¿Utiliza el software herramientas conocidas para la creación? ¿Cual o cuales?* Con estas preguntas se abogaría por herramientas familiares en el sistema para la creación. En caso negativo habría que estudiar si sería capaz entonces de soportar la adaptación de dichas herramientas.
2. *Tipos de archivos que el software puede almacenar:* Lo más ideal sería un sistema que tuviera soporte para almacenar archivos creados en cualquier formato y por cualquier herramienta.
3. *Tipos de archivos a los que el software puede hacerles un proceso de reconocimiento y conversión para el almacenamiento en bases de datos:* Por lo general los procesos de conversión necesarios dentro del subsistema de colección de contenidos, tienden a complicarse para algunos archivos como los de tipos *media*, generalmente desde un principio con el reconocimiento del formato.
4. *Flexibilidad y sencillez del software para autores novatos o casuales:* Actualmente los usuarios tienen la posibilidad de crear contenidos aún sin estar registrados en el sistema si así de desea. El CMS debe ser muy simple e intuitivo garantizando esto.
5. *Posibilidad de mostrar elementos del subsistema según los roles de los usuarios:* Esto permitiría personalizar el subsistema para que determinados usuarios creen determinados contenidos
6. *Soporte de descarga y agregación de imágenes u otros archivos por parte de autores:* Puede ser que los autores al crear contenidos añadan también archivos extras como imágenes entre otros, en ese caso sería recomendable conocer con antelación si el sistema lo soportaría.
7. Posibilidad de crearse automáticamente personalizaciones.
8. *Posibilidad de modificación de las plantillas:* Esto es muy importante ya que siempre se necesita personalizar plantillas HTML propias de acuerdo a políticas de diseño de la organización.

9. Capacidad de creación de contenidos aún estando el usuario en estado offline.
10. Posibilidades de que los autores tengan vistas previas de los contenidos creados antes de enviarlos al sistema.
11. Posibilidad a los autores de realizar revisiones de ortografía en plena creación de contenidos.

- **Criterios relacionados con la adquisición de contenidos.**

Los criterios descritos en esta sección son de importancia dada la tendencia a la adquisición de contenidos desde Portales Web informativos u otras herramientas. . Los mismas son:

1. *Herramientas que emplea el CMS para la adquisición de contenidos:* En la actualidad suelen usarse herramientas que exploran fuentes de información y automáticamente capturan los contenidos. También existen las de sindicación, las cuales se encargan de aceptar contenidos enviados por otras fuentes, entre otras funciones.
2. Integración del CMS con portales y sistemas de publicación existentes en la organización.
3. Posibilidad de recuperar contenidos existentes en viejas bases de datos en la entidad, de forma automática.

- **Criterios relacionados con la conversión de contenidos.**

1. *Herramientas de conversión en el subsistema de colección:* Algunos productos no se preocupan mucho por una adecuada conversión, sin embargo otros ven esto como muy importante para un adecuado almacenamiento e implementan o usan herramientas para este fin.
2. *Forma en que se soporta la conversión de textos y archivos como imágenes o medias:* Esta no es una funcionalidad que se requiera siempre, pero algunos CMS pueden convertir contenidos a varios formatos y mostrarlos según la petición del usuario.

- **Criterios relacionados con la agregación de contenidos.**

El CMS necesita para su fácil manejabilidad de la utilización del concepto de metadatos. Casi se puede afirmar que la mayoría de estos sistemas han optado por los metadatos reconociendo sus ventajas. En esta sección del trabajo se exponen criterios para conocer que tanto y de que forma el software candidato agrega los contenidos. Los mismos son:

1. *Herramientas para el manejo de metadatos*: Dichas herramientas se basan en diseñar listas de metadatos y encontrar los mismos en diferentes archivos.
2. Posibilidad del sistema de obtener metadatos de otras fuentes.
3. Posibilidad de llegar a ofrecer listas de metadatos a otros CMS o bases de datos externas.
4. *Procesos inversos de revisión y validación de listas de metadatos*: Esto se explica cuando un CMS puede dar la posibilidad de manejar las listas de metadatos, una vez hechas, en cuanto a revisión periódica de las mismas por ejemplo.
5. Tipos de metadatos que soporta el sistema.

3.3.4 Criterios basados en el subsistema de gestión de contenidos.

Unas de las cuestiones principales a conocer es todo lo relacionado con el manejo que los Portales CMS realizan con los contenidos. Dicho manejo viene dado por el correcto almacenamiento y la administración de los mismos.

- **Criterios relacionados con el almacenamiento de los contenidos.**

Para conocer la estructura del almacenamiento de un CMS, se debe evaluar entorno a los siguientes criterios:

1. *Forma en que el CMS almacena todos los contenidos*: Lo importante sería conocer exactamente que medios se usan como diferentes tipos de bases de datos, sistemas de archivos, etc.
2. *Capacidad del sistema de comprobar la validez de enlaces y referencias dentro de los contenidos*: Esto se refiere a que muchos sistemas pierden referencias a imágenes y links expuestos dentro de los textos de información.

3. *Sistemas de búsqueda e indexación que soporta el sistema:* Lo mejor es que el sistema sea capaz de soportar indexaciones y consultas, metadatos, tipos de búsquedas, etc.

- **Criterios relacionados con la administración de los contenidos.**

Por lo general todos los CMS cumplen con requisitos de una adecuada administración, siendo esto algo que caracteriza a estos sistemas. La diferencia tiene que buscarse en estudiar cual de las mismas se adapta más a los roles que se definieron en la entidad y la forma de administrar los contenidos en la misma. Los criterios que ayudarían en esta selección serían:

1. Manera en que el CMS permite a usuarios no expertos manejar sus propios contenidos dentro del sistema.
2. *Utilidades que provee el sistema para la administración del mismo por el personal encargado:* Se trataría de identificar flujos de trabajo (Workflow), etc.
3. Posibilidades del sistema de verificar y manejar automáticamente enlaces internos y externos rotos.
4. *Roles del usuario que se utilizan y herramientas que provee el sistema para el manejo y la manipulación de contenidos mediante los mismos:* Hay muchas opciones de manejar el contenido mediante los diferentes roles de usuarios. Los sistemas incluyen herramientas para la configuración de estos roles y la delegación de derechos, etc.

- **Criterios relacionados con el flujo de trabajo (Workflow).**

Los Portales CMS pueden variar en no tener soporte para flujos de trabajo o tener integradas herramientas que fortalezcan módulos que requieran de ellos. Es importante hacer un estudio para saber si se necesitan realmente herramientas para estas cuestiones y luego comparar detalladamente cada sistema por separado mediante criterios como estos:

1. Herramientas de workflow que incluye el CMS.
2. Posibilidad de integración de otras herramientas con el sistema. Ayuda de como realizar la integración.

3. *Métodos que accionen los pasos del flujo de trabajo:* Algunos sistemas tienen acciones como el arribo de un mensaje, e-mail, determinado archivo, etc.
4. Actividades del sistema que se realizan siguiendo flujos de trabajo.
5. *Cualidades del contenido que se utilicen por el sistema para dirigir los flujos de trabajo:* Lo ideal sería que el sistema de workflow pudiese utilizar cualquier elemento de los contenidos que se creen.
6. *Conocimientos requeridos para la creación de determinados flujos de trabajo:* Los sistemas mejor concebidos realizan flujos de trabajo mediante ambientes visuales muy intuitivos.
7. Posibilidad de integración de los servicios de e-mail a los procesos del workflow.

- **Criterios relacionados con la integración.**

Es muy importante conocer de antemano si la entidad tiene vínculos con otros sistemas externos y se piense en ampliarlos mediante una integración con el Portal CMS a implantar. Lógicamente se tiene que tener una descripción detallada del funcionamiento de dichos sistemas. Los mismos pudieran ser:

1. Sistemas de Gestión de documentos.
2. Sistemas de Gestión de archivos media.
3. Bases de datos de catálogos, etc.

En cualquier caso, algunas de los criterios importantes para la selección teniendo en cuenta esto, pudieran ser:

1. Capacidades del CMS de conexión a otros sistemas.
2. Posibilidades de agregación de alguna funcionalidad que permita dichas integraciones.
3. Estándares que se siguen para los accesos.

- **Criterios relacionados con el control de versiones.**

Por lo general un buen CMS debe tener la capacidad de controlar versiones de contenidos. Son funcionalidades que tal vez no sean imprescindibles como necesidad,

pero si muy útiles. Cabría entonces seleccionar un sistema dándole respuesta a las siguientes preguntas:

1. Posibilidad de los usuarios de controlar los contenidos almacenados.
2. Capacidades de hacer salvos de contenidos que pudieran requerir ser restaurados.
3. Forma en que se maneja la fijación del contenido: Esto se refiere a la jerarquía de los contenidos.
4. Posibilidad de los autores de almacenar trabajos inconclusos para ser terminados en otro momento.
5. Capacidad de comparación de contenidos: Estas capacidades pueden ser cómodas si se busca la forma de poner las versiones una junto a la otra y así monitorear la comparación.

- **Criterios relacionados con los reportes.**

La elaboración de informes y reportes a partir de funcionalidades del CMS, son fundamentales para muchas estrategias de entidades. Hay que tener en cuenta por supuesto las actividades que se quieren supervisar y a partir de ahí basarse en aspectos como estos:

1. *Actividades que supervisa el sistema:* En los mejores casos se pueden escoger entre Portales CMS que van muy profundo al punto de reportar todas las actividades hechas por los usuarios, zona geográfica de los mismos, sistemas operativos que acceden al mismo, navegadores Web, etc.
2. *Generación de mapas de las páginas que produce:* Este tipo de funcionalidad es útil para reportes de publicación.

- **Criterios relacionados con la seguridad.**

La seguridad es muy importante y tiene que ser sigilosamente atendida. Nunca está de más excederse. Para comparar portales CMS en cuanto a su seguridad se recomienda el detallado estudio de un especialista. Entre aspectos básicos para una selección se pudieran citar:

1. Cómo el sistema maneja los permisos mediante roles.

2. Implementación de una forma jerárquica de agrupación de usuarios.
3. Posibilidad del CMS de controlar el acceso a contenidos secretos o bajo copyright.

- **Criterios relacionados con el funcionamiento del software en la gestión.**

Antes de comenzar una implantación de un Portal CMS libre u open source, es fundamental tener en cuenta desde su selección que tantos cambios se pueden añadir a dicho sistema existente y concebido por otras personas. Por eso es importante conocer el funcionamiento del mismo en cuanto a:

1. Requerimientos de hardware y capacidades de red: El truco es básicamente basarse en el nivel de uso del software. El uso puede ser de las publicaciones en fin, o del que haga el personal del sistema.
1. Escalabilidad arquitectónica.
2. Limitaciones de almacenamiento.
3. Capacidad de réplicas según la región.

3.3.5 Criterios relacionados con el subsistema de publicación de contenidos.

En esta sección se trazarán bases de comparación de acuerdo a criterios relacionados con la publicación de los contenidos de la entidad. Básicamente lo principal es identificar el estado de este subsistema y que tanto se puede adaptar a los requerimientos de la organización.

- **Criterios relacionados con las plantillas.**

Los siguientes aspectos evaluarían las capacidades de las plantillas de los subsistemas de publicación:

1. Lenguajes de programación utilizados para el desarrollo de plantillas.
2. Soporte del CMS de creación de páginas estáticas para publicación de determinados contenidos.
3. *Métodos para la creación dinámica de componentes de navegación:* Estos componentes suelen ser formas de búsquedas, tablas de contenidos, enlaces, ayudas, etc.

4. Soporte de integraciones con sistemas externos por parte de las plantillas.
5. *Forma en que se realiza la personalización en las plantillas:* Es importante llegar a conocer su funcionamiento para la realización de futuras plantillas propias.

- **Criterios relacionados con el Despliegue de los Contenidos.**

El despliegue de los contenidos puede verse de diferentes formas según las capacidades de los CMS. Por estar tan ligado el despliegue con el subsistema de publicación, se presentan los siguientes criterios que ayudan a verificar las mencionadas capacidades:

1. Forma en que el CMS puede, de acuerdo a los tipos de contenidos, desplegar los mismos a varios destinos siendo estos los apropiados.
2. Establecimiento de horarios de publicación y expiración del contenido.

- **Criterios relacionados con las Plataformas de Publicación.**

En algunos casos la entidad requiere de capacidades en su Portal para publicar o convertir informaciones en varios formatos. En estos casos, la evaluación de criterios relacionados con estas capacidades será:

1. Posibilidad del sistema de integrarse con sistemas de E-Mail para la transmisión de contenidos mediante este medio.
2. Posibilidad de enviar E-Mails personalizados a determinados usuarios como estrategias de la organización.
3. Capacidades del software para producir publicaciones impresas.
4. *Posibilidades de asociar contenidos desde diferentes dispositivos de almacenamiento:* Ejemplo: CD-ROMs, Servidores FTP, etc.
5. Posibilidades de crear documentos descargables a partir de contenidos en varios formatos.

- **Criterios relacionados con la Personalización de los Contenidos.**

Las personalizaciones en este tipo de portales se pueden hacer mediante formas muy variadas que incluyen herramientas. Es importante llegar a comprender las formas de

llevar a cabo las mismas y decidir cuál es más conveniente para la entidad. Los criterios a tener en cuenta en las personalizaciones serían:

1. Utilización de herramientas de personalización.
2. *Tipos de interfaces se provee para llevar a cabo por parte de usuarios la modificación de reglas de personalización:* En estos aspectos existen sistemas que incluyen interfaces muy intuitivas para la creación de reglas, otros menos avanzados requieren de trabajos manuales.
3. Capacidad del sistema de almacenar datos del usuario como acciones del mismo, respuestas a preguntas, etc.
4. *Soporte del manejo de datos de perfiles de otros clientes, provenientes de otros CMS similares:* Sindicación de perfiles de usuarios.
5. Forma en que interactúan los perfiles de usuarios con las personalizaciones.

3.4 Sistemas de referencia.

De las anteriores secciones pueden surgir dudas lógicas teniendo en cuenta el desconocimiento o la poca práctica de búsquedas en Internet, por parte de los desarrolladores de estos temas. La búsqueda puede ser larga y tediosa e incluso no arrojar los mejores resultados. Las dudas más generales pueden venir en el orden de cuestionarse lo siguiente:

1. ¿Donde y cómo encontrar los posibles candidatos?
2. ¿Tendrían que evaluarse uno a uno para comprobar todas sus funcionalidades?
3. ¿Cómo encontrar listados de funcionalidades de mano de expertos y no de medios no confiables del todo, como los foros de discusión?

En esta sección este trabajo brinda referencias a organizaciones y sitios que con ayuda de algunos servicios eliminarán cualquier pregunta de este tipo.

En primer lugar desde el año 2001 existe una organización internacional que agrupa a desarrolladores y usuarios de sistemas manejadores de contenidos open source, la OSCOM (Open Source Content Management)(oscom.org). Dicha organización se encarga de organizar eventos, definir y promover estándares y registrar proyectos de este tipo.

Para facilitar precisamente la búsqueda de sistemas gestores de contenidos, la OSCOM promueve y enlaza varios sitios dedicados a este fin. Entre los mismos se encuentran:

1. *CMS Matrix* (cmsmatrix.org): Este sitio cuenta, según datos recogidos del mismo en el momento de la realización de este trabajo, con exactamente 532 CMS registrados (no todos libres u open source). Además brinda reportes para facilitar y hacer óptima la selección como los softwares más comparados y más vistos por los miles de usuarios que frecuentan el mismo. Se basa principalmente en comparaciones de los CMS registrados según 145 criterios organizados por categorías.
2. *Open Source CMS* (opensourcecms.com): Se encarga de facilitar la prueba de más de 130 CMS de todos tipos mediante demos o referencias a los mismos. Entre las categorías que agrupan a los CMS en este sitio están los portales, los blogs o bitácoras, sistemas de comercio electrónico, sitios para comunidades o grupos de trabajo, foros de discusión, sistemas de aprendizaje electrónico, galerías de imágenes, wikis, entre otros.
3. *CMS Info* (cmsinfo.org): Sitio dedicado a la referencia y búsqueda de información de casi 100 CMS. Basa sus búsquedas en comentarios y referencias de trabajos relacionados con la gestión de contenidos de autores reconocidos mundialmente.
4. *CMS Watch* (cmswatch.com): Se especializa en brindar reportes, y evaluaciones particulares de CMS para necesidades específicas. Además brinda soluciones mediante canales de tecnologías donde agrupa artículos y libros de referencias.

Las listas de funcionalidades y documentación de estos softwares hacen que se usen mucho como referencias estos sitios en procesos de selección.

Capítulo 4: Tecnologías libres en la modificación e implantación.

4.1 Introducción.

La selección de la herramienta más adecuada, no acaba basándose en una serie de criterios. La visión de modificación debe ser tal, que en ella debe ir mucho conocimiento. Debe de haberse tenido cuidado en no haber obtenido herramientas dedicadas a problemáticas diferentes. Tras una buena definición de objetivos seguramente la decisión fue correcta, logrando obtener el software deseado.

La verdadera solución a los problemas de la entidad comienzan una vez adquirido el software y pasando a la transformación o modificación del mismo. En algunos casos pudiera suceder que:

1. El Portal CMS no supera las expectativas.
2. Mala definición de objetivos.
3. Se hubieran resuelto las problemáticas con otras tecnologías más sencillas.

Es vital, una vez seleccionada la herramienta, plantear pasos a seguir para la modificación e implantación final. Los pasos tendrían que seguir la siguiente base:

1. *Identificación de elementos prioritarios*: En la actualidad muchos Portales CMS vienen con demasiadas funcionalidades extras, por lo que es necesario centrarse en aquellas identificadas como las ideales para el aporte de soluciones.
2. *Subcontratación de personal o empresas especializadas*: Esto por supuesto se sugiere para casos donde el personal mismo de la entidad no asuma, por problemas de preparación, dichos procesos de modificación.
3. *Identificación y estudio del hardware puesto a disposición para la implantación de la herramienta*: Es necesario conocer las capacidades de hardware con que consta la entidad donde se implantará o publicará finalmente el portal CMS. Esto ayudaría a trazar planes futuros de rendimiento, entre otras posibilidades.
4. *Trazar estrategias de desarrollo sobre el software para el aumento de capacidades del mismo según necesidades futuras de la entidad*: Nuevas

- funcionalidades, nuevos flujos de trabajo, más capacidad de personalización, etc.
5. *Capacitación del personal de la entidad para el uso del software según cada rol en los flujos de trabajo:* Muchos roles como el de revisión y aceptación de contenidos creados, requieren de personal capacitado y tal vez en la entidad no exista ese nivel de rigurosidad o cultura de trabajo digital, por lo que se hace necesario capacitar a las personas en cuanto a sus roles.
 6. *Familiarización con herramientas añadidas tales como editores de texto online y foros de discusión:* En algunos casos las herramientas de edición de texto y los foros de discusión, entre otras; requieren de una familiarización previa para poder entender su forma de trabajo y realizar para los usuarios secciones explicativas como los sistemas de preguntas frecuentes dentro del mismo portal.

Para cualquiera de los casos es imprescindible el conocimiento de las alternativas tecnológicas para la modificación e implantación final (Ver Figura 9). Más aún si la propia entidad decide hacerse cargo de dichos procesos con su propio personal. El conocimiento previo de las características de las distintas plataformas de desarrollo de estos sistemas puede ayudar a la toma de decisiones en cuanto a ventajas y desventajas de las mismas.

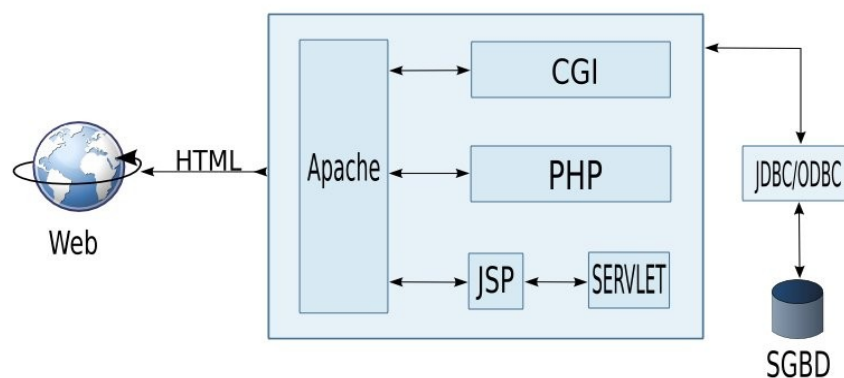


Fig 9: Alternativas Tecnológicas.

En este capítulo se abordarán las tecnologías y tendencias libres u open source más usadas en el desarrollo de Portales CMS, sirviendo de base no solo para tomas de decisiones, sino también como punto de partida para la preparación del personal que asuma la modificación y adaptación de dichos sistemas.

4.2 Plataforma LAMP.

Los CMS se han desarrollado sobre distintas arquitecturas a pesar de ser sistemas relativamente nuevos y no llegar prácticamente a una década desde su creación. Sin dudas la plataforma mas extendida en el desarrollo de CMS libres u open source es la plataforma LAMP.

El término LAMP es empleado para definir el trabajo conjunto del sistema operativo GNU/Linux, el servidor Apache, el gestor de bases de datos MySql y los lenguajes de programación PHP, Perl o Python. Lo más interesante es que cada tecnología fue creada por separada y sus características son las que hacen el perfecto acople de las mismas.

Entre las características más relevantes de LAMP, se tiene que la misma puede funcionar en una amplia variedad de hardware, con requerimientos relativamente pequeños y una alta estabilidad. LAMP es una alternativa muy eficiente para medianas y pequeñas empresas por sus bajos costes en hardware y software. Para el desarrollo de Portales Web es genial debido a su robustez y solidez.

Algunas de la ventajas de usar LAMP son:

1. Soporte a gran cantidad de arquitecturas.
2. Código relativamente sencillo y con pocos cambios de una plataforma a otra.
3. Posibilidad de incremento de servicios y funciones desde el propio código fuente.
4. Actualizaciones mediante internet.

Como posibles desventajas se pudieran citar:

1. Dado el uso del sistema operativo GNU/Linux, pudiesen existir problemas de conocimiento del mismo al no ser el sistema operativo más conocido y menos en nuestro país.
2. Las actualizaciones de las tecnologías por lo general sugieren conocimientos profundos.
3. Personal capacitado para la integración y mantenimiento de su funcionamiento.

La plataforma LAMP sin dudas consigue un nivel muy profesional en soluciones de desarrollo de Portales Web CMS con muy bajo coste de software y licencias, pero a

cambio requiere de un nivel de administración un poco más complejo que con Microsoft Windows como sistema operativo por ejemplo. El uso de soluciones libres como estas, ofrece libertades de copias y redistribuciones, además de posibilidades de obtención mediante internet y sin coste alguno. También ofrece posibilidades de adaptación según necesidades al constar con el código fuente de cada parte de la plataforma.

La forma más común de usar LAMP, como se mencionó anteriormente, es mediante el cuarteto Linux+Apache+Mysql+Php, pero bien soporta variaciones como por ejemplo el uso de otros sistemas operativos como OpenBSD, otro gestor de bases de datos como PostGreSql y otros lenguajes de programación como Python o Perl.

4.2.1 Sistema Operativo GNU/Linux.

El estudio a fondo de las prestaciones de este cada vez más popular sistema operativo, no es como tal un objetivo de este capítulo. Como parte de la mencionada plataforma LAMP solo compromete hacer mención de sus principales funciones dentro de la misma, así como sus principales ventajas y desventajas en dichas funciones.

GNU/LINUX es el sistema operativo por excelencia en la plataforma LAMP, es desarrollado y difundido por el proyecto GNU. El nombre de Linux viene dado por el nombre del kernel que usa y este a su vez viene de su creador Linus Torvalds. Por otra parte las siglas GNU corresponden al proyecto fundado por Richard Stallman tras la idea de desarrollar un sistema operativo completamente libre.

GNU/LINUX es un sistema operativo, de libre distribución y por lo general gratuito, basado en UNIX, este a su vez tiene características notables como su portabilidad a diferentes tipos de ordenadores, características que hereda GNU/LINUX. Otra característica importante es su flexibilidad, no fue ideado desde un principio para su fácil manejo, pero se realizan grandes esfuerzos, entre muchos resultados alentadores, para lograrlo. Ejemplo de esto se pueden citar las interfaces gráficas GNOME y KDE, las cuales han hecho que este sistema operativo rompa los esquemas de utilización solamente en servidores y suban los niveles de utilización en entornos Desktop, o sea más personalizados e intuitivos. Dentro de los últimos avances a resaltar de GNU/LINUX, se pudieran citar la inclusión en el mismo de la suite ofimática OpenOffice.org y el navegador Mozilla, entre otros.

La forma de encontrar este sistema operativo es mediante Distribuciones. Una Distribución no es más que el agrupamiento del kernel Linux y otras muchas aplicaciones de uso general o no. Entre las distribuciones más conocidas están RedHat,

Debian, Gentoo, etc. Cada una tiene características propias que las diferencian en menor o mayor medida.

Las prestaciones que GNU/LINUX presta vienen dadas mayormente en el orden del uso por parte de usuarios de conocimientos avanzados, lo que lo ha convertido en ideal para muchos en trabajos a través de redes, proyectos de programación y servidores. Su principal característica es la estabilidad, de ahí que importantes compañías (Intel, IBM, etc.) apoyen y financien estos proyectos y cada vez más empresas apuesten por su uso. Las últimas versiones del kernel han obtenido una clara modularidad, resultando estar bien diseñadas para añadir o remover componentes y así lograr una optimización para una plataforma en particular.

La inclinación por el uso de Linux viene dada debido a los mecanismos y formas básicas de funcionamiento que garantizan que cada programa pueda acceder de la forma más adecuada a los recursos del ordenador. También por el buen control de memoria y el manejo de ejecuciones de programas, así como la creación y control de interfaces gráficas de usuarios, etc.

4.2.2 Servidor Web Apache.

La piedra angular de cualquier portal es, con toda seguridad, el servidor web; el software encargado de atender las peticiones de los clientes y enviarles las páginas web solicitadas. Existen multitud de paquetes software para montar servidores web, muchos de ellos distribuidos como software libre y siendo, sin lugar a dudas, el más popular de todos Apache. (GAYO, LÓPEZ, VINUESA, LABRA, CUEVA, 2001).

El servidor HTTP Apache es un servidor HTTP de código abierto desarrollado originalmente para plataformas Unix (GNU/LINUX, BSD, etc.), Windows y otras que implementen HTTP. Fue implementado por la Apache Software Foundation dentro del proyecto HTTP Server. En un principio era concebido para que aplicase parches a otro servidor, el NCSA. En la actualidad presenta muchas mejoras claves como un mejoramiento en el soporte para plataformas como Windows y soporte también de IPv6.

El objetivo principal es el de proporcionar un servidor web seguro, eficiente y extensible, que provea servicios HTTP acorde con los estándares de dicho protocolo. Sus características de poseer mensajes de error altamente configurables, negociación de contenido y bases de datos de autenticación; cumplen los objetivos requeridos al punto de ser el servidor web más utilizado no solo en la actualidad, sino desde 1996 según

estudios de NetCraft (netcraft.org). Ver el siguiente gráfico tomado de NetCraft que demuestra el uso creciente de Apache (Figura 10):

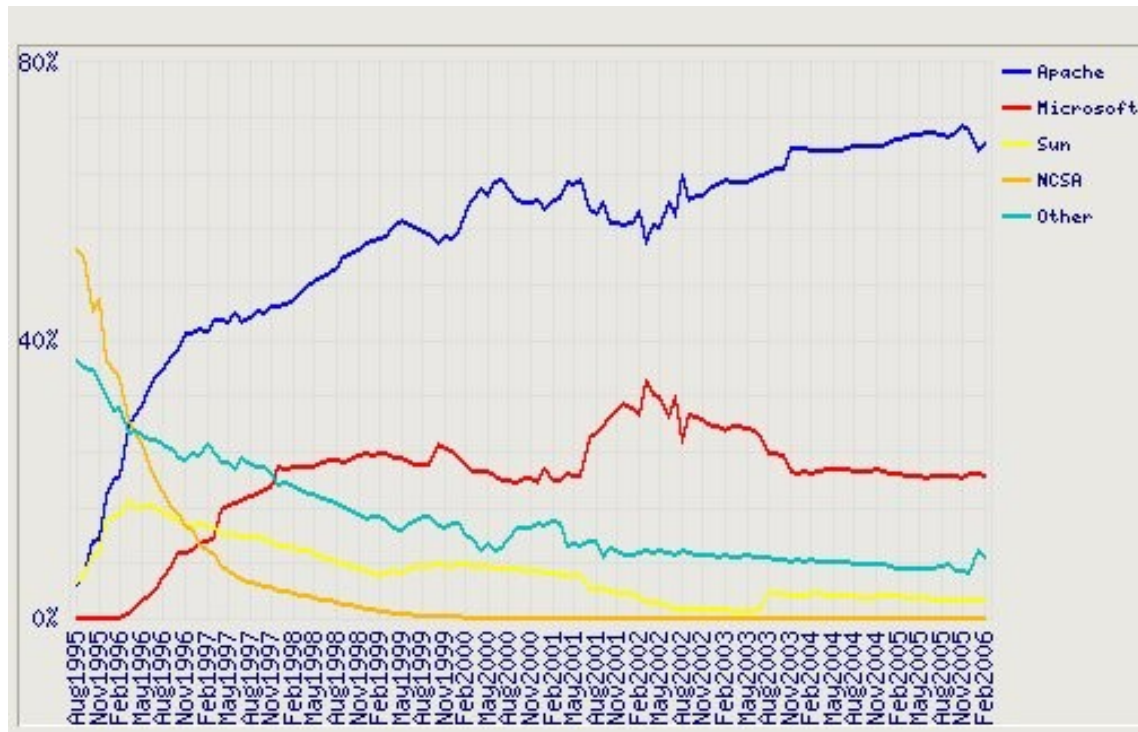


Fig 10: Uso del servidor Apache frente a otros en el periodo 1995-2006.

Su principal característica es su modularidad, lo cual hace que conste de muchas funcionalidades provistas por módulos. Los más notables son:

1. `mod_auth_ldap`: Permite la autenticación contra un servidor LDAP.
2. `mod_proxy_ajp`: Se encarga de enlazar con el servidor jakarta Tomcat (JSP, servlets).

Y otros módulos que pudieran ser añadidos para el trabajo con diversos lenguajes como:

1. `mod_perl`: Para el trabajo de páginas dinámicas en Perl.
2. `mod_php`: Páginas dinámicas en PHP.
3. `mod_aspdotnet`: Páginas dinámicas en .NET
4. `mod_python`: Páginas dinámicas en el lenguaje Python.

Su robustez y su gran lista de prestaciones justifican que se combine perfectamente con las distribuciones de Linux, al punto de que Apache viene integrado en la mayoría de los casos con las mismas. Su característica de ser software open source brinda la posibilidad de que una gran cantidad de desarrolladores de todo el mundo colaboren en su mejoramiento.

4.2.3 Sistemas de Bases de Datos.

Por lo general los CMS en su condición de libres u open source, descartan la posibilidad del uso de gestores de bases de datos como Oracle y DB2, aún cuando sean de las más usadas en el mundo. En la actualidad se nota como opciones libres están llegando al mercado para quedarse dadas sus prestaciones.

Entre las alternativas más utilizadas dentro del software libre se encuentran:

1. MySQL.
2. PostgreSQL.
3. InterBase.

La documentación disponible sobre las mismas son sobradas. Este trabajo solo intentará brindar los principales criterios de comparación de las mismas ante cada situación.

4.2.3.1 MySQL.

MySQL es un Sistema Gestor de Bases de Datos desarrollado y mantenido en la actualidad por la empresa MySQL AB. No obstante se encuentra bajo la llamada filosofía del software libre, o sea, obtención gratuita y código fuente disponible bajo la licencia GNU GPL. Es en la actualidad uno de los más usados dentro de esta rama de softwares.

En un principio este software tenía como objetivo el ser muy veloz, cosa que logra sacrificando elementos esenciales como las transacciones y la integridad referencial. A pesar de ello, muchos desarrolladores de CMS obtaron por su uso apostando por su simplicidad, no obstante en la actualidad esos elementos que faltaron ya se han ido agregando gracias a colaboradores de la comunidad del software libre.

Entre las características más notables que justifican su uso en el desarrollo de Portales CMS están:

1. Capacidad de estar disponible en varias plataformas y sistemas.

2. Opciones según las necesidades: Es configurable en su almacenamiento según las necesidades de velocidad o gran número de prestaciones.
3. Amplio subconjunto del lenguaje SQL.
4. Búsqueda e indexación de campos de textos.
5. Replicación.
6. Herramientas de administración intuitivas.
7. Conectividad segura.

Según la propia empresa MySQL AB (mysql.com), deben de existir en la actualidad más de seis millones de gestores de este tipo en pleno funcionamiento, superando la disponibilidad de algún otro. Este SGBD también incluye una licencia comercial para poder ser utilizada en softwares no libres.

4.2.3.2 PostgreSQL.

PostgreSQL es un SGBD relacional liberado bajo licencia BSD. Es resultado de trabajos y proyectos que datan de los años 80 y que fueron abandonados a principios de la década del 90. Precisamente gracias a su licencia el proyecto fue retomado por desarrolladores de software libre.

El uso en Portales CMS viene dado gracias a sus características, entre la cuales se destaca su buena capacidad en sistemas verdaderamente grandes y con muchos contenidos a manejar. Otras características son:

1. Herencia de tablas.
2. Llaves Foraneas (Foreign keys).
3. Vistas.
4. Tipos de datos y operaciones geométricas.
5. Triggers: Disparadores
6. Capacidad de albergar programas en el servidor en varios lenguajes.
7. Integridad transaccional.

Una limitación que se le señala es el alto consumo de recursos en comparación con MySQL por ejemplo. Pero realmente, si se logra aumentar la memoria disponible y el número de CPUs, escala muy bien debido a su arquitectura de diseño. Otra limitación

era la lentitud, pero según la página oficial (postgresql.org) ha sido resuelta en últimas versiones, logrando una rapidez a la par de otros SGBD similares como MySQL.

4.2.3.3 InterBase.

Nace como un SGBD propietario, pero a partir de su versión 6, la empresa propietaria proporcionó una versión en código libre. Se caracteriza por ser una base de datos consolidada y a sus últimas versiones se le han añadido características de SQL avanzado y se le han mejorado las herramientas de configuración y administración.

Debido a su arquitectura no compartida, este SGBD se comporta rápido frente a pocos usuarios, pero el rendimiento del mismo suele caer una vez que aumente el número de usuarios. Debido a que se le han ido uniendo desarrolladores al proyecto, se prevé que limitaciones como estas sean solucionadas próximamente. Un elemento a tener en cuenta como parte de un CMS, es que puede presentar problemas de almacenamiento de determinados contenidos debido a limitaciones en el tamaño máximo de las filas en sus tablas.

Entre sus principales características están:

1. Arquitectura multigeneracional: Es la forma de administrar el manejo de transacciones y las concurrencias en las actualizaciones de datos. Genera una copia de los datos para cada usuario que abra una transacción, garantizando que no ocurran bloqueos.
2. Triggers: Permiten realizar acciones cada vez que se agreguen, modifiquen o eliminen registros.
3. Procedimientos almacenados.
4. Integridad referencial.
5. Seguridad integrada.
6. Funciones definidas por el usuario.

4.2.4 Lenguajes de Programación.

Un elemento imprescindible en el desarrollo de un Portal CMS es el lenguaje de programación a usar. Las características de los mismos varían y de ahí la importancia de su conocimiento previo ante una selección. En la modificación es tan importante como en la implantación final del software para conocer sus principales opciones sobre todo de compatibilidad con sistemas operativos y servidores Web.

En la plataforma LAMP, existen dos posibilidades de lenguajes a usar:

1. *Lenguajes incrustados*: El código ejecutable no está separado del código HTML, sino que es incrustado en el mismo mediante etiquetas específicas. La ejecución es llevada a cabo mediante el servidor web. Dentro de los lenguajes incrustados esta PHP, desarrollado para generar páginas dinámicas e interpretado por plug-ins añadidos a los servidores Web.
2. *CGI*: No es como tal un lenguaje, sino una especificación para la transferencia de información entre un servidor Web y un programa que puede estar escrito en lenguajes como Perl o Python.

4.2.4.1 PHP.

PHP: Hypertext Procesor, es un lenguaje de programación producto de un proyecto de la fundación de software Apache. Es un lenguaje de scripts, de propósito general pero muy usado generalmente en desarrollos Web. Su uso es variado, al punto de que es utilizado para la creación de aplicaciones para servidores, creación de contenidos dinámicos para sitios Web y creación de otros programas como aplicaciones con interfaz gráfica usando librerías como la GTK+.

PHP puede ser utilizado en muchos de los principales sistemas operativos, incluyendo GNU/Linux, variantes de Unix (incluyendo Solaris, OpenBSD, etc.), Mac OS, entre otros. Además soporta la mayoría de servidores Web de hoy en día como Apache, Personal Web Server, etc. Esta capacidad viene al tener módulos disponibles en la mayoría de estos servidores, para otros que soporten el estándar CGI, PHP puede usarse como procesador de CGI.

De modo que si se va a usar PHP, se puede tener la posibilidad de elegir entre el sistema operativo y el servidor Web de preferencia. También se tiene la posibilidad de usar programación procedimental ó programación orientada a objetos.

La característica quizás más potente de PHP es su soporte para una gran cantidad de bases de datos, haciendo relativamente fácil el hecho de escribir una interfaz web para las mismas. Actualmente según su página oficial (php.net) las siguientes bases de datos están soportadas:

MySQL, InterBase, PostgreSQL, Adabas D, Oracle(OC17 y OC18), dbase, Informix, Sybase, IBM DB2, Velocis, Unix dbm, Solid Hyperwave, mSQL.

También se cuenta con una extensión DBX de abstracción de bases de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Y por si fuera poco, soporta ODBC (estándar abierto de conexión a bases de datos), haciendo posible que se conecte a cualquier base de datos que soporte el estándar.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM, entre otros. Soporta WDDX para el intercambio de datos entre lenguajes de programación Web y puede utilizar objetos Java de forma transparente como objetos PHP.

PHP tiene características muy útiles para el procesamiento de texto, desde expresiones regulares POSIX extendidas o tipo PERL hasta procesadores de documentos XML. Para el acceso a documentos XML soporta los estándares SAX y DOM, además de poder transformar documentos XML mediante la extensión XSLT.

PHP es dentro del campo de los Portales CMS libres u open source quizás el lenguaje de más uso, al tener magnífica conexión con MySQL y Apache.

4.2.4.2 Perl.

Perl (Practical Extraction and Report Language) es un lenguaje de programación desarrollado principalmente para la administración de tareas propias de sistemas Unix y sus derivados. Entre sus características están las de escanear archivos de texto para la extracción de información de los mismos, así como imprimir reportes basados en esa información.

Este lenguaje combina características de lenguajes como C, sed, awk y sh. Su optimización para trabajos con textos es excelente, puede trabajar con datos binarios y puede hacer que archivos DBM se vean como arreglos asociativos. Este lenguaje no es ni un intérprete ni un compilador, sino que está en un punto intermedio.

Es basado en scripts y es portable a muchas plataformas. Sus principales funciones vienen para escribir CGIs, siendo las expresiones regulares uno de sus principales elementos. Básicamente ha sido concebido para que sea lo más natural e intuitivo posible en sus sentencias de código.

Su uso en la implementación de CMS ha ido en incremento, principalmente en sistemas de Blogs o Bitácoras como WebAPP (web-app.org) y Blosxom (blosxom.com).

4.2.4.3 Python.

Python es un lenguaje de programación interactivo capaz de ejecutarse en gran cantidad de plataformas. Este lenguaje es de scripting y usualmente es comparado con otros como Ruby, Java, TCL y PHP. Es administrado por la Python Software Foundation, aunque es mantenido por una gran comunidad de usuarios al ser un proyecto open source.

Es un lenguaje interpretado, lo que ahorra tiempo en el desarrollo del programa, pues no es necesario compilar ni enlazar. La capacidad de utilizar el intérprete de modo interactivo, da la posibilidad de experimentar con características propias del lenguaje, probar funciones en pleno desarrollo del programa o escribir programas desechables.

Este lenguaje viene con una gran colección de módulos estándar y permite dividir los programas en módulos reutilizables desde otros programas. También hay módulos incluidos que proporcionan llamadas al sistema, sockets e incluso interfaces como GTK y QT entre otras.

Python tiene la ventaja de enriquecerse dada su característica de ser un lenguaje de código abierto. Por lo tanto cuenta con módulos/librerías para la realización de cualquier programa incluyendo para la programación Web, donde existen módulos diferentes que de varias maneras básicas permiten llevar a cabo las mismas tareas.

El sitio oficial de Python en la sección dedicada a los temas de programación Web (python.org/topics/web/) enumera, en un largo listado, dichos módulos, detallando sus principales funciones. Entre los principales se encuentran:

1. *Programación CGI Básica:*
2. Módulo CGI de la librería estándar de Python. CGI (Common Gateway Interface) es un mecanismo estándar para la ejecución de código por un servidor Web y la obtención de los resultados de dicha ejecución.
3. Módulo Cookie para crear y procesar cookies. Una cookie no es más que un mecanismo para mantener estados entre las peticiones HTTP de una sesión Web. Una cookie es una cabecera HTTP que permite la identificación en el servidor Web.
4. *Programación CGI avanzada:* En CGI, un nuevo proceso es creado por cada petición HTTP recibida y luego es eliminado cuando la petición es resuelta. De ahí que se hayan creado tecnologías que permitan una integración superior con

el servidor Web y que obtengan mayor eficiencia. Ejemplos de estas tecnologías son PHP, Java Servlets y JSPs y ASPs. En el caso de Python su contribución viene con `mod_python`: Se trata de un módulo Apache (como se veía en la sección 3.2.2) que integra el intérprete Python dentro del servidor, de modo que las aplicaciones se ejecuten de forma más rápida que CGI.

5. *Servidores de aplicaciones*: Van más allá de la clásica generación de páginas dinámicas, asisten al programador en tareas dentro de la lógica del negocio, tales como la persistencia de datos, la gestión de transacciones y la seguridad, etc. Ejemplos de los mismos son Webware y Zope.

No es un lenguaje tan utilizado en desarrollos de CMS como PHP, pero aún así cuenta con excelentes ejemplos de su uso en algunos como Zope (zope.org) y Plone (plone.org).

1. *Python Server Pages (PSP)*: Es un mecanismo para incluir sentencias Python en documentos HTML o XML. El servidor se encarga de interpretar el código Python incluido. A pesar de que a menudo se vea como una mala práctica de programación el violar el paradigma Modelo-Vista-Controlador e introducir lógica del negocio en la capa de presentación, muchos programadores en la actualidad se inclinan por este estilo de programación Web.
2. *DB-API*: Python ofrece el estándar DB-API, muy parecido al JDBC ofrecido por Java para la programación de bases de datos. Con esta API, se consigue aislar el código fuente de la base de datos. El código Python se acomodará a las interfaces de la DB-API, a la vez que la implementación correspondiente de esta API para cada sistema de bases de datos, se encargará de traducir las invocaciones de DB-API en llamadas de las bases de datos.

4.2.5 WAMP.

WAMP no es más que una plataforma que al igual que LAMP, integra elementos como el servidor Apache, el gestor de bases de datos MySQL y lenguaje PHP, solo que todo sobre el sistema operativo Microsoft Windows.

WinLAMP por ejemplo es un programa que instala de forma muy sencilla los elementos de la plataforma WAMP. Por otra parte WAMP5 instala automáticamente Apache 1.3.31, PHP5, MySQL, PHPmyadmin y SQLitemanager.

Esta plataforma es mayormente utilizada por usuarios que prefieren el uso del sistema operativo Microsoft Windows en aplicaciones Web o por algunos que pretenden migrar al uso del software libre y ven este uso como un paso de transición.

4.2.6 MAMP.

Por otra parte se tiene a MAMP, lo cual es una forma muy sencilla de instalar los elementos de LAMP sobre el sistema operativo MAC OS. Tiene la característica de ser muy fácil e intuitiva y no tener que tocar para nada al sistema operativo.

MAMP al ser una combinación de software libre, es lanzada mediante la licencia GNU GPL, permitiendo la distribución de la misma, entre otras posibilidades.

4.3 Plataforma Java.

La plataforma Java es quizás una de las más amplias en el desarrollo de Portales CMS. Presenta, de forma general, soluciones para elaborar cualquier tipo de programa dada cualquier problemática. Fue desarrollado por la compañía Sun Microsystem y siempre ha intentado estar en la delantera en cuanto a técnicas novedosas.

Una de las principales características, que hacen que java goce de popularidad, es su independencia de la plataforma. Esto se refiere a que una vez que un programa es realizado en este lenguaje, el mismo podría funcionar en cualquier ordenador. Esto constituye una buena ventaja para desarrolladores al no tener que implementar para cada sistema operativo por separado. Esta característica se lleva a cabo gracias a que existe una Máquina de java para cada sistema operativo que hace de puente entre el mismo y el programa implementado en java facilitando el entendimiento entre ambos. Esta capacidad le viene muy bien a los usuarios en general teniendo en cuenta que, por ejemplo, en el acceso a internet, no todos se conectan con los mismos tipos de ordenadores.

4.3.1 JSPs, Servlets.

JSP (Java Server Pages) es una especificación proporcionada por la misma compañía Sun y le añade funcionalidad a la especificación de servlet. Los servlets y los JSPs son métodos con el objetivo de la creación de páginas web dinámicas usando lenguaje Java.

Los servlets se ejecutan en el servidor e intersectan las solicitudes del navegador, actuando como una capa intermedia entre clientes y aplicación de menor nivel. Su

principal función no viene dada en generar contenidos, sino en manejar las solicitudes de clientes e invocar otros objetos del lado del servidor. Entre sus principales características están:

1. Se ejecutan en una máquina virtual dentro del proceso del servidor.
2. Cada petición se ejecuta en un hilo, invocando un método servlet.
3. El motor de servlets es capaz de ejecutar los servlets.
4. Además de servidores HTTP, su uso puede extender servidores como el FTP.

Dentro de las ventajas del uso de los servlets están:

1. Portabilidad: Entre servidores y plataformas.
2. Potencia: APIs de java (Manipulación de imágenes, acceso a bases de datos, internacionalización, etc.).
3. Eficiencia: Carga cada instancia permanentemente en memoria por cada servlet. Ejecuta las peticiones mediante la invocación de un método. Cada petición se ejecuta en un hilo. Mantiene automáticamente las conexiones, etc.
4. Seguridad: Mediante la máquina virtual, la gestión de memoria, excepciones, etc.
5. Extensibilidad y flexibilidad: Mediante filtros (cadenas de servlets). Integración con JSP.

Las páginas JSP por lo general se diseñan como páginas web y no como programas, siendo ideales para mostrar anotaciones con contenido dinámico integrado. Cada página es automáticamente compilada a servlet por el motor JSP, primero se recoge y luego se ejecuta.

Básicamente se trata de un archivo de texto simple que integra contenido HTML con elementos JSP. Su integración con los servlets hace que se fortalezca la relación Vista-Controlador.

4.3.2 Motores de Plantillas.

El patrón MVC (Modelo-Vista-Controlador), muy de moda en la programación Web, trata básicamente de separar los datos de la presentación de los mismos. Las plantillas o los motores de plantillas se aproximan a resolver este paradigma.

Básicamente lo que hace un motor de plantilla es leer un fichero de texto que contenga la forma de la presentación en HTML e inserta en el mismo el contenido dinámico indicado por una clase controladora. También suelen tener un pequeño lenguaje de script que permita generar listas o cierto comportamiento condicional.

La principal ventaja del uso de esta tecnología es la separación que se realiza del código HTML y el código Java. Permitiendo a programadores de cada lenguaje no tener que verse en la necesidad de manipular código donde no son expertos.

4.3.2.1 Velocity, webmacro y freemarker.

Entre los principales motores de plantillas open source y dentro de los más usados están Velocity, Webmacro y Freemarker. Los dos primeros se basan en la misma filosofía de trabajo y son muy similares al estar Velocity basado en Webmacro.

Los primeros realizan sus funciones mediante *reflection* y el tercero mediante una serie de interfaces definidas. Los métodos de reflexión aplicados a las plantillas no son del todo eficientes, no obstante han obtenido, por su facilidad de uso, una mayor aceptación por parte de desarrolladores. El trabajo mediante interfaces es más eficiente pero requiere a su vez de la implementación de la mismas en los objetos del modelo de negocio o técnicas como los wrappers.

Otra diferencia viene del hecho de que el mayor uso de los primeros sobre Freemarker, hace que exista una mayor documentación sobre el uso de los mismos, principalmente de Velocity al ser un subproyecto de Jakarta y tener un mayor número de colaboradores.

Dentro de las características que los hacen similares están: La lucha contra las JSPs y tecnologías similares para encontrar un mayor uso dentro del desarrollo de Webs, la utilización de frameworks para su uso desde servlets y el soporte para XML.

4.3.3 Apache Tomcat.

Jakarta Tomcat o Apache Tomcat, son los nombres con los que se le conoce a este contenedor de servlets que fue desarrollado por la Apache Software Foundation y más específicamente por el subproyecto Jakarta. A grandes rasgos, implementa las especificaciones de la compañía Sun Microsystem de los servlets y los JSPs.

Su funcionamiento puede darse en cualquier sistema operativo que disponga de una Máquina Virtual de Java. En la mayoría de los casos Tomcat es combinado con el

servidor web Apache. Pudiera funcionar, bajo pocas exigencias de velocidad y transacciones, como servidor web.

Básicamente Tomcat ofrece un ambiente donde habitan los servlets y los JSPs, en dicho ambiente es donde se contemplan gran parte de sus funcionalidades como la mantención de sesiones, conectividad con el servidor web, threading, etc.

Por su característica de software open source, es mantenido por muchos colaboradores de la Apache Software Foundation.

Otra de las muestras de lo estandarizado del uso de Tomcat es su uso como utilización como servidor de Servlets y JSP por parte de los entornos de desarrollo tanto libres como comerciales. Ejemplos de esto son Borland JBuilder o Eclipse que integran un servidor Tomcat para poder desarrollar nuestras páginas JSP y Servlets. (TOURIS, MARIÑAN, 2002)

4.3.4 Servidores de Aplicaciones.

Los servidores de aplicaciones, a grandes rasgos, son los encargados de manejar componentes tales como objetos accesibles remotamente y páginas web. En los mismos se encuentran piezas como los contenedores de EJB que se encargan del control de los componentes.

En la plataforma Java el servidor de aplicaciones por excelencia para el desarrollo de software libre u open source es JBoss. Es implementado en su totalidad en Java y está distribuido bajo la licencia LGPL.

Lo más interesante de JBoss es la buena integración capaz de lograrse con Apache y Tomcat en desarrollos web. Otros servidores de aplicaciones son JOnAS y OpenEJB, entre otros.

4.4 Estándares. La W3C.

Teniendo en cuenta que el término *estándar* se refiere a una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad, y que este trabajo integra elementos claves para todo el proceso de implantación de un Portal CMS desde su selección, se pasan a explicar los principales estándares promovidos por la W3C (World Wide Web Consortium).

La W3C es un consorcio internacional que se ha dedicado desde más de una década a publicar más de 90 estándares para la Web llamados *Recomendaciones*.

Algunas de estas recomendaciones son:

1. *HTML*: Básicamente esta recomendación trata de exponer las especificaciones de HTML/XHTML, así como las mejores formas de su uso.
2. *Hojas de estilo en cascada (CSS)*: Integra elementos sobre el uso de las CSS en diferentes navegadores, las principales herramientas para su desarrollo, etc.
3. *XML*: Explica y establece elementos claves para el desarrollo con XML.
4. *WSDL (Web Services Description Language)*: Establece elementos de su uso, sus diversas estructuras, sus operaciones, etc.

La W3C cuenta con expertos que evalúan constantemente las recomendaciones, también con herramientas como *Markup Validator Service* que permiten verificar la calidad técnica de un Portal Web. Los Servicios proporcionados por el Consorcio incluyen un almacén de información sobre el World Wide Web para desarrolladores y usuarios, y varios prototipos y aplicaciones de ejemplo para demostrar el uso de estas nuevas tecnologías. Hasta la fecha, según su sitio oficial (w3.org), cerca de 400 organizaciones son miembros del Consorcio. Cualquier Portal CMS con los logos de las recomendaciones W3C, informa sobre estándares cumplidos en calidad de su implementación, de ahí la importancia de su conocimiento (Ver ejemplo de logo en la Figura 11).



Fig 11: Ejemplo de logo de la W3C.

4.4.1 WAI.

WAI (Web Accessibility Initiative) como su nombre lo dice, es una iniciativa para la accesibilidad a la Web del mayor número de personas posibles teniendo en cuenta personas con discapacidad. El grupo de trabajo permanente de la WAI junto a entidades de todo el mundo, persiguen el logro de la accesibilidad a la Web mediante cinco áreas de trabajo:

1. Tecnologías.
2. Directrices.

3. Herramientas.
4. Formación y Difusión.
5. Investigación y Desarrollo.

De los trabajos desarrollados por la WAI se desprenden directrices que son consideradas por la Unión Europea como normas y son citadas como referencia obligada en la mayoría de las legislaciones sobre Tecnologías de la Información de todo el mundo. Estas directrices son llamadas de accesibilidad para:

1. El contenido Web.
2. XML.
3. Herramientas de Autor.
4. Navegadores.

Para la revisión por parte de entidades de Portales Web en cuanto a las directrices existen, por parte de la WAI, listas de puntos de control. Otra forma de revisión puede ser mediante herramientas como HERA o HERA-XP (sidar.org).

Los usuarios pueden también revisar o verificar la accesibilidad de cualquier página web mediante herramientas de evaluación automática como el TAW (Web Accessibility Test) y como Bobby (desarrollado por CAST), revisor online que ejecuta un test de revisión de accesibilidad según las pautas redactadas por la WAI.

Una vez que se haya verificado el diseño en cuanto a la accesibilidad del contenido web del Portal CMS seleccionado, se puede alcanzar algún nivel de accesibilidad y colocar en dicho software el logo que provee la W3C declarando la conformidad con las directrices (Ver Figura 12).



Fig 12: Logo del estándar de accesibilidad WAI.

4.4.2 JCP.

Con el objetivo de lograr una mayor participación de expertos en el desarrollo y mantenimiento de la plataforma Java, surge la JCP (Java Community Process). Una de sus funciones es controlar la evolución de las especificaciones de las plataformas

basadas en Java. La conforman, entre otros, empresas de todo el mundo y desarrolladores expertos junto a la Sun Microsystem. Ejemplos de especificaciones son aquellas que tienen que ver con los portlets:

4.4.2.1 Portlet.

Básicamente son mini-aplicaciones Web interactivas que devuelve fragmentos HTML. Más específicamente:

Son componentes web gestionados por un contenedor que tras la petición de un usuario generan y presentan contenidos dinámicos de forma identificable en el interfaz de usuario del portal como componentes de contenido. El portlet permite la personalización, la presentación, y la gestión de la seguridad. Los portlets se diferencian de los servlets por ser estos componentes de servidor con perspectiva funcional.(VILLA, 2003).

El portlet, al igual que los contenidos tiene su propio ciclo de vida. Los portales actúan como gestores de presentación de los portlets. Los usuarios realizan peticiones durante su sesión de navegación, y los portlets generarán contenidos dependiendo del perfil del usuario que realice las peticiones.

4.4.2.2 JSR.

JSR es el acrónimo de Java Specification Request. Cuando una persona o entidad cree que es necesario la presencia de una determinada tecnología dentro de las plataformas basadas en Java, lo que hace es crear un JSR y presentarlo para su aprobación. Dentro de este documento se relata por que es necesaria dicha tecnología, por que no se pueden abordar los problemas que soluciona con las tecnologías existentes, etc. (TOURIS, MARIÑAN, 2002).

En caso de aprobarse los JSR, se crean documentos que explican los elementos de dicha tecnología, así como los roles de los desarrolladores en la misma, etc. También se ha de proporcionar un test de compatibilidad y ejemplos de referencia.

Según la JCP en su sitio jcp.org:

1. JSR 168 es un grupo de trabajo que finalizó en octubre de 2003 la definición del concepto de portlet estándar, que permitirá que cualquier desarrollo realizado en un Servidor de portales pueda ser instalado en cualquier plataforma del mercado.

2. JSR 170 (finalizado durante 2005) define un interfaz estándar para acceder a un repositorio de contenidos desde un portlet: Cualquier portlet desarrollado podrá utilizarse independientemente del proveedor de gestor de contenidos implantado.

Las especificaciones de portlets:

1. Estandariza el API que ofrece el contenedor de portlets.
2. El diseño del API tiene cierto parecido con el API de servlets.
3. Permite delegar la generación de markup en páginas JSP.

Por otra parte, en más a largo plazo, JSR 170 ofrecerá el potencial para la verdadera infraestructura del repositorio de contenidos en la cual los desarrolladores pudieran construir sus aplicaciones sin asociaciones comerciales o pagos de honorarios. (NUESCHELER, BOYE; 2005).

Capítulo 5. Portal NovaLNX.

5.1 Introducción.

La tendencia por parte de entidades en nuestro país al uso de Portales CMS libres para necesidades de comunicación e información se extiende al ámbito de organizaciones y proyectos productivos de nuestra universidad. Se pudieran citar ejemplos como el sitio de la Federación Estudiantil Universitaria (feu.uci.cu), el proyecto productivo SIMPRO (simpro.uci.cu) y el foro de discusión de Inter-Nos (inter-nos.uci.cu), la mayoría de los cuales se basan en soluciones llamadas “*Nukes*” y se implementan bajo PHP o la plataforma LAMP en general. En el ámbito de Portales CMS para distribuciones de Linux se puede mencionar el de la distro de la Universidad de la Habana Linuhx (linuhx.uh.cu).

Este último capítulo lleva a cabo en la práctica la selección, modificación e implantación de una herramienta de este tipo como base para la distro NovaLNX. Intenta dar cumplimiento a los objetivos planteados.

5.2 Necesidad de NovaLNX.

Por parte de una iniciativa de una facultad de la Universidad de las Ciencias Informáticas se crea un proyecto con la participación de estudiantes y profesores de la misma. El objetivo de este proyecto es la creación de una distribución del sistema operativo GNU/Linux. El proyecto, como parte de resultados obtenidos y debido a decisiones de la dirección de la facultad, ha logrado desde el principio del presente curso poner a prueba a NovaLNX instalándola en varios laboratorios docentes y aulas de la facultad.

En la actualidad existe en crecimiento una comunidad de software libre en todo el país. La misma está conformada por usuarios expertos o no en el uso de herramientas libres para el desarrollo de software. Dicha comunidad cuenta con facilidades como listas de distribución donde se tratan, mediante e-mails, temas de interés o dudas en cuanto a programación o al uso de determinadas herramientas, también cuenta con un portal de software libre, donde el uso por ejemplo de los foros de discusión ha aglutinado una buena cantidad de información sobre temas que en conclusión le dan razón de ser a ese Portal.

La comunidad del software libre no está ajena al desarrollo de la distribución NovaLNX, de hecho en el portal de software libre existe cierta documentación fundamentalmente recogida en un tópico dedicado a este fin. No obstante se considera que el uso y manejo de información entorno al proyecto está creciendo en cantidad y cada vez es más exigida por los ya mencionados usuarios.

Por tanto, se hace necesaria la implantación de una herramienta que llegue a la mayor cantidad de usuarios con el máximo de prestaciones e información. Esta herramienta se decidió que fuese un Portal Web, con esta decisión se aseguraría, dadas las características de la universidad y el creciente acceso que se va teniendo desde cualquier parte del país a la Web, un máximo de efectividad en tratar de llegar a todos. También serviría para darle seguimiento a este importante proyecto en un momento crucial para el software libre en nuestro país al estar el mismo en plena migración.

5.2.1 Contenidos a gestionar.

En capítulos anteriores se mencionaba la importancia del conocimiento previo de todo lo referente a los contenidos de la organización. Como es lógico, los contenidos a gestionar por el Portal CMS están íntimamente ligados a los objetivos en sí del proyecto. Los contenidos identificados vienen en el orden de:

1. Informaciones, provenientes del proyecto, sobre el estado del desarrollo de la distribución de interés para el usuario: En el proyecto se desarrollan constantemente varias herramientas que marcan un sello propio para la distribución, y que le agregan funcionalidades a la misma de acuerdo a necesidades propias de la universidad para diversos casos de usos. Se ha comprobado que el interés por el estado de dichas herramientas y el deseo de colaborar de los usuarios es alto. También son importantes las informaciones actualizadas en cuanto a últimas versiones de las herramientas, bugs reportados y solucionados, entre otras.
2. Documentación sobre las herramientas a nivel de desarrolladores: Es muy usual que trabajen en conjunto desarrolladores propios del proyecto y otros usuarios colaboradores incluso desde fuera de la universidad o del país. Para estos casos el Portal Web debería gestionar niveles de documentación.
3. Materiales y manuales: Al tener en cuenta que la distribución ha sido instalada en laboratorios docentes y aulas de la facultad, se identificó el manejo de contenidos provenientes de materiales o manuales de usuarios.

4. Contenidos provenientes de repositorios: El Portal definitivamente pudiera ser un punto de acceso a ciertos ficheros o paquetes de software que se necesiten publicar para su descarga.
5. Contenidos provenientes de informaciones por parte de usuarios: Los usuarios, en el desarrollo de la distribución, juegan un importantísimo papel. De ellos suelen venir las mayores experiencias de uso y se reportan los errores no erradicados de las nuevas herramientas en su funcionamiento. Las informaciones provenientes de usuarios pueden darse en diferentes formas, entre ellas, mediante foros de discusión, sugerencias, noticias enviadas, blogs o bitácoras, etc.

5.3 Definición de requerimientos.

En el tercer capítulo de este trabajo se señalaban las ventajas de una adecuada definición de requerimientos por la organización para llevar a cabo una buena selección de un determinado Portal CMS. Dadas las sobradas funcionalidades, muchas de ellas estándares, encontradas actualmente en sistemas de este tipo, se hace necesario listar los requerimientos en categorías para lograr no adaptar las necesidades a las herramientas, sino encontrar las herramientas adecuadas para adaptarlas a las necesidades.

Las mencionadas categorías donde se agrupan los requerimientos, no son los llamados requerimientos funcionales y no funcionales que se especifican en el desarrollo de un software, sino que son requerimientos teniendo en cuenta un software a seleccionar dentro de un grupo de candidatos para un tiempo después llevar a cabo un proceso de adaptación a las necesidades más específicas de la entidad. Dichos requerimientos están delimitados teniendo en cuenta los principales subsistemas explicados en el segundo capítulo apoyados en los criterios y aspectos descritos en el tercero, también teniendo en cuenta que se sigan los ciclos de vida de los contenidos antes expuestos.

Como requerimientos a cumplir por el software a seleccionar se identificaron:

- **Subsistema de Colección de contenidos:**
 1. Interfaz consistente y fácil de utilizar para la creación de contenidos: De ser posible mediante editores WYSIWYG.
 2. Posibilidad de establecer múltiples autores y editores.

3. Agregación de contenidos desde otras fuentes.
 4. Introducción de metadatos.
 5. Control de enlaces entre los contenidos.
- **Subsistema de Gestión de Contenidos:**
 1. Control del flujo de publicación a través de workflow.
 2. Seguridad para la integración de los contenidos.
 3. Autenticación única.
 4. Integración con sistemas externos.
 5. Posibilidad de generación de reportes para la administración del proyecto.
 - **Subsistema de Publicación de contenidos:**
 1. Utilización de hojas de estilos (CSS).
 2. Publicación en formato HTML.
 3. Posibilidad de impresión de información desde la publicación de la misma.
 4. Personalizaciones de acuerdo a mostrar determina información a determinado usuario.
 5. Estadísticas que identifiquen aciertos o no entorno a las publicaciones creadas.
 - **Presentación del Contenido.**
 1. Accesibilidad: Deben medirse estándares como el WAI.
 2. Soporte de varios navegadores Web y versiones de los mismos.
 3. Buena velocidad de descarga.
 4. Navegación comprensible y consistente.
 5. Metadatos: Importante para soportar indexación en buscadores externos.
 - **Requerimientos generales.**
 1. Multiplataforma.

2. Documentación abundante y de fuentes confiables.
3. Licencia GNU GPL u otra similar promovida por la FSF o la OSI.
4. Foros de discusión.
5. Sistemas de Bitácoras o Blogs.
6. Debido a las experiencias y conocimientos del personal encargado, se requiere que esté implementado en lenguaje Java.
7. Estructura modular.
8. Soporte a varios SGBD.
9. Buenas referencias de los productores.

5.4 Selección de Jetnuke.

En el tercer capítulo de este trabajo se conformó y expuso una guía sobre los pasos a seguir ante una selección de un Portal CMS para las necesidades de una organización. Una vez elaborado un listado de requerimientos, se procede a determinar los candidatos.

Para la elección de candidatos se utilizó la vía de búsqueda de datos mediante sistemas de referencias como CMS Matrix y OpenSourceCMS. También mediante la búsqueda de proyectos en el SourceForge. Las directrices principales se basaron en los requerimientos generales antes mostrados, y luego más detalladamente en cada requerimiento por su debida categoría.

Los Portales CMS candidatos que surgieron tras una minuciosa búsqueda basada en la lista de requerimientos generales fueron:

1. OpenCMS.
2. Magnolia.
3. Jetnuke.
4. Cofax.
5. eXo Platform.
6. Harmonise.
7. inCMS.

8. Liferay.
9. Lutece.

Siguiendo una selección se describieron a grandes rasgos los mismos teniendo en cuenta los puntos expuestos en el tercer capítulo. En un primer corte de candidatos surgieron elementos que condicionaron desechar algunos de la lista de candidatos, los mismos fueron:

1. Liferay Portal/CMS: Se encuentra bajo un tipo de licencia única, la cual dice ser permisiva y amigable y se autodeclara como las de tipo open source, no obstante no era promovida ni por la FSF ni por la OSI. Este sistema no presentaba forma de conocer estadísticas propias de la Web y tenía capacidades limitadas de seguimiento de un flujo de trabajo.
2. Harmonise: No cumple con funcionalidades como un editor WYSIWYG, ni foros de discusión, elementos estos primordiales en el sistema a implantar.
3. Cofax: Está basado fundamentalmente en tareas de publicación y manejo de contenidos multimedias o con fines editoriales, además de no presentar posibilidad de sindicación de contenidos.
4. eXo Platform: No soporta manejo de ficheros desde servidores FTP y tiene poco delimitados los conceptos de metadatos.
5. inCMS: Presenta limitaciones en cuanto al manejo de preguntas frecuentes y posibilidades de ayudas al usuario, también un limitado flujo de trabajo poniendo en riesgo los contenidos y la seguridad de los mismos.

Para llegar a la selección de los finalistas se intentó obtener respuestas de fuentes especializadas y de los mismos productores o colaboradores, obteniéndolas solamente del proyecto Jetnuke de parte de su autor Andrea Mazzolini. Identificado como una necesidad el hecho de poder integrar funcionalidades extras, se adopta la posición de optar por una arquitectura verdaderamente modular.

La elección final de Jetnuke como Portal CMS base para el proyecto NovaLNX, se llevó a cabo tras analizar las convenientes características que presenta el mismo.

5.4.1 Características de Jetnuke.

Las principales características que fundamentalmente cumplen con los requerimientos expuestos son:

- Editor WYSIWYG para la creación online de contenidos. En este caso Jetnuke presenta un acoplamiento con FCKeditor (Ver Figura 13). Este editor presenta características tales como:
 1. Compatibilidad con múltiples navegadores Web (Internet Explorer, Netscape, Mozilla, Firefox).
 2. Detección automática del navegador y adaptación al mismo para mejor rendimiento.
 3. Formato de textos y formato de fuentes (Tipo, Tamaño, Color, etc.).
 4. Acciones (Copiar, Cortar y Pegar textos, etc.).
 5. Creación de Links desde el contenido.
 6. Inserción de imágenes junto al contenido.
 7. Creación y Edición de tablas.
 8. Corrección ortográfica.
 9. Integración con lenguajes como Java, PHP, Perl y Javascript.
 10. Personalizable mediante Skins.
 11. Soporta CSS para una mejor integración.

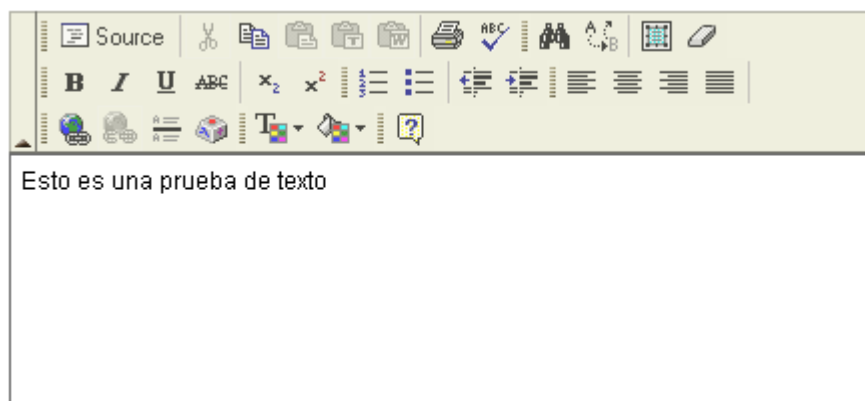


Fig 13: Vista del FCKeditor

- Posibilidad de creación de roles entre los usuarios. No es un objetivo ni una necesidad constar con roles de editores, pero sí con la capacidad de tener usuarios con permisos de revisión de contenidos creados por usuarios antes de su publicación, también con permisos estrictos para ciertos usuarios a la hora de crear contenidos. Jetnuke ofrece esta capacidad, brindando la posibilidad de separar roles entre administradores, editores, usuarios usuales e invitados (Ver Figura 14).



Modificar Autor

Nombre prueba

Permisos Artículos Tópicos

Usuario Encuestas

Secciones Enlaces WEB

Efemérides Preg. más Frec.

Descarga Revisar

Boletín Enciclopedia

Contenido **Administrador**

Salvar

Fig 14: Establecimiento de Roles a los usuarios.

- Capacidad de introducción de metadatos. La interfaz a la hora de crear contenidos es separable para la introducción lógica de metadatos que provean datos específicos sobre el contenido como la fecha de publicación, el autor, etc.
- Mediante el editor FCKeditor se logra insertar enlaces que a su vez funcionan entre diferentes contenidos, estableciendo conexión entre los mismos.
- Controla los flujos de publicación mediante un workflow que incluye actividades de creación, aprobación y publicación de los contenidos.
- Seguridad de los contenidos al haber roles de administradores protegidos con contraseñas para que sean los únicos en borrar los mismos.

- Fácil administración de todos los elementos del portal mediante páginas de administración (Ver Figura 15).

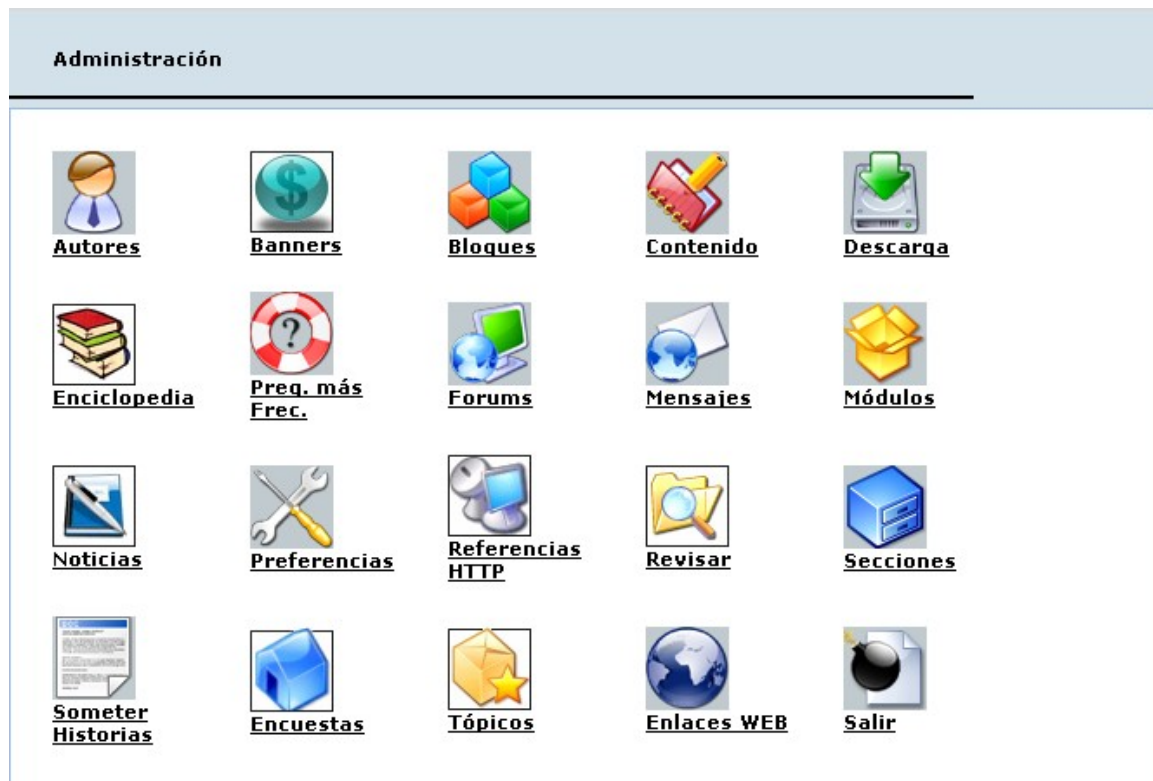


Fig 15: Elementos a administrar en la página de administración.

- La integración de las herramientas como el Jforum, posibilita una única autenticación en toda la navegación del Portal por lo que el usuario requiere de una sola autenticación para el acceso a cualquier módulo.
- Elabora reportes para los administradores sobre los accesos al Portal, los contenidos, las descargas, etc.
- Soporta los estándares CSS y HTML promovidos por el W3C (Ver Figura 16).



Fig 16: Estándares soportados.

- Desde el propio Portal se manejan posibilidades de impresión de contenidos.
- El sistema de administración provee posibilidades de mostrar los contenidos en bloques, donde los mismos pueden ser estructurados y puestos en cualquier parte de la página según las preferencias de los usuarios.
- Uso del motor de plantilla Velocity.
- Posee un total de siete estilos, dando posibilidad a los usuarios de mostrar el Portal de la forma que deseen cada vez que naveguen por el mismo.
- Brinda estadísticas que muestran la cantidad y el tipo de los contenidos publicados, los usuarios registrados, los tipos de navegadores Web que han accedido al Portal, entre muchas otras.
- Cumple estándares de accesibilidad de la WAI. Jetnuke fue procesado por el test Taw3 arrojando capacidades de cumplir con estándares de la WAI.
- Jetnuke como tal soporta varios navegadores, los mismos son: Netscape, Microsoft Internet Explorer, Mozilla, Firefox, Opera, Konqueror y Lynx.
- La generación de las páginas es ejecutada en menos de un segundo, garantizando una buena descarga de la misma en la presentación.
- Su implementación en lenguaje Java le da a Jetnuke la capacidad de ser multiplataforma, corriendo esta herramienta sobre lenguajes operativos Microsoft Windows y GUN/Linux.
- Bajo licencia GNU GPL. En el segundo capítulo de este trabajo se documentaba sobre la misma dándole a Jetnuke la posibilidad de ser un software siempre libre.

- La documentación sobre esta herramienta no es abundante producto de ser un proyecto relativamente nuevo y con muy poco personal dedicado a su mantenimiento. Por lo que la misma es producto de las coordinaciones directas realizadas con su productor principal.
- Foros de discusión. Jetnuke presenta una integración con otra herramienta bajo licencia libre, el foro de discusión JForum (Ver Figura 17). Dicha integración a partir de las últimas versiones viene con autenticación única, no siendo así en un principio al necesitar de varias cuentas dentro del Portal.

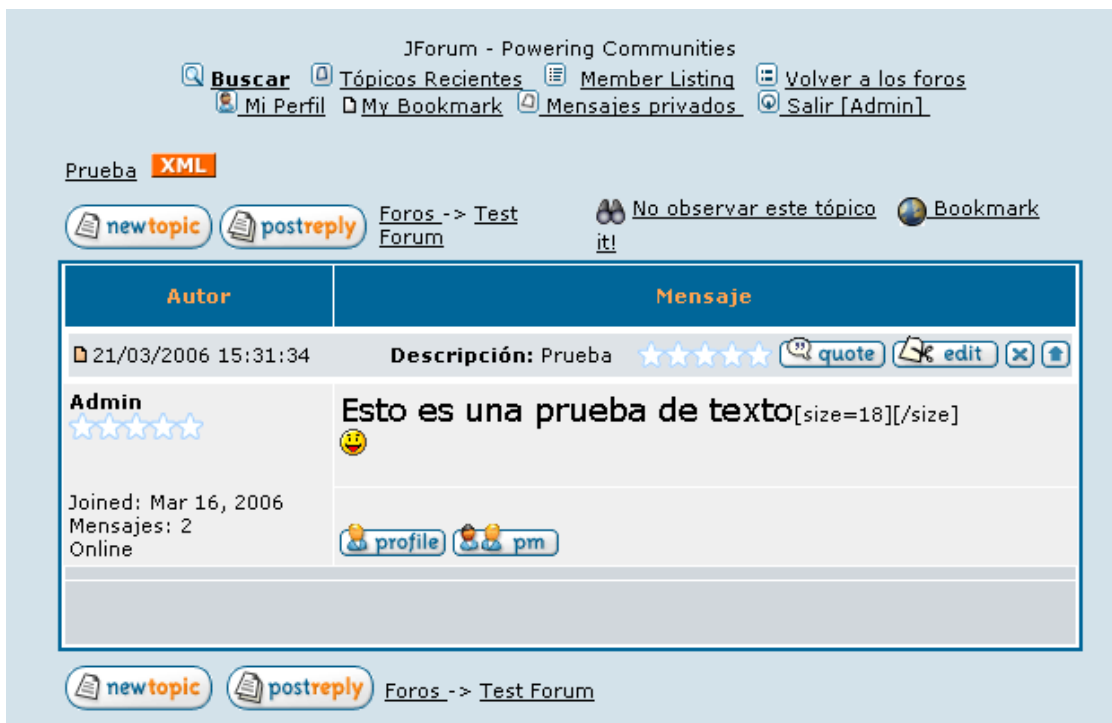



Fig 17: Vista del foro de discusión Jforum.

- Sistema de Blogs. Los usuarios pueden mantener un sistema de diarios o blogs, donde las informaciones de uso de herramientas, experiencias, etc, son mantenidas (Ver Figura 18).

Diario de Usuarios


















































Diario de Usuarios

[Directorio de Diarios](#) | [Tu Diario](#)

[[Ultimos 20 Autores](#) | [Listar Todos los Diarios](#) | [Buscar un Miembro](#)]

Título

Source           **B** *I* U ~~ABC~~ x₂ x²

- Estructura modular. Jetnuke está implementado pensando en la adición constante de módulos. La posibilidad de la creación de los mismos viene integrada al Portal.
- Soporte a varios SGBD. Se recomienda por parte de su productor el uso de MySQL, pero también soporta gestores de bases de datos como PostgreSQL.

Las características notables que incorpora Jetnuke, cumplen a grandes rasgos con los requerimientos identificados antes de su selección. Algunas de ellas necesitarían de un proceso de adaptación al proyecto para que cumpliesen cabalmente las necesidades del mismo.

5.5 Adaptación e implantación del Portal NovaLNX.

La adaptación de una solución libre a las necesidades de una entidad o un proyecto como el caso de NovaLNX, requiere de un escenario sobre el cual mostrar los niveles de aceptación del mismo. En este caso las capacidades a medir concuerdan con las necesidades propias de la entidad especificadas en los requerimientos.

Debido a que las características propias de estos sistemas están formuladas de forma estándar al tratar de cumplir con las necesidades primarias sobre las secundarias, se tiene que especificar las cualidades a cumplir por cada funcionalidad mediante ejemplos concretos. El hecho de que ninguna entidad sea igual ni tenga los mismos objetivos o necesidades, obliga a llevar a cabo un proceso de adaptación de dichas funcionalidades.

En el caso propio del Portal CMS Jetnuke, el mismo fue concebido para el fácil montaje y administración en un ambiente donde los contenidos a gestionar sean del tipo de documentos e informaciones. El escenario para su explotación está formulado teniendo en cuenta una intranet u otro donde se maneje una media de accesos en comparación con grandes portales.

Para la adaptación de Jetnuke hacia el Portal NovaLNX, se siguen esquemas prácticos, así como una serie de pasos lógicos que en general pueden variar de acuerdo al tipo de entidad.

Los objetivos que llevaron al proyecto al uso de un Portal CMS, son los mismos a tener en cuenta para su adaptación. Por tanto nuevamente se formulan y especifican más exactamente los contenidos del proyecto a gestionar por el sistema.

Las actividades que en cuestión se llevaron a cabo para la modificación e implantación fueron:

1. Definición de roles. La dirección del proyecto es quien establece los roles de sus desarrolladores dentro del Portal. Los desarrolladores propiamente no deben ser administradores como tal del Portal, ni manejar todos los permisos teniendo en cuenta que pueden afectar en determinado momento la disposición íntegra del mismo. En el caso de NovaLNX los roles tienen que adaptarse a las necesidades de revisar información provenientes de usuarios para ser mostrada en diferentes canales o tópicos.
2. Establecimiento de tópicos o canales según el manejo de tareas del proyecto. Una de las características del proyecto es que distribuye el personal en varias tareas o funciones según las herramientas en implementación o en cualquier otra fase de su desarrollo. Cada tarea maneja contenidos de acuerdo a documentación propia de las mismas. Por tanto en la adaptación del portal NovaLNX se emplean las mismas como base para el establecimiento de tópicos. Los tópicos van más allá de las herramientas o tareas del proyecto, y se definen también teniendo en cuenta algunas temáticas relacionadas con el mundo del software libre, etc.
3. Política de personalización del proyecto. El establecimiento de una distribución de Linux conlleva al trazado de políticas de acuerdo a la personalización de la misma. El Portal Web como punto de acceso a cualquier información sobre este proyecto, sigue cabalmente dichas políticas declarando pasos para la personalización del mismo. En dicha personalización hay que tener en cuenta: logos de la distribución, colores predefinidos, estilos para nombrar los elementos del portal, etc.
4. Adaptación de la localización. Los términos a usar dentro de los contenidos que tengan que ver específicamente con la distribución, conllevan a la especificación de los mismos dentro del escenario de la localización. Los contenidos, los componentes de los mismos y los metadatos, tienen que especificarse llevando un estudio de la terminología. En el Portal NovaLNX por ejemplo se identificaron términos que tendían a confundir teniendo en cuenta los objetivos y el tema principal del proyecto, por lo que se procedió a su traducción a lenguajes nativos como el español y otros como el inglés. También se definió renombrar algunos

- componentes que especificaban determinados contenidos para su mejor manejo.
5. Políticas de compartimiento de ficheros y el manejo de adjuntos por parte de usuarios. El establecimiento de políticas que aseguren estándares a la hora de manejar ficheros dentro del portal de parte de usuarios del mismo, tiene las cualidades de dejar definido por ejemplo: Tipos de archivos a gestionar, tamaños de los mismos, tiempo de publicación, etc. En el caso particular de NovaLNX, la sección de las descargas de archivos necesitaba de una definición alrededor de los mismos, la misma se realizó teniendo en cuenta los tipos de paquetes y distintos contenidos a publicar.
 6. Conformación de manuales que respondan a preguntas frecuentes de los usuarios. Como parte de los requerimientos presentados para la selección, se opta por la conformación de manuales online que cumplan con necesidades o dudas de usuarios tanto del uso del Portal entorno a fases del ciclo de vida de los contenidos como de la distribución en general.

Implantación teniendo en cuenta el software requerido. Para la implantación desde su fase de prueba hasta la actual, se utilizó el SGBD MySQL, un servidor de aplicaciones Web Apache Tomcat 5.5, la máquina virtual de Java (JVM) mayor que 1.5 para el mejor manejo de imágenes y como sistema operativo, la propia distribución NovaLNX. En la actualidad debido al poco uso por encontrarse en fase de prueba aún, los requerimientos de hardware no sobrepasan los disponibles en el propio laboratorio del proyecto (Pentium 4, 256 RAM, 2.6 GHZ, 80 GB).

Las pruebas que se le han hecho al Portal NovaLNX desde su implantación son desde el punto de vista de soporte técnico y capacidad de asimilación y manejo de un número alto de conexiones simultáneas. Hasta al momento las mismas han revelado resultados satisfactorios y alentadores, no dejando dudas a la buena decisión de la implantación de una herramienta de este tipo.

Conclusiones.

Con la realización de este trabajo se da cumplimiento a los objetivos planteados y se logra la profundización en temáticas que aportan, antes que nada, conocimientos a los autores.

Al concluir el mismo se ha conformado, mediante una investigación, un material que integra elementos necesarios para la selección, adaptación e implantación de Portales CMS libres u open source. Elementos conformados por estudio del ciclo de vida de los contenidos, la estructura básica de los Portales CMS teniendo en cuenta dicho ciclo, un estudio de las licencias, todo un modelo a seguir en una selección y un cúmulo de tecnologías libres a tener en cuenta para la modificación e implantación.

Dichos elementos demuestran su validez dando cumplimiento a la modificación e implantación de una herramienta que responde a la gestión de contenidos para la distribución NovaLNX.

Recomendaciones.

Este trabajo integra elementos para la selección, adaptación e implantación de Portales CMS libres u open source, dados los elementos que aquí se brindan se recomienda:

- La profundización del estudio sobre distintos modelos de gestión de contenidos para un mejor planteamiento de procesos a seguir de acuerdo a variadas necesidades en nuestra universidad.
- La integración de NovaLNX con sistemas similares existentes en nuestra universidad.
- La adaptación de nuevos módulos al Portal NovaLNX de acuerdo a nuevas necesidades que surjan en la distribución y en la universidad.
- El estudio de variantes específicas de Portales CMS libres para la adaptación en nuestra universidad en distintos proyectos.

Bibliografía Consultada

CUERDA, X. Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. 2004. [Disponible en: <http://www.uoc.edu/mosaic/articulos/cms1204.html>].

TRAMULLAS, J. Herramientas de software libre para la gestión de contenidos. 2005. [Disponible en: http://www.betsime.disaic.cu/secciones/tec_ja_05.htm].

BOIKO, B. Content Management Bible. Wiley. 2001.

STALLMAN, R. Software libre para una sociedad libre. 2004.

GLASSHAUS. Content Management Systems. 2003.

IPIÑA, D. Programación Web Open Source con LAMP. 2005.

GAYO D, LÓPEZ B, VINUESA L, LABRA J, CUEVA, J. Utilización de software libre como única tecnología para el desarrollo de portales web. 2001. [Disponible en: <http://www.di.uniovi.es/~dani/?publications:papers>].

TOURIS A, MARIÑAN M. Arquitectura Empresarial y Software Libre. 2002.

VILLA, L. Servidores de portales: ¿usabilidad empaquetada? 2003.

NUESCHELER D, BOYE J. JSR-170: What's in it for me? 2005. [Disponible en: <http://cmswatch.com/Feature/123>].

Bibliografía

BOIKO, B. (2001), Content Management Bible . Wiley.

BROWNING P, LOWNDES M. JISC TechWatch Report: Content Management Systems . TechWatch Report TSW 01-02, The Joint Information Systems Committee.2001.

MCGOVERN G, NORTON R. Content Critical. 2001.

NAKANO, R. Web Content management. A Collaborative Approach. 2002.

HACKOS, J. Content Management for Dynamic Web Delivery. 2002.

PEÑALVO, F. Estado actual de los sistemas e-learning. 2005. [Disponible en: http://www3.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm].

STALLMAN, R. Software libre para una sociedad libre. 2004.

ROBERTSON, J. How to evaluate a content management system. 2002. [Disponible en: (http://steptwo.com.au/papers/kmc_evaluate/index.html).

ROBERTSON, J. Looking towards the future of CM. 2da Edición. 2003. [Disponible en: http://www.steptwo.com.au/papers/cmb_future/index.html].

JACOBSON R. Information Design. 1999.

FSF The Free Software Foundation. [Disponible en: <http://www.fsf.org>][Consulta: Enero 2006].

Open Source Initiative OSI. [Disponible en: <http://www.opensource.org>][Consulta: Enero 2006].

GLASSHAUS. Content Management Systems. 2003.

The CMS Matrix. The Content Management Comparison Tool. [Disponible en: <http://www.cmsmatrix.org>][Consulta: Septiembre 2005].

OpenSourceCMS [Disponible en: <http://www.opensourcecms.com>][Consulta: Septiembre 2005].

CMS Info [Disponible en: <http://www.cmsinfo.org>][Consulta: Septiembre 2005].

CMS Watch: Content Management, Records Management, Enterprise Search, and Portal Reports [Disponible en: <http://www.cmswatch.com>][Consulta: Octubre 2005].

Linux Online [Disponible en: <http://www.Linux.org>][Consulta: Diciembre 2005].

MySQL Hispano – La comunidad de usuarios de MySQL [Disponible en: <http://www.mysql-hispano.org>][Consulta: Diciembre 2005].

MySQL AB. MySQL [Disponible en: <http://www.mysql.com>][Consulta: diciembre 2005].

Apache Software Foundation. Apache [Disponible en: <http://www.apache.org>][Consulta: Enero 2006].

PHP Development Team. PHP: Hypertext Preprocessor [Disponible en: <http://www.php.net>][Consulta: Diciembre 2005].

O'Reilly & Associates, Inc. Online Documentation. Perl [Disponible en: <http://www.perl.org>][Consulta: Noviembre 2005].

Python Software Foundation. Python [Disponible en: <http://www.python.org>][Consulta: Enero 2006].

IPIÑA, D. Programación Web Open Source con LAMP. 2005.

WAMP Server. WAMP [Disponible en: <http://www.wampserver.com>][Consulta: Febrero 2006].

Fundación SIDAR – Acceso Universal [Disponible en: <http://www.sidar.org>][Consulta: Diciembre 2005].

JetNuke [Disponible en: <http://www.jetnuke.org>][Consulta: Junio 2005].

Java.net. JetNuke [Disponible en: <https://jetnuke.dev.java.net>][Consulta: Enero 2006].