UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 8

Título: Repositorio de componentes para el desarrollo de software educativo y multimedia.

Trabajo de Diploma para optar por el título de Ingenieros en Ciencias Informáticas.

Autores: Teddy Ayax Bravo Saavedra
Renier Ochoa López

Tutor: Ing. Abduly Díaz García
Ing. Michel Miranda Cairo

Ciudad de la Habana, Junio 2009

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Ernesto Che Guevara

Declaración de autoría

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Teddy Ayax Bravo Saavedra

Firma del autor

Renier Ochoa López

Firma del autor

Resumen

El presente trabajo que lleva por título: **Repositorio de componentes para el desarrollo de software educativo y multimedia**, consiste en el análisis, diseño e implementación de una aplicación web que almacene y distribuya componentes de software educativo y multimedia para los proyectos de la facultad 8. Para el desarrollo del sistema, se siguieron los pasos que propone la metodología eXtreme Programming (XP), en sus distintas fases: exploración, planificación, implementación y pruebas.

En el documento se recoge el resultado del trabajo realizado para el desarrollo del sistema antes mencionado, conteniendo el estudio del estado del arte de aplicaciones con similares objetivos existentes en el mundo, así como el estudio y definición de las características del sistema.

Índice

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 3

1.1 Introducción 3
1.2 Repositorios de Información 3
1.3 Herramientas para el desarrollo de la aplicación 7
1.3.1 Servidores Web 7
1.3.1.1 Apache 8
1.3.2 Lenguajes de programación para la Web 11
1.5.1 Selección del Sistema de Gestión de Contenido a utilizar 26
1.6 Metodologías para el desarrollo de la aplicación 29
1.7 Lenguaje de modelado empleado para el desarrollo de la aplicación 36
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 38

2.1 Introducción 38
2.2 Descripción de los procesos vinculados al campo de acción 38
2.2.1 Flujo actual del proceso 38
2.2.2 Objeto de automatización 38
2.3 Propuesta del sistema 39
2.3.1 Personas relacionadas con el sistema 39
2.3.2 Requerimientos funcionales del sistema 40
2.4 Conclusiones del capítulo 43
CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN 44

3.1 Introducción 44
3.2 Fase de exploración. 44
3.2.1 Historias de Usuario 44
3.3 Planificación 47
3.3.1 Estimación de esfuerzos por Historias de Usuario 48
3.3.2 Plan de Liberaciones 48
3.3.3 Plan de Iteraciones 49
3.4 Conclusiones 50
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS 51

4.1 Introducción 51
4.2 Diseño del sistema 51
4.2.1 Módulos de Drupal 52
4.2.2 Paquete 1: Drupal 53
4.2.3 Sub paquete 1.1: Modules 54
4.3 Diseño de la base de datos 56
4.4 Fase de implementación 57
4.5 Pruebas 62
4.6 Cocluciones del capítulo 64
CONCLUSIONES GENERALES 65

Índice

RECOMENDACIONES 66

REFERENCIAS BIBLIOGRÁFICAS 67

ANEXOS 69

Introducción

La Universidad de las Ciencias Informáticas como entidad productora de software busca insertarse en el mercado nacional e internacional, para ello utiliza proyectos productivos con el objetivo de fortalecer y aumentar la calidad de los productos informáticos para la obtención de mejores resultados.

Durante este proceso productivo la Facultad 8 ha adquirido diferentes experiencias, que han sido tomadas en cuenta para agilizar la entrega del producto en tiempo y reducir el trabajo informático por parte del equipo de desarrollo de los diferentes proyectos.

Nuestra facultad enfocada hacia el perfil del desarrollo de software educativo y multimedia utiliza diferentes elementos de arquitectura e implementación de otros proyectos anteriores, pero no existe **una herramienta que almacene de forma organizada los componentes** que pudiera generar un proyecto y que luego puedan ser utilizados en los próximos proyectos.

Para darle solución a esta situación se propone desarrollar un sistema digital que ayudaría al almacenamiento organizado de los componentes generados por los proyectos productivos de la facultad, haciendo énfasis en los proyectos de software educativo y multimedia.

Haciendo un punto de escala a partir de la situación problémica se define el **problema científico: ¿Cómo facilitar acceso y distribución a componentes que agilicen el proceso de desarrollo en los proyectos de software educativo y multimedia de la Facultad 8 de de la UCI?**

A partir de este problema, se define como **objeto de estudio:** Desarrollo del análisis, diseño e implementación del Repositorio de componentes para software educativo y multimedia.

El **campo de acción** es el Proceso de recolección de la información de la actividad productiva de los proyectos de software educativo y multimedia de la Facultad 8.

La **idea a defender** sería la siguiente: si se crea en la Facultad un Repositorio de componentes de software educativo y multimedia, a partir del cual se puedan reutilizar los componentes generados por los proyectos productivos, se podrá lograr una reducción del tiempo de investigación y desarrollo de proyectos venideros.

El **objetivo general** es: Crear un Repositorio de componentes de software educativo y multimedia en la Facultad 8 que permita el desarrollo orientado a componentes, para lo cual se trazaron **objetivos específicos** de la investigación que a continuación se muestran:

- Realizar un reconocimiento global de las tendencias mundiales sobre los Repositorios de Información.

- Estudiar Repositorios existentes en la universidad.
- Proponer la estructura del Repositorio en la Facultad 8.

Capítulo 1: Fundamentación teórica

1.1 Introducción

El capítulo actual muestra una descripción general del estado del arte a nivel mundial, en la Universidad de las Ciencias Informáticas (UCI) y en la Facultad 8 con relación a los Repositorios de Información existentes. Se hace énfasis, además, la necesidad que hay en la Facultad 8 de recolectar toda la información productiva de los proyectos de software educativo y multimedia, explicando la gran importancia de la existencia de un Repositorio central para la Facultad.

1.2 Repositorios de Información

Los repositorios también conocidos como Archivos de Acceso Abierto, son archivos digitales accesibles a través de Internet, que reúnen la producción intelectual de una disciplina o de una institución. Una de las características fundamentales de los repositorios es su carácter abierto e interoperable con otros sistemas [1].

A nivel mundial existen diferentes iniciativas para compartir información, una de ellas es establecida en la declaración de Budapes Open Access Iniciative (BOAI) 2002: (libre acceso a través de Internet a los textos completos, su lectura, impresión, vaciado y distribución respetando las leyes de copyright existentes).

El repositorio institucional es un servicio para organizar, gestionar, preservar, difundir y ofrecer acceso libre a la producción científica y académica en soporte digital, generada por los miembros de la institución.

El contar con repositorios da acceso abierto, propicia la democratización en cuanto al acceso a la información como un pilar fundamental dentro de los procesos de desarrollo del país en todos los ámbitos, a saber: económico, cultural, político y social.

La lista de repositorios que se presenta a continuación se creó a partir de recomendaciones de investigadores, profesores y estudiantes, así como la búsqueda en Internet de estos enlaces. Se encuentran ordenados alfabéticamente por especialidad, permitiendo una fácil localización de los mismos.

REPOSITORIO	ESPECIALIDAD
NACA (National Advisory Committee for Aeronautics)	Aeronaútica
Organic Eprints	Agricultura
@rchivSIC	Bibliotecología y Ciencias de la Información
Caltech Library System Papers and Publications	
DLIST (Digital Library of Information Science and Technology)	
E-LIS (Eprints for Library an Information Science)	
LDL - Librarians' Digital Library	
mémSIC (Memorias en Ciencias de la Información y de la Comunicación)	
http://www.biologia-en-internet.com/	Biología, Ciencias Experimentales y de la Salud
Centro Rajiv Gandhi para Biotecnología	Biotecnología
Computer Science Teaching Center	Ciencias de la Computación
Reportes Técnicos de Ciencias de la Computación, Caltech	
CNR Bologna Research Library	Ciencias de la Información y Ciencias de la Computación
OdinPubAfrica	Ciencias marinas
REPEC	Economía
Disertaciones y Tesis Electrónicas, Caltech	Electrónica

CERN Document Server	Física
PhysNet Document Server	
ArXiv	
Academic Archive On-line	Multidisciplinario
Academic Archive On-line	
AIM25 - Archives in London	
ArchivesEIAH	
Caltech Accelerated Strategic Computing Initiative Technical Reports	
CCSU Digital Archive	
Dspace at MIT	
ERPA European Research Papers Archive	
eScholarship Repository	
GSI OAI Repository	
Hong Kong University Theses Online	
http://eprints.anu.edu.au	
http://www.osti.gov/eprints/	
Humboldt University of Berlin, GERMANY, Document Server	
MUNDUS - UK Missionary collections	
NDAD - UK National Archive of Datasets	
NTRS (NASA Technical Report Server)	
OpenVideo	
Publicaciones de Investigadores y Docentes de la Universidad de Costa Rica (PUBLIDOC-UCR)	

Servidor de publicaciones del Instituto Francés de Estudios Andinos	
Servidor de Tesis Doctorales en Xarxa	
The University of Tennessee Library, Knoxville	
Universidad Británica de Glasgow	
Universidad Británica de Nothingham	
Universitat Politècnica de Catalunya (UPC) - PFC/TFC/Tesines	
University of Tennessee Sunsite Open Archives Initiative	
University de Trento - Italy - UNITN-Eprints	
USF Electronic Thesis and Dissertation Archive	
Latindex y las Revistas Científicas de la Universidad de Costa Rica	
IMDI to OAI bridge	Psicolingüistica
Chemistry Preprint Server	Química
CNR ISOF (Istituto per la Sintesi Organica e la Fotoreattivita')	
Caltech Control and Dynamical Systems Technical Reports	Reportes técnicos

Los **factores** que provocan el crecimiento de un repositorio de información son:

• **Accesibilidad:** la disponibilidad a texto completo supone que los documentos sean mucho más accesibles, se eliminen las barreras monetarias o de tiempo y se simplifique el caos que algunas veces supone localizar documentos en la red.

• **Visibilidad:** los documentos se hacen mucho más visibles para la comunidad científica internacional al formar parte de una misma base de datos.

• **Diseminación:** los trabajos pueden estar en repositorios incluso antes de ser publicados, por lo que la difusión es inmediata, y la reacción de la comunidad científica, de existir, ocurrirá en un plazo breve de tiempo.

• **Citación:** las tres primeras ventajas conllevan a otra ventaja aún más interesante para un autor. Si un documento de investigación, además de ser interesante o novedoso, está visible en Internet, es de fácil accesibilidad y ha sido difundido con inmediatez, inevitablemente tendrá más probabilidad de ser citado que otro con las mismas características de contenido pero con poca visibilidad o acceso restringido.

1.3 Herramientas para el desarrollo de la aplicación

Dentro del mundo de la Web se encuentran diversos productos como PHP, lenguaje de programación interpretado e interactivo, capaz de ejecutarse en gran cantidad de plataformas, en el ámbito de los sistemas gestores de bases de datos se encuentra PostgreSQL que está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo.

1.3.1 Servidores Web

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting u hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la

entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

En la web, un servidor web es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo [2].

1.3.1.1 Apache

Apache es el servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.
Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.
La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente siempre que les reconozcas su trabajo.
¿Por qué es tan popular este software libre grandemente reconocido en muchos ámbitos empresariales y tecnológicos, a continuación se muestran algunas razones:

- Corre en una gran multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto.

• Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

• Apache trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script. Teniendo todo el soporte que se necesita para tener páginas dinámicas.

• Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

• Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

El resto de características importantes de Apache son:

Soporte del último protocolo HTTP 1.1

Apache es uno de los primeros servidores Web en integrar el protocolo HTTP 1.1. Es totalmente compatible con el nuevo estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0.

Apache está preparado para todas las novedades del nuevo protocolo. Por ejemplo, antes de HTTP 1. 1, un navegador Web tenía que esperar una respuesta del servidor Web antes de poder emitir otra petición.

Con el surgimiento de HTTP 1. 1, esto ha dejado de ser así. Un navegador Web puede enviar solicitudes en paralelo, las cuales ahorran ancho de banda dejando de trasmitir las cabeceras HTTP en cada solicitud. De algún modo estamos recibiendo un estímulo del lado del usuario final porque los archivos solicitados en paralelo aparecerán antes en el navegador.

Configuración basada en un poderoso archivo

El servidor Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado httpd.conf que se puede utilizar para configurar Apache. Únicamente necesita su editor de texto favorito. Sin embargo, es lo suficientemente flexible para permitirle repartir la configuración de su host virtual en múltiples archivos para no sobrecargar un

único archivo httpd.conf con toda la gestión de las múltiples configuraciones de servidores virtuales.

Soporte para CGI (Common Gateway Interface)

Apache soporta CGI utilizando los módulos mod cgi y mod cgid. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.

Soporte de FastCGI

No todo el mundo escribe sus CGI en Perl. Haciendo uso del módulo mod _f cgi se puede implementar un entorno FastCGI dentro de Apache y hacer que sus aplicaciones FastCGl arranquen rápidamente.

Soporte de host virtuales

Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.

Soporte de autentificación HTTP

Apache soporta autentificación básica basada en la Web. Está también preparado para autentificación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autentificación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autentificación.

Perl integrado

Perl se ha convertido en el estándar para la programación de scripts CGI. Apache es seguramente uno de los factores que hacen de Perl un lenguaje de programación CGI tan popular. Apache se encuentra más cerca de Perl que nunca. Puede bajar un script CGI basado en Perl a la memoria utilizando su módulo mod perl, y reutilizarlo tantas veces como necesite. Este proceso elimina las desventajas del arranque que se encuentran asociadas a menudo con los lenguajes de interpretación como Perl.

Soporte de scripts PHP

Este lenguaje de script ha comenzado a ser muy utilizado y Apache tiene un amplio soporte para PHP utilizando el módulo mod php.

Soporte de servlets de Java

Los servlets de Java y las Java Server Pages (JSP) se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache.

Servidor Proxy integrado

Apache puede convertirse en un servidor Proxy caché. Sin embargo, la implementación actual del modulo opcional de Proxy no soporta HTTP Proxy o el ultimo protocolo HTTP 1.1. Se está planeando actualizar este módulo muy pronto.

Estado del servidor y adaptación de registros

Apache le da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web. Además, puede adaptar sus archivos de registro a su gusto.

Soporte de Server Side Includes (SSI)

Apache ofrece un conjunto de Server Side Includes que añade una gran flexibilidad para el desarrollador del sitio Web.

Soporte de Secured Sockect Layer (SSL)

Puede crear fácilmente un sitio Web SSL utilizando OpenSSL y el modulo mod ssl de Apache.

1.3.2 Lenguajes de programación para la Web

Las máquinas en general, y las computadoras en particular, necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para que nosotros podamos controlar su comportamiento. Ese lenguaje que permite esta relación con las computadoras es el lenguaje de programación (Basic, Java, JavaScript, ActionScrip, Logo, etc.).

En este sentido, hay que diferenciarlo del lenguaje informático, con el que se lo suele confundir, ya que este es mucho más amplio, abarcando desde los lenguajes de programación hasta los lenguajes que dan formato a los textos, como el HTML.

Asimismo, el lenguaje de programación está conformado por una serie de reglas sintácticas y semánticas que serán utilizadas por el programador y a través de las cuales creará un programa o subprograma. Por otra parte, las instrucciones que forman dicho programa son conocidas como código fuente.

La particularidad es que ese lenguaje que utiliza le permite hacer las especificaciones en forma precisa, esto significa que todo se interpreta de la misma manera, sea quien fuere el programador que lo realice. Esto lo diferencia, por ejemplo, del lenguaje humano, en el que no siempre las especificaciones se interpretan de la misma manera.

Los lenguajes de programación pueden clasificarse de diversas manera, como por ejemplo según su nivel de abstracción: lenguaje de bajo nivel (es el código fuente de la máquina, es decir el que la máquina puede interpretar); lenguaje de nivel medio (un término entre el lenguaje de la máquina y el lenguaje natural) y lenguaje de alto nivel (los que están compuestos por elementos del lenguaje natural, es decir el humano, especialmente el inglés).

También puede haber lenguajes según la forma de ejecución, encontrándonos así con el compilador (programas que permiten traducir un programa del lenguaje natural al lenguaje de bajo nivel) y lenguajes interpretados (los que sólo hacen la traducción de los datos que se van a utilizar en ese momento y no los guarda para usar posteriormente)[3].

1.3.2.1 Python

Es un lenguaje de scripts independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, incluso, páginas Web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

El lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.

Entre las **características** del lenguaje tenemos:

Propósito general: Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la Web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

Interpretado: Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

Interactivo: Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

Orientado a Objetos: La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

1.3.2.2 Perl (Practical Extraction and Report Language)

El objetivo para el cual se diseña Perl es el disponer de un lenguaje de propósito general que permita simplificar la mayoría de las tareas de administración de un sistema tipo UNIX.

Perl es multiplataforma, podemos encontrar versiones de Perl disponibles para un amplio número de sistemas operativos como Linux, UNIX, MVS, VMS, MS/DOS, Macintosh, OS/2, Amiga.

Perl provee herramientas muy sencillas de uso para la generación de páginas dinámicas vía procesamiento de información de entrada/salida a través del Web. Este hecho masificó el uso de Perl en la creación de scripts para la Web así como un sinnúmero de otras aplicaciones. La utilidad de Perl en la Web ha ido aumentando notoriamente con el tiempo. Su afinidad con las aplicaciones necesitadas en Internet ha hecho sumar esfuerzos en la creación de distintos módulos y bibliotecas orientadas a distintos temas Web.

Algunas de las **ventajas** del uso del lenguaje Perl son las siguientes:

- Construcción de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas.
- Se puede utilizar en varios entornos, como puede ser Windows 95, OS/2, etc. sin realizar cambios de código, siendo únicamente necesario la introducción del intérprete Perl correspondiente a cada sistema operativo.
- También es uno de los lenguajes más utilizados en la programación de CGI scripts, que son guiones o scripts que utilizan el interfase CGI (Common Gateway Interface), para intercambio de información entre aplicaciones externas y servicios de información.
- El mantenimiento y depuración de un programa en Perl es mucho más sencillo que la de cualquier programa en C.
- El desarrollo de aplicaciones es muy rápido.
- Perl es gratuito. Mucho más que eso, es software libre. Esto quiere decir que el código fuente está disponible para que cualquiera lo pueda ver o modificar.

1.3.2.3 Selección del lenguaje de programación a utilizar

PHP es el acrónimo de Hypertext Preprocessor. Es un lenguaje de código abierto sumamente popular especialmente utilizado para desarrollar aplicaciones que se ejecutan en servidores Web y puede ser integrado en HTML. [4].

Ventajas

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Desventajas

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño **Modelo Vista Controlador** (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Los principales **usos** de PHP son los siguientes:

• Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
• Programación en consola, al estilo de Perl o Shell scripting.

• Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

Siempre que se habla de PHP lo primero que se hace es presentar el gran número de gestores de bases de datos a los que puede acceder:

- Adabas D
- InterBase
- LDAP
- Microsoft SQL Server
- mSQL
- MySQL
- ODBC
- Oracle
- PostgreSQL
- Solid
- Sybase

1.4 Sistemas Gestores de Bases de Datos (SGBD)

Los **sistemas de gestión de base de datos**; son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Existen distintos **objetivos** que deben cumplir los SGBD:

- **Abstracción de la información**. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.
- **Independencia**. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia**. En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por

otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

- **Seguridad**. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad**. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. Los SGBD proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente (ver Consistencia, más arriba) conocido en forma automática.
- **Respaldo**. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia**. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- **Manejo de Transacciones**. Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que el estado luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- **Tiempo de respuesta**. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Ventajas

- Proveen facilidades para la manipulación de grandes volúmenes de datos. Entre éstas:
    - Simplifican la programación de chequeos de consistencia.
    - Manejando las políticas de respaldo adecuadas garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores en el disco, o hay muchos usuarios accediendo simultáneamente a los mismos datos, o se ejecutaron programas que no terminaron su trabajo correctamente, etc.
    - Permiten realizar modificaciones en la organización de los datos con un impacto mínimo en el código de los programas.
    - Permiten implementar un manejo centralizado de la seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- Las facilidades anteriores bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Desventajas

1. Típicamente, es necesario disponer de una o más personas que administren de la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren de los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.
2. Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una planilla de cálculo.
3. Complejidad: los SGBD son software muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
4. Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder correr.

5. Coste del hardware adicional: los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero[5].

1.4.1 MySQL

MySQL es el sistema de gestión de bases de datos SQL Open Source más popular.

MySQL es un servidor multi-hilos de bases de datos de código abierto, confiable, rápido, compacto, poderoso y multiplataforma podemos hacer las bases de datos a código abierto.

Esta base de datos, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio, una gran ventaja es que se puede utilizar gratis y su código fuente siempre está disponible.

Además de que MySQL es la base de datos de código fuente abierto más usada del mundo, su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo.

Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Seguridad

• Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

Escalabilidad y límites

• Soporte a grandes bases de datos. Usamos MySQL Server con bases de datos que contienen 50 millones de registros. También se conocen usuarios que usan MySQL Server con 60.000 tablas y acerca de 5.000.000 de registros.

• Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).Un índice puede usar prefijos de una columna para los tipos de columna char, varchar, blob, o text.

Conectividad

• Los clientes pueden conectarse con el servidor MySQL usando sockets TCP/IP en cualquier plataforma.

1.4.2 Selección de SGBD a utilizar

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofrece control de concurrencia multi-versión, soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (que incluye C, C++, Java, perl, tcl y python). PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características:

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.

Altamente-Extensible

PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Soporte-SQL-Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Cliente/Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse a PostgreSQL.

Ahora se establecerá una comparación entre los gestores PostgreSQL y MySQL.

MySQL

• Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

• Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria.

Ventajas:

• Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects.

• Mejores utilidades de administración (backup, recuperación de errores, etc.)

• Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

• El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

• No hay límites en el tamaño de los registros.

• Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con qué permisos.

Inconvenientes:

• No soporta transacciones, roll-backs ni subselects.

• No considera las claves ajenas. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.

• No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

PostgreSQL

PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Ventajas:

• Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

• Implementa el uso de roll-backs, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.

• Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

• Tiene mejor soporte para triggers y procedimientos en el servidor.

• Soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

Inconvenientes:

- Consume gran cantidad recursos y carga más el sistema.

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.

- Es de 2 a 3 veces más lento que MySQL.

- Menos funciones en PHP.

Como conclusión a la comparación entre MySQL y PostgreSQL, parece aceptado que MySQL junto con Apache y PHP forman un buen equipo para servir páginas Web con contenido dinámico, discusiones, noticias, etc. En general, sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante puede bastar con hacer backups periódicos que se restaurarán tras una caída del servidor. En cambio, para sistemas más serios en los que la consistencia de la base de datos sea fundamental, PostgreSQL es una mejor opción pese a su mayor lentitud.

Además ninguno de estos dos gestores es totalmente perfecto, por lo que no hay que obcecarse en una elección única, como se suele hacer en muchos casos. Simplemente se trata de escoger el más conveniente en cada caso.

1.5 Sistema de Gestión de Contenido

Un Sistema de Gestión de Contenidos (CMS) es una aplicación que sirve para **simplificar** el proceso de crear, publicar y gestionar los contenidos de un sitio web.

Los Sistemas de Gestión de Contenidos permiten renovar los contenidos de una web de una forma muy fácil a usuarios que desconozcan el lenguaje XHTML. También hacen posible de forma extremadamente sencilla la carga y descarga de archivos de texto o imagen. Estos cambios en los contenidos se realizan accediendo a una página en la que tras introducir un nombre de usuario y una contraseña se encontrará con un **menú de opciones muy sencillo** e intuitivo (Insertar Modificar, Borrar...) a través del cual podrá cambiar los contenidos de su web en cuestión de segundos.

Características de los Sistemas Gestión de Contenidos

Los Sistemas Gestión de Contenidos ofrecen la posibilidad de realizar las siguientes funciones de forma extremadamente fácil:

- **Añadir** nuevos contenidos.
- **Editar** contenidos.
- **Borrar** contenidos antiguos.
- Mantener una sección de **Noticias**.
- **Subir** imágenes al servidor.
- Cargar archivos en el servidor (**PDFs, DOCs**...) y ponerlos a disposición de los usuarios para su descarga.
- Insertar **enlaces** hacia otras páginas.
- El sistema puede ser mantenido por **varias personas** que pueden acceder de forma independiente al Administrador de Contenidos.

¿Qué son exactamente los contenidos?

Al hablar de contenidos nos referimos a la **información** de cualquier tipo publicada en una web. Los contenidos atraen visitantes hacia una web. Está más que comprobado que la inserción y/o renovación periódica de nuevos contenidos provocan un aumento de las visitas a una web. Las webs de mayor éxito son las que ofrecen contenidos interesantes y atractivos que se renuevan con frecuencia. Los visitantes de dichas webs vuelven con frecuencia a visitarlas porque tienen la **expectativa** de encontrar nuevos contenidos de interés en cada nueva visita. Además los buscadores como Google otorgan más "puntuación" a las webs cuyos contenidos se renuevan y sitúan estás páginas en mejor posición en los resultados de búsqueda.

Por el contrario, una página web **estática**, que mantiene siempre los mismos contenidos y nunca los modifica, atrae a pocos visitantes y su curva estadística de visitas irá con el tiempo en declive.

Si usted quiere que **aumenten las visitas** a su sitio web, empiece por dedicar unas horas a la semana o al mes a renovar los contenidos de su página e inserte contenidos que crea que puedan ser interesantes para otros internautas. La **especialización** es la mejor arma: tanto si se trata de una web de empresa como la de un colectivo o asociación, siempre habrá un campo en el que

alguno de los miembros de la empresa o del colectivo tengan un **conocimiento especializado** que será interesante para muchos internautas.

Si dispone de un Sistema de Gestión de Contenidos, la tarea de renovar los contenidos de su web le resultará grata y le "enganchará" con el tiempo, ya que comprobará que las visitas a su web aumentan y que se ve reconocido por la comunidad de internautas. Y ello sin tener que aprender ningún tipo de lenguaje informático.

Tipos de CMS

Hay diversos tipos de Sistemas de Gestión de Contenidos. Unos son programados a medida y otros son aplicaciones "prefabricadas". Un ejemplo de los CMS "prefabricados" son la gran mayoría de los **blogs**, cuyo número está aumentando en los últimos meses a velocidad de vértigo en Internet. Cada sistema tiene sus **ventajas e inconvenientes**. El desarrollo de aplicaciones a medida implica lógicamente más tiempo de trabajo y mayor presupuesto que la instalación de sistemas "prefabricados" ya desarrollados por otros programadores.

Sin embargo, un Sistema de Gestión de Contenidos desarrollado expresamente para una web podrá ser modificado fácilmente en el futuro (por ejemplo, instalando nuevas secciones actualizables dentro de una web, modificando la disposición estética y estructural de los contenidos actualizables) mientras que los CMS "prefabricados" son mucho más rígidos y permiten pocas variaciones en su estructura [6].

1.5.1 Selección del Sistema de Gestión de Contenido a utilizar

Drupal es un sistema de gestión de contenido modular y muy configurable.

Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web.

Características generales

Ayuda on-line Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.

Búsqueda Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

Código abierto El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

Módulos La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.

Personalización Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.

URLs amigables Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

Gestión de usuarios

Autenticación de usuarios Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.

Permisos basados en roles Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles [Anexo I].

Gestión de contenido

Control de versiones El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior [Anexo II].

Enlaces permanentes (Permalinks) Todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.

Objetos de Contenido (Nodos) El contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto.

Plantillas (Templates) El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

Sindicación del contenido Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un 'Agregador de Noticias', tal como *NetNewsWire* o *Radio UserLand* visualice el contenido publicado en la web desde el escritorio.

Blogging

Agregador de noticias Drupal incluye un potente Agregador de Noticas para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

Soporte de Blogger API La API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que pueden ser 'herramientas web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.

Plataforma

Independencia de la base de datos Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgresSQL, aunque permite incorporar fácilmente el soporte para otras bases de datos.

Multiplataforma Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Multiples idiomas y Localización Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como *GNU ettext*

Administración y Análisis

Administración vía Web La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional [Anexo III].

Análisis, Seguimiento y Estadísticas Drupal puede mostrar en las páginas web de administración informes sobre *referrals* (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan por el sitio [7].

1.6 Metodologías para el desarrollo de la aplicación

Uno de los temas más comunes en el mundo de la informática hoy en día es el de las metodologías de desarrollo de software: cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Una metodología tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso; producir lo esperado en el tiempo esperado y con el coste esperado.

1.6.1 Programación eXtreme (XP)

XP (eXtreme Programing) nace como nueva disciplina de desarrollo de software hace aproximadamente unos seis años, y ha causado un gran revuelo entre el colectivo de programadores del mundo. Kent Beck, su autor, es un programador que ha trabajado en múltiples empresas y que actualmente lo hace como programador en la conocida empresa automovilística DaimlerChrysler. Con sus teorías ha conseguido el respaldo de gran parte de la industria del software y el rechazo de otra parte.

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. (KIOSKEA 2008)

Que es XP

- Metodología para un ágil desarrollo de software.
- Programación basada en los deseos del cliente.
- El equipo lo conforman los jefes de proyecto, desarrolladores y el cliente.
- Se rige por valores y principios.

Ventajas y desventajas de Extreme Programing

Ventajas:

- Programación organizada.
- Menor taza de errores.
- Satisfacción del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar. (VALVERDE. D. M 2007)

Objetivos de la programación extrema

El objetivo primordial de la XP es el bienestar del cliente. Se le trata de dar al cliente lo que quiere y cuando quiere. Por tanto, se debe responder rápidamente a las necesidades del cliente, aunque realice cambios en fases avanzadas del proyecto. Como metodología ágil que es, se pueden originar modificaciones de los requisitos del proyecto a lo largo de su desarrollo, sin que esto cause grandes tropiezos.

Otro de los objetivos es el trabajo en grupo. Tanto los jefes del proyecto, clientes y desarrolladores forman parte del equipo y deben estar envueltos en el desarrollo.

Prácticas de la programación extrema

12 son las prácticas de la XP:

- El juego de la planificación.
- Pequeñas entregas.
- Metáfora**.**
- Diseño sencillo.
- Pruebas.
- Refactorización.
- Programación por parejas.
- Propiedad colectiva.
- Integración continua**.**
- 40 horas semanales**.**
- Cliente en casa.
- Estándares de codificación.

1.6.2 Proceso Unificado de Rational (RUP)

El **Proceso Unificado de Rational** (**RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el ***Rational Unified Process***, que se vendiera como producto independiente.

El RUP está basado en **5 principios** clave que son:

Adaptar el proceso

El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

Balancear prioridades

Los requerimientos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. *Debe encontrarse un balance que satisfaga los deseos de todos*. Debido a este balanceo se podrán corregir desacuerdos que surjan en el futuro.

Demostrar valor iterativamente

Los proyectos se entregan, aunque sea de un modo interno, en **etapas iteradas**. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requerimientos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones

arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en **todos** los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Ciclo de vida

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones [13].

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se

realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso)....

Fases

- Establece oportunidad y alcance

- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

Proceso: Las etapas de esta sección son: (Revise nuevamente la gráfica)

- Modelado de negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

Soporte: En esta parte nos conseguimos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que este sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio(También llamado Incepción)
- Elaboración
- Desarrollo(También llamado Implementación, Construcción)
- Cierre (También llamado Transición)

Artefactos

RUP en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema (entre otros). Estos artefactos (entre otros) son los siguientes:

Inicio:

- Documento Visión
- Especificación de Requerimientos

Elaboración:

- Diagramas de caso de uso

Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

Vista Lógica:

- Diagrama de clases
- Modelo E-R (Si el sistema así lo requiere)

Vista de Implementación:

- Diagrama de Secuencia
- Diagrama de estados
- Diagrama de Colaboración

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware.

1.7 Lenguaje de modelado empleado para el desarrollo de la aplicación

Como lenguaje de modelado se utilizará UML que es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

La decisión de utilizar UML como lenguaje de modelado para la aplicación se debe a que ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento.

Características de UML:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con los tamaños propios de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

1.8 Conclusiones del capítulo

En el capítulo se realizó un estudio de algunos de los repositorios de información existentes. De esta manera, quedan expuestas las características principales de las herramientas propuestas para el desarrollo de la aplicación, así como las ventajas y los inconvenientes de las mismas. De este mismo modo, las características de XP demuestran las ventajas de elegir esta metodología. Y llevar a cabo el desarrollo teórico de la aplicación.

Capítulo 2: Características del Sistema

2.1 Introducción

El capítulo actual tiene como objetivo hacer una apreciación de las características principales del sistema a desarrollar, para lo cual se tendrá en cuenta la situación problémica que dio origen al mismo. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que serán objeto de automatización. Finalmente se presentará una propuesta del software a implementar, especificando detalladamente los requerimientos funcionales y no funcionales.

2.2 Descripción de los procesos vinculados al campo de acción

Actualmente los proyectos enfocados a la multimedia desarrollados en la Facultad 8 utilizan elementos de arquitectura e implementación de proyectos anteriores. Dichos elementos o componentes no están salvados de manera centralizada lo que se traduce en la demora de proyectos futuros.

2.2.1 Flujo actual del proceso

El desarrollo de este proceso se realiza de la siguiente manera: Cuando un proyecto comienza tiene que hacer su arquitectura o implementar desde cero, en algunos casos buscan información en distintos servidores.

2.2.2 Objeto de automatización

Durante el período de desarrollo de esta actividad existen varios procesos que deben ser automatizados, puesto que consume una valiosa porción de tiempo a los desarrolladores. Muchos de estos procesos son de vital importancia para lograr un nivel de calidad óptimo en los productos finales.

Se automatizarán los procesos de búsqueda, subida y eliminación de componentes de software educativo y multimedia mediante una aplicación Web, la cual mostrará los bloques de contenidos que serán ofrecidos a todas las personas interesadas en el desarrollo de este tipo de aplicaciones, proporcionando en gran medida ahorro de tiempo, pues se brindará la posibilidad de escoger a través de la tipología del componente que le son necesarios o que desea.

2.3 Propuesta del sistema

El presente trabajo propone la implementación de un sistema que provea las mismas funcionalidades que el flujo de trabajo actual. Dicho sistema debe permitir la gestión (subir, buscar y eliminar) de componentes dentro del sistema, recogiendo los datos que son de interés para el usuario. El administrador del sistema será el encargado de gestionar toda la información referente al sistema y dará soporte en cada una de las instancias necesitadas.

El sistema será una aplicación Web administrada con el CMS Drupal, y cada una de sus funcionalidades será implementada como parte de un módulo Drupal de propósito específico para la posterior integración de todos en lo que se denomina “Repositorio de componentes de software educativo y multimedia”.

2.3.1 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellas que se encuentran implicadas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 2.1 Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Invitado	Es la persona que navega por el sistema sin haberse creado una cuenta aún, pudiendo navegar dentro de éste sin privilegios. Tiene la posibilidad de hacer

	búsquedas y descargas de componentes.
Jefe de proyecto	Es la persona encargada de subir y eliminar componentes. Cuenta además con los privilegios de un invitado.
Administrador	Es la persona facultada para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación. Cuenta además con los privilegios del Invitado y el Jefe de proyecto.

2.3.2 Requerimientos funcionales del sistema

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran a través de requerimientos funcionales las prestaciones que el sistema será capaz de brindar. Dentro de ellas se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema.

Teniendo en cuenta los objetivos planteados, el sistema debe ser capaz de:

R1. Autenticar

R1.1. Mostrar el formulario de autenticación de usuario.

R1.2. Pedir nombre de usuario y contraseña.

R1.3. Validar los datos introducidos por el usuario.

R1.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R1.4. Verificar qué rol cumple dentro del sistema para asignar permisos.

R1.5. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos asignados una vez autenticado correctamente.

R2. Buscar componentes

R2.1. Mostrar el formulario de búsqueda.

R2.2. Permitir al usuario teclear la palabra o frase que desee buscar dentro de la base de datos de componentes.

R2.4. Mostrar los datos relacionados con la búsqueda realizada.

R3. Subir componentes

R3.1. Mostrar el formulario para subir un componente.

R3.2. Permitir al usuario entrar los datos referentes a: tipología, lenguaje, sistema operativo, una pequeña descripción y adjuntar el componente correspondiente.

R3.3. Validar los datos introducidos por el usuario.

R3.3.a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R4. Eliminar componentes

R4.1. Mostrar el formulario de los componentes subidos al servidor.

R4.2. Permitir al usuario eliminar el componente que desea.

R5. Ver perfil de usuario

R5.1. Mostrar el formulario con datos del perfil del usuario.

R6. Promover usuario

R6.1. Mostrar el formulario con datos del perfil del usuario.

R6.2. Permitir al usuario dar o quitar permisos a otros usuarios dentro del sistema.

2.3.3 Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Diseño e implementación

Aplicación Web escrita sobre el lenguaje de programación PHP 5.2.6.

Usar como gestor de Base de Datos MySQL 5.0.51b.

Utilizar como servidor web Apache 2.2.8.

Desarrollar bajo CMS Drupal 6.10.

Utilizar como metodología de desarrollo XP.

Correr sobre sistemas operativos de la familia Windows o Linux.

Apariencia o interfaz externa

Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

No debe tener animaciones ni imágenes pesadas que obstaculicen la rapidez de las transiciones del ancho de banda.

Diseño perfectamente encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.

El diseño garantizará la selección de un esquema de colores a la vez atractivo pero que no canse, conjugando equilibrio entre los mismos y un contraste suficiente pero no agresivo.

El diseño debe permitir el uso de los colores corporativos.

Usabilidad

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Funcionalidad

Capacidad de búsqueda con un tiempo menor de 5 segundos.

Mínima cantidad de páginas para ejecutar todas las funciones posibles, es decir, agrupar funciones afines en las mismas páginas.

Rendimiento

La eficiencia de esta aplicación debe ser óptima en cuanto a la velocidad de procesamiento, disponibilidad, tiempo de respuesta y aprovechamiento de los recursos, entre otros.

Seguridad

Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.

Proteger la información manejada por el sistema de accesos no autorizados.

Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

Proteger la información manejada por el sistema contra la corrupción de ficheros o estados inconsistentes.

Verificación sobre acciones irreversibles (eliminaciones).

Ayuda en línea.

El usuario podrá ver una ayuda que será publicada en el sitio que le servirá para ubicarse en lo que puede hacer dentro de la aplicación.

2.4 Conclusiones del capítulo

Se inicia el desarrollo de la propuesta de solución que se desea implementar, tras el análisis de los flujos de trabajos actuales descritos por el cliente. Se obtuvo un listado de funcionalidades que debe tener el sistema, expresados en los requerimientos funcionales. Partiendo de este punto, base de todo proceso de desarrollo, se puede comenzar con la construcción de la propuesta, velando por el cumplimiento de todos los requerimientos y funcionalidades consideradas.

Capítulo 3: Exploración y Planificación

3.1 Introducción

El fin de este trabajo es llevar a cabo la implementación de la aplicación web repositorio de componentes de multimedia. Para lograr dicha fase (implementación) con éxito es necesario realizar previamente las fases de exploración y planificación que propone la metodología de desarrollo utilizada. También se mostrarán los artefactos generados en estas fases.

3.2 Fase de exploración.

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. El equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Además se produce el contacto necesario con las herramientas y tecnologías que se emplearán para construir el sistema y algunas ideas experimentales en cuanto a su arquitectura son consideradas. Se pueden explorar soluciones puntuales también cuando no se tenga una concepción transparente de alguna funcionalidad.

3.2.1 Historias de Usuario

Una de las mejores prácticas adoptadas en el desarrollo de software es la administración de requerimientos. XP propone en este sentido hacer uso de las Historias de Usuario como técnica para especificar las funcionalidades que brindará el sistema y constituye una manera muy dinámica de realizar esta actividad. Como su nombre lo indica son especificadas por los propios usuarios y por tanto redactadas en su lenguaje; de manera sencilla y breve, evitando tecnicismos innecesarios que puedan crear confusión, aunque los programadores pueden contribuir en la tarea. Además son la base para realizar las pruebas de aceptación, así como la estimación y planificación del proyecto.

Una vez identificadas las historias de usuario necesarias para liberar una primera versión operativa los programadores proceden a descomponer cada una en tareas específicas, las denominadas tareas de programación que están escritas técnicamente, que darán solución a la historia correspondiente.

En este proceso se definieron las historias de usuario que serán presentadas a continuación:

Tabla 3.2.1.1: Autenticar usuario.

Historia de Usuario	
Número: 1	**Usuario:** Usuario
Nombre historia: Autenticar usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 1	**Iteración asignada: 1**
Programador responsable: Teddy Bravo Saavedra	
Descripción: Se brinda la posibilidad de que la persona que acceda al sistema introduzca sus datos (usuario y contraseña del dominio UCI) con la finalidad de verificar y otorgarle los permisos según el rol que cumpla dentro de la aplicación.	

Tabla 3.2.1.2: Administración.

Historia de Usuario	
Número: 2	**Usuario:** Usuario
Nombre historia: Administración	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 1	**Iteración asignada: 1**
Programador responsable: Teddy Bravo Saavedra	
Descripción: Se le brinda al administrador del sistema de modificar o eliminar una cuenta de usuario de la aplicación, de esta forma se tendrá un mayor control en	

cuanto a las personas que acceden a la aplicación y poder asignarle los permisos que le corresponden.

Tabla 3.2.1.3: Insertar componentes.

Historia de Usuario	
Número: 3	**Usuario:** Administrador
Nombre historia: Insertar componentes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Renier Ochoa López	
Descripción: Se brinda la posibilidad a usuarios con permisos de administración sobre el sistema de insertar o subir componentes.	

Tabla 3.2.1.4: Buscar componentes.

Historia de Usuario	
Número: 4	**Usuario:** Usuario
Nombre historia: Buscar componentes	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Media
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Teddy Bravo Saavedra - Renier Ochoa López	
Descripción: Una vez dentro del sistema el usuario debe seleccionar la opción buscar, revisa si tiene alguna respuesta, y en caso de existir alguna respuesta, puede acceder a descargarla.	
Observaciones: Existe un administrador en cada proyecto.	

Tabla 3.2.1.5: Eliminar componentes.

Historia de Usuario	
Número: 5	**Usuario:** Administrador
Nombre historia: Eliminar componentes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Teddy Bravo Saavedra - Renier Ochoa López	
Descripción: Se brinda la posibilidad de realizar búsquedas a usuarios con los permisos requeridos y luego elimine el o los componentes deseados.	

3.3 Planificación

A lo largo de esta fase los clientes establecen la prioridad de las historias de usuario de acuerdo a sus necesidades más inmediatas para luego asignarlas, por orden de relevancia, a las iteraciones planificadas. A partir de las historias se realiza la estimación del esfuerzo que se requiere por parte del equipo para su posterior implementación.

El método escogido para realizar la estimación tiene como elemento el punto, el cual equivale a una semana perfecta de trabajo, esto se refiere a que solamente el equipo se dedica a labores relacionadas con la construcción del sistema sin la influencia o el retraso provocado por otros factores lo que en la práctica es complejo lograr.

3.3.1 Estimación de esfuerzos por Historias de Usuario

Las estimaciones del esfuerzo para implementar las historias de usuario permiten tener una medida real de la velocidad de progreso del proyecto y brindan una guía razonable a la cual ajustarse. Los resultados estimados se muestran seguidamente:

Puntos de Estimación por historias de usuario

No.	Historia de Usuario	Puntos de Estimación
1	Autenticar usuario	1
2	Administración	1
3	Insertar componentes	2
4	Buscar componentes	2
5	Eliminar componentes	2

3.3.2 Plan de Liberaciones

El plan de liberaciones (entregas) tiene como objetivo definir el número de *releases* que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una. El cliente se encarga de decidir cuáles historias de usuario comprende la primera entrega según sus prioridades para darle valor a su negocio y que por tanto justifique su ejecución y así sucesivamente para las demás.

Para realizar el plan de entregas, el sistema propuesto se dividió en módulos que contienen las historias de usuario relacionadas lógicamente de acuerdo a su propósito. Cada una de las entregas se correspondió con una iteración de desarrollo.

Historias de Usuario por Módulos

Módulos	Historias de Usuario
Gestión de componentes	Insertar componentes
	Buscar componentes
	Eliminar componentes

Gestión de Usuario	Autenticar usuario
	Administración

El plan de liberaciones previsto finalmente quedó estructurado en 2 iteraciones a lo largo de las que se trabajó en los diferentes módulos creados y como consecuencia de esa labor se cosecharon los resultados, en forma de versiones operacionales, expuestos en la siguiente tabla:

Plan Entregas

Módulos	**Iteración 1**	**Iteración 2**
Gestión de componentes	3	
Gestión de Usuario		2

3.3.3 Plan de Iteraciones

En el plan de liberaciones se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su consecución. Sin embargo, se precisa establecer el contenido de trabajo para todas y cada una de ellas y es aquí donde hace acto de presencia el plan de iteraciones, regulando la cantidad de historias de usuario a implementar dentro del rango establecido por la estimación efectuada. Tomando como referencia los aspectos antes tratados la aplicación que se pretende construir se desarrollará en dos iteraciones, explicadas detalladamente a continuación:

Iteración 1

La iteración presente tiene como finalidad implementar las historias de usuario que se identificaron en el primer módulo: Gestión de componentes, y que a su vez se consideraron las más necesarias atendiendo a su relevancia e impacto para el sistema.

Iteración 2

La actual iteración se centra en darle solución a las historias de usuario identificadas en el segundo módulo: Gestión de usuario, que es considerada de gran importancia para el sistema.

A modo de resumen se presenta la siguiente tabla que muestra las dos iteraciones analizadas previamente con las historias de usuario que incluyen y su duración:

Plan de duración de las iteraciones

Iteraciones	Historias de Usuario a implementar	Duración
Iteración 1	Insertar componente	8 semanas
	Buscar componente	
	Descargar componente	
Iteración 2	Autenticar usuario	2 semanas
	Administración	

3.4 Conclusiones del capítulo

Se efectuó un análisis exhaustivo de los conceptos relacionados con la situación problémica planteada. Se abordaron las peculiaridades de las fases de Exploración y Planificación y los artefactos que se generaron durante su desarrollo, entre ellos las historias de usuario y los planes de entregas e iteraciones. Ambas fases se repiten en cada iteración lo que posibilita realizar una estimación más exacta y real del esfuerzo necesario para cumplir con las historias de usuario negociadas y con las que el equipo de trabajo se ha comprometido.

Capítulo 4: Implementación y Pruebas

4.1 Introducción

La Metodología XP plantea que la implementación de un producto debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. Se detallan las dos iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

4.2 Diseño del sistema

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración) [15]. No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de las comunicaciones, no constituyan un peso su mantenimiento, no sean extensos y se enfoquen en la información importante [16]. Con el objetivo de la comprensión de este trabajo se hace una breve explicación del funcionamiento de Drupal como plataforma de publicación.

El CMS Drupal contiene un tipo de contenido genérico llamado *Node* que puede ser extendido por cualquier desarrollador, este tipo de contenido tiene las propiedades básicas para cualquier publicación como son título, autor, fecha de creación y contenido, además Drupal proporciona los mecanismos para la creación, edición y publicación de este tipo de contenido. Cualquier desarrollador que desee una publicación personalizada sólo debe extender este tipo de contenido y de esta manera aprovechar sus propiedades.

Drupal provee al desarrollador de un potente sistema de seguridad basado en roles, el mismo Sistema de Gestión de Contenidos se encarga de la creación de usuarios y roles, así como del control de accesos a los diferentes módulos según los permisos definidos por el administrador. El desarrollador se limita a exportar en su módulo los tipos de acceso que desea definir, el resto lo maneja Drupal, es decir, cuando un usuario trata de acceder a un módulo la plataforma chequea que el usuario autenticado tenga acceso al módulo en cuestión.

4.2.1 Módulos de Drupal

Los módulos son extensiones para Drupal que amplían las funcionalidades del núcleo. Un módulo es la unión de varias funciones que se juntan en Drupal y ayudan a ofrecerle mayor funcionalidad a la Web. Módulo para Drupal consta de uno o más ficheros, el fichero principal con extensión .module debe implementar una interfaz definida por el propio Drupal. Básicamente existen dos tipos de módulos: los módulos de contenido, son los que definen un nuevo tipo de contenido personalizado y la funcionalidad para su creación, edición y publicación y los módulos funcionales, estos tienen disímiles propósitos dependiendo del objetivo con el que se desarrolla. La tarea de estas funciones es actuar como enganche, al ser llamadas por Drupal a la hora de construir una página Web y gestionar el contenido.
El sistema de bloques de Drupal consiste en una serie de bloques (ya sea definidos por un módulo o en la misma interfaz de Drupal) que se activan o desactivan para ser mostrados en las áreas de menú de la plataforma. A continuación se utilizará el término “paquete” para referirse a módulos del Sistema de Gestión de Contenido en cuestión.

Paquetes de Drupal:

1. Paquete 1: Drupal
   a. Sub paquete 1.1: Modules
   b. Sub paquete 1.2: Includes
   c. Sub paquete 1.3: Themes

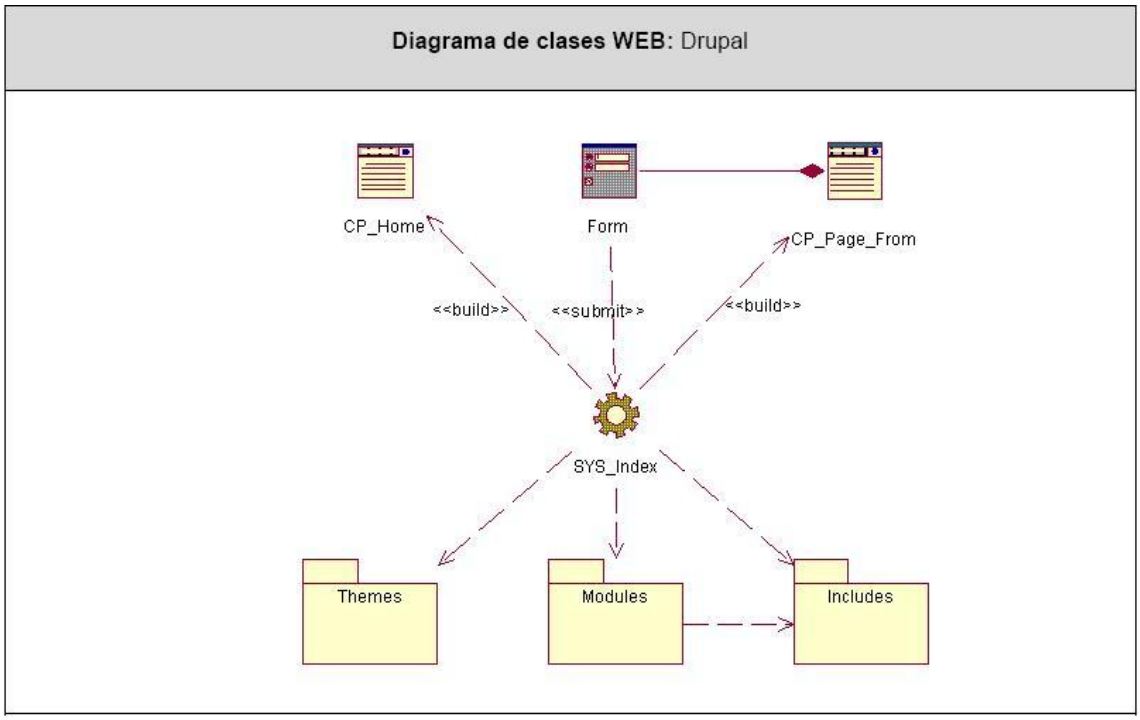
Los paquetes desarrollados son:

a. Sub paquete 1.1.1: Subir Componentes
b. Sub paquete 1.1.2: Eliminar Componentes
c. Sub paquete 1.1.3: Buscar Componentes

4.2.2 Paquete 1: Drupal

El Paquete 1: Drupal, consiste en la distribución del CMS Drupal. Básicamente contiene un paquete *Themes*, donde se encuentran los mecanismos que soportan el sistema de plantillas, de modo que cuando usted desee cambiar el diseño de la interfaz que presentará el sistema, sólo tiene que definir una nueva plantilla en este archivo; un paquete *Includes*, donde se encuentran ficheros de configuración y ficheros utilitarios, es este paquete donde se incluyen las API de acceso a datos; y por último un paquete *Modules*, que proveen a Drupal de sus funcionalidades, de forma tal que cuando desee agregar un nuevo módulo, sólo debe copiarlo dentro de esta carpeta y activarlo a través de la interfaz de Drupal. Este CMS contiene una única página de servidor, la cual basándose en el sistema de clases genera el contenido de la página final, teniendo en cuenta los argumentos con que se realiza la petición. Las páginas generadas pueden o no contener formularios, esto depende del módulo en cuestión y del propósito del mismo. Debido a la naturaleza de los Sistemas de Gestión de Contenidos y a su estructura modular en esta sección se modelarán por separado cada módulo, puesto que son independientes uno del otro en cuanto a funcionalidad. Abordados los principales aspectos que influyen en el diseño del diagrama de clases Web que representa el funcionamiento de Drupal, se está en condiciones de presentar el mismo, recordando siempre que la terminología para referirse a un módulo Drupal será la de “paquete”. El diagrama de clases se muestra en la siguiente figura:

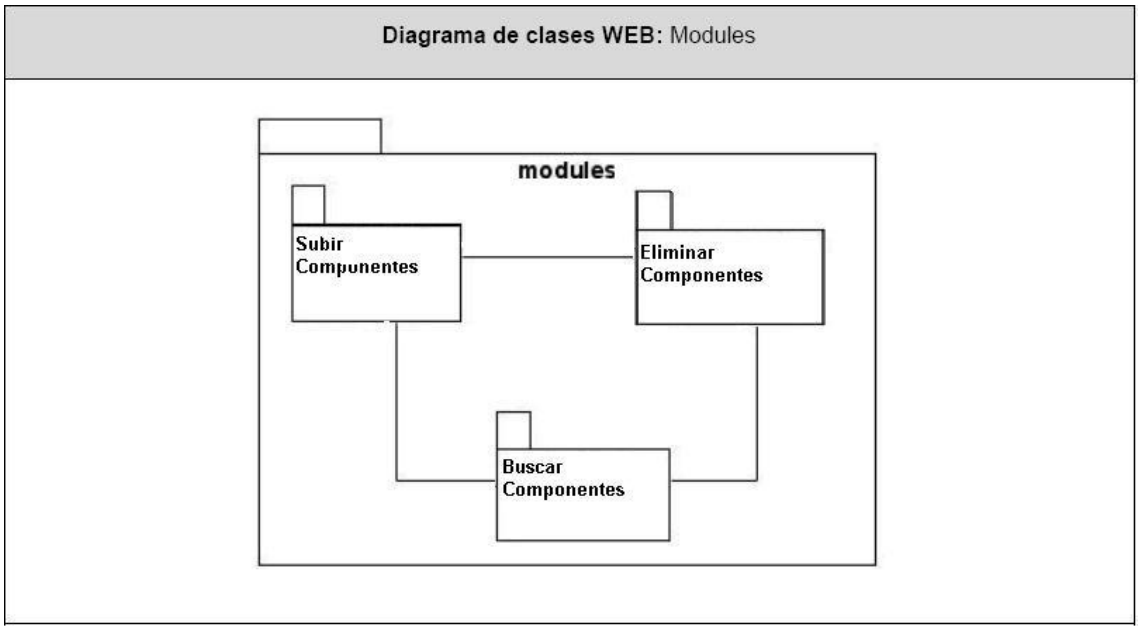
Figura 1 Diagrama de clases del diseño: Paquete Drupal



4.2.3 Sub paquete 1.1: Modules

El paquete Modules contiene los módulos que dan soporte a las funcionalidades de Drupal, y los módulos desarrollados en este trabajo. Para simplificar el diagrama no se modelan todos los módulos de Drupal, sino solamente los desarrollados.

Figura 2 Diagrama de clases del diseño: Sub paquete Modules



Con el objetivo de hacer entendible las funcionalidades encapsuladas en estos módulos, se definen una tarjeta CRC por cada uno, con la finalidad de obtener un diseño simple y no incurrir en la implementación de características que no son necesarias.

Tabla 4.1 Tarjeta CRC Módulo Gestión de Componentes

Módulo Gestión de Componentes	
Funcionalidades	**Colaboraciones (Módulos)**
Subir Componente Buscar Componente Eliminar Componente	Node Upload Menu Help Block User

Tabla 4.2 Tarjeta CRC Módulo Gestión de Usuarios

Módulo Gestión de Componentes	
Funcionalidades	**Colaboraciones (Módulos)**
Autenticar usuario Administración	Node Upload Menu Help Block User

4.3 Diseño de la base de datos

Diseñar la Base de Datos es algo que puede pasar por alto un equipo de desarrollo, producto de que uno de sus objetivos fundamentales es brindar la persistencia al modelo que se describe en el epígrafe anteriormente desarrollado.

El modelo de datos del problema en cuestión posee un nivel de complejidad bajo, producto a que las entidades son manejadas por el CMS Drupal (por lo que no han sido contempladas en el modelo de datos). A continuación se muestra el diagrama de clases persistentes y el modelo de datos que se utilizó:

Figura 3 Diagrama de clases persistentes

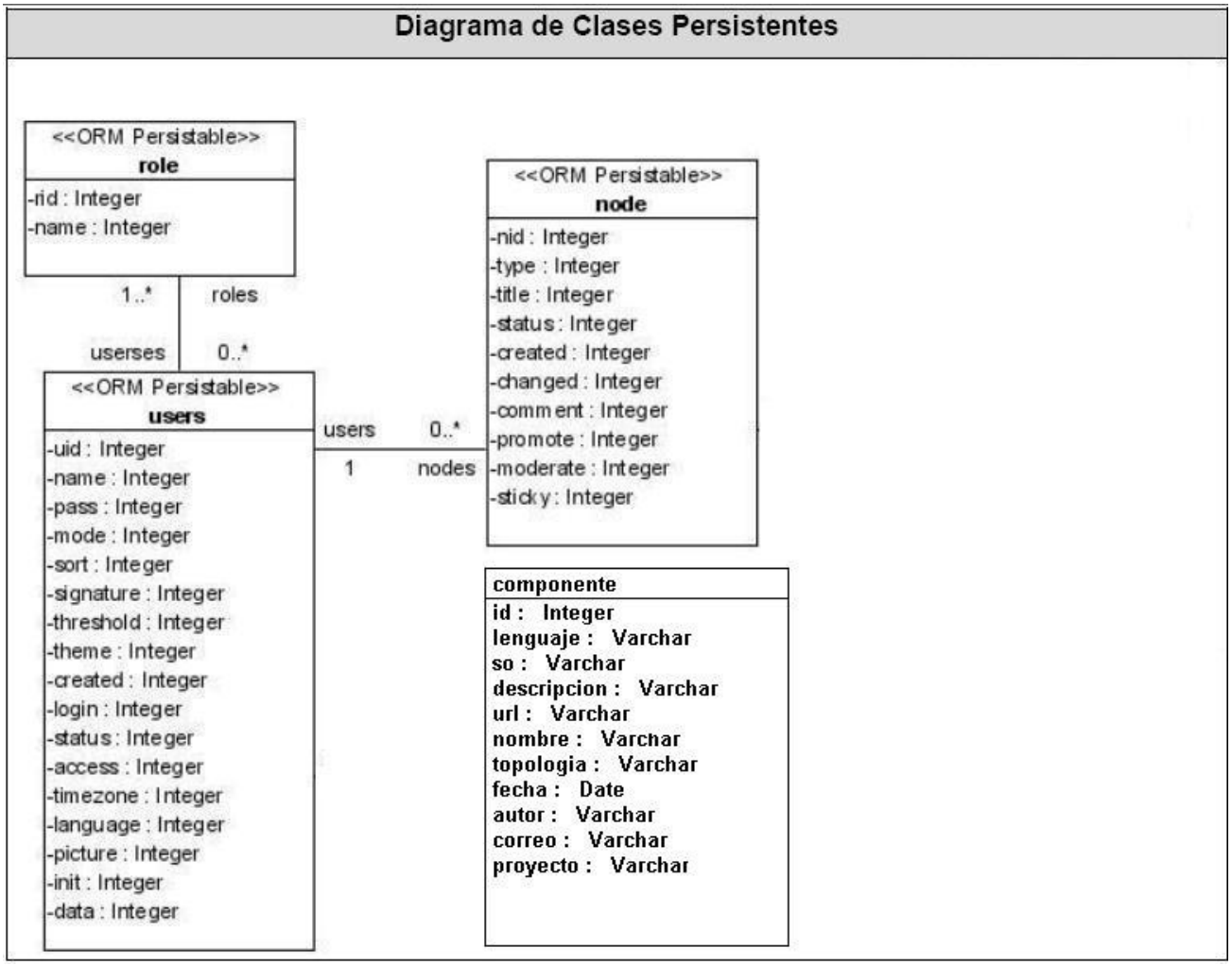
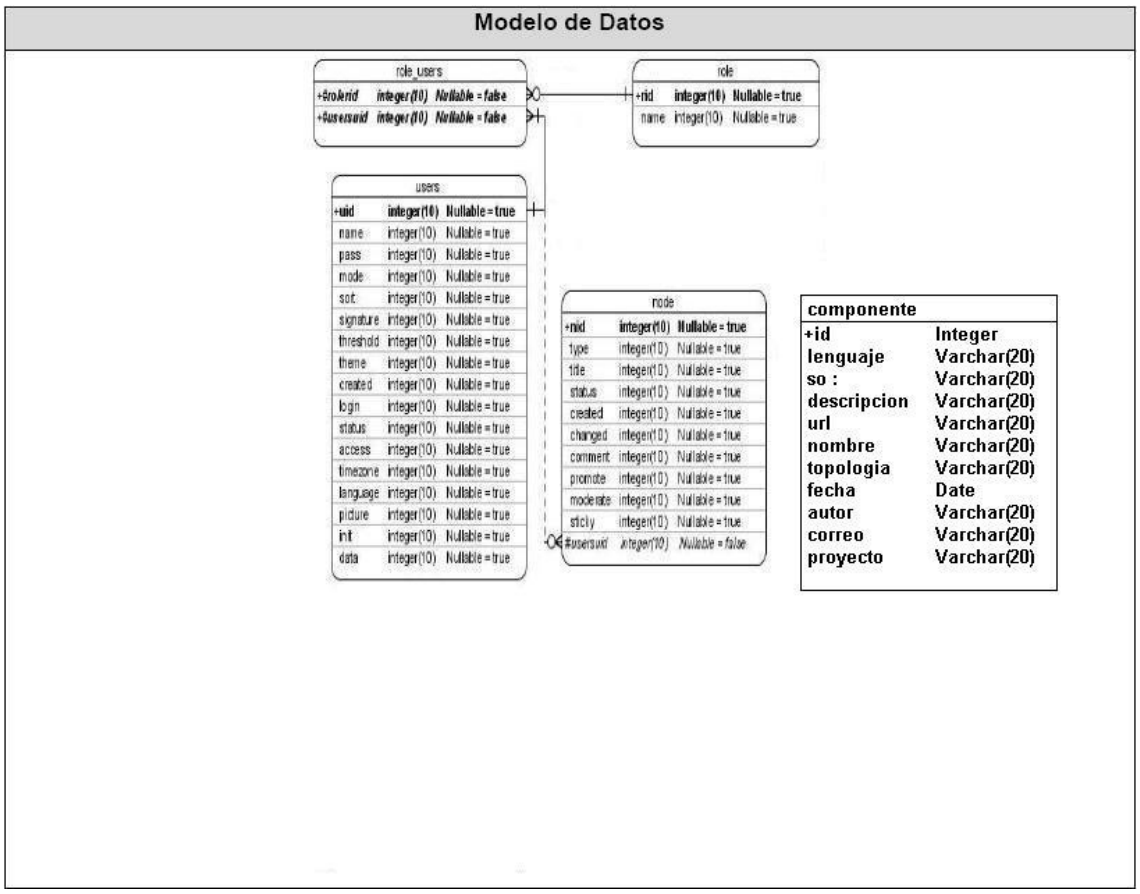


Figura 4 Modelo de datos



4.4 Fase de implementación

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o una persona), responsable de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo 2 iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan cada una de las iteraciones.

4.4.1 Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad, con el fin de obtener una versión del producto con algunas de las funcionalidades críticas para ser mostrado al cliente y tomar nuevas iniciativas de forma rápida.

Tabla 4.3 Módulos abordados en la primera iteración

Módulo	Historias de usuario	Tiempo de Implementación (semanas)	
		Estimación	Real
Gestión de Usuario	Registrar usuario	1	0.5
	Autenticar usuario	1	0.2
Gestión de Componentes	Insertar componentes	2	2

A continuación se muestran las tareas efectuadas para cada uno de los módulos implementados en esta iteración: **Módulo Gestionar Usuario**

Tabla 4.4 Tarea 1 del módulo Gestionar Usuario

Tarea	
Número de tarea: 1	**Número de HU: 1**
Nombre de la tarea: Configuración del módulo Gestionar Usuario	
Tipo de tarea: Configuración	**Puntos estimados:** 0.29
Fecha inicio: - 15 de marzo del 2009	**Fecha fin:** - 16 de marzo del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se especifica el rol al cual estará sujeta la cuenta. Los datos serán almacenados en la Base de Datos de forma persistente.	

Tabla 4.5 Tarea 2 del módulo Gestionar Usuario

Tarea	
Número de tarea: 2	**Número de HU: 1**
Nombre de la tarea: Manejo de persistencia en la BD	
Tipo de tarea: Desarrollo	**Puntos estimados: 0.29**
Fecha inicio: - 17 de marzo del 2009	**Fecha fin:** - 18 de marzo del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se gestionará la verificación de los datos introducidos con los almacenados en la Base de Datos. Se darán de alta y baja en el sistema a diferentes roles, reconociendo solamente los que están registrados en él. Un rol puede estar vinculado a varios usuarios.	

Tabla 4.6 Tarea 3 del módulo Gestionar Usuario

Tarea	
Número de tarea: 3	**Número de HU: 2**
Nombre de la tarea: Gestionar autenticación	
Tipo de tarea: Configuración	**Puntos estimados:** 0.29
Fecha inicio: - 19 de marzo del 2009	**Fecha fin:** - 20 de marzo del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se especificará el modo de autenticación por parte del administrador. Se gestionará la verificación de los datos introducidos con los almacenados en la Base de Datos.	

Tabla 4.7 Tarea 4 del módulo Gestionar componente

Tarea	
Número de tarea: 3	**Número de HU: 3**
Nombre de la tarea: Insertar componente	
Tipo de tarea: Desarrollo	**Puntos estimados:** 2
Fecha inicio: - 21 de marzo del 2009	**Fecha fin:** - 22 de marzo del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se mostrará un formulario que debe ser llenado donde se brinda la opción de seleccionar el componente de una dirección física de la máquina. Los datos se salvarán en la base de datos.	

4.4.2 Iteración 2

Durante el transcurso de la presente iteración se concluyó la implementación y configuración de las funcionalidades del módulo Gestionar Componente y se continúa en el perfeccionamiento de las historias de usuarios definidas para la iteración anterior, teniendo como resultado de dicha iteración otra versión funcional del producto más acabada que la obtenida anteriormente.

Tabla 4.7 Módulos abordados en la primera iteración

Módulo	Historias de usuario	Tiempo de Implementación (semanas)	
		Estimación	Real
Gestión de Componentes	Buscar componentes	2	2
	Eliminar componentes	2	2

Tabla 4.8 Tarea 5 del módulo Buscar componente

Tarea	
Número de tarea: 5	**Número de HU: 4**
Nombre de la tarea: Realizar búsqueda de componentes	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.6
Fecha inicio: - 24 de marzo del 2009	**Fecha fin:** - 29 de marzo del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Con esta tarea se persigue la realización de una búsqueda correcta de componentes basada en los criterios de búsqueda establecidos.	

Tabla 4.9 Tarea 6 del módulo Buscar componente

Tarea	
Número de tarea: 6	**Número de HU: 4**
Nombre de la tarea: Mostrar resultado de la búsqueda	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha inicio: - 30 de marzo del 2009	**Fecha fin:** - 4 de abril del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se mostrarán los resultados de la búsqueda realizada.	

Tabla 4.10 Tarea 7 del módulo Eliminar componente

Tarea	
Número de tarea: 7	**Número de HU: 5**
Nombre de la tarea: Realizar búsqueda de componentes	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha inicio: - 5 de abril del 2009	**Fecha fin:** - 6 de abril del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se realizará una búsqueda correcta de componentes basada en los criterios de búsqueda establecidos, para luego brindar la opción de eliminar el componente buscado.	

Tabla 4.11 Tarea 8 del módulo Eliminar componente

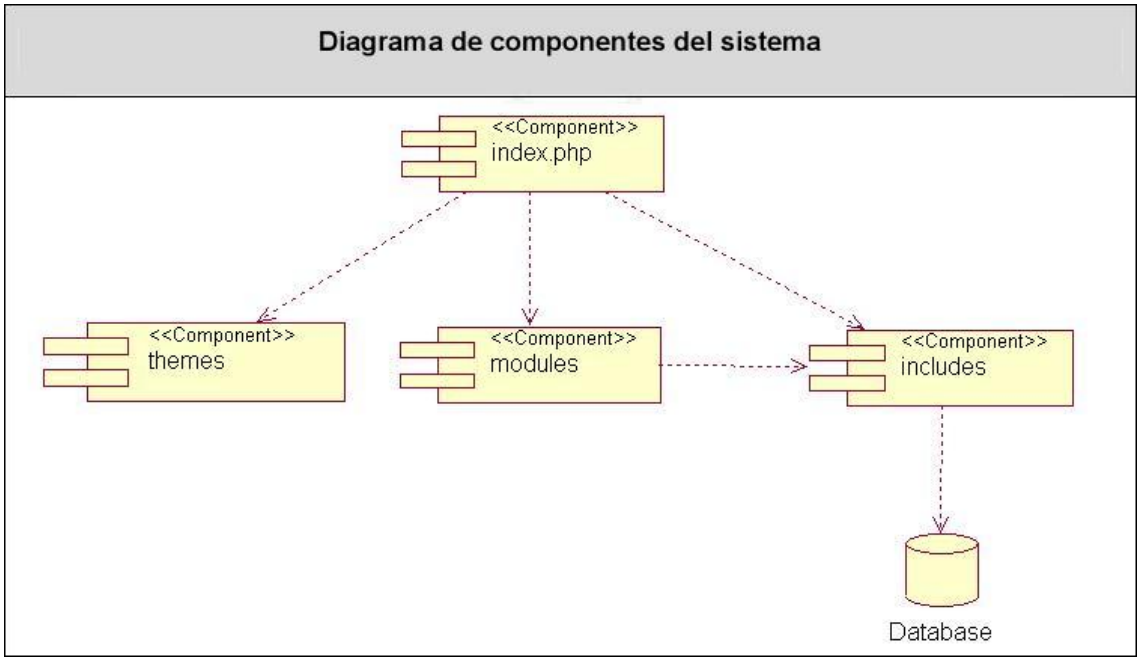
Tarea	
Número de tarea: 8	**Número de HU: 5**
Nombre de la tarea: Mostrar resultado de la búsqueda-eliminación	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.6
Fecha inicio: - 7 de abril del 2009	**Fecha fin:** - 8 de abril del 2009
Programador responsable: Renier Ochoa López – Teddy Bravo Saavedra	
Descripción: Se mostrarán los resultados de la búsqueda realizada y puede eliminar.	

4.4.4 Diagrama de Componentes

Un diagrama de componentes representa la separación del sistema de software en componentes físicos por ejemplo: archivos, módulos, paquetes, etc, y muestra las dependencias entre estos componentes. Se utiliza para modelar la vista estática de un sistema, además de mostrar la organización y las dependencias entre un conjunto de componentes.

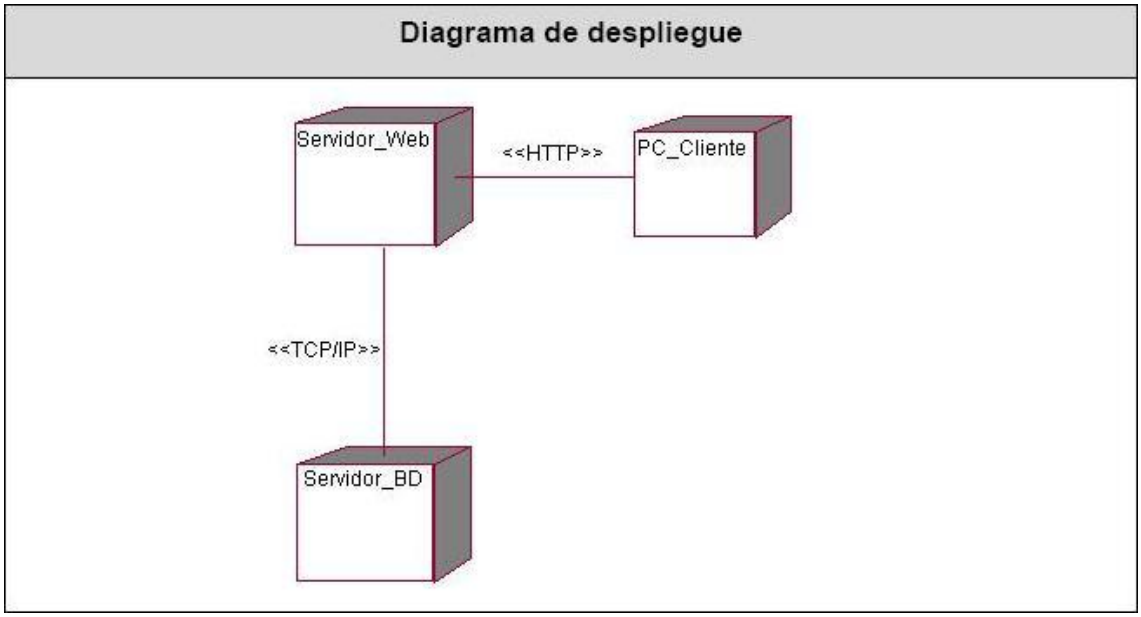
Uno de los usos principales, sirve para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

Con el propósito de brindar un mejor entendimiento del sistema se muestra el diagrama de componentes del sistema y de cada uno de los módulos implementados.



4.4.5 Diagrama de Despliegue

El diagrama de despliegue permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso el usuario accede al sistema desde una PC cliente a través de un navegador Web por medio del protocolo http. La aplicación se encuentra hospedada en un servidor Web, el cual se conecta al servidor de base de datos (MySQL) mediante el protocolo TCP/IP.



4.5 Pruebas

Uno de los pilares fundamentales de XP es el proceso de pruebas [14], el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección [13]. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. [13]

4.5.1 Pruebas de aceptación

Estas pruebas son pruebas de caja negra que se crean a partir de las historias de usuario [13]. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los

requerimientos han sido cumplidos y que el sistema es aceptable [15]. Una HU no se considera completa hasta que no ha pasado por sus pruebas de aceptación.

Tabla 4.12 Prueba 1 al módulo Gestionar Componentes

Caso de Prueba de Aceptación	
Código: HU3_P1	**Historia de Usuario: 3**
Nombre: Insertar Componente.	
Descripción: Prueba para la funcionalidad de insertar un componente en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y debe tener permisos para realizar esta operación.	
Entrada/ Pasos de ejecución: Se intenta publicar una conferencia por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: El componente es subido sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.13 Prueba 2 al módulo Gestionar Componentes

Caso de Prueba de Aceptación	
Código: HU4_P2	**Historia de Usuario: 4**
Nombre: Buscar Componente.	
Descripción: Prueba para la funcionalidad de buscar un(os) componente(s) en el sistema.	
Condiciones de Ejecución: No es necesario que el usuario esté autenticado.	
Entrada/ Pasos de ejecución: Se intenta realizar una búsqueda de un(os) componente(s) por un usuario.	
Resultado Esperado: La búsqueda es realizada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.14 Prueba 3 al módulo Gestionar Componentes

Caso de Prueba de Aceptación	
Código: HU5_P3	**Historia de Usuario: 5**
Nombre: Eliminar Componente.	
Descripción: Prueba para la funcionalidad de eliminar un(os) componente(s) en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos requeridos para realizar esta operación.	

Entrada/ Pasos de ejecución: Se intenta eliminar un(os) componente(s) por un usuario que cumple rol de administrador.
Resultado Esperado: El componente es eliminado sin errores.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 4.15 Prueba 4 al módulo Gestionar Componentes

Caso de Prueba de Aceptación	
Código: HU3_P4	**Historia de Usuario: 3**
Nombre: Descargar Componente.	
Descripción: Prueba para la funcionalidad de descargar un(os) componente(s) en el sistema.	
Condiciones de Ejecución: El usuario no necesariamente tiene que estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta descargar algún componente de los se encuentran en el servidor, sin necesidad de estar autenticado en el sistema.	
Resultado Esperado: El componente es descargado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

4.6 Conclusiones del capítulo

Se construyó el modelo necesario para llevar a cabo el proceso de implementación del sistema. También se diseñaron las clases persistentes que permiten hacer el diagrama de entidad-relación en el sistema de gestión de bases de datos que se utilizará. Se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y las pruebas de aceptación que propician al cliente conformidad y seguridad ante el sistema. Con la realización de este tópico se da por terminada la propuesta que trae este trabajo.

Conclusiones Generales

Con el desarrollo de la propuesta de solución se entregó una aplicación web que permite el almacenamiento y distribución de componentes para proyectos que desarrollen productos educativos y multimedia.

La utilización de la metodología seleccionada permitió un desarrollo ágil del trabajo por parte de los implicados en el mismo, además se pudo documentar todo el trabajo desde el principio, lo cual permitirá una mejor comprensión del funcionamiento del sistema desarrollado.

Las herramientas y tecnologías puestas en práctica permitieron el desarrollo claro y fluido de un sistema construido sobre bases sólidas y un entorno bien definido.

Recomendaciones

Como resultado del proceso de investigación y realización del sistema han brotado ideas que serían recomendables tener en cuenta para una futura mejoría del producto, a continuación se listan las mismas:

- ✓ Promover la utilización de metodologías ágiles en el desarrollo de futuros proyectos a partir de las experiencias positivas obtenidas.

- ✓ Estudiar la posibilidad de realizar un sistema similar para ser usado en el país, específicamente para proyectos enfocados al desarrollo de software educativo y multimedia.

Referencias bibliográficas

[1] http://sibdi.bldt.ucr.ac.cr/repositorios.htm Consultado 22/01/2009

[2] http://www.masadelante.com/faq-servidor.htm Consultado 23/01/2009

[3] http://www.mastermagazine.info/termino/5560.php Consultado 23/01/2009

[4] http://www.e-mexico.gob.mx/wb2/eMex/eMex_Glosario_de_terminos _Seguridad?page=25 Consultado 23/01/2009

[5] http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos Consultado 28/01/2009

[6] http://www.atrioweb.com/cms/ Consultado 02/02/2009

[7] http://drupal.org.es/caracteristicas Consultado 04/02/2009

[8] Ivar Jacobson, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo de software Consultado 05/02/2009

[9] http://www.mug.cl/articulos.php?id=287 Consultado 06/02/2009

[10] http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm Consultado 06/02/2009

[11] http://www.monografias.com/trabajos24/herramientas-case/herramientas - case.shtml Consultado 06/02/2009

[12] http://www.slideshare.net/vivi_jocadi/rational-rose Consultado 06/02/2009

[13] http://metodologiaxpvsmetodologiarup.blogspot.com/ Consultado 23/01/2009

[14] Beck, K. Extreme Programming Explained. s.l.: Addison Wesley, 2000 Consultado 12/02/2009

[15] http://www.extremeprogramming.org Consultado 10/02/2009

Bibliografías

1. http://menatronica.blogspot.com/2009/05/proceso-unificado-rational-rup.html Consultado 22/01/2009

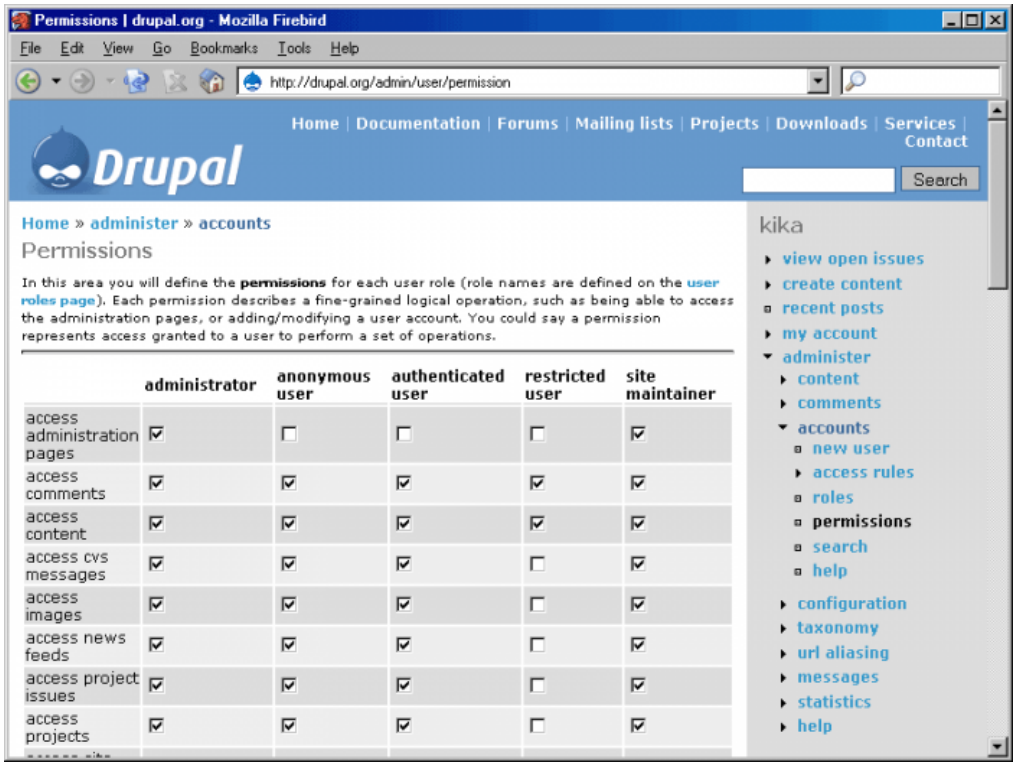
2. http://scruz334.blogspot.es/1194151560/ Consultado 22/01/2009

3. http://metodologiaxpvsmetodologiarup.blogspot.com/ Consultado 23/01/2009

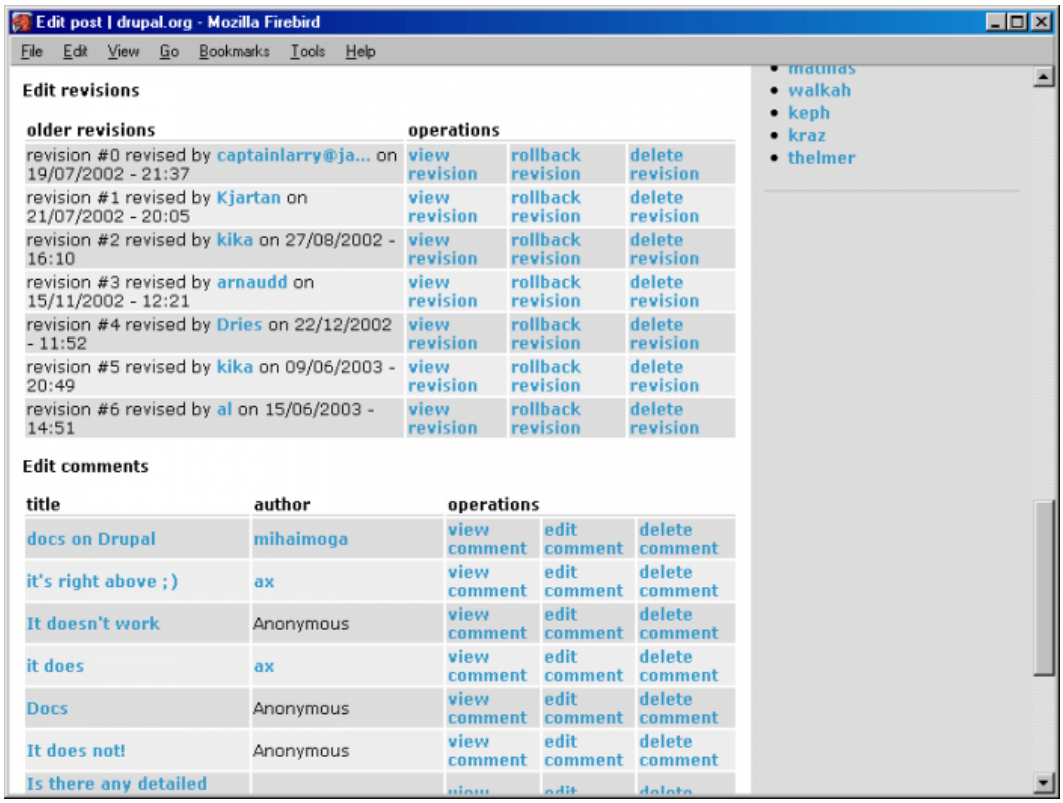
4. http://drupal.org.es/dh Consultado 15/02/2009

Anexos

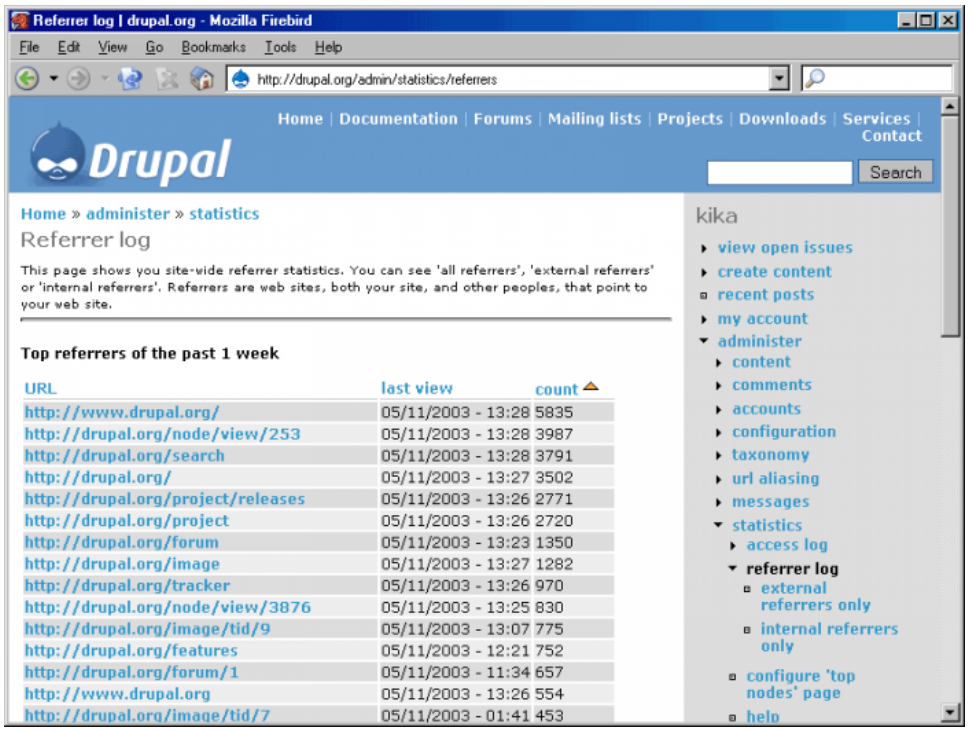
Anexo I



Anexo II


Edit post | drupal.org - Mozilla Firebird
File Edit View Go Bookmarks Tools Help
Edit revisions
older revisions
operations
revision #0 revised by captainlarry@ja... on 19/07/2002 - 21:37
revision #1 revised by Kjartan on 21/07/2002 - 20:05
revision #2 revised by kika on 27/08/2002 - 16:10
revision #3 revised by arnaudd on 15/11/2002 - 12:21
revision #4 revised by Dries on 22/12/2002 - 11:52
revision #5 revised by kika on 09/06/2003 - 20:49
revision #6 revised by al on 15/06/2003 - 14:51
view revision
rollback revision
delete revision
Edit comments
title
author
operations
docs on Drupal
mihaimoga
it's right above ;)
ax
It doesn't work
Anonymous
it does
Docs
It does not!
Is there any detailed
view comment
edit comment
delete comment
walkah
keph
kraz
thelmer

Anexo III


Referrer log | drupal.org - Mozilla Firebird
File Edit View Go Bookmarks Tools Help
http://drupal.org/admin/statistics/referrers
Drupal
Home | Documentation | Forums | Mailing lists | Projects | Downloads | Services | Contact
Search
Home » administer » statistics
Referrer log
This page shows you site-wide referrer statistics. You can see 'all referrers', 'external referrers' or 'internal referrers'. Referrers are web sites, both your site, and other peoples, that point to your web site.
Top referrers of the past 1 week
URL last view count
http://www.drupal.org/ 05/11/2003 - 13:28 5835
http://drupal.org/node/view/253 05/11/2003 - 13:28 3987
http://drupal.org/search 05/11/2003 - 13:28 3791
http://drupal.org/ 05/11/2003 - 13:27 3502
http://drupal.org/project/releases 05/11/2003 - 13:26 2771
http://drupal.org/project 05/11/2003 - 13:26 2720
http://drupal.org/forum 05/11/2003 - 13:23 1350
http://drupal.org/image 05/11/2003 - 13:27 1282
http://drupal.org/tracker 05/11/2003 - 13:26 970
http://drupal.org/node/view/3876 05/11/2003 - 13:25 830
http://drupal.org/image/tid/9 05/11/2003 - 13:07 775
http://drupal.org/features 05/11/2003 - 12:21 752
http://drupal.org/forum/1 05/11/2003 - 11:34 657
http://www.drupal.org 05/11/2003 - 13:26 554
http://drupal.org/image/tid/7 05/11/2003 - 01:41 453
kika
view open issues
create content
recent posts
my account
administer
content
comments
accounts
configuration
taxonomy
url aliasing
messages
statistics
access log
referrer log
external referrers only
internal referrers only
configure 'top nodes' page
help