



Universidad de las Ciencias Informáticas

Facultad 4

Título: Sistema de Gestión de Calidad del Grupo de Calidad de FORTES. Módulo de Capacitación.

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

Autor: Julio Cruz Castro

Tutor: Ing. Lizardo Ramírez Tabuada.

Ciudad de la Habana

2011

Declaración de autoría.

Declaramos que somos los únicos autores del trabajo “**Sistema de Gestión de Calidad del Grupo de Calidad de FORTES. Módulo de Capacitación.**” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año_____.

Autor:

Tutor:

Julio Cruz Castro

Ing. Lizardo Ramírez Tabuada

Dedicatoria

A mis padres... por apoyarme en todas las decisiones que he tomado e impulsarme a seguir adelante.

A mis hermanos... que han compartido conmigo muchas cosas y siendo el mayor he tratado de serles de ejemplo.

A toda mi familia... que siempre me apoyó en todo lo que pudo.

A todos mi amigos....por haberme dado su apoyo y ayuda en todo momento.

Agradecimientos

A mis padres y hermanos por su apoyo incondicional.

A mis tías y mis abuelos por estar siempre preocupados por mi desempeño aquí en la Universidad

Al tribunal por ser tan exigente.

Al tutor por ayudarme en el difícil trabajo con la tesis.

A la oponente por ser bien exigente y ayudarme a mejorar en todos los aspectos.

A la profesora Yisell Góngora por ayudarme y guiarme en estos 5 años en la UCI.

A mis compañeros de aula y de apartamento por darme su apoyo.

A Wisel y Ancel por ayudarme con la programación cuando tuve problemas.

A todo mi equipo de DotA que no solo sabe jugar sino también ayudaron con mi tesis con sus conocimientos.

A todos los que me han dado su apoyo en esta escuela y en la vida.

MUCHAS GRACIAS A TODOS.....

Índice

Índice	5
Introducción	7
Capítulo 1. Fundamentación teórica	10
1.1. Estado del arte	10
1.2. Tecnologías	10
1.2.1. Sitios Web.	10
1.2.2. Diferencia entre sitio web y página web	11
1.2.3. Sistema de Gestión de Contenido (CMS)	12
1.2.4. Tipos de CMS	13
1.2.5. Lenguaje de programación.....	17
1.3. Metodologías	19
1.3.1. Programación Extrema (XP)	19
1.3.2. Detalles de XP	21
1.3.3. Proceso XP	21
1.3.4. Proceso Unificado de Rational (RUP)	22
1.3.5. UML	22
1.3.1. RUP vs XP	24
1.4. Herramientas	25
1.4.1. Visual Paradigm	25
1.4.2. Postgres	26
1.4.3. Servidor de aplicaciones Apache	27
1.5. Conclusiones	28
Capítulo 2 Análisis y Diseño del Sistema.....	28
2.1. Definición de los objetivos del sitio.....	29
2.2. Objetivos del Módulo de Capacitación	29
2.3. Análisis de sitios similares.....	29
2.4. Exploración	29
2.5. Requerimientos funcionales del sistema.....	30
2.6. Requerimientos no funcionales:	32
2.7. Historias de Usuario	36
2.8. Planificación	39
2.8.1. Estimación de esfuerzos por historias de usuario.....	39
2.9. Plan de iteraciones	40
2.9.1. Iteración 1	40

2.9.2.	Iteración 2	40
2.9.3.	Iteración 3	40
2.9.4.	Plan de duración de las iteraciones	40
2.10.	Plan de entregas	41
2.10.1.	Plan de duración de las entregas	41
2.11.	Conclusiones	42
Capítulo 3 Implementación y Pruebas		43
3.1.	Introducción	43
3.2.	Diseño del sistema	43
3.2.1.	Módulos de Drupal	43
3.2.2.	Paquete 1: Drupal.....	44
3.2.3.	Sub paquete 1.1: Modules	45
3.2.4.	Tarjetas CRC.....	46
3.3.	Fase de Implementación	47
3.3.1.	Iteración 1	48
3.3.2.	Iteración 2.....	51
3.3.3.	Iteración 3.....	54
3.3.4.	Diagrama de Componentes	56
3.3.5.	Diagrama de Despliegue	57
3.4.	Pruebas	60
3.4.1.	Pruebas de aceptación.....	60
3.5.	Resultado de las pruebas.....	69
3.6.	Conclusiones	69
Conclusiones		70
Bibliografía.....		71

Introducción

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) es incuestionable y forma parte de la cultura tecnológica con la que se convive; amplía las capacidades físicas y mentales, así como las posibilidades de desarrollo social. Las TIC tienen, día a día, una mayor presencia en todos los aspectos de la vida laboral y personal, ofreciendo un nuevo espacio de innovación en ámbitos como la industria, los servicios, la salud, la administración, el comercio y la educación. La evolución de las tecnologías de la información tiende a incorporar, dentro de sus estructuras informáticas, la gestión de la documentación que tradicionalmente se ha conservado en los archivos, en papel.

En la actualidad, coexisten en el mundo los más diversos sistemas de gestión: desde el simple registro manual de la correspondencia, hasta los más sofisticados sistemas informáticos que manejan no sólo la documentación administrativa propiamente dicha, venga ella en papel o en formato electrónico, sino que además controlan los flujos de trabajo del proceso de tramitación de expedientes, capturan información desde bases de datos, contabilidad y otros, enlazan con el contenido de archivos, bibliotecas, centros de documentación y permiten realizar búsquedas sofisticadas y recuperar información.

El sistema de gestión de una organización es el conjunto de elementos (estrategias, objetivos, políticas, estructuras, recursos y capacidades, métodos, tecnologías, procesos, procedimientos, reglas e instrucciones de trabajo) mediante el cual la dirección planifica, ejecuta y controla todas sus actividades en aras al logro de los objetivos preestablecidos. (1)

“Aquellos que creen que la capacitación es costosa, pueden probar con la ignorancia.” (2)

El concepto de calidad es hoy moneda corriente en las empresas y organizaciones de muchos países. En muchas empresas cada año el tema se reaviva en ellas y surgen ideas y planes que la mayoría de las veces quedan en la nada. La creencia generalizada es que para acceder a una alta calidad sólo se puede hacer a través de costosos y complicados procesos. La mayoría de las veces esto es cierto y suele desalentar a aquellos empresarios que por diferentes razones no pueden acceder a dichos procesos.

Existen algunas herramientas de aplicación sencilla y de costo accesible, que al ser aplicadas mejoran enormemente la calidad de los servicios. Una de esas herramientas es la capacitación de las personas que por sus méritos accedieron a puestos de gerencia, jefatura o supervisión. Es habitual que éstas accedan a puestos superiores porque se destacan por su eficiencia en el cumplimiento de sus tareas.

“La meta de un buen programa de capacitación y desarrollo es hacer florecer las capacidades de los participantes”. (3)

En la Universidad de las Ciencias Informáticas (UCI) se está llevando a cabo la informatización de muchos sectores, lo que posibilita la agilidad y rapidez de trabajo por parte del personal. La facultad y la dirección del proyecto productivo están encargadas de asignar e impartir los temas de capacitación. Esta tarea se hace actualmente de forma manual lo que imposibilita un control estricto y facilidad en el manejo de los datos. De los procesos llevados a cabo por esta entidad depende en gran manera la calidad del trabajo que se realice así como la satisfactoria preparación del personal.

Dentro de la UCI hay proyectos productivos en los cuales la capacitación se realiza de forma manual lo que ocasiona que el proceso sea lento y más difícil de controlar. En el Centro de Tecnologías para la Formación (FORTES) perteneciente a la facultad 4 también se manifiesta este problema.

FORTES dentro de sus proyectos cuenta con el Grupo de Calidad de FORTES (GCF) el cual es un proyecto que ofrece servicios de aseguramiento de la calidad a los proyectos productivos del centro. En él se realizan actividades en las que intervienen diferentes roles como: Ingeniero en Pruebas de Sistema (o probador), y Diseñador de Pruebas e Ingeniero en Componentes, se realizan además, actividades relacionadas con la formación y capacitación del personal interno del proyecto. Las actividades relacionadas con la capacitación se han visto afectadas por disímiles razones:

- Poca disponibilidad de profesores que respalden la impartición de asignaturas electivas sobre temas de Calidad.
- Los cursos impartidos hasta el momento solo garantizan una formación parcial y muy generalizada como miembro del GCF, sin prácticamente contribuir a la capacitación para el desempeño de roles y competencias dentro del proyecto.
- Dentro del proyecto todo el trabajo y control de la capacitación se hace de forma manual.
- No existen datos históricos que muestren la evolución de los miembros del GCF desde su entrada al proyecto.
- No hay forma de dar seguimiento a la atención de los roles asignados, por parte de los profesores del GCF.
- Los roles son asignados sin tenerse en cuenta las aptitudes y competencias adquiridas por los estudiantes durante su formación dentro del GCF.
- Existe una marcada desorientación por parte de los estudiantes, respecto a los roles que ocupan dentro del GCF.

Por cuanto el **problema de investigación** radica en la orientación y centralización de la información referente a la capacitación dentro del Grupo de Calidad de FORTES.

Esto trae como consecuencia mejorar la preparación del personal dentro del proyecto así como la calidad a la hora de realizar las actividades productivas.

Se establece entonces como **objeto de estudio** el proceso de desarrollo de sitios web para la gestión de la capacitación. Donde surge como **objetivo general de la investigación** desarrollar una solución informática que permita la gestión de la capacitación del Grupo de Calidad de FORTES.

El **campo de acción** se centra en los sitios web basados en tecnologías CMS para la gestión de la capacitación del personal del Grupo de Calidad de FORTES.

La **idea a defender** que se plantea es: Si se desarrolla un sitio web que se caracterice por gestionar la capacitación del Grupo de Calidad de FORTES y que se actualice con periodicidad, se logrará la informatización de los cursos¹ que este ofrece así como el control e información por parte del personal del proyecto productivo.

Se definen como **tareas de investigación** las siguientes:

- Realización de entrevistas sobre los métodos de capacitación dentro del GCF.
- Redacción del diseño teórico de la Investigación.
- Realización del estudio del estado del arte acerca del objeto de investigación.
- Realización de la descripción del Modelo de negocio.
- Realización del levantamiento de los requisitos del sistema.
- Realización análisis y el diseño del sistema para el módulo de capacitación para el Sistema de Gestión de la Calidad del GCF.
- Realización de la implementación del sistema.
- Realización de pruebas funcionales a nivel de sistema al software para validarlo.
- Redacción del documento de tesis.

El contenido del presente documento se estructura en cuatro capítulos, donde se incluye todo lo relacionado con el trabajo desarrollado.

Capítulo 1 “Fundamentación Teórica”. Aborda el estudio del estado del arte del tema relacionado con el trabajo que se desarrolla. Se destacan además las tendencias y tecnologías actuales sobre las cuales se apoya la propuesta del sistema informático, así

¹ Cursos: Cursos de capacitación que reciben los miembros del proyecto para su superación.

como las diferentes metodologías de desarrollo de software y herramientas analizadas hasta mostrar las escogidas.

Capítulo 2 “Solución propuesta, Exploración y Planificación”. En este capítulo se definirán los objetivos del sitio y los servicios que brindará, se especificarán los requisitos funcionales y los no funcionales. Se dará a conocer el alcance del producto a desarrollar y se estimarán los tiempos de entrega década versión.

Capítulo 3 “Implementación y Pruebas”. Se explican temas referentes a la implementación de la solución. También se puntualiza cuáles fueron las pruebas realizadas para comprobar que el sistema brinde las funcionalidades acordadas.

Capítulo 1. Fundamentación teórica

En este capítulo se expondrán elementos de la investigación realizada acerca de sitios web existentes y del estado del arte para mayor entendimiento del tema. También se abordan temas referentes a las tecnologías, herramientas y lenguajes que se emplean en el desarrollo del sistema propuesto.

1.1. Estado del arte

La capacitación en las empresas se ha convertido en un factor fundamental para incrementar la producción y la preparación del personal. En el mundo hay compañías que implementan su sistema de capacitación a través de sitios web lo cual permite mayor control y efectividad a la hora de preparar y superar al personal.

1.2. Tecnologías

1.2.1. Sitios Web.

En inglés **website** o **web site**, un **sitio web** es un sitio (localización) en la World Wide Web que contiene documentos (páginas web) organizados jerárquicamente. Cada documento (página web) contiene texto y o gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos.

Cada sitio web tiene una página de inicio (en inglés Home Page), que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un

navegador. El sitio normalmente tiene otros documentos (páginas web) adicionales. Cada sitio pertenece y es gestionado y por un individuo, una compañía o una organización.

Como medio, los sitios web son similares a las películas, a la televisión o a las revistas, en que también crean y manipulan imágenes digitales y texto, pero un sitio web es también un medio de comunicación. La diferencia principal entre un sitio web y los medios tradicionales es que un sitio web está en una red de ordenadores (Internet) y está codificado de manera que permite que los usuarios interactúen con él. Una vez en un sitio web, puedes realizar compras, búsquedas, enviar mensajes, y otras actividades interactivas.

1.2.2. Diferencia entre sitio web y página web

A veces se utiliza erróneamente el término **página web** para referirse a **sitio web**. **Una página web es** parte de un sitio web y es un único archivo con un nombre de archivo asignado, mientras que un sitio web es un conjunto de archivos llamados páginas web.

Si se compara con un libro, un sitio web sería el libro entero y una página web de ese sitio web sería un capítulo de ese libro. El título del libro sería el nombre del dominio del sitio web. Un capítulo, al igual que una página web, tiene un nombre que lo define. Se dice que sería un capítulo y no una página del libro porque a menudo es necesario desplazarse hacia abajo en la pantalla para ver todo el contenido de una página web, al igual que en un libro se *desplazan* a través de varias páginas para ver todo el contenido de un capítulo. El índice de los capítulos del libro sería el equivalente al **mapa del sitio web (sitemap en inglés)**. (4)

Sitios de ejemplo:

www.masterclass.com.mx, es una compañía que se especializa en la oferta de cursos de superación y capacitación del personal de acuerdo a las necesidades existentes. Su objetivo ha sido alcanzado a través de la oferta de cursos especializados que se ofertan de acuerdo a las necesidades del país o empresas que soliciten. (5)

es.pro169.org, es un sitio web que se especializa en dar información y capacitación sobre los derechos de los pueblos indígenas y de los convenios de la OIT. (6)

www.desarrollodeweb.com.ar, es un sitio web que ofrece ayuda y capacitación para los que comienzan a trabajar con Joomla y además ofrece actualizaciones para este CMS así como guías para trabajar con ellas. (7)

1.2.3. Sistema de Gestión de Contenido (CMS)

Realizar un sitio web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En el pasado las herramientas eran básicamente editores que permitían generar una página, que evolucionaron para incorporar el control de la estructura de la web y otras funcionalidades, pero en general estaban enfocadas más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de sistema de gestión de contenidos (Content Management Systems o CMS). Se trata de herramientas que permiten crear y mantener un sitio web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las webs.

Teniendo en cuenta el ahorro que supone la utilización de estas herramientas, y el coste de desarrollarlas, sería lógico esperar que su precio fuera muy elevado. Eso es cierto para algunos productos comerciales, pero existen potentes herramientas de gestión de contenidos de acceso libre, disponibles con licencias de código abierto.

Los gestores de contenidos proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. En cualquier entorno virtual ésta es una característica importante, que además puede ayudar a crear una comunidad cohesionada que participe más de forma conjunta. (8)

¿Qué es un CMS?, Los sistemas de gestión de contenidos (Content Management Systems o CMS) son un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM). (2)

Los Sistemas Gestión de Contenidos presentan las siguientes características:

- ✓ Intuitivo: fácil de entender y utilizar.
- ✓ Flexible en la personalización.
- ✓ Extensible a través de *plugins* y módulos.
- ✓ Sin necesidad de tener conocimientos de programación.
- ✓ Están optimizados para un alto rendimiento y velocidad.
- ✓ Ofrece seguridad.
- ✓ Posee amplia documentación y apoyo comunitario.
- ✓ Énfasis en estándares web y buenas prácticas.

Dando respuesta a la problemática planteada se decide utilizar los CMS como herramienta de desarrollo dada la facilidad y las ventajas que estos ofrecen sobre otras herramientas de desarrollo

web porque vienen con funcionalidades implementadas, además ofrecen facilidades de administración y actualización.

1.2.4. Tipos de CMS

Drupal, 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 diferentes tipos 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.

Hooks: La documentación de Drupal tiene una sección dedicada a los Hooks. Los Hooks son el mecanismo que provee Drupal para interactuar con los distintos procesos que se ejecutan en un sitio web. Conocer su funcionamiento es fundamental para cualquier programador de módulos así como también para aquellos diseñadores o Themers que deseen modificar aspectos al parecer imposibles de lograr.

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

Módulos: La comunidad de Drupal ha contribuido con infinidad de módulos que proporcionan funcionalidades como página de categorías, autenticación mediante chat, mensajes privados y bookmarks, entre otros.

Personalización: Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo con 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 y 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.

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

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

Sindicación del contenido. Blogging. Agregador de noticias: Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un Agregador de Noticias, visualice el contenido publicado en la web desde el escritorio, además pueda leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

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

Plataforma Independiente 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 PostgreSQL, aunque permite incorporar fácilmente 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.

Múltiples 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.

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.

Registros e informes: Toda la actividad y los sucesos del sistema son capturados en un registro de eventos, que puede ser visualizado por un administrador.

Características de comunidad. Comentarios enlazados: Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión de un tema publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.

Encuestas: Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.

Foros de discusión: Drupal incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.

Libro Colaborativo: Esta característica es única de Drupal y permite crear un proyecto o libro cuyo contenido será la contribución de los mismos usuarios. El contenido se organiza en páginas cómodamente navegables.

Rendimiento y escalabilidad. Control de congestión: Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

Sistema de Cache: El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor. (9)

Joomla es un sistema de gestión de contenido premiado que permite construir sitios Web y aplicaciones en línea de gran alcance. Muchos aspectos, incluyendo su facilidad de uso y extensibilidad, con Joomla han hecho software, los sitios web más populares disponibles. Lo mejor de todo, Joomla es una solución de código abierto que está disponible gratuitamente para todo el mundo. (10)

Características generales:

Gestión de usuarios: Joomla permite almacenar datos de usuarios registrados y también la posibilidad de enviar e-mails masivos a todos los usuarios. La administración de usuarios es

jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio.

Organización del sitio web: Joomla está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel administrador de Joomla usted podrá crear, editar y borrar las secciones y categorías de su sitio de la manera en que más le convenga.

Publicación de Contenidos: Con Joomla CMS podrá crear páginas ilimitadas y editarlas desde un sencillo editor que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.

Escalabilidad e implementación de nuevas funcionalidades: Joomla ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de su sitio web, por ejemplo: galerías de imágenes, foros, newsletters, clasificados, etc.

Diseño y aspecto estético del sitio: Es posible cambiar todo el aspecto del sitio web tan solo con un par de clic, gracias al sistema de templates que utiliza Joomla.

Navegación y menú: Totalmente editables desde el panel administrador de Joomla.

Administrador de Imágenes: Joomla posee una utilidad para subir imágenes al servidor y usarlas en todo el sitio.

Disposición de módulos modificable: En un sitio creado con Joomla, la posición de módulos puede acomodarse como se prefiera.

Encuestas: Joomla posee un sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.

Feed de Noticias: Joomla trae incorporado un sistema de sindicación de noticias por RSS/XMS de generación automática.

Publicidad: es posible hacer publicidad en el sitio usando el Administrador de Banners.

Estadísticas de visitas: contiene información del navegador y detalles de los documentos (páginas) más vistos.

Dentro de los CMS de código abierto Joomla y Drupal son los mejores por lo cual es algo difícil seleccionar uno de ellos.

- ✓ En cuanto a la construcción del sitio, Drupal es significativamente más potente y mucho más flexible. El Views, CCK y Panels, le dan a Drupal una enorme ventaja sobre Joomla.
- ✓ Curva de aprendizaje: Joomla es de largo más fácil de configurar y poner en marcha. Incluso con los todos vídeos gratis y blogs, Drupal todavía es un desafío.
- ✓ Plantillas: Joomla gana por un amplio margen. Debido a que las empresas de temas para Drupal son muy malas. Por ejemplo, empresas para Joomla como Joomlaart, JoomlaLashack hacen un buen trabajo.
- ✓ Admin: El panel de control de la administración de Drupal es malo. La separación entre el front-end y el backend es débil y confusa. Joomla es mucho mejor.
- ✓ Editor de contenido: El sistema de taxonomías de Drupal es excelente. La aproximación de Joomla es pobre (los tipos de contenido están confinados a una sección/categoría). La consola de administración de Joomla es mucho más fácil para organizar y encontrar contenido. El WYSIWYG de Joomla es mucho mejor que las opciones que existen en Drupal.
- ✓ Construcción de contenido: Las herramientas de Drupal son increíblemente mejores que las de Joomla, CCK y Views. Con Joomla puedes conseguir componente que ofrezcan el 60-80% de lo que necesitas. Por ejemplo, Joomla tiene un tremendo componente de noticias, pero nunca llegará a las capacidades de CCK, Views, etc. (10)

Si lo que se quiere lograr es un sitio con una amplia estética y lento a la hora de gestionar los contenidos entonces definitivamente seleccionaríamos el Joomla con gestor de contenidos.

Pero si lo que queremos lograr es un alto rendimiento, mejor escalabilidad, mejor gestión de los contenidos y una significativa flexibilidad del diseño esto se puede lograr con Drupal.

Por tanto, debido a su rendimiento, taxonomía y la calidad en la gestión de contenidos de Drupal, se propone el uso de este CMS para desarrollar la propuesta de solución del Grupo de Calidad de FORTES.

1.2.5. Lenguaje de programación

PHP: Las siglas significan "PHP Hypertext Pre-processor" (*inicialmente PHP Tools, o, Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Es gratuito e independiente de plataforma, muy rápido, con una gran librería de funciones y mucha documentación. (11)

PHP es ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la

mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Ventajas

- ✓ Es un lenguaje multiplataforma.
- ✓ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ 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 aplicar técnicas de programación orientada a objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).
- ✓ 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.

Inconvenientes

- ✓ La ofuscación de código es la única forma de ocultar las fuentes.

Seguridad

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

PHP se utilizará como el lenguaje de programación por su facilidad de conectividad con base de datos MySQL y Postgres, además, que Drupal está desarrollado en PHP por lo que sería muy oportuno su uso. (12) (11)

1.3. Metodologías

Para el desarrollo del software se debe contar con un proceso bien detallado y para esto se necesita aplicar una metodología que sea capaz de llevar a cabo el control total del producto. De esta forma se cumple con el objetivo principal de asegurar la producción de software de alta calidad, con el fin de satisfacer las necesidades de los usuarios finales. Esto se realiza dentro de un cronograma y con presupuestos bien definidos.

Algunas de las metodologías que se ponen en práctica a la hora de desarrollar un software son las siguientes: metodología orientada a objetos planteada por Rational (la metodología RUP) y la metodología XP. Se ha definido como metodología de desarrollo la conocida metodología basada en XP.

1.3.1. Programación Extrema (XP)

Es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

XP está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. (1)

Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Consiste en un conjunto de prácticas que a lo largo de los años han demostrado ser las mejores prácticas de desarrollo de software, llevadas al extremo y fundamentadas, en un conjunto de valores.

Reglas y prácticas

Planificación

- ✓ Se escriben las historias de usuario.
- ✓ El plan de entregas crea el cronograma.
- ✓ Entregas pequeñas muy frecuentes.
- ✓ Se mide la velocidad del proyecto.
- ✓ Se divide el proyecto en fases de iteración.
- ✓ La planificación de la iteración inicia en cada fase.
- ✓ El personal rota por las diferentes áreas del sistema.
- ✓ Reuniones de pie, todos los días.
- ✓ Corregir el proceso si falla (cambiar las reglas por consenso).

Diseño

- ✓ Simplicidad.
- ✓ Elegir una metáfora de sistema, un conjunto de nombres ilustrativos de la realidad.
- ✓ Usar tarjetas CRC en las sesiones de diseño.
- ✓ Crear soluciones rápidas puntuales para reducir el riesgo.
- ✓ No agregar funcionalidad antes de tiempo.
- ✓ Refactorizar donde y cuando sea posible.
- ✓ Codificación.
- ✓ El cliente siempre al alcance.
- ✓ El código debe respetar las normas aceptadas.
- ✓ Codificar primero las pruebas de unidad.
- ✓ Todo el código se escribe en parejas de programadores.
- ✓ Integra código sólo una pareja por vez.
- ✓ Integrar seguido.
- ✓ El código es propiedad colectiva.
- ✓ No optimizar hasta el final del proyecto.
- ✓ No trabajar horas extra.

Prueba

- ✓ Todo el código debe tener pruebas de unidad.
- ✓ Todo el código debe pasar las pruebas de unidad antes de la entrega.
- ✓ Crear pruebas toda vez que aparezcan errores.
- ✓ Correr frecuentes pruebas de aceptación y publicar la puntuación.

1.3.2. Detalles de XP

Historias de usuario: Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcional. Similares a los casos de uso pero no lo mismo; estiman el tiempo con vistas al plan de entrega.

Plan de iteración: Incluye los relatos de usuario y las pruebas de aceptación. Ambas se dividen en tareas de programación. Una tarea de programación abarca de 1 a 3 días de trabajo ideales, sin interrupciones ni otras actividades.

Pruebas de unidad: Las pruebas de unidad evolucionan junto con el código. No pueden crearse al final, ni dejar de escribirse. Las pruebas de unidad posibilitan la propiedad colectiva de código, la refactorización, la integración frecuente. El agregado de funcionalidad incluye el agregado de pruebas. Crear las pruebas antes de escribir el código. No puede integrarse código sin sus pruebas de unidad.

Prueba de aceptación: Las pruebas de aceptación se crean a partir de los relatos de usuario. El cliente define los escenarios de prueba para verificar si el relato de usuario ha sido correctamente implementado. Un relato de usuario puede tener una o varias pruebas de aceptación. El cliente es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas.

1.3.3. Proceso XP

El ciclo de desarrollo consiste en los siguientes pasos:

- ✓ El cliente define el valor de negocio a implementar.
- ✓ El programador estima el esfuerzo necesario para su implementación.
- ✓ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- ✓ El programador construye ese valor de negocio.
- ✓ Vuelve al primer paso.
- ✓ En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. (13) (14)

1.3.4. Proceso Unificado de Rational (RUP)

Es un marco de trabajo extensible que puede ser adaptado a proyectos específicos o a organizaciones. Es el eje orientador que permite controlar el desarrollo iterativo e incremental del sistema, a través de un ciclo de vida. (15)

Es basado en el modelo en espiral que organiza las iteraciones por etapas y fases para obtener una estructura que se ajuste a las necesidades particulares de cada organización y para facilitar la administración del proyecto.

RUP es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software, como lenguaje de modelado utiliza el UML.

1.3.5. UML

Por sus siglas en inglés, (Unified Modeling Language) es el lenguaje gráfico para modelado de sistemas con tecnología orientada a objeto que permite especificar, visualizar, construir y documentar. Soporta todo el ciclo de vida de desarrollo de un software: especificaciones de analistas, arquitectura, diseño, implementación e implantación. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (1)

RUP se caracteriza por:

- ✓ Iterativo e incremental.
- ✓ Centrado en la arquitectura.
- ✓ Guiado por los casos de uso.

Incluye artefactos y roles. RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante.

Fases dentro del ciclo de vida

Inicio: Se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto.

Elaboración: Se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema.

Construcción: Se crea el producto.

Transición: El producto se convierte en la versión beta.

Flujos de trabajo

- ✓ Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✓ Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✓ Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.
- ✓ Instalación: Produce liberación del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ✓ Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ✓ Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Conclusiones sobre RUP

- ✓ Es un esqueleto del proceso a desarrollar.
- ✓ Iterativo e incremental.
- ✓ Maneja casos de uso.
- ✓ Es diseñado para ser flexible y extensible.
- ✓ Permite una variedad de estrategias de ciclos de vida.
- ✓ Elegir que artefactos construir.
- ✓ Define actividades y trabajadores.

- ✓ No es un proceso universal.

1.3.1. RUP vs XP

XP es adecuada para equipos muy chicos, de dos a cuatro personas, y para proyectos de corta duración y metas modestas. En el caso del RUP, sin embargo, la implementación de los procesos para equipos pequeños o proyectos chicos se convierte en un gasto de tiempo y dinero innecesario.

La mayoría de los equipos que utilizan RUP lo han hecho por varios años y tienen especialistas del proceso que facilitan a los miembros sin experiencia el conocimiento de las prácticas y metodologías. La metodología XP es mucho más fácil de implementar y de aprender, por lo que los equipos jóvenes pueden incorporarla de manera más natural.

RUP hace un uso intensivo de artefactos de muy diversos tipos, entre ellos, el uso de artefactos de documentación es quizá una de los factores que lo hacen tedioso para algunos. El uso intensivo de documentación es una buena práctica que no debe abandonarse incluso en XP, aunque esta tiene sus propios artefactos, más ágiles y menos protocolarios, pero igual de exhaustivos.

Los ciclos de vida de un proyecto en XP y en RUP no son exactamente iguales, aunque sin duda tienen bastantes similitudes, ambas son metodologías iterativas con probado éxito en el desarrollo de software.

En un proyecto con XP, la primera fase es llamada *Ápice arquitectónico* corresponde bastante con la *Incepción* de RUP. El *ápice arquitectónico* de la XP suele ser mucho más rápida que en RUP, donde la *inceptión* puede tener varias iteraciones, sin embargo, ambas buscan lo mismo, conceptualizar de manera general el proyecto.

La fase de *Plan de entregas* de XP podría verse como la *Elaboración* de RUP, en ambas se presenta los guiones de usuario (XP) o casos de uso (RUP) y se establecen con más claridad los requerimientos del sistema generales. Una de las mayores diferencias es la documentación asociada a estas fases y el estilo de la misma.

Después de esto comienza propiamente el desarrollo en sí. En RUP se le llama *Construcción*, en XP se le considera como *Iteración*. Aquí las iteraciones cobran su verdadera importancia y ambas metodologías comienzan cada iteración con guiones de usuario o casos de uso que deberán cumplirse al final de la iteración y con un trabajo de arquitectura.

Finalmente en XP se debe cumplir las pruebas de aceptación definidos también en los guiones de usuario donde se cotejan los resultados actuales con lo que se esperaba del sistema. En RUP esta fase se contempla dentro de la *Construcción* y la *Transición*, su fase final.

En la Transición de RUP o Entrega de XP las diferencias pueden ser mayores, para RUP la entrega final debe ser algo mucho más definido, mientras que en XP se realizan entregas continuas y discretas que permiten evaluar el sistema conforme se colocan las versiones finales.

La diferencia está, en el nivel de protocolo, la agilidad en el desarrollo, conseguida mediante una comunicación intensiva del equipo en una confianza en todos los desarrolladores y de las jerarquías dentro y fuera del equipo y una autoevaluación intensiva incluso a nivel de cada desarrollador. (16)

Por lo cual se optó por utilizar la metodología XP debido a que el equipo de desarrollo es pequeño y se necesita desarrollar el sitio web en poco tiempo. Lo cual solo es posible con el uso de esta metodología ágil.

1.4. Herramientas

1.4.1. Visual Paradigm

Es una herramienta visual para el modelado UML que brinda soporte a todo el ciclo de vida de un producto software: análisis y diseño, construcción, pruebas y despliegue. Está disponible en varias ediciones, cada una destinada a una necesidad diferente: Enterprise, Professional, Community, Standard, Modeler y Personal.

Ofrece un entorno de creación de diagramas para UML, un diseño centrado en casos de uso y enfocado al negocio que propicia un software de calidad y provee a todo el equipo de desarrollo de un lenguaje estándar común que facilita la comunicación y capacidades de ingeniería directa e inversa.

Entre las características fundamentales de esta herramienta se puede destacar:

- ✓ Brinda la posibilidad de integrarse con los principales Entornos de Desarrollo Integrado (IDEs) y que es multiplataforma.
- ✓ Funciona en los sistemas operativos más importantes como Linux, Windows y Mac OS X.
- ✓ Está diseñada para una amplia gama de usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocio, arquitectos y desarrolladores.
- ✓ Está orientada a la creación de diseños utilizando el paradigma de programación orientada a objetos.
- ✓ Garantiza la disponibilidad de múltiples versiones de acuerdo a la necesidad.
- ✓ Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo.
- ✓ Es muy útil para la generación de código fuente en PHP.

Aunque Visual Paradigm es software propietario esta es la herramienta usada en la universidad para modelar y documentar los productos de software que se desarrollan ya que se cuenta con la licencia que aprueba su utilización. Por tanto se decide utilizar el Visual Pradigm en su versión 6.0 como herramienta CASE par el modelado del sitio web. (17)

1.4.2. Postgres

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. (18)

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Este potente gestor de base de datos implementa el estándar SQL92/SQL99 y soporta distintos tipos de datos, al mismo tiempo que brinda soporte para los tipos de datos base soporta además datos de tipo fecha, monetarios, elementos gráficos y cadenas de bits, permite la creación de tipos propios e incorpora una estructura de datos array.

Características:

- ✓ Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- ✓ Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para números de precisión arbitraria, texto de largo ilimitado, figuras geométricas, direcciones IP (IPv4 e Ipv6), bloques de direcciones estilo CIDR, direcciones MAC y arrays.
- ✓ Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.
- ✓ Claves ajenas: también denominadas llaves ajenas o claves foráneas (*foreign keys*).
- ✓ Disparadores (*triggers*): Un disparador o *trigger* se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.
- ✓ Integridad transaccional.

- ✓ Herencia de tablas.
- ✓ Tipos de datos y operaciones geométricas.
- ✓ Soporte para transacciones distribuidas. Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.
- ✓ Soporta el uso de índices, vistas y procedimientos almacenados en múltiples lenguajes.
- ✓ Tiene interfaces de programación nativos para C/ C++, Java, Perl, Python, Ruby, Tcl, ODBC, entre otros.
- ✓ Es multiplataforma, funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX y Windows.

Por lo cual se decide utilizar este potente Sistema de gestión de base de datos para soportar la información de sitio web dada la funcionalidades y características de este SGBD.

1.4.3. Servidor de aplicaciones Apache

Es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo.

Los módulos del Apache se pueden clasificar en tres categorías:

- ✓ Módulos Base: Módulo con las funciones básicas del Apache.
- ✓ Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- ✓ Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un

módulo, de forma que no es necesario volver a instalar el software. Por todo lo antes expuesto se decide utilizar Apache como servidor de aplicación en su versión 2.0.59.

Las principales ventajas que nos brinda este servidor son:

- ✓ Corre en una gran multitud de sistemas operativos por lo que se considera multiplataforma lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código abierto.
- ✓ Es altamente configurable y de diseño modular. Es sencillo ampliar las capacidades de este servidor.
- ✓ Trabaja con Perl, PHP y otros lenguajes de script.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Por su gran utilidad y su versatilidad se decide utilizar este servidor web para montar el sitio.

1.5. Conclusiones

En la primera parte de este capítulo se hizo referencia a conceptos relacionados con el objeto de estudio definido en la introducción del documento. En la segunda parte, se desarrolla el estudio de algunas de las tecnologías actuales que serán empleadas durante el desarrollo del software, mencionando características y ventajas que las mismas poseen; la decisión de utilizarlas fue tomada siguiendo fundamentalmente la política de uso de herramientas con soporte multiplataforma y licencias de utilización libre las cuales fueron escogidas por nuestros clientes.

Capítulo 2 Análisis y Diseño del Sistema

En el presente capítulo, se realizará todo el modelamiento de la información del sitio de capacitación del proyecto Calidad de FORTES (GCF), este consiste en la definición de los objetivos del sitio, la organización de los contenidos y los servicios que se brindarán, especificándose los requisitos funcionales y no funcionales que debe cumplir así como la estructura y diseño del sitio. También se hará alusión a las fases de exploración y planificación, las dos primeras de la metodología de desarrollo XP. El objetivo de estas es conocer el alcance del producto a desarrollar así como estimar los tiempos de entrega de cada versión. En este capítulo se exponen además los artefactos que se generan a partir de los requerimientos expuestos por el cliente.

2.1. Definición de los objetivos del sitio.

La primera tarea que debe realizarse en la producción de un producto electrónico es la definición de los objetivos. Entre las informaciones deben estar: los objetivos que se persiguen con el producto a realizar, definición de la problemática que genera la creación del producto como solución, la misión de la institución creadora, los usuarios finales que va a tener el producto y las expectativas que se tienen con el producto.

2.2. Objetivos del Módulo de Capacitación

El entrenamiento en un proyecto productivo es fundamental, contribuye a la superación del personal y mejora los servicios que se brindan. También se controla la participación de los miembros del Grupo de Calidad de FORTES (GCF) y se mejora el control de la evaluación de los mismos.

1. Automatizar el proceso de entrenamiento del colectivo de trabajo del proyecto GCF.
2. Mejorar el control sobre las tareas realizadas y a realizar por los miembros del GCF.
3. Brindar mejor información sobre los entrenamientos a realizar y los resultados de los mismos.

2.3. Análisis de sitios similares

El análisis de sitios similares consiste en la comparación de productos similares al que se va a desarrollar, constituyendo una herramienta muy útil. El resultado obtenido sirve de guía en el seguimiento de aquellos que muestran las mejores prácticas adaptables a las necesidades del producto en cuanto a tipos de contenido, estructuras, niveles de información y otros elementos.

Para el modelamiento de información se identificaron Sitios o Portales Web que responden a características similares a las necesidades del GCF para gestionar su información.

- ✓ Entorno Virtual de Aprendizaje (eva.uci.cu)
- ✓ Akademos (akademos.uci.cu)
- ✓ Masterclass.com.mx
- ✓ Desarrollodeweb.com.ar

2.4. Exploración

La metodología de desarrollo *Extreme Programming* comienza con su fase de exploración. Es la etapa del proceso de desarrollo de software que propone XP para comenzar la construcción de un producto. Una vez que los clientes entregan su propuesta al equipo de trabajo, comienza el análisis en grupo. Un aspecto clave en el esclarecimiento de las dudas sobre los procesos a automatizar es que, desde el mismo inicio del proyecto, un miembro del equipo del trabajo del cliente, se adiciona al de desarrolladores.

Se define el alcance general del proyecto, se exploran las diferentes formas de resolver problemas concretos de implementación que puedan presentarse y sobre todo se definen historias de usuario, que son la forma de definir los requisitos del sistema a implementar. Los programadores estiman los tiempos de desarrollo sobre la base de esta información la cual podría variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado.

2.5. Requerimientos funcionales del sistema

Una vez conocido el objetivo del trabajo y los principales conceptos se puede analizar que debe hacer el sistema para que se cumplan los objetivos planteados en 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 funciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema.

RF1- Gestionar programa de la Asignatura

RF1.1- Agregar nuevo programa de la asignatura.

RF1.2- Modificar programa de la asignatura. (Depende de las necesidades del proyecto)

RF1.3- Eliminar programa de la asignatura.

RF1.4- Mostrar el programa de la asignatura.

RF2- Consultar estructura del programa de la asignatura.

RF3- Gestionar profesor (nombre, apellidos, facultad, número de solapín).

RF3.1- Agregar un nuevo profesor.

RF3.2- Eliminar profesor.

RF3.3- Modificar profesor.

RF3.4- Mostrar profesor.

RF4- Conocer datos de un profesor.

RF5- Gestionar estudiante (nombre, apellidos, año, facultad, número de solapín, usuario).

RF5.1- Agregar un nuevo estudiante.

RF5.2- Eliminar estudiante.

RF5.3- Modificar estudiante.

RF5.4- Mostrar estudiante.

RF6- Gestionar evaluaciones de los estudiantes en los cursos.

RF6.1- Agregar evaluación del estudiante.

RF6.2- Modificar evaluación del estudiante.

RF6.3- Eliminar evaluación del estudiante.

RF6.4- Mostrar evaluación del estudiante.

RF7- Gestionar registro de asistencia.

RF7.1- Agregar asistencia al estudiante.

RF7.2- Modificar asistencia al estudiante.

RF7.3- Mostrar asistencia del estudiante.

RF8- Gestionar curso.

RF8.1- Agregar asignatura al curso.

RF8.2- Agregar profesor al curso.

RF8.3- Cambiar profesor del curso.

RF8.3- Mostrar curso.

RF9- Consultar datos de los cursos.

RF10- Gestionar planificación de horario de cursos.

RF10.1- Insertar curso al horario.

RF10.2- Modificar curso en el horario.

RF10.3- Eliminar curso del horario.

RF10.4- Mostrar horario.

2.6. Requerimientos no funcionales:

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; por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado.

Existen múltiples categorías para clasificar los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

➤ Restricciones en el diseño y la implementación.

Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Ejemplos de ellas son:

1. Estándares requeridos.
2. Lenguajes de programación a ser usados para la implementación.
3. Uso obligatorio de ciertas herramientas de desarrollo.
4. Restricciones en la arquitectura o el diseño.

➤ Requerimientos de apariencia o interfaz externa.

Este tipo de requerimiento describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto.

➤ Requerimientos de Seguridad.

Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes, estos son:

1. Confidencialidad: La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
2. Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
3. Disponibilidad: Los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

Dentro de los requerimientos no funcionales del sistema propuesto, se encuentran:

1. Usabilidad.

- El sistema podrá ser usado por todas aquellas personas de los proyectos de la facultad 4, fundamentalmente integrantes del grupo de calidad de la facultad y calidad UCI.
- La aplicación deberá poseer una interfaz y navegación asequibles, y funcionales tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.
- Mantener informado al usuario del resultado de las operaciones.

2. Rendimiento.

- Operaciones normales en dos hilos.
- Respuestas rápidas en segundos al usuario.
- Debe estar disponible las 24 horas del día.

3. Portabilidad.

- El sistema debe ser Multiplataforma. Haciendo énfasis en el sistema operativo Linux.

4. Seguridad.

- Se permitirá la autenticación, la cual será una contraseña de acceso para los usuarios autorizados por el administrador. El sistema informático debe garantizar que la información sea vista únicamente por quien tenga permiso para esto. El sistema debe permitir solo el acceso a cada uno de sus servicios al personal requerido para esto.
- El sistema informará quien es el usuario correspondiente con la sección que esté activa.
- Confiabilidad: Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.
- Disponibilidad: El sistema debe permitirle al usuario conectarse desde cualquier PC de la red.
- Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción de datos.

5. Apariencia o interfaz externa.

- El sistema debe poseer una interfaz Web amigable y sencilla, fácil para la interacción del usuario. Con colores en su interfaz que distingan a la facultad, sin saturación de colores ni imágenes.
- El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de esta, facilitando el uso de la aplicación.
- El sistema implementará la ejecución de acciones de una manera rápida, minimizando los pasos a ejecutar en cada proceso.
- En la interfaz de presentación se mostrará el nombre del producto en este caso: “Capacitación de Calidad”.

6. Soporte.

- La aplicación tendrá un periodo de tiempo en prueba para ver su funcionamiento y que cumpla con lo requerido dándole mantenimiento para luego brindar servicio de instalación.
- Documentación necesaria para el uso o desarrollo de las funcionalidades de la herramienta.

7. Interfaz.

- Las interfaces deben ser lo más entendible posible.
- Todas las interfaces tienen que tener una forma de cerrarse o retornar al paso anterior.
- Ninguna interfaz puede ir cargada de muchos colores evitando la desconcentración del usuario.

8. Requisitos Legales, de Derecho de Autor y otros.

- Es un deber proteger la información por parte de las personas que tienen derecho de administración u otros que pongan en peligro la integridad y seguridad del sistema.

9. Confiabilidad.

- Está prohibida la diseminación pública de la información por cualquier usuario a cualquier nivel.

10. Políticos-culturales.

- Se deben poner las palabras lo más acorde a su entendimiento.

2.7. Historias de Usuario

Del inglés *User Stories* (en adelante HU), las historias de usuarios son la forma en que se especifican en XP los requisitos del sistema, las mismas no deben ser descritas en más de tres líneas e idealmente es el cliente quien las redacta y prioriza por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Luego de esto se le suma un tiempo estimado de desarrollo que lo define el propio equipo del proyecto. En la claridad de su descripción radica el éxito del proyecto motivo por el cual el contenido que ellas abarcan debe ser concreto y sencillo.

Tabla 1: Historias de usuario

No:1	Nombre: Gestionar Programa de la asignatura
Usuarios	Administrador
Prioridad: alta	Complejidad: alta
Estimación: 7d	Iteración: 1
Descripción:	El administrador podrá agregar, modificar, eliminar, exportar o mostrar el programa de las asignaturas.
Observaciones:	Se le da cumplimiento al requisito 1
No: 2	Nombre: Consultar Programa de la asignatura
Usuarios	Usuarios Autenticados.
Prioridad: baja	Complejidad: baja
Estimación: 1d	Iteración: 1
Descripción:	El usuario podrá visualizar el programa de la asignatura.
Observaciones:	Se le da cumplimiento al requisito 2
No: 3	Nombre: Gestionar Profesor
Usuarios	Administrador, Editor
Prioridad: alta	Complejidad: alta
Estimación: 3d	Iteración: 1

Descripción:	El administrador podrá agregar, elimina, modificar o mostrar el profesor que imparte los cursos.
Observaciones:	Se le da cumplimiento al requisito 6
No: 4	Nombre: Mostrar datos de un profesor.
Usuarios	Usuarios Autenticados.
Prioridad: baja	Complejidad: baja
Estimación: 1d	Iteración: 2
Descripción:	Los usuarios podrán ver los datos del profesor que imparte el curso.
Observaciones:	Se le da cumplimiento al requisito 7
No: 5	Nombre: Gestionar estudiantes.
Usuarios	Administrador.
Prioridad: alta	Complejidad: alta
Estimación: 3d	Iteración: 2
Descripción:	El administrador podrá agregar, eliminar, modificar o mostrar los datos de los estudiantes en los cursos.
Observaciones:	Se le da cumplimiento al requisito 8
No: 6	Nombre: Gestionar Evaluación de los estudiantes.
Usuarios	Administrador, Editor
Prioridad: alta	Complejidad: alta
Estimación: 3d	Iteración: 2
Descripción:	El administrador podrá agregar, eliminar, modificar o mostrar las evaluaciones de los estudiantes en los cursos.
Observaciones:	Se le da cumplimiento al requisito 9
No: 7	Nombre: Gestionar Registro de Asistencia
Usuarios	Administrador, Editor

Prioridad: alta	Complejidad: alta
Estimación: 3d	Iteración: 2
Descripción:	El administrador, Editor del curso podrá agregar, modificar, eliminar o mostrar la asistencia de los estudiantes.
Observaciones:	Se le da cumplimiento al requisito 11
No: 8	Nombre: Gestionar Curso
Usuarios	Administrador, Editor
Prioridad: alta	Complejidad: alta
Estimación: 7d	Iteración: 3
Descripción:	El administrador, Editor del curso podrá agregar estudiantes, profesores y asignaturas al curso, cancelar la matrícula del curso, cambiar el profesor del curso, mostrar la matrícula del curso, imprimir el registro de asistencia y evaluaciones y mostrar el curso.
Observaciones:	Se le da cumplimiento al requisito 13
No: 9	Nombre: Consultar datos de los cursos
Usuarios	Administrador, Editor
Prioridad: media	Complejidad: media
Estimación: 2d	Iteración: 3
Descripción:	El administrador, Editor del curso podrá consultar los datos de los cursos.
Observaciones:	Se le da cumplimiento al requisito 14
No: 10	Nombre: Gestionar planificación de horario de cursos.
Usuarios	Administrador, Editor
Prioridad: alta	Complejidad: alta
Estimación: 3d	Iteración: 3
Descripción:	El administrador, Editor del curso podrá insertar, modificar,

	eliminar o mostrar los cursos en el horario.
Observaciones:	Se le da cumplimiento al requisito 15

2.8. Planificación

XP plantea la planificación como un permanente diálogo entre el cliente y los desarrolladores del proyecto, en la que los primeros decidirán el alcance, qué es lo realmente necesario del proyecto, la prioridad con qué debe ser desarrollado en primer lugar. En cuanto a los desarrolladores, son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, de organizar la cultura de trabajo y, finalmente, de realizar la planificación detallada dentro de cada versión.

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar todas las historias de usuario siendo uno de los artefactos generados en esta fase. Una vez terminado esto, se procede a organizarlas en las iteraciones correspondientes, en dependencia de la prioridad especificada por el cliente y del tiempo de desarrollo de cada una.

2.8.1. Estimación de esfuerzos por historias de usuario.

Tabla 2: Estimación de esfuerzos por historias de usuario.

Historias de Usuario	Estimación (días)
Gestionar Programa de la asignatura	7
Consultar Programa de la asignatura	1
Gestionar Profesor	3
Mostrar datos de un profesor.	1
Gestionar estudiantes.	3
Gestionar Evaluación de los estudiantes.	3
Gestionar Registro de Asistencia	3
Gestionar Curso	7
Consultar datos de los cursos	2

Gestionar planificación de horario de cursos	3
Consultar datos del horario	2

2.9. Plan de iteraciones

Después de haber identificado las historias de usuarios y realizado una previa estimación de esfuerzos se procede a la planificación de la etapa de implementación del sistema. En este plan se especifica la prioridad con que se irán implementando las historias de usuarios organizadas por iteraciones así como posibles fechas de liberación. Este paso es de suma importancia en el ciclo de desarrollo de XP. Las funcionalidades son planificadas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración.

Una iteración no es más que un mini-proyecto en el que se obtiene un resultado en software con un valor para el cliente. Claro está que no se tendría la versión completa del producto, ya que este concluye al finalizar la última iteración.

2.9.1. Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario de mayor prioridad. Al finalizar se contará con las funcionalidades descritas en las historias de usuario asignadas.

2.9.2. Iteración 2

El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad alta, además, de otras con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las historias de usuario asignadas.

2.9.3. Iteración 3

En esta iteración serán implementadas las funcionalidades de prioridad baja. Estas funciones tienen el propósito de brindar al cliente comodidad en la gestión de otras tareas asociadas a las de alta prioridad, entre ellas tenemos: mostrar el perfil del usuario con sus datos personales. Estas funciones están descritas en las historias de usuario asignadas.

2.9.4. Plan de duración de las iteraciones

Siguiendo el desarrollo de la metodología XP se crea el plan de duración de las iteraciones. En este plan se especifica más detalladamente el orden de desarrollo de las historias de usuarios dentro de cada iteración así como la estimación completa de dicha iteración.

Tabla 4: Plan de duración de las iteraciones.

Iteraciones	Orden de las Historias de Usuarios a implementar.	Duración de las iteraciones.
-------------	---	------------------------------

Iteración 1	Gestionar Programa de la asignatura Consultar Programa de la asignatura Gestionar Profesor Mostrar datos de un profesor.	3 semanas
Iteración 2	Gestionar estudiantes. Gestionar Evaluación de los estudiantes. Gestionar Registro de Asistencia	2 semanas
Iteración 3	Gestionar Curso Consultar datos de los cursos Gestionar planificación de horario de cursos Consultar datos del horario	2 semanas

2.10. Plan de entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (clientes, gerentes, desarrolladores). Este plan se acopla las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la fase de implementación.

2.10.1. Plan de duración de las entregas

Historia de usuario	Fin 1ra iteración 1ra semana de abril	Fin 2da iteración 3ra semana de abril	Fin 3ra iteración 2 semana de mayo
Gestionar Programa de la asignatura	V 0.1	finalizado	finalizado
Consultar Programa de la asignatura	V 0.1	finalizado	finalizado

Gestionar Profesor	V 0.1	finalizado	finalizado
Mostrar datos de un profesor.	-	V 0.1	finalizado
Gestionar estudiantes.	-	V 0.1	finalizado
Gestionar Evaluación de los estudiantes.	-	V 0.1	finalizado
Gestionar Registro de Asistencia	-	V 0.1	finalizado
Gestionar Curso	-	-	V 0.1
Consultar datos de los cursos	-	-	V 0.1
Gestionar planificación de horario de cursos	-	-	V 0.1
Consultar datos del horario	-	-	V 0.1

2.11. Conclusiones

Durante el desarrollo del capítulo se expusieron los artefactos generados en las fases de exploración y planificación de la metodología XP. Asumiendo de esta forma una implementación por etapas correctamente descrita y detallada anteriormente. Durante esta fase aparecen factores primordiales para el éxito del producto como el diálogo entre clientes y desarrolladores los cuales exponen puntos clave en la decisión de si se va a continuar o no con el proyecto, argumentando entre lo deseable y lo posible.

Capítulo 3 Implementación y Pruebas

3.1. Introducción

La Metodología XP plantea que la implementación de un software 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. En el presente capítulo se detallan las tres iteraciones realizadas 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.

3.2. 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). No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

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.

3.2.1. 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. Un 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, éstos tienen disímiles propósito 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: Estudiantes
- b. Sub paquete 1.1.2: Cursos
- c. Sub paquete 1.1.3: Profesores
- d. Sub paquete 1.1.4. Horario
- e. Sub paquete 1.1.5. Registro
- f. Sub paquete 1.1.6. Evaluación

Casi todos los módulos tienen relación con algunos componentes, que constituyen capas intermedias; como pueden ser componentes de acceso a la base de datos, de lógica de negocio, etc.

3.2.2. Paquete 1: Drupal

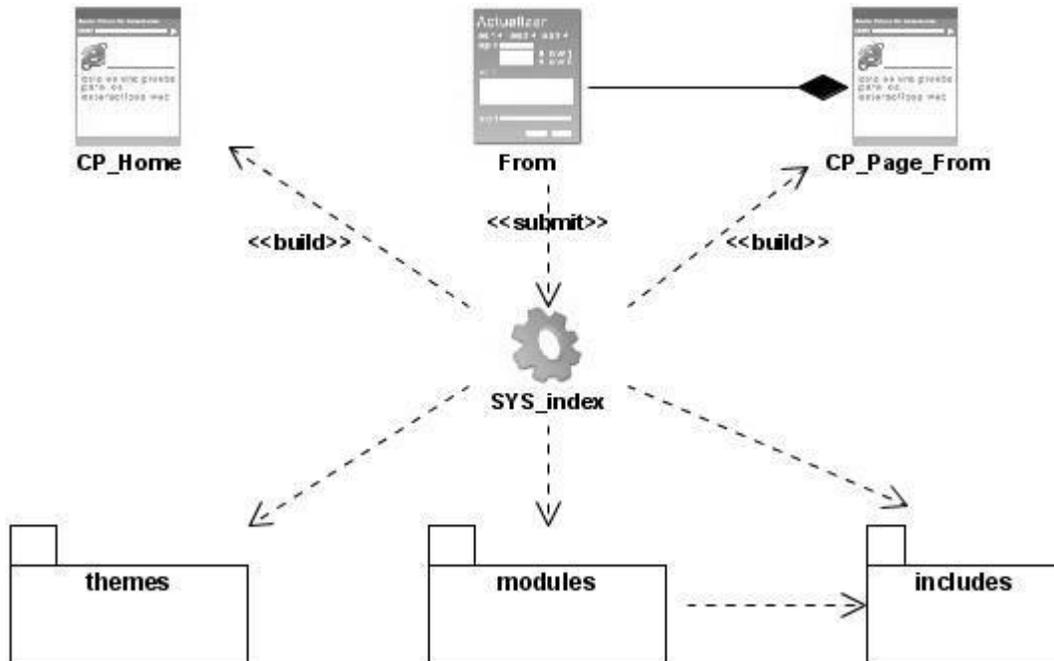
El Paquete 1: Drupal, consiste en la distribución del CMS Drupal. Básicamente contiene un paquete *Themes*, donde se encuentran los mecanismos que soportan el sistema de plantillas, de modo que cuando usted desee cambiar el diseño de la interfaz que presentará el sistema, sólo tiene que definir una nueva 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, ésto 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 modelará cada módulo por separado, 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

Diagrama de clases WEB: Drupal

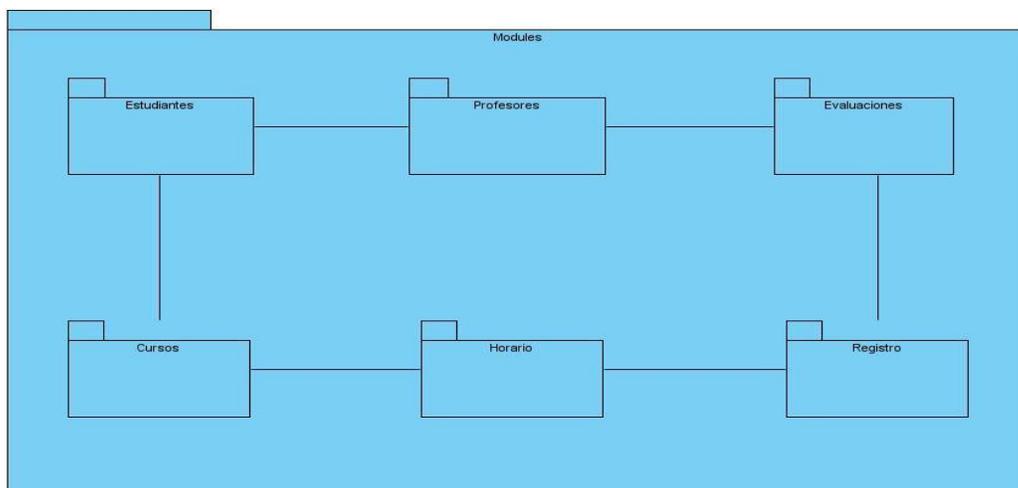


3.2.3. Sub paquete 1.1: Modules

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 sólo los desarrollados.

Figura 2 Diagrama de Clases del Diseño: Sub Paquete Modules

Diagrama de clases WEB: 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.

3.2.4. Tarjetas CRC

Módulo Profesores	
Funcionalidades	Colaboraciones
Crear Profesor	Menu
Ver Profesor	Node
Modificar Profesor	Block
Eliminar Profesor	User
	Help

Módulo Estudiantes	
Funcionalidades	Colaboraciones
Crear Estudiante	Menu
Ver Estudiante	Node
Modificar Estudiante	Block
Eliminar Estudiante	User
	Help

Módulo Cursos	
Funcionalidades	Colaboraciones
Crear Curso	Menu
Ver Curso	Node
Modificar Curso	Block
Eliminar Curso	User
	Help

Módulo Evaluación	
Funcionalidades	Colaboraciones
Agregar Evaluación Ver Evaluación	Menu Node Block User Help

Módulo Horario	
Funcionalidades	Colaboraciones
Agregar Horario Ver Horario	Menu Node Block User Help

Módulo Registro	
Funcionalidades	Colaboraciones
Crear Registro Mostrar Registro	Menu Node Block User Help

3.3. 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 tres 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.

3.3.1. Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad, que son sin dudas el esqueleto del negocio anteriormente descrito, 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.

Módulos abordados en la primera iteración

Módulo/Componente	Historia de Usuario	Tiempo de Implementación	
		Estimación	Real
Profesores	Insertar Profesor	2	2
	Ver Profesor	1	1
Estudiantes	Insertar Estudiante	2	2
	Ver Estudiante	1	1
Cursos	Crear Curso	2	2
	Ver Curso	1	1

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

Tareas de Ingeniería Iteración 1

Tarea	
No. de tarea: 1	No. de HU: 13
Nombre: Creación del Módulo Curso	
Tipo de tarea: Desarrollo	Puntos de estimación: 0.50

Fecha Inicio: 12/01/2011	Fecha Fin: 15/01/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades necesarias para que los usuarios puedan visualizar los contenidos creados.	

Tarea	
No. de tarea: 2	No. de HU: 13
Nombre: Manejo de persistencia en la BD.	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 16/01/2011	Fecha Fin: 16/01/2011
Programador responsable: Julio Cruz Castro	
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.	

Tarea	
No. de tarea: 1	No. de HU: 14
Nombre: Mostrar Curso	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 17/01/2011	Fecha Fin: 19/01/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que los usuarios visualicen las funciones	

implementadas.

Tarea	
No. de tarea: 1	No. de HU: 8
Nombre: Insertar Estudiante	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 17/01/2011	Fecha Fin: 19/01/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o el editor realicen la funcionalidad descrita.	

Tarea	
No. de tarea: 1	No. de HU: 10
Nombre: Ver Estudiante	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 20/01/2011	Fecha Fin: 22/01/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el usuario autenticado visualice la funcionalidad implementada.	

Tarea	
No. de tarea: 1	No. de HU: 6
Nombre: Insertar Profesor	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 23/01/2011	Fecha Fin: 30/01/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor inserte los datos que requiere la funcionalidad.	

Tarea	
No. de tarea: 1	No. de HU: 7
Nombre: Ver Profesor	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 31/01/2011	Fecha Fin: 10/02/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el usuario autenticado visualice la funcionalidad implementada.	

3.3.2. Iteración 2

Durante el transcurso de la presente iteración se concluyó la implementación y configuración de las funcionalidades del módulo curso, se continúa en el perfeccionamiento del módulo bibliografía y se comienza con la elaboración del módulo matrícula.

Módulos abordados en la segunda iteración

Módulo/Componente	Historia de Usuario	Tiempo de Implementación

		Estimación	Real
Profesores	Eliminar Profesor	2	2
	Modificar Profesor	1	1
Estudiantes	Modificar Estudiante	2	2
	Eliminar Estudiante	1	1
Cursos	Modificar Curso	2	2
	Eliminar Curso	1	1

Se continúa con la elaboración de las tareas correspondientes a cada uno de los módulos tratados en la iteración:

Tarea	
No. de tarea: 1	No. de HU: 7
Nombre: Eliminar Profesor	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 10/02/2011	Fecha Fin: 14/02/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor eliminen los datos correspondientes.	

Tarea	
No. de tarea: 2	No. de HU: 7
Nombre: Modificar Profesor	

Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 15/02/2011	Fecha Fin: 20/02/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor modifiquen los datos correspondientes.	

Tarea	
No. de tarea: 1	No. de HU: 10
Nombre: Modificar Estudiante	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 20/02/2011	Fecha Fin: 25/02/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor modifiquen los datos correspondientes.	

Tarea	
No. de tarea: 2	No. de HU: 10
Nombre: Eliminar Estudiante	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 26/02/2011	Fecha Fin: 3/03/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor eliminen los datos	

correspondientes.

Tarea	
No. de tarea: 1	No. de HU: 14
Nombre: Modificar Curso	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 3/03/2011	Fecha Fin: 10/03/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor modifiquen los datos correspondientes.	

Tarea	
No. de tarea: 2	No. de HU: 14
Nombre: Eliminar Curso	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 11/03/2011	Fecha Fin: 18/03/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor eliminen los datos correspondientes.	

3.3.3. Iteración 3

En esta iteración se implementaron las funcionalidades de baja prioridad. Éstas tienen la finalidad de propiciar un ambiente más afable y cómodo al cliente. Al finalizar se cuenta con un producto listo para poner en funcionamiento.

Módulos abordados en la tercera iteración

Módulo/Componente	Historia de Usuario	Tiempo de Implementación	
		Estimación	Real
Horario	Crear Horario	2	2
	Ver Horario	1	1
	Modificar Horario	2	2
	Eliminar Horario	1	1

Tarea	
No. de tarea: 1	No. de HU: 15
Nombre: Crear Horario	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 18/03/2011	Fecha Fin: 30/03/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor creen los datos correspondientes.	

Tarea	
No. de tarea: 1	No. de HU: 16
Nombre: Ver Horario	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 31/03/2011	Fecha Fin: 6/04/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que los usuarios autenticados visualicen los datos	

correspondientes.

Tarea	
No. de tarea: 2	No. de HU: 16
Nombre: Modificar Horario	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 7/04/2011	Fecha Fin: 15/04/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor modifiquen los datos correspondientes.	

Tarea	
No. de tarea: 3	No. de HU: 16
Nombre: Eliminar Horario	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Fecha Inicio: 15/04/2011	Fecha Fin: 25/04/2011
Programador responsable: Julio Cruz Castro	
Descripción: Se implementarán las funcionalidades para que el administrador o editor modifiquen los datos correspondientes.	

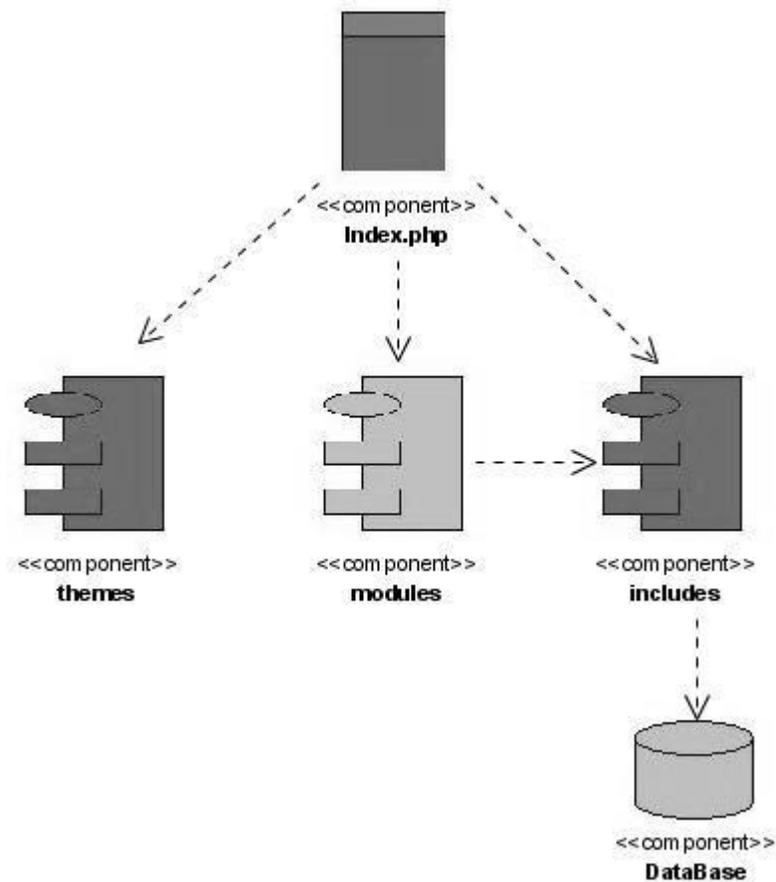
3.3.4. Diagrama de Componentes

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 es que puede servir 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.

Diagrama de componentes del Sistema



3.3.5. 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 la aplicación se encuentra hospedada en un servidor Web y la misma se comunica con un sistema de gestión de base de datos (PostgreSQL).

Diagrama de Componentes del sistema

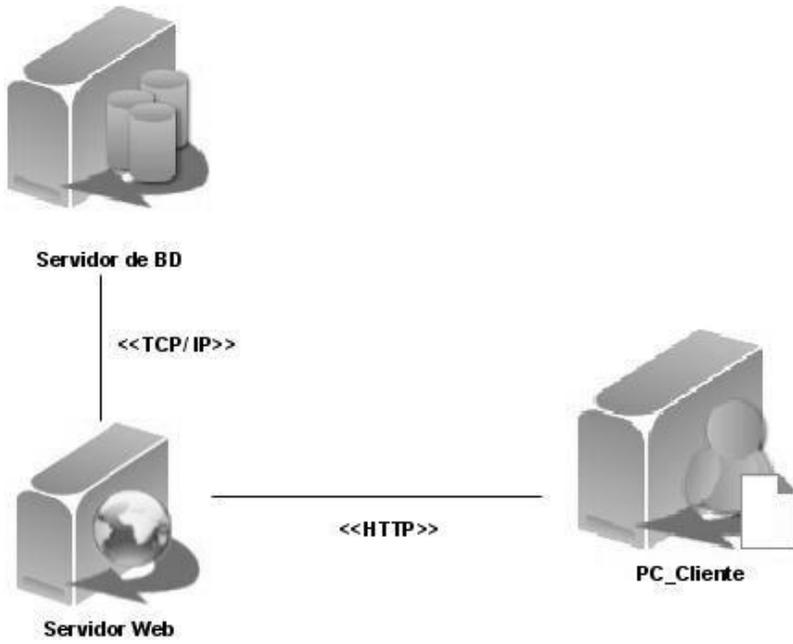
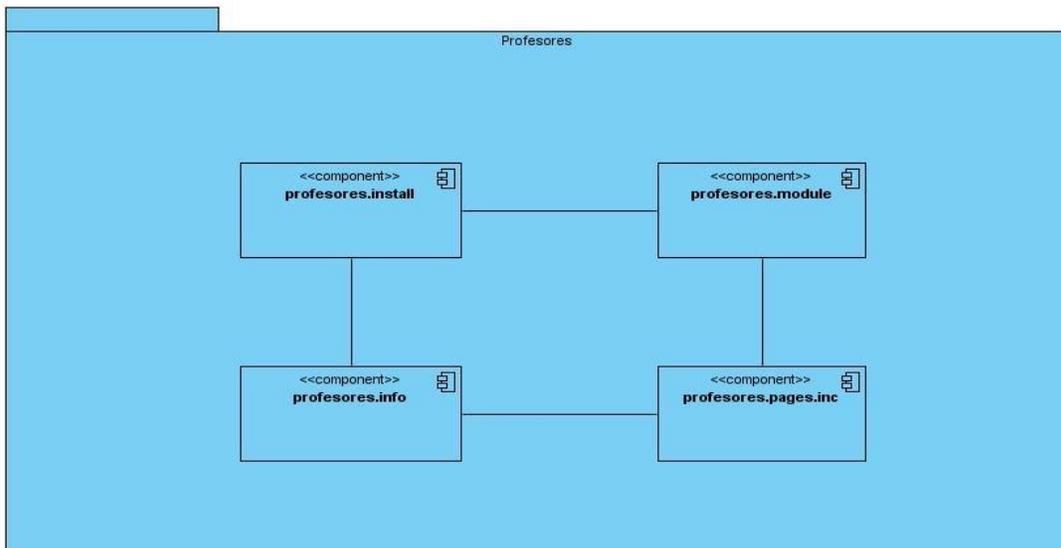
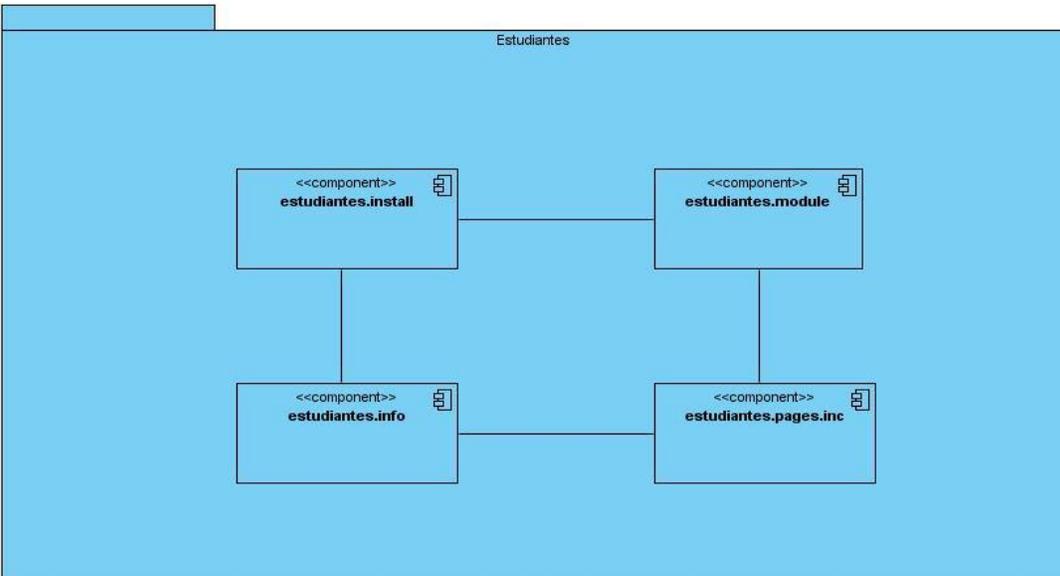
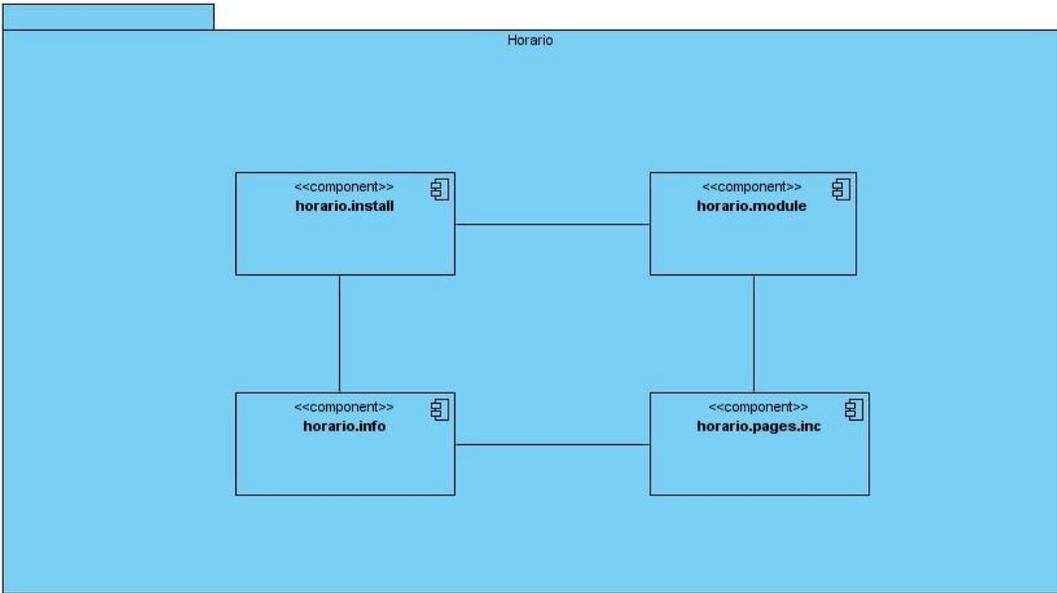
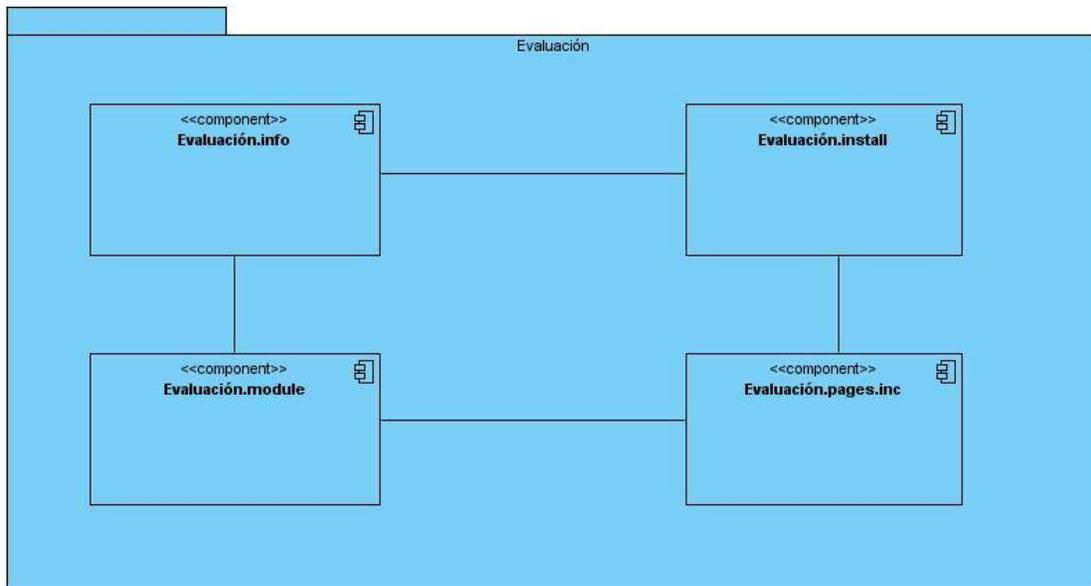


Diagrama de Componentes de los módulos







3.4. Pruebas

Uno de los pilares fundamentales de XP es el proceso de pruebas, 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. 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. (1)

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. (1)

3.4.1. Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. 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. Una HU no se considera completa hasta que no ha pasado por sus pruebas de aceptación. (1)

Pruebas de aceptación realizadas

Casos de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Crear Curso	
Descripción: Prueba la funcionalidad de crear un curso.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta crear recursos.	
Resultado esperado: Que la aplicación logre crear un curso.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Ver Curso	
Descripción: Prueba la funcionalidad de ver un curso.	
Condiciones de Ejecución: El usuario debe estar autenticado	
Entrada/Pasos de Ejecución: Se intenta visualizar recursos.	
Resultado esperado: Que la aplicación logre mostrar un curso.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Modificar Curso	
Descripción: Prueba la funcionalidad de ver un curso.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta modificar recursos.	
Resultado esperado: Que la aplicación logre modificar un curso.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU9_P3	Historia de Usuario: 9
Nombre: Eliminar Curso	
Descripción: Prueba la funcionalidad de eliminar un curso.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta eliminar recursos.	
Resultado esperado: Que la aplicación logre eliminar un curso.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Crear Profesor	
Descripción: Prueba la funcionalidad de crear profesor.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta agregar recursos.	
Resultado esperado: Que la aplicación logre agregar datos de un profesor.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Modificar Profesor	
Descripción: Prueba la funcionalidad de Modificar profesor.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta modificar recursos.	
Resultado esperado: Que la aplicación logre modificar datos de un profesor.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Eliminar Profesor	
Descripción: Prueba la funcionalidad de Eliminar profesor.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta eliminar recursos.	
Resultado esperado: Que la aplicación logre eliminar datos de un profesor.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Ver Profesor	
Descripción: Prueba la funcionalidad de Ver profesor.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de Ejecución: Se intenta visualizar recursos.	
Resultado esperado: Que la aplicación logre visualizar los datos de un profesor.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Crear Estudiante	
Descripción: Prueba la funcionalidad de Crear Estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta agregar recursos.	
Resultado esperado: Que la aplicación logre agregar datos de un estudiante.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Modificar Estudiante	
Descripción: Prueba la funcionalidad de Modificar Estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta modificar recursos.	
Resultado esperado: Que la aplicación logre modificar datos de un estudiante.	
Evaluación de la prueba:	

Prueba Satisfactoria.

Casos de Prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Eliminar Estudiante	
Descripción: Prueba la funcionalidad de Eliminar Estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta Eliminar recursos.	
Resultado esperado: Que la aplicación logre Eliminar datos de un estudiante.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU5_P4	Historia de Usuario: 5
Nombre: Ver Estudiante	
Descripción: Prueba la funcionalidad de Ver Estudiante.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de Ejecución: Se intenta visualizar recursos.	
Resultado esperado: Que la aplicación logre visualizar datos de un estudiante.	
Evaluación de la prueba:	

Prueba Satisfactoria.

Casos de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Crear Horario	
Descripción: Prueba la funcionalidad de Crear Horario.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta crear recursos.	
Resultado esperado: Que la aplicación logre crear un horario.	
Evaluación de la prueba: Prueba Satisfactoria.	

Casos de Prueba de Aceptación	
Código: HU10_P2	Historia de Usuario: 10
Nombre: Modificar Horario	
Descripción: Prueba la funcionalidad de Modificar Horario.	
Condiciones de Ejecución: El usuario debe estar autenticado. El usuario debe tener el rol de administrador o editor.	
Entrada/Pasos de Ejecución: Se intenta modificar recursos.	
Resultado esperado: Que la aplicación logre modificar un horario.	

Evaluación de la prueba:

Prueba Satisfactoria.

Casos de Prueba de Aceptación**Código:** HU10_P3**Historia de Usuario:** 10**Nombre:** Eliminar Horario**Descripción:** Prueba la funcionalidad de Eliminar Horario.**Condiciones de Ejecución:**

El usuario debe estar autenticado.

El usuario debe tener el rol de administrador o editor.

Entrada/Pasos de Ejecución:

Se intenta eliminar recursos.

Resultado esperado:

Que la aplicación logre eliminar un horario.

Evaluación de la prueba:

Prueba Satisfactoria.

Casos de Prueba de Aceptación**Código:** HU10_P4**Historia de Usuario:** 10**Nombre:** Ver Horario**Descripción:** Prueba la funcionalidad de Ver Horario.**Condiciones de Ejecución:**

El usuario debe estar autenticado.

Entrada/Pasos de Ejecución:

Se intenta visualizar recursos.

Resultado esperado:

Que la aplicación logre visualizar un horario.

Evaluación de la prueba:

Prueba Satisfactoria.

3.5. Resultado de las pruebas

Durante el desarrollo de las pruebas se obtuvo como resultado:

- Las funcionalidades propuestas se ejecutaban correctamente.
- Se detectaron 2 faltas de ortografía.
- Se detectaron 2 errores en validaciones de campos.
- Se comprobó que el sistema resiste varias conexiones simultáneas.
- Se probó que el sistema brinda una fácil navegación.
- Se comprobó que el sistema es multiplataforma.

3.6. Conclusiones

En este capítulo se construyó el modelo necesario para desarrollar el proceso de implementación del sistema. 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 el fin de este capítulo se da por terminada la propuesta que trae este trabajo.

Conclusiones

La gestión de información hoy en día se ha convertido en tarea de todos. El rápido avance de las tecnologías en el campo de la informática y su gran aceptación a nivel mundial hace que cada día se dé más importancia a la docencia de forma digital, así como al uso de las tecnologías de la informática en general, debido a la gran cantidad de usuarios que le han dado su preferencia a este sector.

El estudio realizado permitió tener un conocimiento de la situación actual, las herramientas y la metodología para el desarrollo de una solución informática basándose en las tendencias de los sistemas de gestión de cursos con contenidos informáticos. Se desarrolló una aplicación Web que permite la publicación de cursos por bloques y niveles sobre temas de calidad, brindando así al usuario un punto de acceso donde este puede obtener toda la información que necesite acerca de los cursos publicados y contará con la bibliografía adecuada para cumplir con sus expectativas. Además permite la constante actualización de todo su contenido. Un aspecto novedoso, es que se realizarán las diferentes funcionalidades como módulo Drupal, lo que permite que sea altamente configurable y adaptable en dependencia del centro donde se utilice el sistema.

El desarrollo del trabajo utilizando la metodología XP permitió documentar el mismo desde el comienzo, lo que facilitará su estudio por parte de futuros miembros del equipo de desarrollo u otros miembros del proyecto, permitiendo de esta forma, una comprensión más rápida y fácil de la concepción general del sistema. La versión del sistema obtenida, constituye una base para el trabajo del equipo, así como para futuras versiones del sistema.

Bibliografía

1. **Camisión, César.** www.mailxmail.com. [En línea] 16 de 12 de 2009. <http://www.mailxmail.com>.
2. **Press, Eduardo.** sht.com.ar. [En línea] [Citado el: 16 de 01 de 2011.] <http://www.sht.com.ar/>.
3. **Sanchez, Sonia Ymaris.** *Gestión de la capacitación para el desarrollo del talento.* 2005.
4. www.masadelante.com. [En línea] [Citado el: 16 de 01 de 2011.] <http://www.masadelante.com/>.
5. masterclass.com.mx. [En línea] [Citado el: 10 de 01 de 2011.] <http://www.masterclass.com.mx>.
6. es.pro169.org. [En línea] [Citado el: 16 de 01 de 2011.] <http://es.pro169.org>.
7. www.desarrollodeweb.com.ar. [En línea] [Citado el: 16 de 01 de 2011.] <http://www.desarrollodeweb.com.ar>.
8. **Cuerda, Xavier García.** mosaic.uoc.edu. [En línea] 29 de 11 de 2004. [Citado el: 16 de 01 de 2011.] <http://mosaic.uoc.edu/>.
9. drupal.org. [En línea] [Citado el: 8 de 12 de 2010.] <http://drupal.org/>.
10. joomlaspanish.org. [En línea] [Citado el: 5 de 12 de 2010.] <http://www.joomlaspanish.org/>.
11. php.net. [En línea] [Citado el: 10 de 01 de 2011.] <http://www.php.net>.
12. www.linuxcentro.net. [En línea] [Citado el: 10 de 01 de 2011.] <http://www.linuxcentro.net>.
13. **García, Yurien Ramos.** *“Propuesta de diseño de una aplicación Web para el Control de Trabajos de Diploma”.* Ciudad de la Habana : s.n., 2008.
14. **Escribano, Gerardo Fernández.** *Introducción a.* 2002.
15. www.informatizate.net. [En línea] [Citado el: 10 de 12 de 2010.] <http://www.informatizate.net>.
16. www.usmp.edu.pe/. [En línea] [Citado el: 12 de 12 de 2010.] <http://www.usmp.edu.pe/>.
17. www.visual-paradigm.com/. [En línea] [Citado el: 10 de 12 de 2010.] <http://www.visual-paradigm.com/>.
18. www.postgresql.org. [En línea] [Citado el: 12 de 12 de 2010.] <http://www.postgresql.org/>.
19. eternalfight.com. [En línea] [Citado el: 16 de 01 de 2011.] <http://www.eternalfight.com>.
20. **Acosta, Idalmis Alba y León Pradera, Edgar.** *Sistema de Gestión de la Información en el Proceso de Formación Posgraduada en Salud Módulo de Capacitación y Superación Profesional.* Ciudad de la Habana : s.n., 2008.
21. **Veranes, Maité Sosa y González Guerra, Vladimir.** *Sistema de Gestión de Tesis Facultad 2.* Ciudad de la Habana : s.n., 2009.