

**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS
FACULTAD 10**



**Proceso de desarrollo de aplicaciones Web
sobre la plataforma Drupal**

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autores:

Yeleny Serrano Barrero
Ekaterina Perdigón Torres

Tutor:

Ing. Yanier Castellanos González

Ciudad de la Habana, Junio del 2007

Declaración de autoría

Yeleny Serrano Barrero y Ekaterina Perdigón Torres se declaran como únicas autoras de este trabajo y autorizan al proyecto Portales de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, se firma la presente a los ____ días del mes de _____ del 2007.

Yeleny Serrano Barrero

Ekaterina Perdigón Torres

Yanier Castellanos González

Datos de Contacto

El tutor de este trabajo investigativo es el Ing. Yanier Castellanos González, profesor de la facultad 10. Desempeña el papel de líder del proyecto “Portales PHP” desde hace 2 años y se especializa en las líneas de investigación del proyecto, Desarrollo web y Sistemas de Gestión de Contenido (CMS).

“Cuanto más alto coloque el hombre su meta, tanto más crecerá” .

Johann Christoph Friedrich von Schiller

Agradecimientos

A mi mamá Gladys Neyda por apoyarme de forma incondicional durante toda la carrera y darme fuerzas para seguir adelante. A mi hermana Galina por poder siempre contar con ella. A mi primo Yoan por siempre tenerme presente y preocuparse por mi mamá, mientras yo he estado lejos. A mi tía Lily que se ha preocupado por mí y ha guiado mis pasos para que yo salga adelante. A mis tíos, Cuchito y Eva, por el apoyo que siempre me han brindado. A mis abuelos que siempre están al pendiente de mí y me han apoyado en todo momento. A Angel que está conmigo en las buenas y malas, que me ha apoyado incondicionalmente y que ha sido protagonista de este momento tan importante de mi vida. A la familia de Angel por acogerme como un miembro más de la familia y preocuparse por mi desarrollo profesional.

Ekaterina

A Dios por serme fiel, por infundirme las fuerzas necesarias para seguir adelante en todo momento. A mi querida madre del alma, Lilian, que lo ha dado todo por mí, que no ha cesado de transmitirme su amor, y a quien le debo la vida. A mi hermanita Yele, que más que hermana ha sido mi mejor y más fiel amiga. A mi querido padre Gabriel, que me brindó su apoyo incondicional. A mi tía Ely, que estuvo siempre al tanto de mis estudios. A mi queridísimo amigo Adrian de quien he recibido un gigantesco apoyo y a quien agradezco inmensamente toda su ayuda, a mi peluche Bocacita. A Luis M. Teijón que me ha apoyado en las buenas y en las malas, que me ha dado su amor, su cariño, su comprensión, y con el cual he compartido muchos momentos de alegría.

Yeleny

A la Universidad de las Ciencias Informáticas por formarnos tanto ideológica como profesionalmente, al tutor de este trabajo investigativo, el Ing. Yanier Castellanos González. A nuestros compañeros de estudios por compartir tantos momentos inolvidables, y por ser partícipes de este maravilloso sueño. En especial a Yaima Viltres, Annarelys, Yaima, Yanexis, Yaquelin, Yunior, Yaisel y Sayli.



Dedicatoria

Dedico este trabajo investigativo a mi madre Gladys Neyda, a mi hermana Galina, a mi sobrinita Stephany, a mi primo Yoan, a mi tía Lily, a mis abuelos Ana Delia, Gladys y Juan Manuel, a mis tíos Eva y Cuchito, a mi novio Angel. A ellos que día a día me brindan su amor y apoyo incondicional. En especial quiero dedicar todo el esfuerzo impregnado en este trabajo a mi padre Juan Esteban, que aunque la vida no le dio la oportunidad de disfrutar este momento, siempre estuvo presente y sé que estará orgulloso de mí.

Ekaterina Perdigón Torres

Dedico esta investigación a Dios, a mis queridos padres Lilian y Gabriel, a mi hermana Yelena, A ellos que siempre esperaron el día en que me hiciera ingeniera, que me apoyaron con amor y me guiaron por un buen camino.

Yeleny Serrano Barrero

Resumen

Actualmente el desarrollo de aplicaciones Web se ha convertido en un reto para sus desarrolladores, debido a la demanda de aplicaciones más rápidas, ligeras y robustas. En la Universidad de las Ciencias Informáticas se desarrollan diversos proyectos dedicados a este fin, entre ellos se encuentra el proyecto Portales. En el cual no se sigue un mismo modelo de desarrollo, ni se trabaja de forma organizada, por lo tanto se hace necesario organizar y estandarizar el desarrollo de aplicaciones Web para así lograr la creación de portales de manera rápida y eficiente. En la presente investigación se realizará un estudio sobre la gestión de proyectos con el fin de formalizar y establecer un proceso estándar de desarrollo de aplicaciones Web utilizando la herramienta Drupal, permitiendo a estudiantes y profesores involucrados en el proyecto trabajar de forma organizada y eficiente. Los resultados obtenidos fueron: una guía de desarrollo adaptada al proyecto, que permite seguir un mismo estándar en la creación de portales Web y una guía donde se describe cómo realizar el montaje de un repositorio de módulos de Drupal en la herramienta de desarrollo colaborativo GForge, que posibilita un acceso más organizado a dichos módulos.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica	5
Introducción.....	5
1.1 Aplicaciones Web.....	5
1.1.1 Áreas de diseño de una aplicación Web.....	6
1.1.2 Páginas Web	6
1.2 Gestión de Proyectos.....	7
1.2.1 Proceso de planificación.....	8
1.2.2 Herramientas que facilitan la Gestión de Proyectos	9
1.2.2.1 Características de Drupal.....	10
1.2.2.2 Arquitectura de Drupal	12
1.2.2.3 Requerimientos del sistema.....	12
1.2.3 Metodologías de desarrollo	14
1.2.3.1 Proceso Unificado de Rational (RUP).....	14
1.2.3.2 Programación eXtrema (XP)	14
1.2.3.3 Desarrollo Manejado por Rasgos (FDD)	15
1.3 Desarrollo orientado a Componentes.....	15
1.3.1 Programación orientada a componentes	16
1.3.2 Principales tecnologías de componentes	17
Conclusiones.....	18
Capítulo 2. Tendencia y tecnologías actuales utilizadas.	19
Introducción.....	19
2.1 Metodologías y herramientas	19
2.1.1 Fundamentación de la metodología a utilizar	19
2.1.1.1 Comparación entre las metodologías RUP, XP y FDD	20
2.1.1.2 Decisión	22

2.1.2 Control del proyecto.....	23
2.1.2.1 Características de GForge	23
2.1.2.2 Arquitectura de GForge.....	24
2.1.2.3 Funcionalidades de GForge	25
2.1.2.4 ¿Qué hacer para controlar un proyecto en GForge?	25
2.1.3 Control de versiones.....	26
2.1.3.1 Características de Subversión	26
2.1.3.2 Arquitectura de Subversión	26
2.1.3.4 Funcionalidades de Subversión	27
2.2 Drupal.....	27
2.2.1 Desarrollo de módulos.....	27
2.2.1.1 Instalación de módulos en Drupal.....	27
2.2.2 Desarrollo de plantillas	28
2.2.2.1 ¿Cómo crear una plantilla en Drupal?.....	29
2.2.2.2 Tipos de plantillas	30
2.3 Propuesta de solución.....	33
2.3.1 Repositorio de módulos	33
2.3.1.1 Reglas para usar módulos	34
2.3.2 Fases de desarrollo para Portales Web.....	34
2.3.2.1 Fase de inicio	34
2.3.2.2 Fase de diseño e Implementación	36
2.3.2.3 Fase de ensamblaje.....	37
2.3.2.4 Fase de transición.....	38
Conclusiones.....	39
Conclusiones	40
Recomendaciones	41
Referencias Bibliográficas	42
Bibliografía.....	43
Anexos.....	45
Glosario de Términos	47



Introducción

Uno de los problemas que afronta actualmente la esfera de la informática es lo referente a la calidad del software, debido a que las aplicaciones son cada vez más complejas y simulan procesos del mundo real. Además existe una mayor probabilidad de ocurrencia de errores en las fases de desarrollo de un proyecto, ya que se requiere de un mayor período de tiempo para su implementación. Para minimizar estos errores se adoptan medidas tales como: la implementación y cumplimiento de estándares de desarrollo, metodologías para elevar la productividad y a su vez unificar el trabajo de los equipos de desarrollo en este sentido.

Es por eso que surge la gestión de proyectos, como una vía para organizar y administrar recursos, a fin de cumplir los plazos de tiempo establecidos para el desarrollo de un software y simultáneamente aminorar los efectos de la crisis de software afrontada por la humanidad.

La gestión de proyectos transita por 5 procesos: inicio, planificación, ejecución, control y cierre, los cuales deben llevarse a cabo de la mejor manera posible para así obtener un software con la calidad requerida. Además se deben tener en cuenta determinados aspectos que se interrelacionan entre sí, como son el tiempo, costo y alcance, donde uno no puede ser modificado sin impactar a los otros. El manejo adecuado de los mismos debe conducir a la obtención de un producto final con la calidad solicitada, teniendo en cuenta que el tiempo empleado y el costo resultante sean los menores posibles.

No obstante, ocasionalmente se produce un incremento tanto del tiempo como del costo al ser necesario desarrollar proyectos de gran envergadura y alta complejidad. Debido a esto, la gestión de proyectos debe proporcionar herramientas y técnicas que permitan al equipo de desarrollo organizar el trabajo, de forma tal que siguiendo un mismo estándar, la magnitud de un proyecto no constituya un obstáculo.

La Universidad de las Ciencias Informáticas lleva a cabo numerosos proyectos productivos entre los que se encuentra el proyecto Portales, que se encarga del desarrollo de aplicaciones Web



mediante diversas herramientas. Para llevarlo adelante, se hace necesario organizar y estandarizar el desarrollo de aplicaciones con el objetivo de lograr una mayor organización y solidez dentro del proyecto. Es por ello que se debe hacer un excelente proceso de planificación para así lograr una buena gestión de proyectos y luego obtener un producto de alta calidad.

La necesidad de realizar una gestión de proyectos apropiada condujeron a determinar el problema siguiente: ¿cómo lograr que el desarrollo de aplicaciones Web en el proyecto Portales fluya de manera organizada y coherente?

A fin de establecer los límites de la presente investigación se precisa, primeramente, analizar la tendencia actual de las aplicaciones Web, que consiste en generar las páginas de forma dinámica y gestionar el contenido con sistemas de gestión de bases de datos.

La organización y coherencia del proyecto Portales se alcanzan con el auxilio de determinadas herramientas que facilitan el trabajo, entre las que se encuentran los gestores de contenidos libres, los cuales van simplificando paulatinamente la acción de compartir contenidos en la Web. Entre ellos, se hallan los sistemas de administración de contenidos que permiten al usuario administrar el contenido de los sitios Web, brindando grandes facilidades, integrando todo tipo de módulos y modificando el código adaptándolo a las necesidades, debido a que su código es abierto y se encuentra muy bien estructurado. Este es el caso de Drupal, el sistema de Administración de Contenido libre empleado en el proyecto Portales, que permite publicar artículos, imágenes y servicios añadidos tales como: foros, encuestas, votaciones y administración de usuarios y permisos.

El objeto de estudio de esta investigación está relacionado específicamente con la gestión de proyectos para portales Web y sus tendencias actuales.

El campo de acción es el proceso de planificación de portales corporativos dentro de la gestión de proyectos, utilizando la herramienta Drupal.

La hipótesis de la investigación es: Si se logra llevar a cabo una investigación científica satisfactoria de la gestión de proyectos, entonces el equipo de desarrollo del proyecto Portales



podrá trabajar de manera organizada y eficiente. Por lo que la variable Independiente es el estudio de la gestión de proyectos y la dependiente persigue lograr una investigación lo más amplia posible para que el equipo de desarrollo del proyecto Portales pueda desarrollar aplicaciones web sobre la plataforma Drupal de manera organizada y eficiente.

El objetivo principal de la investigación es elaborar una estrategia de desarrollo de aplicaciones Web en el proyecto Portales, de forma tal que el trabajo se haga más organizado y eficiente utilizando la herramienta Drupal. Se tienen como objetivos específicos orientar y brindar información al personal del proyecto acerca de cómo lograr eficiencia y organización a la hora de desarrollar una aplicación Web, para de esa forma hacer más factible el trabajo.

Para dar cumplimiento a los objetivos propuestos se han llevado a cabo las siguientes tareas de la investigación:

- Estudio del entorno de trabajo.
- Estudiar los antecedentes y tendencias actuales de las diferentes herramientas que se utilizan para el desarrollo de aplicaciones Web.
- Identificar las deficiencias en el desarrollo de aplicaciones Web dentro del proyecto.
- Estudio de la gestión de proyectos.

Para el desarrollo de las tareas definidas se utilizaron los siguientes métodos teóricos y empíricos:

Métodos teóricos

Analítico-sintético: Se utilizará este método con el objetivo de buscar todo lo referente a la gestión de proyectos de portales Web, pues da la posibilidad de analizar las teorías y documentos relacionados, permitiendo de esa forma extraer los elementos más importantes y significativos.

Inductivo-deductivo: Se utilizará este método con el objetivo de obtener conocimientos generalizadores, pues permite ir de lo particular a lo más general, además de poder llegar de un



Capítulo 1. Fundamentación teórica

Introducción

En este capítulo se aborda las aplicaciones Web en sentido general, la gestión de proyectos y dentro de ella las herramientas y metodologías que hacen posible su desarrollo, así como el desarrollo orientado a componentes.

1.1 Aplicaciones Web

En un principio las aplicaciones Web estaban diseñadas estáticamente, donde cada aplicación tenía su propio programa cliente y su interfaz de usuario. El cliente y el servidor eran instalados separadamente, de forma tal que si se hacía algún cambio o mejora en el servidor lo requería el cliente también.

Con el transcurso del tiempo Internet ha alcanzado un gran desarrollo y se han abierto las puertas al acceso de la información desde cualquier parte del mundo. Por lo tanto las aplicaciones Web se han convertido en un desafío para sus desarrolladores, ya que el modo de generar las páginas ha ido evolucionando, conjuntamente con la demanda de aplicaciones más rápidas, ligeras y robustas.

Existen dos tipos de páginas Web las estáticas y las dinámicas. Las primeras funcionan en cualquier servidor y no requieren de ninguna instalación, ni configuración de software. Además son creadas en el lenguaje HTML, que solo permite desarrollar textos planos acompañados de imágenes. Las segundas presentan un mayor dinamismo entre las páginas, incluyen contenidos reutilizables y para su desarrollo pueden utilizarse varios lenguajes de programación. Facilitan la actualización constante de la información y la renovación de las imágenes del sitio, así como el control total de la administración de contenidos. Además ofrecen autonomía en la administración de contenidos usando un CMS, presentación de contenidos en diversos dispositivos y formatos, entre otras.



Las páginas estáticas requieren de menor costo de producción y requerimientos técnicos, pero las dinámicas ofrecen un sitio bien estructurado, configurable, extensible, etc.

1.1.1 Áreas de diseño de una aplicación Web

El diseño de una aplicación Web presenta tres áreas fundamentales. El diseño de la interfaz de usuario, que debe encontrar un equilibrio entre los tipos de exploradores admitidos, la estética de la aplicación y su facilidad de uso. Un diseño eficaz de la base de datos, que garantice un tiempo de respuesta rápido por parte del cliente. Para diseñar una base de datos adecuada, es necesario distribuir apropiadamente los datos entre la base de datos y el sistema de archivos. La última área consiste en elaborar un plan de acceso a la base de datos, en el que se especifique qué páginas HTML requieren contenido de base de datos, lo cual le permite al diseñador determinar cuáles páginas deben generarse de forma totalmente dinámica mediante secuencias de comandos de servidor y cuáles pueden crearse pseudo dinámicamente para generar páginas estáticas de acuerdo con una programación predeterminada. (1)

1.1.2 Páginas Web

Las páginas Web pueden clasificarse en informativas (**figura 2**), de negocio (**figura 3**) y portales (**figura 4**). En las informativas solo se muestra información del centro, es decir quienes somos, visión, historia, servicios, noticias, eventos, productos, etc. Estas páginas deben tener un formulario de contacto, además de mostrar información de la ubicación, emails y teléfonos del centro. La página principal debe tener un diseño cómodo y ligero, así como toda la información importante de la Web de manera resumida.

Las páginas Web de negocio tienen un diseño sencillo que combinan con aplicaciones Web básicas y avanzadas. Un ejemplo de estas son las cotizaciones en-línea, donde el cliente llena un formulario y los datos entrados son almacenados en una base de datos, para luego dar respuesta a esta solicitud mediante cualquier vía de contacto que se haya ingresado en el formulario. Existen también las cotizaciones personalizadas automáticas, donde el cliente solicita

organizar el trabajo, asignar tareas, controlar el funcionamiento del proyecto, gestionar la calidad, entre otras. Consta de 5 procesos: inicio, planificación, ejecución, control y cierre. El proceso inicial define el comienzo y necesidad de un proyecto o fase, además establece el compromiso y viabilidad para su puesta en marcha. El de planificación realiza un esquema de trabajo para alcanzar los objetivos propuestos inicialmente, donde se definirán el alcance, los objetivos y resultados, así como la secuencia y duración de las actividades, la planificación de la calidad, la planificación organizativa, la adquisición y asignación de recursos, la planificación de la contratación externa, la planificación de la comunicación, entre otras. El de ejecución pone en marcha el plan del proyecto y se encargará de verificar el alcance, el aseguramiento de la calidad, el desarrollo del equipo de proyecto, la distribución de la información, así como la selección y gestión de la contratación externa. El de control verifica que los objetivos del proyecto están cumpliéndose, mediante la supervisión y medición de los progresos alcanzados y tomando acciones correctivas cuando es necesario. Finalmente se lleva a cabo el proceso de cierre, donde se formalizará la finalización del proyecto y esto incluirá la resolución de contratación externa y el cierre administrativo de la fase o proyecto. (4)

1.2.1 Proceso de planificación

La planificación se divide en dos grandes fases, la primera es en la que se estudia y establece la viabilidad de un proyecto. Además se deben realizar los correspondientes estudios técnicos, así como una estimación de los recursos necesarios y los costos generados. Todo ello le permite al cliente decidir sobre la realización o no del proyecto.

La segunda fase se lleva a cabo si se ha decidido ejecutar el proyecto y a partir de ahí se hace una planificación detallada del mismo.

El proceso de planificación se encarga de definir el alcance, las actividades, costo, recursos, objetivos, alternativas y riesgos de un proyecto. La definición del alcance se realiza para refinar la trayectoria del trabajo a efectuar para obtener el producto, donde se incluye un sumario detallado del producto y de subproductos parciales necesarios tanto internamente, como externamente.

Además se define una estructura jerárquica de las actividades a realizar en el proyecto. La definición de las actividades refina la descripción de las actividades a realizar y estas tienen una secuencia que determina el orden de ejecución de las mismas para así alcanzar los objetivos y tener como resultado el producto. Además se establece una estimación de la duración de las actividades, basándose en la disponibilidad y habilidades de los recursos humanos. La planificación de recursos determina la cantidad y tipo de recursos necesarios para desarrollar el proyecto, luego se debe estimar el costo que se establece para cada actividad, permitiendo así el cálculo del presupuesto del proyecto. La planificación de gestión de riesgos describe los riesgos que pueden producirse en el proyecto y desarrolla procedimientos para darles respuesta.

La planificación organizativa y asignación de recursos se realiza mediante la identificación, documentación y asignación de roles, responsabilidades y relaciones jerárquicas, así como la adquisición de los recursos humanos necesarios. La planificación de la comunicación determina las necesidades de información y comunicación entre los integrantes del proyecto, donde se establecen métodos para la gestión de las comunicaciones entre miembros del proyecto, con el cliente, con los responsables, etc. (5)

1.2.2 Herramientas que facilitan la Gestión de Proyectos

Dentro de las herramientas que facilitan la gestión de proyectos se destacan los Sistemas de Gestión de Contenidos o CMS (Content Management System), que permiten la creación, administración y manejo del contenido de forma independiente.

Un CMS es una herramienta que permite a los usuarios modificar, administrar, manejar y publicar de forma rápida y segura el contenido de una página Web. Da la posibilidad de actualizar, tanto el contenido como el diseño del sitio Web en forma regular sin necesidad de tener los conocimientos técnicos requeridos, además controlar que sólo las personas autorizadas accedan al sitio, mejorando de esta forma la calidad y eficiencia del mismo. Gracias a que posee una estructura modular, permite personalizar el desenvolvimiento de nuevos módulos específicamente para atender a las necesidades tanto de una empresa como de un usuario en

particular.

Los Sistemas de Gestión de Contenido trabajan en conjunto con una base de datos que es utilizada para almacenar documentos, texto e imágenes, lo cual significa que el contenido puede ser reutilizado en muchos lugares dentro del sitio Web. Estos generan automáticamente las páginas nuevas y sub-niveles de navegación, lo cual proporciona una gran interactividad. Además posibilitan la aplicación de un mismo estilo a su entorno visual y una misma estructura mediante patrones de páginas, ofreciendo así una consistencia de la Web.

Dentro de los CMS se encuentra Drupal, que es un Sistema de Administración de Contenido libre, bajo los términos de la Licencia Pública General (GPL), que se destaca por la calidad de su código y por la forma dinámica de generar las páginas. Su diseño es utilizado fundamentalmente para construir y gestionar un gran número de aplicaciones Web. Extiende sus funcionalidades básicas mediante la implementación de módulos. El código fuente de los Sistemas de Administración de Contenido es abierto, lo cual posibilita que los desarrolladores puedan hacer modificaciones y mejoras al código, haciéndolo cada vez más potente. Drupal se caracteriza además por ser un potente gestor de contenido, que se encarga de formar grandes comunidades de usuarios registrados que colaboran en un fin común. (6)

1.2.2.1 Características de Drupal

Gestión de Usuarios

Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como cualquier otro sitio. Según el rol que se tenga se establecen los permisos, es decir los administradores de Drupal no tienen que establecer permisos para cada usuario, porque estos son agrupados por roles.

Gestión de contenido

Drupal permite el seguimiento del contenido en sus modificaciones, haciendo posible llevar a cabo el control de versiones. También permite comentar sobre dichos cambios y deshacer los

Capítulo 1. Fundamentación teórica

mismos recuperando una versión anterior. A todo contenido dado de alta en Drupal se le asigna un enlace permanente, evitando así contenidos perdidos, incluso páginas. Todo contenido en Drupal, funciona como un objeto, llamado “Nodo”. De esta forma todo el contenido es tratado por igual, independientemente del tipo que sea.

Drupal toma por separado el contenido de la interfaz gráfica del sitio, permitiendo así que se pueda cambiar este último usando plantillas con HTML, PHP y CSS. El contenido es exportado en formato RDF/RSS, de esta forma puede ser utilizado por otros sitios Web.

Blogging

Drupal incluye un potente agregador de noticias para leer y publicar enlaces a noticias de otros sitios Web. La API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que proporcionen un entorno de edición más completo.

Plataforma

Drupal permite incorporar soporte para diferentes bases de datos. Ha sido diseñado para ser multiplataforma, por lo que puede funcionar con Apache o Microsoft IIS como servidor Web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Proporciona opciones para crear un portal multilingüe.

Administración y Análisis

La administración y configuración del sistema Drupal se puede realizar enteramente con un navegador y no necesita específicamente de ningún software adicional. Drupal puede realizar un gran análisis y seguimiento, mostrando en las páginas Web de administración informes sobre enlaces entrantes. Toda actividad y sucesos del sistema son capturados en un 'registro de eventos', que puede ser visualizado por un administrador.

Características de comunidad

Drupal proporciona un potente modelo de comentarios enlazados, lo cual posibilita participar fácilmente en la discusión que se esté llevando a cabo en el foro u otra sección del sitio. Incluye

Capítulo 1. Fundamentación teórica



Drupal 5.x trabaja sobre Apache 1.3 o Apache 2.x.

Opcional: IIS, el código de Drupal trabaja usando IIS5 o IIS6 si PHP es configurado correctamente.

Drupal está desarrollado para funcionar independiente del servidor Web.

2. PHP

Se recomienda el uso de las últimas versiones de PHP 4.x o 5.x por seguridad y futura compatibilidad, así como la extensión XML de PHP (para los módulos bloggerapi, drupal, jabber, ping). Esta extensión está habilitada por defecto en una instalación estándar de PHP; la versión de PHP de windows tiene soporte integrado para esta extensión.

PHP necesita las siguientes configuraciones para que Drupal funcione:

- Session_save_handler = user

Además, se recomiendan las siguientes configuraciones:

- Session_cache_limiter = Ninguno

Estas configuraciones están contenidas por defecto en el archivo .htaccess que viene con Drupal, las opciones de configuración de PHP en el .htaccess solo funcionan:

- Con Apache (o un servidor Web compatible).
- Si el .htaccess es actualmente de lectura, por ejemplo, AllowOverride no funciona.
- Si PHP está instalado como un módulo de apache.

Usar una base de datos soportada por PEAR.

3. Un servidor de base de datos soportado por PHP.

Se recomienda MySQL v3.23.17 o superior, pero es fuertemente recomendado el uso de las versiones 4.1 ó 5.0 para futura compatibilidad con el Drupal 6.



Opcional: PostgreSQL 7.3 o anterior. (7)

1.2.3 Metodologías de desarrollo

Para obtener un software con la calidad requerida debe llevarse a cabo una metodología de desarrollo y principalmente cuando los proyectos que se van a desarrollar son de mayor envergadura. Además se debe tener presente cuál metodología usar en dependencia de las necesidades.

Existen tres metodologías que son consideradas las más importantes, RUP, XP y FDD.

1.2.3.1 Proceso Unificado de Rational (RUP)

Es una metodología de desarrollo que puede adaptarse a cualquier proyecto y se divide en 4 fases de desarrollo como son inicio, elaboración, construcción y transición. La fase de inicio determina la visión del proyecto, la de elaboración es la que determina la arquitectura, es decir realiza la definición, análisis y diseño, la de construcción se encarga de la implementación y la de transición es cuando se llega al fin del proyecto y este se pone en marcha. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones y en cada ciclo de iteración se hace exigente el uso de artefactos. Los elementos que componen al RUP son las actividades que se determinan en cada iteración, los trabajadores, que son las personas involucradas en cada proceso y los artefactos que pueden ser un documento, un modelo o un elemento de modelo.

RUP se basa en casos de uso para describir lo que se espera del software. Basándose en el Lenguaje Unificado de Modelado (UML) como herramienta principal. Contiene abundante documentación y con él se trabaja muy organizado.

1.2.3.2 Programación eXtrema (XP)

Es una metodología de desarrollo muy ágil que es utilizada en proyectos de corto plazo. Se basa en pruebas unitarias, que se realizan a los principales procesos y a posibles fallas que pudieran ocurrir, también en la reutilización de código para lo cual se crean patrones o modelos

estándares. Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Lo esencial de esta metodología es llevar a cabo una buena comunicación entre usuarios y desarrolladores. La simplicidad, al desarrollar y codificar los módulos del sistema, así como la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

1.2.3.3 Desarrollo Manejado por Rasgos (FDD)

Es una metodología de desarrollo que es utilizada en proyectos con tiempo de desarrollos relativamente cortos y se basa en un proceso iterativo con iteraciones cortas que producen un software funcional. Se divide en 5 fases de desarrollo como son el desarrollo de un modelo general, la construcción de la lista de funcionalidades, plan de versiones en base a las funcionalidades a implementar, diseñar e implementar en base a las funcionalidades. Las dos últimas fases requieren de un mayor periodo de tiempo durante el desarrollo del proyecto. El trabajo de desarrollo y modelado se realiza a nivel de grupo y las funcionalidades de implementar las versiones se dividen entre los miembros del equipo. Esta metodología define métricas durante el proceso de desarrollo de la aplicación que pueden ser de mucha ayuda en proyectos futuros.

En general se puede decir que estas tres metodologías de desarrollo permiten obtener un producto final con la calidad requerida, pero se diferencian entre sí, en cuanto a funcionalidades y características. RUP está pensado para proyectos y equipos de desarrollo grandes y de mayor período de duración, mientras que XP y FDD están dirigidas a proyectos y equipos más pequeños.

1.3 Desarrollo orientado a Componentes

Desarrollar una aplicación es una actividad costosa en tiempo y dinero, por lo que se hace necesaria la utilización de componentes logrando de esta forma que el sistema sea más flexible. La tecnología de integración de componentes mejora la eficiencia del desempeño, pues hace

del uso y la capacidad de un componente de reemplazar a otro sin violar los contratos; la seguridad, que es la garantía que debe ofrecer un componente que respeta sus interfaces y contratos; la reflexión, que es la habilidad para conocer y modificar el estado de un componente.

1.3.2 Principales tecnologías de componentes

VBX/OCX/ActiveX de Microsoft

La Extensión de Visual Basic (VBX) es un modelo de componente usado en Microsoft Visual Basic, que apareció como una vía para distribuir controles entre los desarrolladores. Brinda la posibilidad de crear un software más poderoso y eficiente, permitiendo reutilizar códigos ya creados.

La Extensión de Controles OLE (OCX) son instrucciones preprogramadas capaces de ser insertadas en cualquier aplicación que está siendo desarrollada, permitiéndole al programador ahorrar tiempo. Está hecho dentro de Visual Basic y por lo tanto su código es compatible únicamente dentro del mismo.

ActiveX es una tecnología de Microsoft para el desarrollo de aplicaciones Web dinámicas. Es un componente que se puede insertar en una página Web y proporciona la funcionalidad de secuencias de animación. Pueden ser implementados en diferentes lenguajes de programación.

JavaBeans de Sun.

Es un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java y son capaces de encapsular la lógica empresarial de una aplicación, que es el código que cumple la finalidad de la aplicación. Puede ser reutilizado y manipulado visualmente por una herramienta de programación en lenguaje Java.

Un bean es una clase que se rige por varias reglas. La primera de estas es tener un constructor por defecto. La segunda es implementar el interface y la tercera tener introspección, que es cuando permite analizar a la herramienta de programación o IDE como trabaja el bean y conocer sus propiedades.



Modelo de Objetos de Componentes Distribuidos (DCOM)

Es un protocolo diseñado por Microsoft que permite a los componentes de software comunicarse directamente a través de una red de manera confiable, segura y eficiente. Funciona tanto con subprogramas de lenguaje Java como con componentes ActiveX gracias a su uso del Modelo de objetos componentes.

Dentro de DCOM se encuentra COM, que es un estándar para modelos basados en el modelo de objetos, el cual está diseñado para lograr que las aplicaciones puedan cooperar entre sí y ejecutarse en arquitecturas o sistemas operativos diferentes.

La tecnología basada en componentes es un método poderoso para la ingeniería de sistemas de información, ya que es una tecnología que está reduciendo el conflicto entre sistemas de alta complejidad, calidad y productividad.

Conclusiones

Con la culminación de este capítulo se puede decir que hacer una excelente gestión de proyectos es indispensable para la calidad del software, también que los Sistemas de Gestión de Contenido son cada vez más potentes y que le brindan gran soporte a los portales Web. Dentro de ellos se encuentra Drupal que tiene un gran número de funcionalidades y características que lo distingue de otras herramientas.



Capítulo 2. Tendencia y tecnologías actuales utilizadas.

Introducción

En este capítulo se aborda las metodologías y herramientas utilizadas, estableciendo una comparación entre las metodologías de desarrollo RUP, XP y FDD para definir cuál usar. Se incluirá el control de proyectos con el auxilio de la herramienta GForge y el control de versiones con Subversión. Se desarrollará una guía para crear módulos y plantillas en Drupal. Finalmente se dará la propuesta de solución.

2.1 Metodologías y herramientas

Existen diferentes metodologías de gestión de proyectos, dentro de ellas se encuentran RUP, XP y FDD que ofrecen herramientas de planificación, análisis y control de proyectos necesarios para una buena administración. Estos métodos son indispensables, ya que para obtener un proyecto de software con la calidad requerida se debe comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, las tareas a llevar a cabo, el plan a seguir, entre otras.

Para lograr una buena gestión de proyectos se debe tener en cuenta cómo llevar a cabo el control del proyecto de manera eficiente, utilizando la herramienta GForge y Subversión. Este último es el encargado de realizar el control de versiones que se vayan desarrollando para así poder recuperar versiones anteriores.

2.1.1 Fundamentación de la metodología a utilizar

En la actualidad la variedad de metodologías de desarrollo ha aumentado considerablemente y se han desarrollado dos corrientes en lo referente a procesos de desarrollo, los métodos pesados y los ligeros.

Las metodologías ágiles o ligeras reconocen las características inherentes de complejidad de software y tratan de mejorar su calidad por medio de una estrecha comunicación con las

hacer cambios en el código, aportando mejor calidad a este. No siendo así en RUP y FDD, ya que optan por la propiedad del código, aunque definen grupos y secciones de trabajo en conjunto.

Conocimiento sobre la arquitectura: En XP se conseguirá mediante la programación en pareja, pues en la creación del código se pueden evitar errores y malos diseños, controlando cada línea de código y decisión de diseño. Mientras que en RUP se intentará disminuir la complejidad del software a desarrollar mediante una planificación intensiva, para no perder el conocimiento de la aplicación. En FDD sin embargo se usan las sesiones de trabajo conjuntas en fase de diseño para conseguir una arquitectura sencilla y sin errores.

Evaluación del estado del proyecto: FDD permite definir métricas para evaluar el estado del proyecto, pues al dividirlo en unidades pequeñas es bastante sencillo hacer un seguimiento de las mismas. Mientras que XP define igualmente esos componentes pequeños, pero la tarea de realizar los reportes recae solo sobre los jefes de proyectos y en FDD está más distribuida en la jerarquía. RUP es tan grande y complejo que manejar el volumen de información que puede generar requiere mucho tiempo.

2.1.1.2 Decisión

En sentido general ninguna de estas tres metodologías esta completa, pues lo que le falta a una lo tiene la otra. Por lo tanto se deben integrar las que están más perfeccionadas como es el caso de RUP y XP, ya que FDD es el punto intermedio entre ellas, para así aplicar una metodología lo más completa posible.

Se ha decidido que la guía de desarrollo propuesta se desarrollará utilizando como base a XP y RUP, teniendo en cuenta las mejoras que nos ofrecen cada una. XP es un proceso orientado a la implementación y no genera mucha documentación. Esto en gran medida sería bueno, porque no se requiere de tanta documentación como en RUP, pero es necesario tener registrados los documentos más importantes del proyecto para así definir las funcionalidades a implementar,

entender el código y diseño con mayor claridad. Esto le permite al cliente comprobar el rendimiento y calidad del software, así como recibirlo en la fecha prevista. RUP define un sin número de roles que deben de minimizarse para que el equipo de desarrollo no sea tan grande y el proceso se haga más rápido. Es por ello que se debe aplicar la programación en pareja como lo hace XP, ya que favorece al trabajo colaborativo, donde cada miembro del equipo puede aportar sus conocimientos y hacer que el trabajo se haga en menor período de tiempo y con mejor calidad. La comunicación con el cliente debe ser más estrecha para así mantenerlo informado de la situación del proyecto y poder corregir el proceso si se aleja de sus necesidades. Se debe llevar a cabo la ejecución automática de pruebas unitarias para garantizar que el código este correcto y prevenir los errores.

2.1.2 Control del proyecto

El control de un proyecto se lleva a cabo con el auxilio de determinadas herramientas como es el caso de GForge, que es una herramienta de desarrollo colaborativo. Permite organizar y administrar un gran número de proyectos e integra un conjunto de herramientas aptas para el desarrollo de software y la gestión de proyectos, tales como: foros, herramientas de gestión y motorización de errores, repositorios de ficheros, así como herramientas de control de versiones de software. Cada proyecto que se registra en GForge contiene sus propios foros de discusión y cuando se registra uno nuevo este se genera automáticamente. Además contiene registros para registrar errores, soporte, parches y solicitud de funcionalidades entre los desarrolladores del proyecto.

2.1.2.1 Características de GForge

GForge es un sitio Web al que todos pueden acceder y a la vez contribuir al desarrollo colaborativo de software. Está patrocinado por la licencia GPL, por lo que presenta código abierto al que se le pueden hacer modificaciones y mejoras. Permite centralizar y homogeneizar la gestión de proyectos por lo que brinda gran utilidad a la hora de realizarla, ya que se pueden

registrar tareas, incidencias, listas de correos, discusiones, documentos, así como repositorio de código y software. Permite además la creación automática de uno o varios repositorios y garantiza el acceso controlado a este, dependiendo de la configuración de permisos en el proyecto. Proporciona un entorno configurable con control de versiones, así como herramientas para la comunicación entre desarrolladores y servidor Web por cada proyecto.

2.1.2.2 Arquitectura de GForge

En la **figura 1** se muestra la arquitectura de GForge, donde se utiliza Apache 2, que es el servidor Web que proporciona una interfaz al usuario mediante php 4.0 y le permitirá acceder a todas las funcionalidades del GForge. Como servidor de base de datos se utiliza Postgre-sql y para controlar todas las versiones de código que se realicen SVN. Postfix permite gestionar el correo electrónico y por lo tanto se encarga de procesar los mensajes de correo que llegan al servidor para enviarlos a donde corresponda y Mailman administra las listas de correo. SSH es un servidor de Shell seguro, que permite a los usuarios iniciar sesión en la máquina. Login/PAM es el sistema de inicio de sección de los usuarios que será configurado para utilizar la base de datos a la hora de aceptar o rechazar el login de un usuario. Por medio de tareas del Crontab los datos almacenados en la base de datos generan listas de correo y repositorios. Libnss-pgsql utiliza la base de datos para traducir los UID's y GID's, por nombres de usuarios normales locales de la máquina. Libpam-pgsql se utiliza para comprobar en la base de datos si un usuario existe y si es así verificar si la contraseña que ha proporcionado es válida, para luego permitirle iniciar sección.

El sistema GForge es administrado en su totalidad desde la interfaz Web y los cambios que se realicen se van a actualizar en la base de datos, la cual es usada por los demás componentes. Es multiplataforma y está implementado en php4.0-html-xml. Soporta todos los navegadores, ya que carga hojas de estilo dinámicas para cada uno. El texto puede ser traducido en múltiples idiomas, pues GForge tiene archivos para guardar las diferentes configuraciones de lenguajes y una clase en php para manejarlos según las preferencias del usuario.



2.1.2.3 Funcionalidades de GForge

GForge facilita el trabajo en equipo y hace que este sea cada vez más colaborativo, permitiendo a los miembros del proyecto desarrollar una base de conocimiento compartida y a su vez colocar librerías de códigos y programas para que otros usuarios puedan utilizarlo. Permite realizar encuestas sobre el proyecto, colocar noticias relacionadas con software, documentación, tareas, archivos, entre otras y a su vez descargar archivos de los proyectos que estén almacenados, administrar listas de correos asociadas al proyecto, así como asignar tareas y tener un seguimiento de las mismas.

Mediante GForge se pueden buscar personas por el nombre o login, por habilidades y grupos de software. Los proyectos que se almacenan en él son clasificados por categorías y mostrados en un árbol de proyectos.

2.1.2.4 ¿Qué hacer para controlar un proyecto en GForge?

En GForge primeramente se debe hacer la subscripción de los miembros del proyecto y el papel fundamental lo jugará el líder del proyecto, que es el que va a tener el rol de administrador y por lo tanto podrá acceder a todas sus funcionalidades. Los integrantes del proyecto pueden ser suscritos por el líder del proyecto o suscribirse ellos mismos.

El administrador registra el proyecto y asigna los roles a los integrantes del mismo. Es el encargado de administrar los registros, documentos, correos, noticias, tareas, foros, ficheros, control de versiones de código, además debe asignar las tareas a los integrantes del proyecto y revisar el cumplimiento de las mismas. El arquitecto de la información y el documentador deben añadir la documentación y actualizarla en el GForge, pues van a subir al repositorio toda la documentación del proyecto, la cual es revisada por el administrador y consultada por todos los miembros del proyecto que quieran acceder a ella. El programador debe subir los ficheros creados a GForge para estos ser utilizados por el montador. El administrador de módulos es el encargado de administrar el repositorio de módulos del proyecto. El líder del proyecto también se

va a encargarse de actualizar las noticias, realizar encuestas al equipo de trabajo y verificar el estado de las tareas asignadas a cada integrante.

2.1.3 Control de versiones

Para llevar a cabo el control de versiones con mayor eficiencia se puede hacer mediante Subversión, que es un software de sistema de control de versiones de código abierto y gratuito. Se encarga de manejar ficheros y directorios que se almacenan en un repositorio central y permite recuperar versiones antiguas de los datos almacenados o examinar el historial de cambios de los mismos.

Subversión puede ser instalado bajo linux, Mac OSX o Windows y también como un módulo de Apache o como un servicio independiente, accesible a través de varios protocolos.

2.1.3.1 Características de Subversión

Subversión proporciona versionado de directorios, implementado un sistema de ficheros versionado que sigue los cambios sobre árboles de directorios completos a través del tiempo; un verdadero historial de versiones, ya que se puede añadir, borrar, copiar y renombrar ficheros y directorios; envíos atómicos que permiten construir y enviar los cambios como fragmentos lógicos e impide que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito; versionado de metadatos con el que se puede crear y almacenar cualquier par arbitrario de clave/valor; elección de capas de red, lo cual ofrece gran estabilidad. Este servidor brinda acceso instantáneo a las características existentes como es la autenticación, autorización y compresión de la conexión.

2.1.3.2 Arquitectura de Subversión

Subversión está compuesto por un repositorio que conserva todos los datos versionados, un programa cliente que administra réplicas parciales de esos datos versionados (llamadas “copias de trabajo”). Entre ellos hay múltiples rutas a través de varias capas de acceso al repositorio.



2.1.3.4 Funcionalidades de Subversión

Una de las principales funcionalidades de Subversión es que nos permite ir guardando los archivos en un repositorio cada vez que se haga algún cambio y el sistema asigna una versión a los cambios realizados. Proporciona realizar el trabajo en equipo de forma más organizada, pues cualquier miembro del equipo puede bajar los últimos archivos y subir nuevos cambios. Cuando se necesite de alguna versión anterior o de la original se le puede pedir al servidor y automáticamente se recupera.

2.2 Drupal

En Drupal se utilizan módulos que son enviados a un repositorio y controlados por un administrador. Se crean plantillas, que hacen que las aplicaciones Web sean más configurables y que sus páginas tengan una estructura bien definida.

2.2.1 Desarrollo de módulos

Los módulos son ficheros con extensión (.module) que contienen funciones en PHP. Estas funciones actúan como hooks (ganchos), que son llamadas por Drupal durante sus procesos de gestión de contenido y construcción de páginas Web. De ahí que los módulos tienen la posibilidad de modificar y adaptar la información a visualizar en las páginas Web antes de que se mande definitivamente al navegador.

2.2.1.1 Instalación de módulos en Drupal

Para instalar un módulo en Drupal se deben seguir los siguientes pasos:

1. Descargar el módulo del repositorio de módulos, verificando que sea estable.
2. Descomprimir el archivo.
3. Mover la carpeta o el archivo .module a la carpeta módulos.
4. Leer las instrucciones de los ficheros ReadMe y Module.Install en los cuales aparece

información útil (dependencias de otros módulos, etc.).

5. Para activar el módulo se debe acceder a: administrar > construcción del sitio > módulos y luego activar el módulo que se desee y hacer clic en "Guardar la configuración".

6. Crear los permisos necesarios para los distintos roles. Para ello acceder a: administrar > usuarios > control de acceso y luego localizar el módulo para asignarle los permisos apropiados a los roles.

7. La configuración de los módulos viene con la instalación a diferencia de versiones anteriores de Drupal. (8)

2.2.2 Desarrollo de plantillas

Una plantilla es un fichero de texto, que define los elementos que deben aparecer en las páginas y su ubicación dentro de la misma. Se utilizan para definir la estructura global de las páginas y el contenido. Los nombres de las plantillas son fijos y no pueden cambiarse.

En el caso de Drupal se pueden crear plantillas en HTML o PHP con extensión (.tpl.php). Estas plantillas son ficheros HTML con código PHP que se interpreta dentro del contexto del engine, que es el que se encarga de extraer información de Drupal y dejarla en variables accesibles desde las plantillas.

Las plantillas son compatibles por defecto con XHTML y CSS. Los únicos nombres válidos que se utilizan para estas son: page.tpl.php, node.tpl.php, block.tpl.php, comment.tpl.php y box.tpl.php, los cuales son utilizados para definir las estructuras de las páginas, nodos, bloques, comentarios y contenedores, respectivamente.

La versión Drupal 5.0 contiene un nuevo sistema de gestión de plantillas llamado Garland, que implementa un gestor de esquema de colores para las plantillas. Ofrece también mejoras en PHPTemplate, ya que introduce la posibilidad de buscar plantillas sugeridas. Drupal 5.0 auto-descubre un amplio rango de plantillas que permite sobrescribir la vista por defecto de diversas páginas y proporciona diversas plantillas para páginas, bloques, nodos etc. Se pueden agregar

estilos de nodos del tipo ' blog ', agregando node-blog.tpl.php preferentemente sobre la plantilla estándar node.tpl.php. Las plantillas de la página principal son más flexibles con esta versión, por lo tanto se pueden crear plantillas específicas para nodos específicos.

En Drupal 5.0 PHPTemplate soporta el uso de plantillas múltiples de la página principal para un solo tema, dependiendo de la trayectoria actual del URL y busca estas plantillas múltiples antes de acceder al archivo page.tpl.php. Por ejemplo si se accede a la pagina: <http://www.example.com/node/1/edit> PHPTemplate buscaría las plantillas siguientes: page-node-edit.tpl.php page-node-1.tpl.php page-node.tpl.php page.tpl.php. (9; 10)

2.2.2.1 ¿Cómo crear una plantilla en Drupal?

Para crear una nueva plantilla se debe crear un nuevo directorio dentro del directorio de temas, por ejemplo temas/mitema. Luego se necesita crear un archivo llamado page.tpl.php en ese directorio para definir la plantilla de la página principal. También se pueden crear los siguientes archivos: block.tpl.php, box.tpl.php, comment.tpl.php, node.tpl.php y se deben agrupar en un paquete que es el que contiene todos los archivos de las plantillas.

Para modificar una plantilla se puede ir a administrar -> contenido -> templates (plantillas) y se muestra la lista de tipos de nodo con un enlace para crear o modificar la plantilla, luego se va al enlace y te aparecerán las plantillas para hacerle cambios o crear una nueva.

Para crear plantillas en Drupal se deben seguir los siguientes pasos:

- Diseñar el boceto para las páginas o secciones principales en una aplicación como Gimp, Photoshop u otra aplicación.
- Pasar el diseño XHTML y CSS a los archivos de PHPTemplate: page.tpl.php, node.tpl.php, comment.tpl.php, block.tpl.php, etc.
- Para obtener mayor personalización se pueden crear funciones en template.php para así usar archivos tpl.php para ciertos tipos de nodo, términos en la taxonomía, regiones. (6)

2.2.2.2 Tipos de plantillas

1- page.tpl.php

La plantilla de la página principal es la primera que se crea y se interpreta dentro del contexto del engine. Este pone a disposición de la plantilla una serie de variables con información del contenido. El número de variables disponibles, y su significado, depende enteramente del engine utilizado y del tipo de plantilla.

El engine redirige el contenido de la plantilla a la salida del servidor Web, es decir hacia el navegador para proporcionar una página bien estructurada. Lo primero que tiene que aparecer en el fichero page.tpl.php es la declaración del tipo de documento: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml" lang="<?php print $language; ?>" xml:lang="<?php print $language; ?>">`. En este caso el código PHP imprime la variable `$language` y el engine inicializa en función del idioma con el que se encuentre configurado Drupal.

Para definir la sección `<head>` se utiliza las variable `$head_title`, que contiene el título de la página, la variable `$head`, que contiene las cabeceras HTML enviadas por Drupal, y la variable `$styles`, que contiene los imports de las hojas (.css) a utilizar por la página.

Para definir la sección `<body>`, que va después de la cabecera sería de la siguiente forma:

`<body <?php $onload_attributes; ?>>`, donde la variable `$onload_attributes` debe imprimirse siempre al principio de la página para permitir la ejecución de los posibles scripts que contenga la página.

Dentro de las plantillas se puede poner cualquier código PHP válido, como las llamadas a la función `url`. La variable `$closure` debe imprimirse siempre al finalizar la página, por si se está utilizando alguna función en JavaScript que deba ejecutarse después de que la página se haya mostrado. (6)

2- node.tpl.php

El fichero node.tpl.php es la plantilla de los nodos y se interpreta una vez por cada nodo que se muestre en la página. Las variables que están inicializadas se imprimen y las que puede que no estén definidas se acompañan de una sentencia if. Se utiliza la variable booleana \$page para verificar si el nodo se está mostrando él solo o dentro de una lista. Si se muestra solo no se imprime el título del nodo, porque ya se imprimió en la plantilla de la página. También se utiliza la variable \$node_url para imprimir la dirección donde va a estar el nodo.

Además cada nodo posee un conjunto de atributos:

\$title: Título del nodo.

\$node_url: Enlace al nodo.

\$terms: HTML para la taxonomía de los términos.

\$name: Nombre del autor con formato.

\$date: Fecha de creación con formato.

\$sticky: Esta variable devuelve verdadero si el nodo está en la cima de las listas.

\$picture: Imprime el nombre del fichero que representa la imagen del usuario.

\$content: Contenido del nodo.

\$links: Enlaces al nodo.

\$taxonomy (array): Arreglos de enlaces HTML para la taxonomía de los términos.

\$page: Esta variable devuelve verdadero si el nodo está siendo mostrado como una página.

\$submitted: Muestra autor y fecha de creación si el nodo tiene habilitada la opción de mostrar información. (11)

3- comment.tpl.php

Esta plantilla corresponde al módulo de comentarios y es la más sencilla de todas, la cual

←—————→

\$zebra: Controla si el bloque es par o impar. Esto es útil para hacer “rayas de cebra” con CSS.

\$block_zebra: Lo mismo que el \$zebra, pero este se resetea para cada región. (13)

2.3 Propuesta de solución

Como primera propuesta de solución se propone una guía que describe el montaje de un repositorio de módulos en la herramienta de desarrollo colaborativo GForge, para tener almacenados por carpetas los módulos de Drupal, de forma tal que se encuentren lo más organizados posible. Para ello se definirán reglas de acceso en dependencia de los roles. Como segunda propuesta de solución se define una guía de desarrollo adaptada al proyecto Portales, ya que para desarrollar una buena gestión de proyectos deben estar bien definidas las fases de desarrollo por las que debe pasar un proyecto, los roles por fases y los artefactos asociados a estos roles.

2.3.1 Repositorio de módulos

Un repositorio de módulos es un sitio centralizado donde se almacenan los módulos de Drupal, el cual va a tener un administrador que será el encargado de verificar la existencia de los mismos dentro del repositorio y de asignar los permisos para acceder a él.

Este repositorio se montará dentro de la herramienta GForge para que los programadores y montadores puedan descargar los módulos desde allí mediante cualquier cliente de subversión. Para ello, primeramente se creará un nuevo proyecto que se llamará “módulo” e incluirá todos los módulos nuevos, los ya desarrollados y los que están en fase de desarrollo, es decir todo el código fuente de estos. Además se subirán los ficheros para el área de descarga, de forma tal que se puedan acceder a ellos. Cada módulo debe llevar implícito un archivo, el cual debe explicar para que se utilizará y su funcionalidad.

Los módulos deberán estar separados por carpetas diferentes dentro del subversión para así diferenciar los que se utilizarán para un proyecto en particular, de los que son en común para varios proyectos. Para almacenar un módulo en el repositorio debe pasar inicialmente por una

fase de prueba que la realizará el programador y el administrador del repositorio para así corregir los errores. Luego se le asigna a dicho módulo un número de versión y se almacena en el repositorio.

2.3.1.1 Reglas para usar módulos

- 1- Para acceder al repositorio de módulos hay que ser miembro del proyecto y tener el rol de montador, programador, administrador del proyecto o administrador de módulos.
- 2- El administrador de módulos es el único encargado de asignar los permisos dentro del repositorio y de verificar la existencia de un módulo.
- 3- El montador solo podrá descargar los módulos que necesite para realizar el montaje del portal Web en el que se encuentre trabajando.
- 4- El programador es el encargado de implementar los módulos, subirlos al repositorio y modificarlos en caso de que requieran algún cambio.

2.3.2 Fases de desarrollo para Portales Web

Inicialmente en un proyecto se deben definir las fases de desarrollo para así lograr una mayor organización, por lo que debe pasar por cuatro fases: inicio, diseño e implementación, ensamblaje y transición. Estas tendrán asociado roles como son: líder del proyecto, arquitecto de la información, documentador, administrador de módulos, diseñador, programador, montador y responsable de calidad.

2.3.2.1 Fase de inicio

En la fase inicial se analizan las necesidades del cliente y a partir de ahí se trazan una serie de metas y objetivos a cumplir. De esta forma se tiene una visión general del proyecto, que permite definir el número de integrantes que conformarán el equipo de desarrollo del mismo, asignando responsabilidades por roles. Los usuarios finales, clientes y desarrolladores deben tener una idea en común para así lograr un desarrollo óptimo del producto. El levantamiento de requisitos,

Capítulo 2. Tendencias y Tecnologías Actuales Utilizadas

dentro de esta fase, juega un papel esencial, ya que muestra más detalladamente los alcances del sistema y lo que se quiere lograr. Además los errores más comunes y más costosos de reparar, así como los que más tiempo consumen, se deben a una inadecuada ingeniería de requisitos. El análisis del problema debe estar enfocado a las perspectivas del usuario y evaluado mediante una definición en detalle de los datos, funciones y comportamiento del sistema. Se debe definir qué requisitos implementar con funciones ya desarrolladas (reutilización de código) y cuáles se implementarán desde el inicio. En esta fase se firma el contrato si ambas partes están de acuerdo con lo obtenido durante el desarrollo de la misma.

Roles involucrados

El **líder del proyecto** establece un primer contacto con el cliente y a partir de ahí define roles y responsabilidades, es decir comienza a distribuir el trabajo en dependencia de la magnitud del proyecto. En un segundo encuentro con el cliente, participan el líder del proyecto y el equipo de desarrollo escogido para el mismo. El **arquitecto de la información** realiza el levantamiento de requisitos para hacer una descripción más detallada del sistema que se quiere obtener. El **administrador de módulos** es el encargado de verificar si los requisitos se pueden implementar con módulos existentes en el repositorio o deben ser implementados. El **líder del proyecto** firma el contrato con el cliente y se establece el convenio.

Artefactos por roles

Líder del proyecto

- Documento visión.

Arquitecto de la información:

- Especificación de requisitos funcionales y no funcionales.
- Descripción de casos de uso.



2.3.2.2 Fase de diseño e Implementación

En esta fase se especifica y describe la forma de implementar los módulos nuevos y los que se basan en funcionalidades ya desarrolladas. Se realiza la parte del diseño para descomponer el trabajo de implementación en partes más manejables y crear un punto de partida para la implementación. Se modela el sistema, es decir se hace el diseño de la aplicación Web para ver cómo quedará finalmente. Se realiza además el diseño y modelo de la base de datos. Al mismo tiempo se procede a la construcción del producto, la cual se divide en dos partes; la extensión, que es donde se van a implementar los nuevos módulos y los basados en funcionalidades ya desarrolladas, así como la implementación de la base de datos y la personalización, que es donde se crea la apariencia de la aplicación, es decir la plantilla.

Roles involucrados

El **arquitecto de la información** analiza la arquitectura a seguir, describiendo la distribución del sistema. El **líder del proyecto** identifica los elementos del diseño para luego diseñar la base de datos. Además especifica y describe la forma de implementar los módulos nuevos y los que se basan en funcionalidades ya desarrolladas. El **diseñador** modela el sistema para así descomponer el trabajo de implementación en partes más manejables. El **administrador de módulos** le informa al programador que implemente un módulo nuevo. El **programador** se encarga de implementar dichos módulos y los basados en funcionalidades ya desarrolladas, además de implementar la base de datos. Concluida la implementación de los módulos se pone en contacto con el **administrador de módulos** para añadirlo al repositorio. El **montador** crea la plantilla de la aplicación. El **documentador** elabora los documentos roles de usuario y diccionario de datos.

Artefactos por roles

Arquitecto de la información:

- Descripción de la arquitectura.

Capítulo 2. Tendencias y Tecnologías Actuales Utilizadas

Líder del proyecto:

- Modelo del diseño de la BD.
- Modelo de análisis.

Diseñador:

- Pautas del diseño de la aplicación.
- Pautas del diseño de la base de datos.

Programador:

- Módulos.
- Modelo de datos.

Montador:

- Plantilla.

Documentador:

- Roles de usuario.
- Diccionario de datos.

2.3.2.3 Fase de ensamblaje

Es en esta fase donde una vez implementados los módulos se lleva a cabo el montaje de los mismos y de la aplicación Web en general para luego obtener una versión funcional del sistema. Esta versión debe pasar por calidad interna, donde un grupo encargado de la calidad certifica si está lista para la transición al cliente.

Roles involucrados

El **montador** realiza el montaje del sitio y de los módulos. Para realizar el montaje de módulos tiene que solicitarlos al **administrador de módulos** y este se los facilita. El **documentador** será



como trabajar en la aplicación.

Roles involucrados

El **equipo de trabajo** es el encargado de realizar las primeras y las terceras pruebas a la aplicación, las segundas las realiza el **equipo de prueba independiente**, luego el **líder del proyecto** se reúne con el cliente y le entrega, configura e instala la aplicación , indicándole como trabajar en ella.

Conclusiones

En este capítulo se analizaron herramientas de control de proyectos y control de versiones. Se establece una comparación entre diferentes metodologías de desarrollo, que permitió fundamentar la elección de la metodología a seguir en el desarrollo de aplicaciones Web dentro del proyecto. Se desarrolla una guía para la creación de módulos y plantillas en Drupal. Finalmente se propone el montaje del repositorio de módulos de Drupal en la herramienta GForge y una guía de desarrollo adaptada al proyecto.



Conclusiones

Hoy en día resulta difícil desarrollar una buena gestión de proyectos, ya que las aplicaciones de software requieren de mayores prestaciones y equipos de desarrollo de gran envergadura. Por lo tanto se debe conformar un detallado plan de trabajo, de forma tal que se realice una correcta distribución del mismo y se siga un estándar de desarrollo, logrando así una mayor organización que permita desarrollar una aplicación bien estructurada, configurable y extensible, en el plazo de tiempo establecido.

En el desarrollo de esta investigación se ha dado una descripción de la herramienta Drupal, como sistema de administración de contenido capaz de construir un sitio Web competente, describiendo sus características con el fin de demostrar su gran flexibilidad y dinamismo en el desarrollo de portales Web.

Para dar cumplimiento a los objetivos propuestos se crea una guía de desarrollo de aplicaciones Web, que permite ganar en organización y calidad de trabajo. Se realiza además un estudio de la herramienta GForge con el fin de integrarla al proyecto Portales, permitiendo desarrollar una guía para el montaje de un repositorio de módulos de Drupal.



Recomendaciones

Se recomienda:

- La presente investigación a todos los proyectos vinculados al desarrollo de aplicaciones Web para que se siga el mismo estándar de desarrollo propuesto y así lograr una mayor eficiencia y organización.
- El uso de la herramienta de desarrollo colaborativo GForge con el fin de lograr una buena gestión y control de los proyectos.
- Hacer un estudio más profundo de la gestión de proyectos, de forma tal que se desarrolle con la mayor eficacia posible.

Referencias Bibliográficas

1. SOLANO, W. Interfaces Web 2007, Disponible en: <http://www.monografias.com/trabajos24/bases-de-datos/bases-dedatos.shtml>
2. ---.Tipos y categorías de una pagina Web. 2007, Disponible en: http://www.tuimagen.net/disenodepaginasWeb/Modulo5_TiposyCategorias.asp
3. ANÓNIMO. Diseño de páginas Web. 2007, Disponible en: http://www.tuimagen.net/html/Servicios_DisenoPaginas_Portales.asp
4. ---. Gestión de Proyectos Europeos de investigación y desarrollo. 2006 Disponible en: <http://www.madrimasd.org/revista/revista25/aula/aula2.pdf>
5. DOLADO, F. J. R.-B. Y. J. Gestión Activa de Eventos en Proyectos Software 2004, Disponible en: <http://www.sc.ehu.es/jiwdocoj/remis/docs/FranAdis2004.pdf>
6. DESARROLLADORES, C. D. SobreDrupal. 2006, Disponible en: http://www.rhinocerus.net/files/About_Drupal.pdf
7. ---. System requirements. 2007, Disponible en: <http://drupal.org/requirements>
8. ---. Installing contributed modules. 2007, Disponible en: <http://drupal.org/node/70151>
9. ---. Different node templates depending on URL aliases. 2007, Disponible en: <http://drupal.org/node/117491>
10. ---. Using different page templates depending on the current path. 2007, Disponible en: <http://drupal.org/node/104316>
11. ---. node.tpl.php. 2007, Disponible en: <http://drupal.org/node/11816>
12. ---. comment.tpl.php. 2007, Disponible en: <http://drupal.org/node/11815>
13. ---. block.tpl.php. 2007, Disponible en: <http://drupal.org/node/11813>



Bibliografía

1. ALTAMIRANDA, J. Plataforma GForge. 2006, Disponible en: <http://sistemas.fsl.funmrd.gov.ve/docman/view.php/16/116/GForge.pdf>
2. ALVAREZ, M. A. Páginas estáticas Vs. Dinámicas. 2007, Disponible en: <http://www.desarrolloWeb.com/articulos/712.php#manuales>.
3. ANÓNIMO. LA PLATAFORMA ACTIVA DE MICROSOFT. Disponible en: <http://acsblog.es/svn/pfc/DCOM/DCOM.html>.
4. ---. Active X 2003, Disponible en: <http://www.mastermagazine.info/termino/3768.php>
5. ---. Control de versiones con Subversión. 2004, Disponible en: <http://svnbook.red-bean.com/nightly/es/>
6. ---. Artículos. 2005, Disponible en: <http://barrapunto.com/articles/05/11/11/0615238.shtml>
7. ---. Diseño de páginas Web estáticas y dinámicas. 2006, Disponible en: <http://www.ldrs.com.ar/?botonera=disenioWeb>
8. ---. Control de versiones Parte I. 2006, Disponible en: <http://blog.innocuo.com/archive/2006/07/15/Subversión-control-de-versiones-parte-i/>
9. ---. Sistemas para administración de contenido de páginas Web. 2006, Disponible en: <http://www.informaticamilenium.com.mx/paginas/mn/articulo98.htm>
10. ---. Aplicaciones Web. 2007, Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_Web

Bibliografía

11. ---. Gestión de Proyectos. 2007, Disponible en: http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos
12. ---. Calidad. 2007, Disponible en: <http://es.wikipedia.org/wiki/Calidad>
13. ---. OCX. 2007, Disponible en: <http://es.wikipedia.org/wiki/OCX>
14. ---. Definición de VBX. 2007, Disponible en: <http://www.alegsa.com.ar/Dic/vbx.php>
15. DANIEL, C. P. G. Z. Y. S. G. Programación basada en Componentes 2003, Disponible en: http://computacion.cs.cinvestav.mx/~sgarrido/cursos/ing_soft/Componentes/index.html
16. GARCÍA, A. F. Introducción a los JavaBeans. Disponible en: <http://www.sc.ehu.es/sbWeb/fisica/cursoJava/applets/javaBeans/fundamento.htm>
17. MENDOZA, M. A. Metodologías de Desarrollo de Software. 2004, Disponible en: http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf
18. MOLPECERES, A. Procesos de Desarrollo: RUP, XP, FDD. 2004, Disponible en: <http://tutorial.emagister.com/frame.cfm?>

Anexos

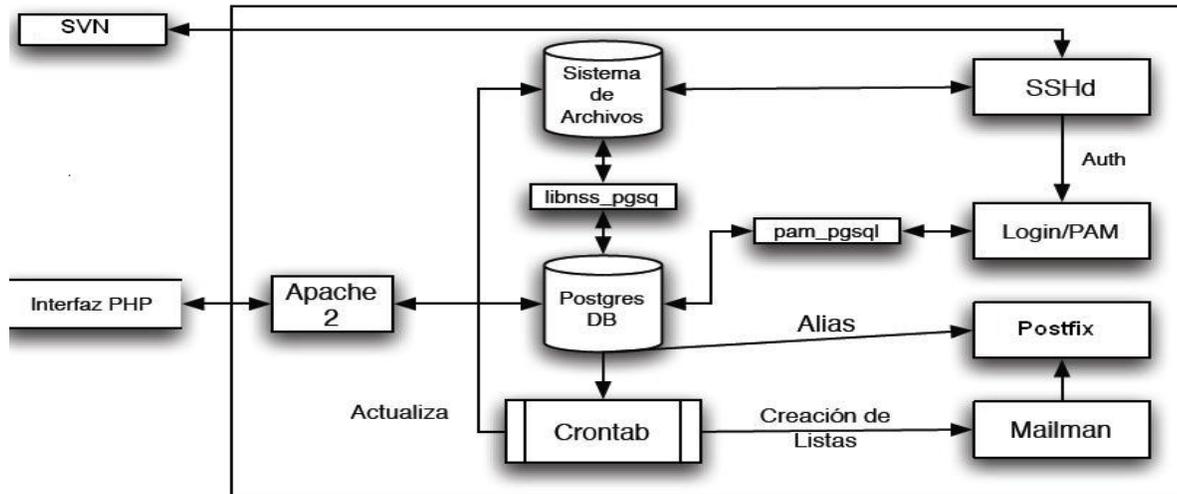


Figura 1: Esquema Arquitectura de GForge.

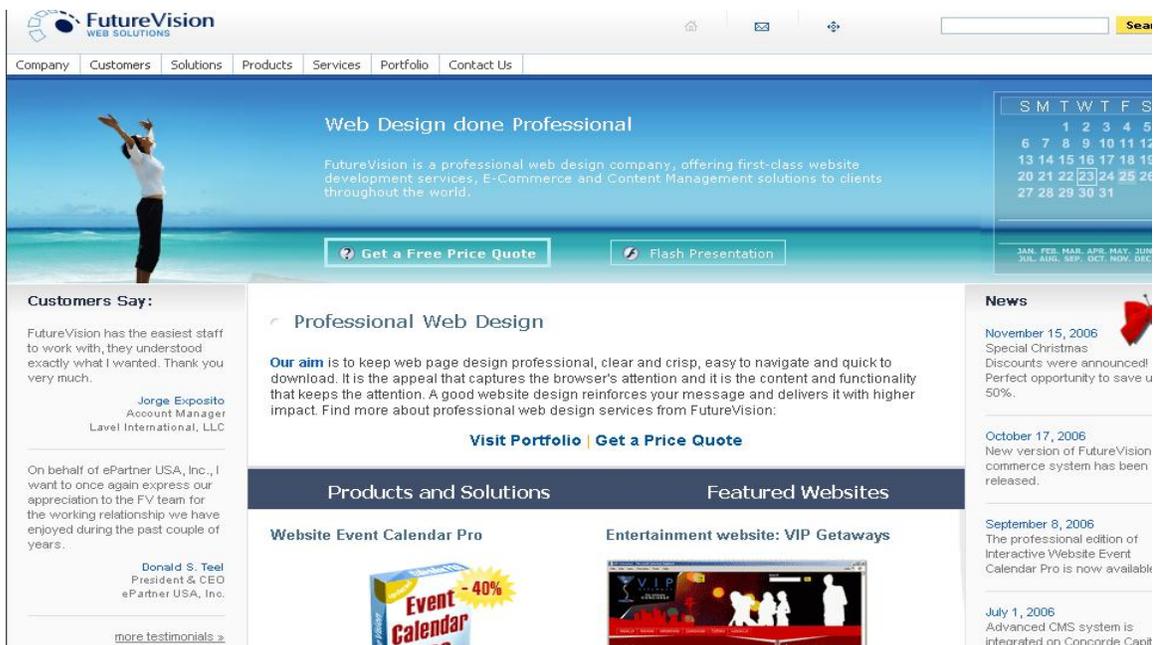


Figura 2: Páginas informativas.



Figura 3: Páginas de negocios.

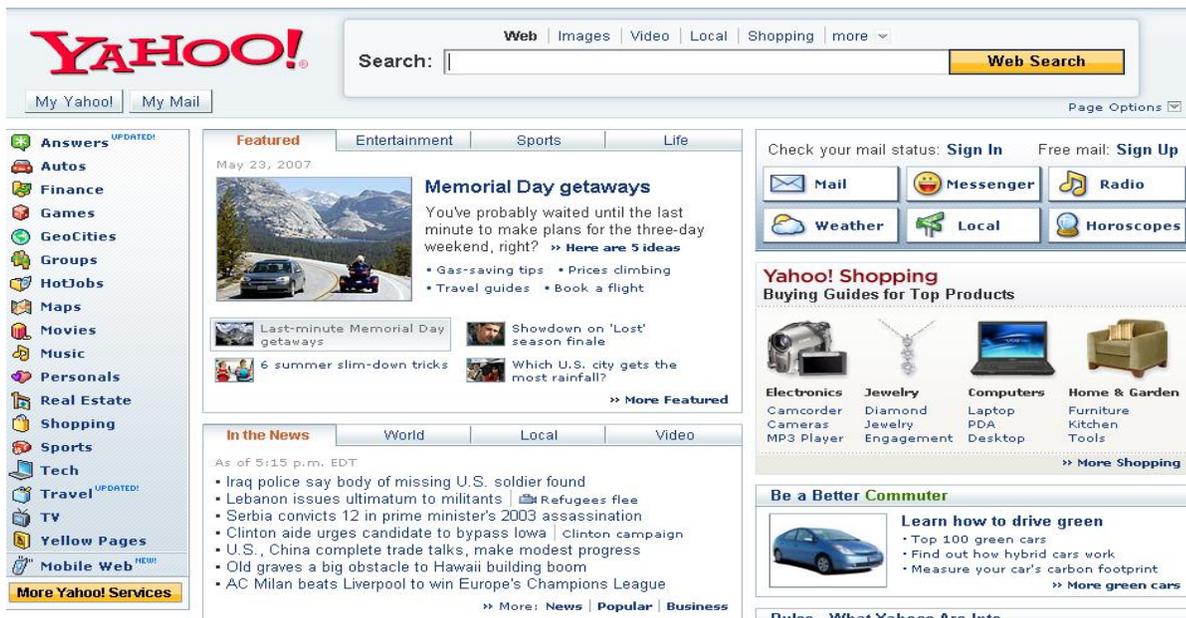


Figura 4: Portales

Glosario de Términos

API: Interfaz de Programación de Aplicaciones (Application Programming Interface). Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

CSS: Hoja de Estilo en Cascada (Cascading Style Sheets). Dentro del diseño de páginas de Internet se presenta esta como la vanguardia en cuanto a definición de estilos dentro de las plantillas de diseño.

GPL: Licencia Pública General (General Public License). Licencia creada por la Free Software Foundation, está orientada a proteger la libre distribución, modificación y uso de software.

Hook: Del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos de las interfaces que deben implementar los módulos desarrollados para este CMS.

HTTP: Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol). Protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

IDE: Ambiente de desarrollo integrado (Integrated Development Environment). Es como se le llama al ambiente que proporciona al usuario una determinada herramienta de desarrollo.

IIS: Internet Information Services (o Server), IIS, es una serie de servicios para los

ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y



HTTP/HTTPS.

Microsoft: Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.- Apache.

OLE: Object Linking and Embedding, un sistema de distribución de objetos y un protocolo desarrollado por Microsoft.

PEAR: PHP Extensión and Application Repository, es un entorno de desarrollo y sistema de distribución para componentes de código PHP

PHP: PHP Hypertext Pre-processor. Lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

RDF: Marco de Descripción de Recursos (Resource Description Framework) es un framework para metadatos en la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C).

RSS: Formato de datos que es utilizado para syndicar (redifundir) contenidos a suscriptores de un sitio Web.

XML: Lenguaje de Marcas Extensible (eXtensible Markup Language), es un metalenguaje extensible de etiquetas desarrollado por Consortium (W3C).