

Universidad de las Ciencias Informáticas
Facultad 1



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

Título: Intranet de la Facultad 1
Implementación del Módulo Producción

Autor: Pedro Luís Loredo Sánchez

Tutores: Ing. Marianny Hernández Batista
Lic. Yoemny González Almaguer

*Ciudad de la Habana,
2009 "Año del 50 Aniversario del Triunfo de la Revolución"*

Declaración De Autoría

Declaración de Autoría _____

Declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ___ días del mes de ____ del año_____.

Pedro Luís Loredó Sánchez
Autor

Ing. Marianny Hernández Batista
Tutor

Lic. Yoemny González Almaquer
Tutor

Resumen

El presente trabajo pretende mejorar todo lo referente a los procesos de producción de la facultad 1 de la Universidad de las Ciencias Informáticas (UCI). Teniendo en cuenta que la forma en que se realizan actualmente dificulta la correcta gestión de los recursos materiales, humanos y de información en sentido general, haciéndose casi imposible la actualización inmediata de la información. Es por ello, que se decide como propuesta de solución la implementación del módulo de Producción de la Intranet de la facultad 1, el cual formará parte de un conjunto de módulos que permitirá que dicha intranet cuente con las funcionalidades necesarias para proveer a la facultad de una herramienta que mejore en gran medida la gestión de sus procesos.

Para su desarrollo se realizó un amplio estudio de los procesos referentes a la actividad productiva, con el objetivo de garantizar un mejor funcionamiento, identificar deficiencias, corregirlas e incluirle mejoras. Se hizo un estudio crítico del análisis y diseño propuesto por los analistas, además de estudiar a profundidad las herramientas, tecnologías y lenguajes que fueron escogidos por el equipo de proyecto para proceder a la implementación. Se generó toda la documentación referente a los flujos de trabajo que propone la metodología de desarrollo escogida para el rol de implementador. Finalmente se hicieron pruebas para verificar la calidad con el objetivo de obtener la aceptación de los clientes y usuarios finales.

Palabras Claves: Producción, Implementación, Actividad productiva, Gestión de información.

Contenido

Declaración De Autoría	II
Resumen.....	III
Contenido	IV
Agradecimientos	VI
Dedicatoria	VII
Introducción	1
Capítulo I Fundamentación Teórica.....	6
Introducción	6
La gestión de actividades productivas. Tendencias actuales	6
Tendencias históricas de los Sistemas de gestión de actividades productivas	7
Conclusiones	28
Capítulo II: Descripción y análisis de la solución propuesta	29
2.1 Introducción	29
2.2 Valoración crítica del diseño propuesto por el analista	29
2.3 Estrategia de integración.....	30
2.3.1 Estándares de codificación.....	30
2.4 Arquitectura del sistema.....	39
2.5 Estudio de los Módulos del CMS Drupal. Análisis de posibles módulos rehusados.....	40
2.6 Creación de Módulos del CMS Drupal.....	42
2.7 Representación del Diseño.....	46
2.8 Descripción de las nuevas clases u operaciones necesarias:.....	50
Conclusiones	56
Capítulo III: Validación de la solución propuesta.	57
3.1 Introducción.....	57
3.2 Pruebas a realizar en tiempo de desarrollo.....	57
3.2 Pruebas después de la implementación.....	58
3.3 Diseño de casos de prueba contra requisitos funcionales.....	60
3.4 Propuesta de despliegue.....	67
3.5 Propuesta de soporte técnico.....	68
Conclusiones	68
Conclusiones generales:.....	69
Recomendaciones:	71
Referencias Bibliográficas	72
Glosario de términos	76
Anexos	79



**"El éxito es aprender a ir de fracaso
en fracaso sin desesperarse."**

Winston Churchill

Agradecimientos

Primero que nada a mi madre y a mi abuela, que han sido también mis padres, por haber estado siempre ahí para mí, por todo el apoyo, el cariño, la comprensión, la confianza que me han brindado...

A mis tíos y mis primos, por haber tenido parte en mi formación en mi educación, en mi vida...

A mis tutores, por su apoyo, su comprensión, y paciencia... especialmente mi tutora sin ella, nada de esto hubiese sido posible...

A Matilde, mi Vicedecana de formación, que no está ahora, pero que siempre tuve fe en mí...

A Mayelin, mi secretaria docente, por su apoyo y su confianza...

A mis amigos, todos ellos...

A todos los que formaron parte de mi vida, tanto social, como docente, y que pusieron su granito de arena, para que pudiese llegar hasta aquí...

A todos los que debí mencionar, y se me quedaron, y incluso a los que no...

A todos muchísimas gracias.

Dedicatoria

A toda mi familia, especialmente a mi madre y mi abuela que las quiero mucho a las dos...

A mi primo Reinier que también lo quiero mucho, aunque ya no está...

A mis amigos de verdad...

A mi compañera de tesis, Karina, que no pudo seguir conmigo, pero el trabajo también es de ella y para ella...

A todos los que desearon que yo llegara hasta aquí, les dedico este trabajo.

Introducción

A lo largo de la historia, incluso desde tiempos en los que no existía la electricidad, el hombre siempre ha desarrollado instrumentos y herramientas que lo ayudan a mejorar y perfeccionar sus actividades productivas, influyendo en su modo de vida. A mediados del siglo XX el desarrollo científico y tecnológico hace posible que una nueva rama: la informática, comience un ascendente camino que llega hasta la actualidad donde se ha generalizado en todos los sectores de la sociedad. A medida que avanza el desarrollo de la ciencia y la tecnología en las últimas décadas ha ido surgiendo convergencia cada vez mayor en la rama de la Informática y las telecomunicaciones. Las mismas que han progresado de la nueva rama del saber denominada Tecnologías de la Información de gran incidencia y modernización los diferentes sectores de la sociedad.

Actualmente el mundo está avanzando a la Era tecnológica, e incluso en el país se está efectuando todo un proceso de informatización. El Comandante en Jefe, uno de los mejores pensadores del siglo XX y del XXI, fue el creador de la Universidad de Ciencias Informáticas. La UCI es una universidad productiva, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. La Producción de Software y Servicios Informáticos se basa en la integración de los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva. Este espacio de integración temática es denominado Polo Productivo y se promueve la formación de pregrado, postgrado, la colaboración nacional e internacional, el fomento de líneas de investigación y desarrollo y la ejecución de proyectos en el marco de acuerdos de trabajo.

Esta integración garantiza la innovación continua que genera y aporta valor a los productos y servicios, promueve la gestión del conocimiento garantizando un mayor rendimiento, logra una mejor utilización y aprovechamiento de los recursos humanos y materiales, generando alta especialización y colaboración.

El Comandante en Jefe expresó: “La idea es convertir la informática en una de las ramas más productivas y generadoras de recursos para la nación. Es el empleo a fondo de la inteligencia y del capital humano que tenemos y principalmente del que podemos crear casi como espina dorsal de la economía ”. La Infraestructura de Productiva de la UCI coordina la actividad productiva y brinda servicios para asegurar el correcto desarrollo y terminación de proyectos productos o servicios. Entre los servicios más importantes se brindan los de calidad de software, arquitectura y tecnología, servicios legales, diseño de comunicación visual.

La producción se concentra en el desarrollo de proyectos en más de 30 Polos Productivos y se destacan resultados en las esferas de salud, educación, software libre, teleformación, sistemas legales, realidad virtual, automatización, bioinformática, procesamiento de imágenes y señales, gestión universitaria entre otras. Promueve el desarrollo de productos y servicios informáticos en aquellas ramas donde Cuba tiene un reconocido prestigio en el mundo a través del concurso de los mejores especialistas del país para lograr una solución de calidad y de impacto internacional. Desarrolla programas de informatización de la sociedad cubana a través de la relación con entidades nacionales, los resultados alcanzados se extienden por todo el país.

En la UCI se producen software para dar soluciones prácticas a distintos problemas sociales tanto dentro como fuera de la misma, es por eso que desde su creación como proyecto se ha informatizado muchos de los servicios que esta posee. Existe un crecimiento notable en la creación de proyectos productivos para la producción de software y la gestión de estos que se ha convertido en una de las disciplinas con mayor grado de prioridad dentro de esta importante institución, tratándose así de un reto para los que tienen la responsabilidad de llevarla a la práctica en las distintas facultades de la Universidad.

Para un mejor funcionamiento de la producción en la Facultad 1, se llevan a cabo una serie de procesos relacionados con la gestión de estas actividades, pero en una minuciosa investigación se detectaron algunos problemas que hace de estas actividades un trabajo engorroso. Primeramente, se trabaja sobre la

base de cronogramas irreales. El control y administración de los polos y proyectos productivos, se realizan manualmente utilizando programas de Microsoft Office lo que ha traído como consecuencia que el trabajo sea mucho más extenso, la planificación se hace mucho más lenta.

Al no existir un espacio organizado, donde se archiven las actas de las reuniones que se realizan en los consejos de producción, entre otros documentos importantes, se suscita la posibilidad de que muchas de ellos se pierdan o que el personal que los necesite no pueda acceder a ellos en el momento deseado.

Todo lo anteriormente planteado origina el siguiente **Problema Científico**: ¿Cómo lograr una eficiente gestión de todas las actividades productivas de la facultad 1?

En consecuencia el **Objeto de Estudio** de dicho trabajo son las Actividades Productivas de la Facultad 1 y más específicamente su **Campo de Acción** son los procesos referentes a la gestión de las actividades productivas de la Facultad 1.

Para darle respuesta a dicha situación problemática planteada anteriormente se trazó el siguiente **Objetivo General** de la presente investigación: implementar un sistema que mejore de la forma eficiente la gestión de las actividades productivas de la Facultad 1.

Partiendo de la necesidad que en la actualidad la Facultad 1 presenta con la gestión de sus actividades productivas se trazó la siguiente **Hipótesis**:

Si se implementa el módulo de la intranet de la facultad 1, entonces se podrá lograr una eficiente gestión de las actividades productivas de la Facultad 1.

Como **Variable Dependiente** se tiene: el módulo de producción de la intranet de la Facultad 1 y como **Variable Independiente**: las actividades productivas de la facultad 1.

Para lograr un eficiente desarrollo de la investigación y darle cumplimiento al objetivo general trazado se pone en práctica las siguientes **Tareas Científicas**:

- Establecer las bases teórico-conceptuales de la investigación.
- Realizar una búsqueda bibliográfica que permita fundamentar el estado del arte.
- Implementar el módulo de producción de la intranet de la Facultad 1
- Diseñar las pruebas que validen el funcionamiento del sistema.

El contenido expuesto en el siguiente trabajo es el resultado de diversos métodos de investigación, cuya práctica fue aprendida a lo largo de todos los cursos de la carrera.

Como **Métodos Teóricos** se utilizó el método **Analítico-Sintético** pues se hizo una breve investigación y estudio detallado de documentos y sistemas gestores de actividades productivas para así identificar el problema concreto existente en la facultad, sus causas y lograr un profundo análisis y diseño. Por medio del **Método Histórico-Lógico** se estudió la evolución y desarrollo de la gestión de actividades productivas a nivel de la UCI, Cuba y en el ámbito internacional; además de estudiar los cambios que surgieron y dieron lugar a la situación actual. Como **Métodos Empíricos** se realizaron una serie de **Entrevistas** a los trabajadores que actualmente llevan los procesos de la gestión de actividades productivas para tener un mejor conocimiento sobre los requisitos que debía cumplir el sistema y saber cómo funcionan las actividades relacionadas con la producción de la facultad. Se empleó la **Observación Científica** como método empírico con carácter cualitativo, la cual permitió la apreciación detallada de los procesos relacionados con los objetivos de la presente investigación. La misma se realiza con el propósito de conocer la evolución de los métodos utilizados para la gestión de las actividades productivas.

El documento consta de **tres capítulos**:

En el capítulo I "**Fundamentación Teórica**", se presenta el estado del arte de las diversas técnicas y tendencias de desarrollo de software que existen tanto en el ámbito internacional como nacional. Se realiza además, una descripción

detallada de las herramientas y metodologías de desarrollo utilizadas para alcanzar los objetivos propuestos.

En el capítulo II “ **Descripción y análisis de la solución propuesta** ”, se introducen los principales conceptos y artefactos tenidos en cuenta a la hora de la implementación del sistema, así como la forma en que cada uno de estos será específicamente empleado en la elaboración del sistema. Se realiza una valoración crítica, desde el punto de vista técnico, del diseño propuesto previamente por el analista, Se recoge además un estudio, a grandes rasgos, de los módulos de Drupal, principalmente los procesos relacionados con la creación e integración de los mismos, así como una valoración de la medida en que pueden ser rehusados algunos ya existentes, o características específicas de ellos.

En el capítulo III “ **Validación de la solución propuesta** ”, se exponen los conceptos y procesos relacionados con las pruebas de validación, así como los resultados de las mismas , aplicadas a cada unos de los requisitos del sistema, incluyendo la propuesta de despliegue. Este abarca además recomendaciones y sugerencias orientadas al soporte técnico del producto.

Capítulo I Fundamentación Teórica

Introducción

En este capítulo se brinda una visión general de los aspectos teóricos relacionados con la gestión de actividades productivas, los conceptos básicos asociados al dominio del problema y que se hacen necesarios para entender el desarrollo de la investigación. Se realiza un estudio crítico de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad para desarrollar sistemas para la gestión de actividades productivas, así como una valoración de las que son más factibles para dar solución al problema en cuestión.

La gestión de actividades productivas. Tendencias actuales

Hay casos en que unas mismas situaciones se repiten durante un período relativamente prolongado de tiempo (planes permanentes), de tal manera que los posibles acontecimientos se conocen con un razonable grado de certeza. En el caso, por ejemplo, de las operaciones repetitivas, de producción en línea o por pedido, donde los parámetros de control, una vez establecidos, permanecen estables por un período de tiempo determinado, solo se modifican cuando se intenta efectuar cambios en el proceso de producción, en el modelo de calidad, etc. Teniendo en cuenta que dentro de las funciones de un directivo de proyectos se encuentra la de dirigir y controlar las operaciones de ejecución de tal modo que el conjunto de acciones ejecutadas se ajusten (en tiempo y calidad) a lo especificado en el proyecto, es de vital importancia para el cabal desarrollo de cualquier proyecto, que el directivo tenga la autoridad, capacidad, sentido de equilibrio y una gran facilidad de comunicación y rapidez para tomar decisiones y para controlar las tareas, teniendo presente la dificultad que esto implica tratándose de proyectos (Ramón, 2007).

Tendencias históricas de los Sistemas de gestión de actividades productivas

1.2.1 Sistemas de gestión de actividades productivas en el mundo

A nivel internacional se han desarrollado varios sistemas para la gestión y control de actividades productivas y más específico para el control de proyectos productivos. A continuación se presenta un estudio de los sistemas de gestión de actividades productivas:

SICOP - Sistema de Control de Proyectos de Investigación

Sistema que fue desarrollado por la Oficina de Informática y Telecomunicaciones –OITEL- (Universidad del Valle, Colombia). Es un software que permite el registro de la información de los proyectos de investigación que adelantan las diferentes dependencias y profesores de la Universidad. Permite el registro y control de la información de los proyectos patrocinados por la Universidad y por entidades externas tanto nacionales como internacionales. Este es un software muy simple, que permite gestionar un grupo muy reducido de información, por tanto no abarca los requisitos del sistema (Oficina de Informática y Telecomunicaciones).

Gestproject

Gestproject es un software de Novo net (Madrid, España) para la organización, gestión y control de proyectos. Es una solución de gestión global de proyectos. Nos posibilita gestionar un proyecto de forma integrada desde el punto de vista del tiempo y costo de desarrollo, además de localizar, visualizar y editar cualquier información necesaria y agilizar el desarrollo del trabajo. Permite

también organizar, coordinar y optimizar el trabajo en equipo. La gestión de toda la documentación del proyecto, permite tener un mayor control y actualización de la información, obteniendo un mayor rendimiento, fiabilidad y disminución de errores y costes (GestProject, 2007). Como se plantea anteriormente GestProject, aunque es un herramienta muy eficaz, se orienta mucho más al plan de proyecto, que a la información que se genera en la gestión de proyecto como tal. Lo que imposibilita proponerlo como solución al problema planteado, aunque cabe destacar que se tomará en cuenta, para la implementación, su eficiente capacidad para controlar el avance de los proyectos productivos.

Microsoft Project

Microsoft Project es una aplicación desarrollada por Microsoft e integrada al paquete de Microsoft Office. Es una eficaz y flexible herramienta de administración de proyectos, válida para controlar proyectos simples o complejos y que podrá ayudar al responsable del proyecto a programar y realizar un seguimiento de todas las actividades para supervisar su progreso. Es importante destacar que es una de las herramientas más usadas para la Gestión de proyectos (Microsoft Corporation, 2009).

Características y ventajas

Entre muchas posibilidades de trabajo, Microsoft Project (MS Project) nos permite:

- Planificar y programar tareas así como asignar recursos a dichas tareas de manera adecuada y sencilla.
- Realizar un control, organización y seguimiento, así como coordinar toda la información que conlleva los requisitos del proyecto, la duración y los recursos asignados a las diferentes tareas.

- Visualizar el Plan de Proyecto en formatos estándar y con un diseño de diagramas muy apropiados y fáciles de interpretar.
- Establecer escenarios dentro del proyecto para crear análisis de hipótesis.
- Planteamientos del tipo "Qué pasaría si...".
- Intercambiar información de proyecto con todos sus participantes a través de una red local, Internet o de una Intranet

Independientemente del carácter propietario de este producto, que no se complementa con el proceso de migración hacia plataforma libre que se lleva a cabo en la UCI, y el país en general, y a pesar de que como antes se afirmara, es uno de los software más usados en el ámbito de la gestión de software, MS Project, orientado hacia una tendencia totalmente comercial de esta rama de la informática, no contempla las interacciones docente-productivas que toman lugar en los diferentes procesos de desarrollo en la Universidad, y que generan información relevante de cada estudiante y trabajador vinculado a un proyecto productivo.

DotProject

Fue creado por dotmarketing.org en el año 2000, con el fin de construir una herramienta para la Gestión de Proyectos. Esta construido por aplicaciones de código abierto y es mantenida por un pequeño pero dedicado grupo de voluntarios. Es una aplicación multiusuario, soporta varios lenguajes y es software libre. Está programada en PHP, y utiliza MySQL como base de datos (aunque otros motores como Postgres también pueden ser utilizados). La plataforma recomendada para utilizar DotProject se denomina LAMP (Linux + Apache + MySQL + PHP). De todas formas, existen binarios para instalar

DotProject bajo otros sistemas operativos tales como Microsoft Windows (NT, 2000, XP) y Mac (Abartia Team, 2008).

Es una aplicación web creada para administración de proyectos, que incluye módulos para compañías, proyectos, tareas (con gráficas de Gantt), foros, archivos, calendario, contactos, tickets de soporte. Es multidioma y soporta distintos niveles de permisos de uso de módulos. El aspecto de mayor interés que brinda DotProject, es su alto nivel de organización del trabajo, aunque como el caso del Gestproject todo su motor de trabajo gira sobre el plan de proyecto, y omite determinados esquemas de información que se generan el transcurso de la implementación de un software, principalmente si los desarrolladores son en su mayoría estudiantes y profesores.

1.2.2 Sistemas de gestión de actividades productivas en la UCI

Rinde Forge

Rinde Forge es un sistema desarrollado en la UCI, posee herramientas para ayudar al equipo de desarrollo a colaborar, como son foros de discusión, listas de correo, herramientas para crear y administrar repositorios de los ficheros de un proyecto utilizando CVS (Concurrent Version System) o Subversion. Permite registrar ítems acerca de los proyectos y darles seguimiento, tales como errores, correcciones, solicitudes de ayuda o solicitudes de nuevas funcionalidades, pudiendo definir más, con "ilimitado" número de categorías, campos de texto, etc. Luego de estudiar las distintas funcionalidades que le brinda al usuario se puede decir que se va más allá de lo que requiere el módulo a desarrollar ya que Rinde Forge básicamente ayuda a gestionar todo el ciclo de vida de desarrollo de proyectos de software, pero se debe tener en cuenta algunas de las funciones que proporciona como son: la gestión de documentos, publicación de noticias, administración de miembros, roles, referencias y la generación de reportes y estadísticas.

Intranet UCI e Intranet de las Facultades

En la UCI existe una intranet de forma centralizada la cual brinda servicios en las diferentes áreas de la misma. El área de Producción e Investigación hace referencia a las diferentes Direcciones de Producción que existen en la universidad. Conjuntamente algunas facultades desarrollan una intranet propia para ofrecer informaciones a sus trabajadores y estudiantes. Estas tienen un apartado para cada área o esfera de la facultad, entre ellas se encuentra Producción e Investigación pero hasta la actualidad solo proporciona informaciones estáticas y no presentan un sistema que automatice como tal los procesos relacionados con dichas esferas.

COPROY- Control de proyectos productivos

Es un sistema desarrollado por estudiantes de la facultad 4 de la UCI para el control de proyectos productivos, se basa en una aplicación web para centralizar y facilitar la búsqueda de la información proveniente de los proyectos productivos, específicamente los datos de los proyectos, integrantes, computadoras, laboratorios, empresas clientes y herramientas. Fue diseñado con metodología RUP, implementado con el lenguaje PHP, para el almacenamiento de los datos utiliza el servidor de base de datos PostgreSQL y el diseño de la arquitectura se realizó mediante el patrón Modelo – Vista Controlador. Actualmente se está utilizando en el centro UCI-FAR para la gestión básica de sus proyectos productivos. Es una aplicación por la cual se puede tomar partida para el desarrollo del nuevo módulo, aunque debemos tener en cuenta las características particulares de las actividades productivas en la Facultad 1 de la UCI.

Debido a las características que presenta la gestión de las actividades productivas en la facultad 1 de la UCI, se puede decir que en el mundo, a pesar de existir múltiples sistemas de gestión y administración de estas actividades, los

cuales presentan magníficas funciones, no se ajustan al módulo que se va a desarrollar, teniendo en cuenta que la mayoría de estos sistemas son privativos o no funcionan en el sistema operativo Linux. Actualmente se usan en algunos de los proyectos de la facultad sistemas de control con el fin de tratar información específica de los mismos pero no existe ningún sistema que permita llevar el control global de toda la producción de la facultad 1. Como se puede apreciar en la UCI se ha hecho necesaria la gestión de las actividades productivas que se desenvuelven a diario, por lo que se han desarrollado varios sistemas para dar cumplimiento a estas necesidades, los cuales cumplen con algunas de las funcionalidades requeridas pero no se integran a la intranet de nuestra facultad lo cual es objetivo primario para la realización del módulo. Para el desarrollo del nuevo módulo se tendrán en cuenta las funcionalidades comunes y los inconvenientes que presentan las aplicaciones anteriormente estudiadas.

1.3 Metodologías a utilizar

El objetivo de una metodología o proceso de desarrollo es subir la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso con el objetivo de reducir el caos y aumentar el porcentaje de proyectos que acaban con éxito. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo. Cuesta tiempo que los trabajadores se adapten al proceso, pero una vez superado la inversión se recupera con creces. Es por ello que no tiene sentido ajustarse a un proceso al pie de la letra, sino que hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo o casi a cada proyecto (JavaConGanas, 2008). Esto debería entenderse también como que no hay uno mejor que otro, puesto que su éxito depende siempre del contexto (tipo de proyecto, experiencia del personal, etc.). En realidad en los últimos tiempos la cantidad y variedad de los procesos de

desarrollo ha aumentado de forma impresionante, sobre todo teniendo en cuenta el tiempo que estuvo en vigor como ley única el famoso desarrollo en cascada. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros (también llamados métodos ágiles) tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen el proceso. Es importante destacar que para el uso de la metodología, lenguajes, herramientas y tecnologías empleadas se tuvo en cuenta la decisión del arquitecto y el líder del proyecto Intranet Facultad 1, basada en la política de migración a software libre llevada en la actualidad en nuestro país y en especial por nuestra Universidad.

SCRUM

Es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basado en las Metodologías Ágiles de desarrollo de software. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums (Proyectos Ágiles).

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutara durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores (Ansueta, 2008).

Durante cada sprint, un periodo entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene

del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.⁴ Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint (Palacio, 2007).

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

XP (eXtreme Programming)

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos (González Hernández, Reyes-Arias, Carbonell-de la Fé, Pérez González, & Ramos-Musibay, 2007-2008).

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Características fundamentales de esta metodología:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas,** frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi y NUnit para la plataforma.NET. Estas dos últimas inspiradas en JUnit.
- **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente **integración del equipo de programación con el cliente** o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- **Refactorización del código,** es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad** en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

1.4 Lenguajes utilizados

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos (Web, M.d, 2007). Además de la existencia del lenguaje de modelado UML el cual nos permite representar visualmente las reglas de creación, estructura y comportamiento de un grupo relacionado de objetos y procesos. Nos permite desarrollar el diseño de un software orientado a objetos.

UML2.0 (Unified Modeling Language)

Es una notación por donde se van construyendo sistemas por medio de conceptos orientados a objetos. Esta prescribe un conjunto de notaciones y diagramas estándares, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

Las características más generales de UML2.0 son:

- Tecnología de orientación a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo incremental e iterativo.
- Participación del cliente en todas las etapas del proyecto.

El UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizable. El UML prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. En resumen UML es el resultado de la unión de tres metodologías, Booch, OMT, y OOSE. Estas han tenido una aplicación extensa en el campo de la POO, tienen su historia, y han sido aplicadas en una gran variedad de industrias y problemas, por lo que pueden ser clasificadas como maduras (Ramón, 2007).

Entre las ventajas que nos brinda el UML2.0 encontramos:

- Mejora el nivel de comunicación formal.
- Se aborda la complejidad con una documentación minimalista.

Se desarrollan los procesos y los productos con una mayor fiabilidad y calidad.

Se define, organiza y comparte conocimiento.

- El esfuerzo de especificación es más eficiente.
- El impacto de las decisiones sobre un proceso o producto es más visible.

HTML (Hypertext Markup Language)

Es el acrónimo de HyperText Markup Language. Es el lenguaje más utilizado para la presentación de textos estructurados en formato hipertexto, estándar de las páginas Web. HTML es utilizado por la práctica totalidad de navegadores Web del mercado con el fin de presentar al visitante de una página Web el contenido de la misma tal como el diseñador quiere que se muestre a su público. No es un lenguaje de programación y no tiene compilador alguno, así que si hay algún error que no detecta lo visualizará de la manera en la que lo ha entendido. Es un sistema de etiquetas que indica al ordenador cuando hay que señalar una cursiva, separar un párrafo o definir el color del texto (Pozo, 2001). De todas maneras, tiene sus limitaciones así que a menudo se utilizan otras herramientas como las hojas de estilo, que le dan mayor libertad al diseñador. En concreto, el HTML le da las indicaciones mencionadas al programa cliente, el browser o navegador para que presente el documento en la pantalla de la manera adecuada. El HTML se hizo popular por su sencillez, era fácil de aprender y eso lo hace accesible a mayor número de personas. Estos documentos web deben

estar escritos con el mismo —lenguaje|| para que diferentes ordenadores puedan leerlos, si alguien utiliza un sistema diferente no podrá compartir su información con los que usan el HTML ni podrá visualizar los de los demás (Antes que nada., 2007).

PHP 5(Hypertext Processor)

Es un potente lenguaje de script del lado del servidor, que se utiliza principalmente para generar páginas de forma dinámica, es tan potente que se utiliza para muchísimas cosas más: generar imágenes "on-the-fly", generar PDF, atacar diferentes protocolos de Internet (LDAP, IMAP, POP3, FTP, TELNET, JABBER), como lenguaje de script genérico. La sintaxis, gramática y funciones del lenguaje son muy sencillas, su "compatibilidad" con otros sistemas se incrementa agregando nuevas funciones. La manera de manejar los datos, las funciones disponibles y la documentación son realmente excepcionales. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a Java Script o a C, para aquellos que conocen estos lenguajes (Alvarez, 2004). PHP es usado por el 24% del total de los sitios Web de Internet. Esto suma un total de 9 millones de servidores y por el momento no cesa de crecer. En los últimos dos años, PHP ha crecido un 6.5% mensualmente. La clave de su éxito es la confianza de los administradores por lo que ahora se conoce como LAMP (Linux + Apache + MySQL + PHP). Además, PHP está diseñado para soportar perfectamente sitios con tráfico muy denso, es muy productivo para las empresas y es código abierto. Hoy en día grandes empresas usan PHP, entre ellas GMC, Motorola, Siemens, Ericsson, Philips, Air Canada, Lycos Europe etc. PHP usa una mezcla entre interpretación y compilación para intentar ofrecer a los programadores la mejor mezcla entre rendimiento y flexibilidad. PHP compila

para tu código una serie de instrucciones (llamadas opcodes) siempre que estas son accedidas. Estas instrucciones son entonces ejecutadas una por una hasta que el script termina. Esto es diferente a la manera convencional de compilación de lenguajes como C++ donde el código es compilado a código ejecutable que es después ejecutado. PHP es recompilado cada vez que se solicita un script. Esta constante re compilación puede parecer una pérdida de tiempo, pero no lo es porque no hay que preocuparse de la re compilación cada vez que realizas cambios. Además, proporciona una regeneración muy rápida durante el desarrollo. Si hay errores en el código, PHP rechazará compilar la página hasta que el programador arregle el problema, y tendrá que ejecutar el código línea a línea hasta encontrar el problema (Antes que nada., 2007).

Ventajas de PHP5:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente rescrita.
- Mejor soporte a XML (XPath, DOM, etc.).
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones.

1.6 Herramientas a utilizar

Visual Paradigm para UML 6.1

Es una herramienta CASE que utiliza —UML como lenguaje de modelaje. Permite diseñar todo el negocio y sistema del presente trabajo dando una mayor precisión y permitiendo obtener exitosos resultados (Programación en Castellano, 2004). Está disponible en varias ediciones, cada una destinada a unas necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal (Visual paradigm, Official Homepage, 2008).

Principales Ventajas de Visual Paradigm para UML6.1:

- Entorno de creación de diagramas para UML 2.0
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

Drupal 5.7

Es un sistema de gestión de contenido para sitios Web que permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web. En general estas son características que motivaron para la utilización de este sistema de gestión de contenido.

Características que hacen de Drupal 5.7 un buen gestor de contenidos:

- Amplia comunidad de usuarios que lo utilizan y lo mantienen, lo que implica que el desarrollo de la aplicación tenga un gran dinamismo y esté mejorándose continuamente.

- Está basado en software libre: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros gestores de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Entorno modular: al estar estructurado en módulos permite una gran flexibilidad, permitiendo crear desde el portal Web de un Instituto hasta un sitio de blogs. Drupal tiene disponibles una gran cantidad de módulos que pueden descargarse de su página Web e instalarlos en el sistema ampliando su funcionalidad.
- Múltiples herramientas de trabajo colaborativo: comentarios enlazados jerárquicamente, foros de discusión, encuestas, libros colaborativos, que permiten a muchos usuarios trabajar conjuntamente en la elaboración del contenido... También existe la posibilidad de que los contenidos sean moderados por los usuarios. En este caso los contenidos pasan por una serie de puntuaciones realizadas por los usuarios antes de ser publicados en la portada.
- Soporte de múltiples idiomas y localización: Drupal permite tener el contenido en varios idiomas, lo cual, para comunidades autónomas bilingües, es un requisito importante.
- Drupal es multiplataforma desde sus inicios. Puede funcionar con Apache o Microsoft IIS como servidor Web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X.

1.7 Gestor de base de datos

PostgreSQL8.2

Este gestor de bases de datos posee una estabilidad y confiabilidad legendaria, nunca ha presentado caídas en varios años de operación de alta actividad. Está disponible para 34 plataformas Unix en la última versión estable, existe una versión para Windows usando la plataforma Cygwin. Fue diseñado para ambientes de alto volumen. Escala muy bien al aumentar el número de CPUs y la cantidad de memoria. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos (Veillard, 2008).

PostgreSQL8.2 ofrece muchas ventajas respecto a otros sistemas de bases de datos:

- Instalación ilimitada.
- Modelos de negocios más rentables con instalaciones a gran escala.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- Mejor soporte que los proveedores comerciales.

- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad legendarias.
- Extensible.
- Multiplataforma.
- Diseñado para ambientes de alto volumen.
- Herramientas gráficas de diseño y administración de bases de datos.

1.7 Tecnologías y principios básicos empleados

1.7.1 ¿Qué se define como Tecnología web?

El concepto de servicios web suele confundirse con el de "servicio interactivo", es decir, con "servicios que se prestan a través de una página web". Sin embargo, los servicios web son en realidad un paradigma de computación distribuida que consiste en un conjunto de protocolos de comunicación que permiten el intercambio de datos entre aplicaciones remotas. El motivo de utilizar tecnología de la Web es porque incluye software servidor, sindicación de contenidos, aplicaciones cliente, navegadores estándares con plugins y extensiones y todas estas tecnologías le permiten ofrecer capacidades distintas de las que ofrecen los sitios Web tradicionales, como son almacenamiento de información, creación y capacidades de diseminación. La tecnología que hay detrás de la Web trata, sobre todo, de tecnologías que están siendo utilizadas por personas en el mundo, y de la velocidad con la que se asimilan muchas de ellas. Esta está formada por un conjunto de tecnologías que permiten que un

gran número de personas se conecten, colaboren y creen entre sí (Sanidad, 2007).

1.6.2 Sistema Operativo

Linux

Linux es un sistema operativo tipo Unix, también conocido como GNU/Linux. Se distribuye bajo la Licencia Pública General de GNU o GPL, es decir que es software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds. Es usado ampliamente en servidores y súper-computadores y cuenta con el respaldo de corporaciones como Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat y Sun Microsystems. Las variantes de estos sistemas se denominan "distribuciones" y su objetivo es ofrecer una edición que cumpla con las necesidades de determinado grupo de usuarios. De esta forma existen distribuciones para hogares, empresas y servidores. Algunas son gratuitas y otras de pago, algunas insertan software no libre y otras contienen solo software libre (Linux Para Todos, 2009).

Siguiendo la decisión tomada por la dirección del proyecto Intranet Facultad 1 de seguir la política de nuestro país y el objetivo que se quiere lograr en la Universidad de las Ciencias Informáticas de adoptar un sistema operativo estándar de trabajo, se decidió emplear Linux por las múltiples características que ofrece:

- Linux es muy robusto, estable y rápido: Ideal para servidores y aplicaciones distribuidas. A esto se añade que puede funcionar en máquinas humildes: Linux puede correr servicios en un x86 a 200 MHz con calidad.

- Linux es libre: Esto implica no sólo la gratuidad del software, sino también que Linux es modificable y que Linux tiene una gran cantidad de aplicaciones libres en Internet. Todo ello arropado por la inmensa documentación de Linux que puede encontrarse en la Red.
- Linux ya no está restringido a personas con grandes conocimientos de informática: Los desarrolladores de Linux han hecho un gran esfuerzo por dotar al sistema de asistentes de configuración y ayuda, además de un sistema gráfico muy potente. Distribuciones Linux como Red Hat/Fedora tienen aplicaciones de configuración similares a las de Windows.
- Linux difícilmente tendrá algún virus: De hecho, un virus en Linux no es imposible pero Linux hace que esto sea bien difícil que pase, por varias razones: La mayoría de personas usa Microsoft Windows y los delincuentes informáticos quieren hacer tanto daño (o controlar) como sea posible: es por esto que ellos apuntan sus ataques a Windows.

Teniendo en cuenta que el uso de la metodología y las herramientas antes mencionadas responde a los lineamientos trazados por la arquitectura del proyecto "Intranet Facultad 1", de cuyo sistema el módulo en cuestión será componente, no se hizo necesario plasmar en el presente documento el marco teórico que fundamenta su uso, aunque si se realiza una detallada descripción de las propiedades y capacidades de cada uno de ellos, las cuales será los medios de desarrollo del sistema a desarrollar.

Conclusiones

A lo largo del capítulo se expuso el estudio del arte y conceptos fundamentales para el desarrollo del presente trabajo, analizando las tendencias históricas y actuales que presentan los distintos sistemas existentes para la gestión de las actividades productivas; así como las herramientas previamente dispuestas para el desarrollo del módulo en cuestión. Dicho esto, se ha llegado a la conclusión de que utilizando correctamente dichas herramientas y siguiendo al pie de la letra todas las normas y procedimientos de implementación se llegará al resultado esperado o sea: El Módulo de Producción de la Intranet de la Facultad 1.

Capítulo II: Descripción y análisis de la solución propuesta

2.1 Introducción

Este capítulo contiene el resultado de la búsqueda y análisis de la información vinculada al objeto de estudio, procesos a automatizar y conceptos asociados al dominio del sistema. Además, se define el diseño de la solución propuesta, los principales elementos tenidos en cuenta a la hora de hacer cada una de las implementaciones para lograr un producto con la calidad y la eficiencia requerida. Se estarán introduciendo además diferentes conceptos tenidos en cuenta a la hora del desarrollo del módulo en cuestión.

2.2 Valoración crítica del diseño propuesto por el analista

El diseñador del sistema ha realizado una propuesta adecuada teniendo en cuenta las condiciones tecnológicas, los requerimientos funcionales y no funcionales del sistema.

En el diseño propuesto para lograr una mejor comprensión de las relaciones internas de los elementos del diseño, se propone una estructuración a través de paquetes de diseño, y valga la redundancia, que contienen de manera lógica, los diferentes procesos del negocio según su responsabilidad en la aplicación. A grandes rasgos la propuesta de diseño fue creada siguiendo patrones, que de manera general, constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño basado en la experiencia y que se ha demostrado que funcionan, permitiendo llevar a cabo la implementación clara y limpia del módulo en cuestión.

2.3 Estrategia de integración

Todo el código dentro un mismo componente se comunica mediante llamadas a métodos o eventos de forma directa. La comunicación entre diferentes componentes se realiza de forma directa a nivel de negocio, en caso de utilizarse servicios web, la información que es transmitida debe cumplir con los estándares internacionales que hay establecidos para facilitar la integración entre nuevos componentes y otros sistemas. La base de datos es accedida de forma directa mediante clases controladoras y los componentes rehusados son integrados mediante interfaces propias del CMS.

2.3.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (Fernández, 2005). Independientemente de que PHP posee su propio estándar de codificación, Drupal brinda la posibilidad de usar técnicas de codificación sólidas y realizar buenas prácticas de programación con el objetivo generar un código de alta calidad. Considerando el alto nivel de flexibilidad que nos brinda PHP como lenguaje interpretado y el cómodo ambiente de desarrollo que brindan el variado grupo de funciones que contiene Drupal, se hizo necesario plasmar en este documento la esencia de los estándares de código utilizados en la implementación, para que se tenga una mayor comprensión de por qué se utiliza la estructura del código fuente que presenta el sistema, y por qué la misma le otorga al módulo y al sitio en general, un alto grado de integración y rendimiento a la hora de la compilación y ejecución de su código fuente.

Sangrado

Utilice un guión de 2 espacios, sin pestañas. No *trailing* blanco.

Estructuras de control

Estos incluyen *if*, *for*, *while*, *switch*, etc Aquí hay un ejemplo de una declaración de *if*, ya que es el más complicado de ellos:

```
if (condition1 || condition2) {
    action1;
}
elseif (condition3 && condition4) {
    action2;
}
else {
    defaultaction;
}
```

Las estructuras de control deben tener un espacio entre las palabras claves de control y la apertura de paréntesis, para distinguirlos de las llamadas a funciones.

Es recomendable utilizar siempre llaves, incluso en situaciones en las que son técnicamente opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores de lógica cuando se introducen nuevas líneas.

Para cambiar las declaraciones:

```
switch (condition) {
```

case 1:

action1;

break;

case 2:

action2;

break;

default:

defaultaction;

}

Llamadas a funciones

Las funciones deberían ser, sin espacios entre el nombre de la función, la apertura de paréntesis, y el primer parámetro, comas y espacios entre cada parámetro, y ningún espacio entre el último parámetro, el paréntesis de cierre, y el punto y coma. He aquí un ejemplo:

```
$var = foo($bar, $baz, $quux);
```

Como se muestra más arriba, debería haber un espacio a ambos lados de un signo de igualdad utilizado para asignar el valor de retorno de una función a una variable. En el caso de un bloque de asignaciones, más espacio se podría introducir para proveerlo de más legibilidad:

```
$short    = foo($bar);
```

```
$long_variable = foo($baz);
```

Declaraciones de funciones

```
function funstuff_system($field) {
  $system["description"] = t("This module inserts funny text into posts randomly.");
  return $system[$field];
}
```

Los argumentos con valores por defecto deben ir al final de la lista de argumentos. Siempre intentando devolver un valor significativo a partir de una función, si uno es apropiado.

Arreglos

Las matrices deben ser formateadas con un espacio que se separe cada uno de los elementos y operadores de asignación, si es aplicable:

```
$some_array = array('hello', 'world', 'foo' => 'bar'); array $some_array =
array('hello', 'world', 'foo' => 'bar');
```

Tenga en cuenta que si la línea se extiende por más de 80 caracteres (con frecuencia el caso de la forma y el menú de declaraciones), cada elemento debe ser roto en su propia línea, y un nivel de sangría:

```
$form['title'] = array(
  '#type' => 'textfield',
  '#title' => t('Title'),
  '#size' => 60,
  '#maxlength' => 128,
  '#description' => t('The title of your node.'),
);
```


Cotizaciones (Comillas)

Cosas como "comillas simples vs dobles" son herramientas que PHP utiliza para hacer más fácil la programación y no existen razones para no utilizar una forma preferentemente a otra. Estas mejores prácticas se quedan por decidir únicamente en el desarrollador. Mientras sea posible, es bueno mantener la coherencia dentro de cada módulo y el respeto del estilo personal de otros desarrolladores.

Con esa advertencia en mente: las cadenas con comilla simple son conocidas por ser más rápidas para leer debido a que el parser no tiene que buscar variables en línea. Su uso se recomienda, excepto en dos casos:

- Variables en línea.
- Traducido de cadenas donde uno puede evitar escapar comillas encerrando la cadena entre comillas dobles. Una de estas cadenas sería "Él es una buena persona." Sería "El \ 'sa buena persona." con comillas simples. Esa no puede escapar de ser manejado correctamente por los generadores de traducción de texto.

Concatenaciones de cadenas

Siempre use un espacio entre el punto y la concatenación de las partes para mejorar la legibilidad.

```
<?php
$string = 'Foo' . $bar ;
$string = $bar . 'foo' ;
$string = bar () . 'foo' ;
```

```

$string = 'foo' . 'bar' ;
?> string <?php
$string = 'Foo' . $bar ;
$string = $bar . 'foo' ;
$string = bar () . 'foo' ;
$string = 'foo' . 'bar' ;
?>

```

Al concatenar las variables simples, puede usar comillas dobles y añadir dentro de la cadena, de otra manera se debe utilizar comillas simples.

```

<?php
$string = "Foo $bar" ;
?> string <?php
$string = "Foo $bar" ;
?>

```

Cuando se utiliza el operador de asignación concatenando ('.='), se debe utilizar un espacio en cada lado con el operador de asignación:

```

<?php
$string .= 'Foo' ;
$string .= $bar ;
$string .= baz ();
?> string <?php
$string .= 'Foo' ;
$string .= $bar ;
$string .= baz ();
?> string <?php
$string .= 'Foo' ;
$string .= $bar ;
$string .= baz ();

```

```
?> string <?php
$string .= 'Foo' ;
$string .= $bar ;
$string .= baz ();
?>
```

Comentarios

Los comentarios no documentados se deben utilizar con mayúscula al principio. Las sentencias deben ser separadas por espacios. Todos los tapones se utilizan en los comentarios sólo cuando existen referencias constantes, por ejemplo *TRUE*. Los comentarios deben ser en una línea separada inmediatamente antes de la línea de código o bloque que referencia. Por ejemplo:

```
// Unselect all other contact categories.
db_query('UPDATE {contact} SET selected = 0');
```

Si cada línea de una lista necesita un comentario aparte, los comentarios podrán ser recogidos en la misma línea y pueden ser formateados a un modelo uniforme con guión para mejorar la legibilidad.

Los comentarios al estilo C */* */* y estándar de C++ *//* son a la vez bien vistos. El uso del shell de Perl (*#*) es desalentador.

Etiquetas de Código PHP

Siempre use `<?php ?>` Para delimitar el código PHP, y no las etiquetas, `<? ?>` `<? ?>` Esto es necesario para el cumplimiento de las normas de implementación en Drupal y también es la forma más portátil para incluir código PHP en diferentes sistemas operativos.

Todos los archivos de código fuente en el núcleo de Drupal contienen el siguiente comentario como bloque de la cabecera:

```
<?php  
// $Id$
```

Esta etiqueta se ampliará para el CVS que contenga información útil

```
<?php  
// $Id: CODING_STANDARDS.html,v 1.15 2008/12/22 15:27:26 keithsmith Exp $  
v <?php  
// $Id: CODING_STANDARDS.html,v 1.15 2008/12/22 15:27:26 keithsmith Exp $
```

Funciones y métodos

Las funciones y los métodos deben nombrarse con minúsculas y las palabras deben estar separadas por un guión bajo. Las funciones además tienen el nombre del módulo como prefijo, para evitar colisiones entre los módulos.

Las estructuras con miembros privados (es decir, los miembros de la clase que van a utilizarse sólo dentro de la misma clase en que se declaró) están precedidas por un solo subrayado. Por ejemplo:

```
_node_get()
```

```
$this->_status
```

Constantes

Las constantes deben ser siempre todas en mayúsculas, con sangría para separar palabras. Esto incluye las pre-definidas como constantes *PHP TRUE*, *FALSE* y *NULL*. Las constantes de módulos definidos también deberían ir

precedidas de la misma forma en que estaban escritos los nombres de los módulos correspondientes.

Variables globales

Si necesita definir las variables globales, su nombre debe comenzar con un solo subrayado seguido por el módulo o tema, y otro nombre subrayado.

Nombres de clase

Aunque las clases se utilizan con poca frecuencia en Drupal, en sus nombres debe utilizarse "studlyCaps". Por ejemplo:

```
$type = new stdClass ();  
type $type = new stdClass();  
new $type = new stdClass();
```

Los nombres de los archivos

Todos los archivos de documentación debe tener la extensión de nombre de archivo". Txt" para ser vistos más fácil por los sistemas operativos. Además, los nombres de archivo para este tipo de archivos deben ser todos capitalizados (por ejemplo, en lugar del archivo, readme.txt, README.txt), mientras que la extensión en sí es todo en minúsculas (es decir, en lugar de txt, TXT).

Ejemplos: archivo *README.txt*, *Install.txt*, *TODO.txt*, etc *CHANGELOG.txt*

Ayudante de secuencias de comandos

Existe un script de línea de comando de ayuda que se incluye con Drupal para comprobar el estilo de código. El código-style.pl es un archivo que se encuentra en la carpeta /scripts Para usar el script simplemente introduzca su archivo como argumento cuando se ejecute el script (asegúrese de que el script es ejecutable):

```
./code-style.pl path/to/file/example.module
```

Esta es una lista de sugerencias para mejorar el código. El usuario tendrá que hacer los cambios reales en el código mismo. (Drupal Coding standards, 2008)

2.4 Arquitectura del sistema

Más allá de clasificar Drupal en uno u otro padrón arquitectónico, digamos que se habla en términos de "modelo-vista-controlador" en el sentido de que Drupal hace una excelente separación *_limpia_* entre la persistencia de los datos y los formularios (modelo), la lógica (ciclo de respuesta al pedido http, o sea, ciclo de vida de un "mensaje" a un sitio Drupal), y la vista que emerge como la respuesta HTML al explorador del internet (navegador). En cada esfera, hay muchas oportunidades para efectuar modificaciones de manera limpia. Drupal es un gestor de contenidos modular, y como todos, sólo hay que instalarlo en un servidor web que disponga de un lenguaje servidor (en este caso PHP) y una base de datos. Además tiene un API (un motor recopilador de funciones, librerías, clases) muy potente. Funciona con bases de datos MySQL o PostgreSQL. Está desarrollado con PHP y tiene ya muchas funciones con jQuery y Ajax (Chin, 2007).

Drupal no está programado orientado a objetos, sin embargo si está diseñado con características de un sistema orientado a objetos, utiliza patrones de diseño que son propios de estos sistemas. En la arquitectura de Drupal se observan comportamientos de patrones de diseño de diferentes categorías como son *Singleton, Decorator, Bridge, Observer, Command, Chain of Responsibility*; y un patrón determinante es el "*Reflection Pattern*", que describe el comportamiento de todo sitio basado en Drupal. Gracias a este patrón el sistema se convierte en una aplicación muy adaptable a diferentes entornos puesto que permite prácticamente la modificación de todo su comportamiento a través de los módulos instalables, sin la necesidad de modificar del núcleo.

2.5 Estudio de los Módulos del CMS Drupal. Análisis de posibles módulos rehusados

Un módulo es la unión de un grupo de varias funciones que se unen en Drupal y ayudan a ofrecerle mayor funcionalidad a la web. Los módulos en Drupal son ficheros con extensión .module que contienen funciones escritas en PHP. 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. Los módulos se pueden desactivar temporalmente para reducir la carga del servidor. Algunos módulos vienen con todas las instalaciones de Drupal (módulos del “núcleo”), mientras que otras pueden ser instaladas. Existe una gran cantidad de módulos disponibles, que tratan diversas temáticas de gran calidad. Muchas características del funcionamiento básico de Drupal, como por ejemplo el poder organizar por categorías el contenido de las páginas, son hechas a través de módulos o la posibilidad de realizar la búsqueda de alguna palabra dentro del contenido de una página o agregar comentarios. Dos tipos de módulos importantes que utiliza Drupal son el módulo de bloque, que da la posibilidad de crear contenido abreviado, que aparecen en la mayoría de los temas en los bloques de la izquierda y la derecha; y están además los módulos de nodo que generan el contenido de la página del sitio, como puede ser la página del foro.

2.5.1 Módulos de Drupal para la realización del diseño

Módulo node

- Acceder al contenido
- Administrar nodos.

- Ver revisiones.
- Volver a revisiones.
- Crear.
- Editar.
- Edit own (creación propia).

Módulo upload.

- Cargar archivos.
- Ver archivos cargados.

Módulo block

- Administrar bloques.

Módulo system

- Acceder a las páginas de administración.

Módulo Page

- Crea contenido

Módulo views

- Crear una vista de los servicios quedando visible para los usuarios

Módulo CCK

- Crear todos los campos asociados a la creación de los tipos de contenidos.

Módulo taxonomy

- Activa la categorización del contenido.

2.6 Creación de Módulos del CMS Drupal.

Para comenzar a crear un módulo es necesario crear un archivo PHP, y guardarlo con ***nombremódulo.module***, otro con el ***nombremodulo.install*** este sería para crear las tablas en las base de datos y un último archivo ***nombremodulo.info*** para ofrecer información acerca del modulo. Todas las funciones del módulo que son usadas por Drupal se nombran como ***{nombremódulo}_{hook}***, donde "hook" es una función predefinida llamada sufijo. Drupal llamará a esas funciones para conseguir datos específicos, así teniendo estos nombres bien definidos. Cada módulo se creará según los requisitos funcionales y a partir los casos de usos. La instalación de un nuevo módulo resulta por lo general muy sencilla. Aunque esto depende de los requerimientos de cada módulo en particular. Como mínimo se debe tener los conocimientos necesarios para acceder al servidor y copiar los ficheros del módulo en el directorio modules.

A continuación se muestran los hook más importantes, que se utilizan a la hora de crear un nuevo módulo:

Hook access

Define restricciones de acceso. Este hook permite a módulos de nodo limitar el acceso. La cuenta administrativa (el usuario ID #1) siempre pasa cualquier comprobación de acceso, no se llama a este hook en este caso. Si este hook no está definido para un tipo de nodo, todas las comprobaciones de acceso fallarán, tan sólo el administrador será capaz de ver el contenido.

Hook perm

Aquí es donde se definen los nombres de los permisos del módulo. Esta función no asigna los permisos, simplemente especifica cuáles estarán disponibles para el módulo. Con esta función se da acceso a quién puede acceder al contenido del sitio o administrar el módulo.

Hook menu

La función *hook_menu* es llamada siempre por Drupal antes de crear las páginas, antes de mandarlas al navegador, y en ella pueden indicarse los puntos concretos en los que el módulo interactuará con la web.

Hook view

Esto es un gancho usado por módulos de nodo. Esto permite a un módulo definir un método de mostrar los nodos.

Hook form

Muestra un nodo para editar formulario. Este hook es usado por módulos de nodo, es llamado cuando se necesita crear o editar algún artículo. Este hook tiene que devolver el título de nodo, el área de texto, y otro campo específico del

tipo de nodo. Un arreglo devuelve los elementos del formulario del nodo que se está modificando.

Hook submit

Es utilizado por módulos de nodo. Es usado a para las acciones que deben ocurrir cuando el módulo es guardado. Es utilizado después de que la validación ha tenido éxito y antes de que insertar o actualizar.

Hook validate

Esto es un gancho usado por módulos de nodo. Es utilizado para permitir al módulo verificar que el nodo se encuentra en un formato válido para añadirse al sitio.

Hook delete

Es un gancho usado por módulos de nodo. Permite al módulo tomar medidas cuando un nodo es eliminado de la base de datos, por ejemplo, borrando la información de tablas relacionadas.

Hook update

Responde cuando un nodo se actualiza. Esto es un gancho usado por módulos de nodo. Es utilizado para permitir al módulo tomar acciones cuando un nodo es actualizado en la base de datos, por ejemplo, actualizando la información en tablas relacionadas.

Hook search

Permite a un módulo realizar búsquedas sobre el contenido que este define cuando una búsqueda de sitio es realizada.

Hook link

Defina los enlaces internos de Drupal. Este hook posibilita añadir enlaces en diversos lugares en Drupal, como en bloques de navegación o en nodos.

Hook settings

Para que el administrador pueda configurar lo que se va a mostrar, se crea la página de configuración. Casi siempre el administrador es quien tiene acceso a esta página.

Hook search item

Este módulo permite modificar los resultados de una búsqueda.

Hook search

Permite a un módulo realizar búsquedas sobre el contenido que este define cuando una búsqueda de sitio es realizada.

Hook nodeapi

Actúa sobre nodos definidos por otros módulos. Este hook no es reservado para módulos de nodo. Al contrario, esto permite a módulos reaccionar a acciones que afectan todas las clases de nodos, independientemente de si aquel módulo definió el nodo.

Hook filter

Define filtros de contenidos. El contenido en Drupal es pasado por todos los filtros permitidos, antes que sea mostrado. Con este se puede modificar el contenido al gusto del administrador de sitio. Terminado este estudio general sobre los módulos, es necesario dejar constancia de cómo se definen estos a la hora de crearlos desde cero o de adaptarlos a la funcionalidad que se persigue, para ello se hace imprescindible crear un artefacto llamado: "Propósito del módulo", por parte del Jefe del Portal y del Programador, donde queden

plasmadas las características y funciones del nuevo módulo para un futuro mantenimiento. Se propone que dicho artefacto tenga esta estructura:

- Nombre del Artefacto
- Nombre del módulo en cuestión.
- Nombre del autor.
- Breve descripción.
- Funcionalidades que abarca (requisitos funcionales y no funcionales).
- Diagramas de secuencia para las funcionalidades.

2.7 Representación del Diseño.

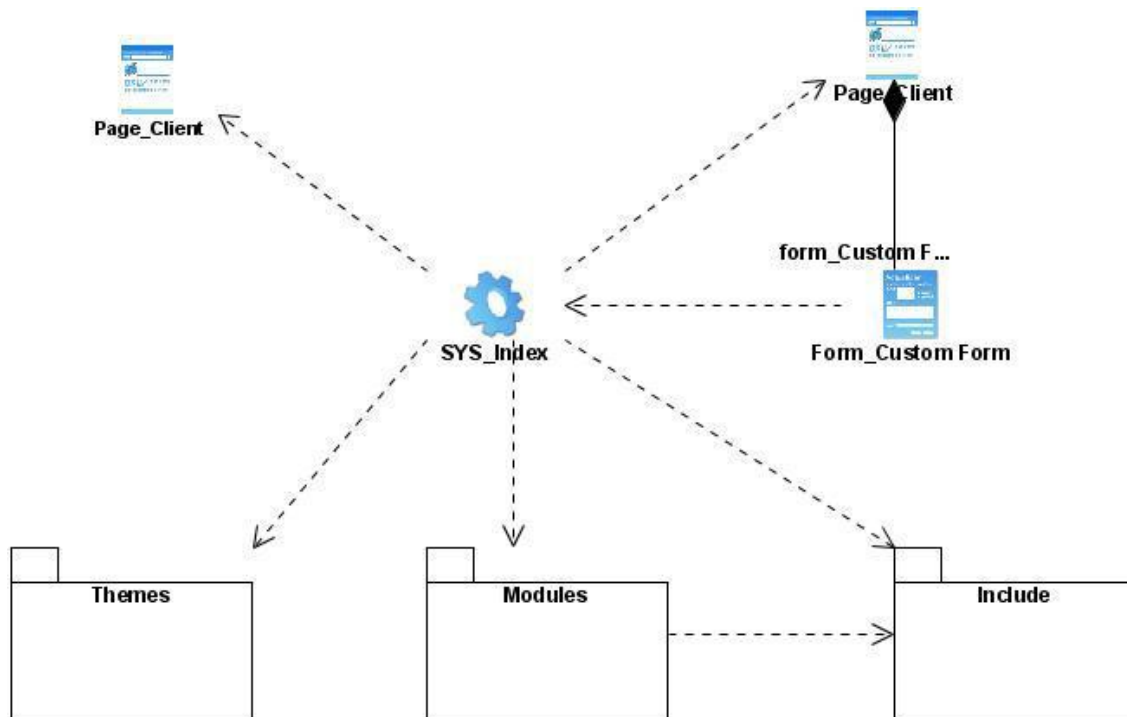


Figura #1 *Diseño General de Drupal*

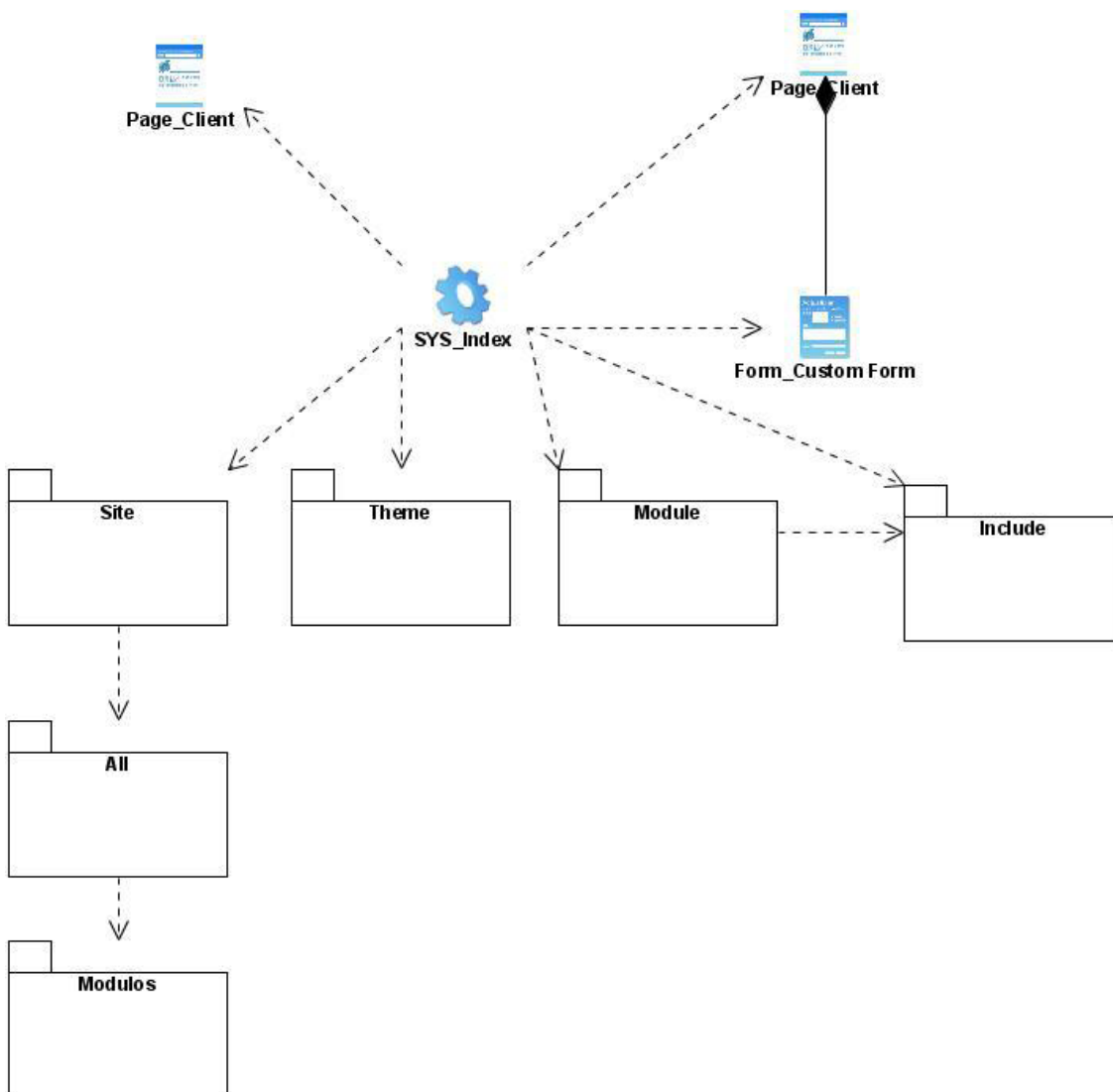


Figura #2 **Estructura propuesta para el diseño**

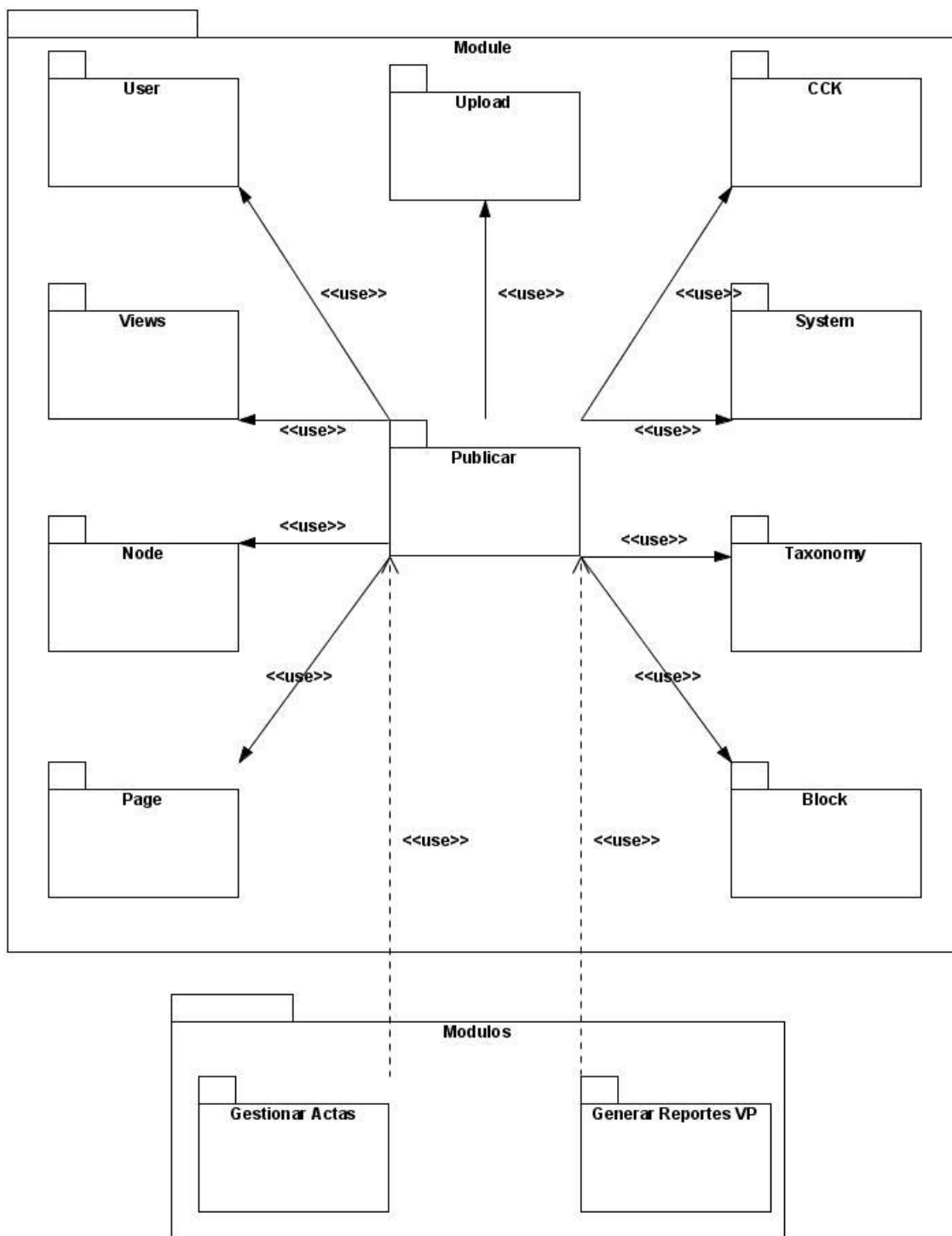


Figura #3 *Diseño de módulos de Drupal*

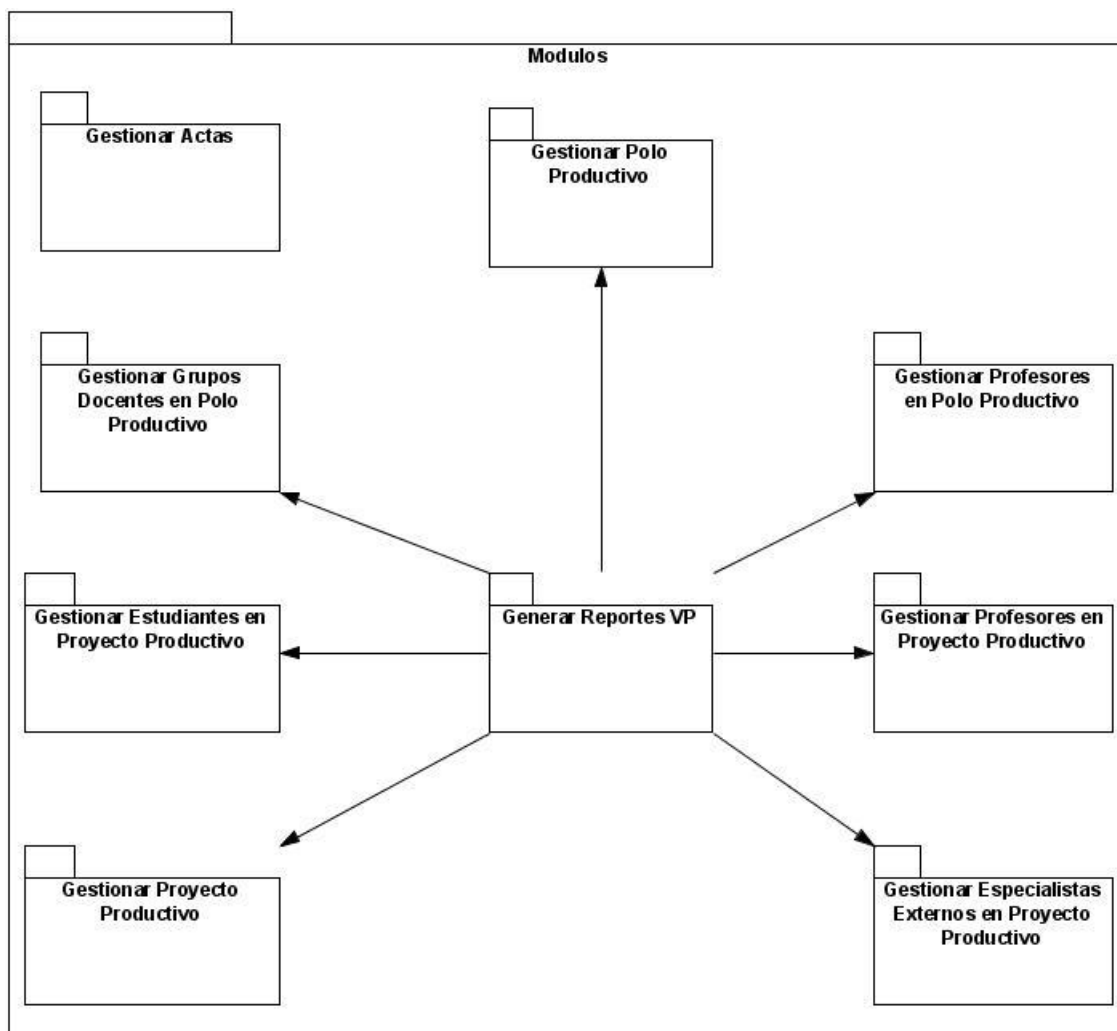


Figura #4 *Estructura del módulo de producción.*

2.8 Descripción de las nuevas clases u operaciones necesarias:

Polo

Nombre: Polo	
Tipo de clase: Entidad	
Atributo	Tipo
Id_polo	Int
Nombre	String
Líneas_trabajo	String
Fecha_creación	String
Usuario_jp	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #5 *Descripción de la clase Polo*

Información_polo

Nombre: Información_polo	
Tipo de clase: Entidad	
Atributo	Tipo
Polo	Node_reference
Grupo	Node_reference
Ci	Int
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #6 *Descripción de la clase Información_polo*

Grupo_docente

Nombre: Grupo_docente	
Tipo de clase: Entidad	
Atributo	Tipo
Id_grupo	Int
Nombre_grupo	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #7 *Descripción de la clase Grupo_docente*

Persona	
Nombre: Persona	
Tipo de clase: Entidad	
Atributo	Tipo
Ci	Int
Nombre	String
1er_apellido	String
2do_apellido	String
Usuario	String
Tipo_persona	String
Solapin	String
Email	String
Sexo	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #8 *Descripción de la clase Persona*

Proyecto	
Nombre: Proyecto	
Tipo de clase: Entidad	
Atributo	Tipo
Código	Int
Nombre_proy	String
Descripción	String
Clasificación	String
Entidad_cliente	String
Polo	Node_reference
Usuario_jproy	String
Fecha_ini	Date
Fecha_fin	Date
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #9 *Descripción de la clase Proyecto*

Información_proyecto

Nombre: Información_proyecto	
Tipo de clase: Entidad	
Atributo	Tipo
Código	String
Proyecto	Node_reference
Ci	Int
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

 Figura #10 *Descripción de la clase Información_proyecto*

Acta

Nombre: Acta	
Tipo de clase: Entidad	
Atributo	Tipo
Lugar	String
Fecha	Date
Asistencia	Int
Texto	String
Id	Int
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

 Figura #11 *Descripción de la clase Acta*

Especialista_externo

Nombre: Especialista_externo	
Tipo de clase: Entidad	
Atributo	Tipo
Persona	Node_reference
Especialidad	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

 Figura #12 *Descripción de la clase Especialista_externo*

Profesor

Nombre: Profesor	
Tipo de clase: Entidad	
Atributo	Tipo
Id_profesor	Int
Persona	Node_reference
Polo	Node_reference

Proyecto	Node_reference
Asignatura	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #13 *Descripción de la clase Profesor*

Estudiante

Nombre: Estudiante	
Tipo de clase: Entidad	
Atributo	Tipo
Id_estudiante	Int
Persona	Node_reference
Polo	Node_reference
Proyecto	Node_reference
Grupo	Node_reference
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #14 *Descripción de la clase Estudiante*

Información

Nombre: Información	
Tipo de clase: Entidad	
Atributo	Tipo
Id_información	Int
Texto	String
Título	String
Para cada responsabilidad:	
Nombre:	(4)
Descripción:	(5)

Figura #15 *Descripción de la clase Información*

2.8.1 Propuesta de diseño de la base de datos

Al crear tipos de contenido en Drupal con el módulo CCK, este genera, en la base de datos, las tablas correspondientes a cada tipo contenido, con sus respectivas relaciones. Pero teniendo en cuenta cuan estrechas y delicadas son las relaciones entre los elementos que componen el modelo de datos del sistema en cuestión, pues se realizaron algunos cambios en cuanto a jerarquías y relaciones entre tablas. Cabe señalar, que aunque este último resultó un trabajo algo engorroso, nos hizo mucho más fácil después el diseño de las vistas para generar los reportes tal y como lo demandaba el cliente. Teniendo en cuenta lo anteriormente dicho, y retomando los elementos generales del modelo de datos diseñado por el analista, se propone el siguiente modelo de datos:

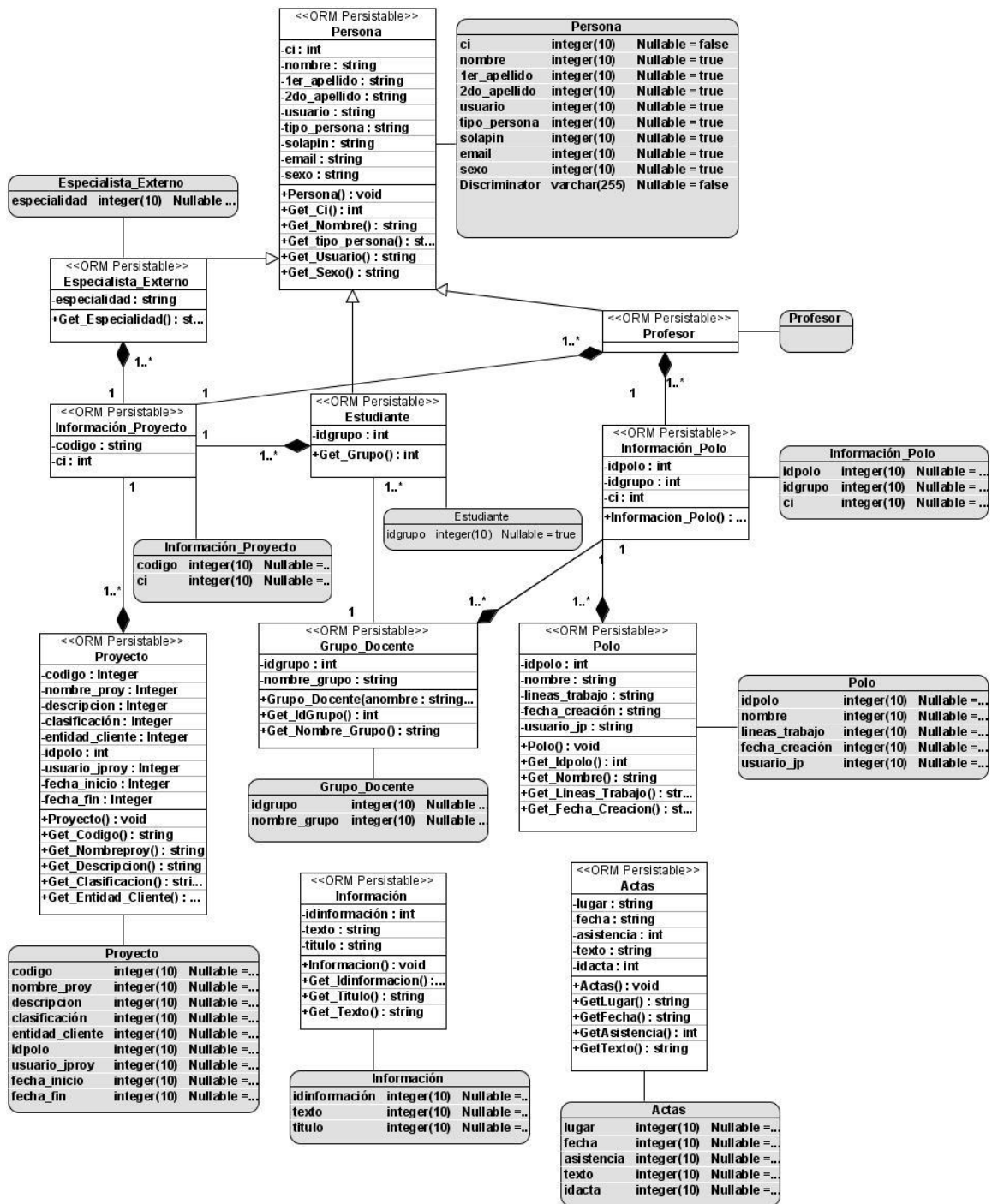


Figura #16 Propuesta de modelo de datos

Conclusiones

De forma general en este capítulo se ha presentado una descripción minuciosa del sistema propuesto, como resultado del análisis y diseño previos propuestos por el analista. Habiendo valorado el trabajo hecho por el mismo, y haciendo uso de las herramientas y metodologías mencionadas en el Capítulo I, en el presente se ha detallado el proceso de implementación, con todas las normas y artefactos que genera la misma, incluyendo la estrategia de integración del sistema, estándares de código utilizados, la propuesta de diseño de la base de datos, así como un análisis del funcionamiento del sistema de gestión de contenidos que se utiliza. Teniendo en cuenta lo anteriormente dicho, y todo el proceso terminado, se ha concluido que la implementación se ha llevado a cabo satisfactoriamente, cumpliendo con las normas y estándares establecidos para el desarrollo de software, y siendo así se encuentra listo para el proceso de pruebas.

Capítulo III: Validación de la solución propuesta.

3.1 Introducción.

Teniendo en cuenta que uno de los mayores problemas que se afronta actualmente en la esfera de la informática es la calidad del software, en el presente capítulo se exponen los casos de pruebas necesarios para validar cada una de las funcionalidades del sistema. Cada test brinda una evaluación objetiva de la eficacia y eficiencia del sistema desde el punto de vista de sus requisitos funcionales, así como del proceso de integración y adaptabilidad a otros sistemas y plataformas. Se darán además recomendaciones acerca del soporte y mantenimiento del mismo.

3.2 Pruebas a realizar en tiempo de desarrollo.

Pruebas informales o fase de prueba informal, son aquellas pruebas que hace el desarrollador en su puesto de trabajo, tienen como objetivo comprobar que el programa compile y ver que todo está yendo como debiera, normalmente se realizan varios cientos de estas pruebas que básicamente consisten en compilar periódicamente durante el desarrollo y ejecutar para ver el resultado.

Dentro de las pruebas en tiempo de desarrollo se encuentran las pruebas de unidades, estas son pruebas de menor escala y consisten en probar cada uno de los módulos que conforman el programa, cuando estos módulos son extensos o complejos se dividen para probar objetivamente partes más pequeñas, este tipo de pruebas es la más común.

Las pruebas de integración tienen por objetivo verificar el conjunto funcionamiento de dos o más módulos, esta se debe poner en práctica desde la creación de dos o más que interactúen entre sí, en el supuesto caso que se necesiten más de dos para efectuar las pruebas, deberán generarse simples

emuladores de módulos que entreguen datos esperados para la prueba individual de cada uno.

También las pruebas de integración pueden ser realizadas en forma ascendente, esto evita tener que crear módulos emuladores, ya que a medida que se va creando la pirámide va siendo probada de abajo hacia arriba (*Down to Top*).

3.2 Pruebas después de la implementación.

Cuando se considera que un módulo está terminado se realizan las pruebas sistemáticas, el objetivo de estas es buscar fallos a través de un criterio específico, estos criterios se denominan "pruebas de caja negra y de caja blanca".

Las pruebas de caja negra son aquellas que se enfocan directamente en el exterior del módulo, sin importar el código, son pruebas funcionales, que se trata de encontrar fallas en las que no se atiene a su especificación, como ser interfaz con el usuario, apariencia de los menús, control de las teclas, etcétera (Rojas & Barrios, 2007).

Este tipo de pruebas no es aplicable a los módulos que trabajan en forma transparente al usuario. Para realizar estas pruebas existe una técnica algebraica llamada "clases de equivalencia", consiste en tratar a todos las posibles entradas y parámetros como un modelo algebraico, y utilizar las clases de este modelo para probar un amplio rango de posibilidades.

Para la generación de las clases no se puede armar un modelo, pero se pueden seguir las siguientes pautas como guía utilizable para la creación de cada clase.

Por ejemplo:

Cuando una entrada es booleana, existen solo dos clases, verdadero o falso.

Para una entrada que está comprendida dentro de un rango, existen tres clases, por debajo, dentro, y por encima del rango.

Utilizando este ejemplo se pueden generar las distintas clases aplicables al módulo en cuestión, luego, se procede a ingresarle al módulo un valor de cada clase.

Las pruebas de caja blanca son mucho más amplias, normalmente se denominan pruebas de cobertura o pruebas de caja transparente, al total de pruebas de caja blanca se le llama cobertura, la cobertura es un número porcentual que indica cuanto código del programa se ha probado. Básicamente la idea de pruebas de cobertura consiste en diseñar un plan de pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya corrido todo o la gran mayoría de él, esto que parece complicado es más aún cuando el programa contiene código de difícil alcance, como por ejemplo manejadores de errores o "código muerto". Entiéndase por código muerto a aquellas funciones y/o procedimientos que se han incluido por encontrarse en recopilaciones pero que estas nunca son ejecutadas por el programa, estas funciones no necesariamente deberán ser removidas pero si probadas por si algún día en revisiones futuras son incluidas.

Para los módulos que no poseen condiciones basta con ejecutar una vez el programa para asegurar una cobertura total. Con respecto a la cobertura en bucles el tema es un poco más delicado, a simple vista un bucle no es mas que un salto condicional que se repite hasta que se cumpla o deje de cumplirse una o más condiciones, en teoría esto es simple, pero en la práctica son una fuente inagotable de versátiles errores, que en su gran mayoría suelen ser catastróficos.

3.3 Diseño de casos de prueba contra requisitos funcionales.

3.3.1 Iteraciones:

R1. Gestionar polo productivo

R1.1. Adicionar polo productivo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un polo que no se encuentra en la base de datos	Inserción de un polo que se encuentra en la base de datos.	Se adiciona el polo correctamente	Se adiciona el polo correctamente	<Observaciones>

R1.2. Modificar polo productivo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un polo que no se encuentra en la base de datos.	Inserción de un polo que se encuentra en la base de datos.	Se modifica el polo correctamente	Se modifica el polo correctamente	Se debe mostrar un listado de los polo creados en la Facultad 1

R1.3. Eliminar polo productivo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un polo que se encuentra en la base de datos.	Inserción de un polo que no se encuentra en la base de datos.	Se modifica el polo correctamente	Se modifica el polo correctamente	Se debe mostrar un listado de los polo creados en la Facultad 1

R2. Gestionar grupos docentes en polo productivo

R2.1 Asignar grupo docente

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un grupo que no se encuentra en la base de datos.	Inserción de un grupo que se encuentra en la base de datos.	Se asigna el grupo correctamente	Se asigna el grupo correctamente	Se debe mostrar un listado de los grupos docentes de la Facultad 1

R2.2 Modificar grupo docente

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un grupo que se encuentra en la base de datos.	Inserción de un grupo que no se encuentra en la base de datos.	Se modifica el grupo correctamente	Se modifica el grupo correctamente	Se debe mostrar un listado de los grupos docentes asignados a un proyecto productivo en la Facultad 1

R2.3 Eliminar grupo docente

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un grupo que se encuentra en la base de datos.	Inserción de un grupo que no se encuentra en la base de datos.	Se elimina el grupo correctamente	Se elimina el grupo correctamente	Se debe mostrar un listado de los grupos docentes asignados a un proyecto productivo en la Facultad 1

R3. Gestionar profesor en polo productivo

R3.1 Asignar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que no se encuentra en la base de datos.	Inserción de un profesor que se encuentra en la base de datos.	Se asigna el profesor correctamente	Se asigna el profesor correctamente	<Observaciones>

R3.2 Modificar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que se encuentra en la base de datos.	Inserción de un profesor que no se encuentra en la base de datos.	Se modifica el profesor correctamente	Se modifica el profesor correctamente	Se debe mostrar un listado de los profesores asignados a un polo productivo en la Facultad 1

R3.3 Eliminar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que se encuentra en la base de datos.	Inserción de un profesor que no se encuentra en la base de datos.	Se elimina el profesor correctamente	Se elimina el profesor correctamente	Se debe mostrar un listado de los profesores asignados a un polo productivo en la Facultad 1

R4. Gestionar proyecto productivo

R4.1 Adicionar proyecto

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un proyecto que no se encuentra en la base de datos.	Inserción de un proyecto que se encuentra en la base de datos.	Se adiciona el proyecto correctamente	Se adiciona el proyecto correctamente	<Observaciones>

R4.2 Modificar proyecto

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un proyecto que se encuentra en la base de datos.	Inserción de un proyecto que no se encuentra en la base de datos.	Se modifica el proyecto correctamente	Se modifica el proyecto correctamente	Se debe mostrar un listado de los proyectos productivos de la Facultad 1

R4.2. Eliminar proyecto

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un proyecto que se encuentra en la base de datos.	Inserción de un proyecto que no se encuentra en la base de datos.	Se modifica el proyecto correctamente	Se modifica el proyecto correctamente	Se debe mostrar un listado de los proyectos productivos de la Facultad 1

R5. Gestionar estudiante en proyecto productivo

R5.1. Asignar estudiante

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un estudiante que no se encuentra en la base de datos.	Inserción de un estudiante que se encuentra en la base de datos.	Se asigna el proyecto correctamente	Se asigna el proyecto correctamente	<Observaciones>

R5.2 Modificar estudiante

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un estudiante que se encuentra en la base de datos.	Inserción de un estudiante que no se encuentra en la base de datos.	Se modifica el estudiante correctamente	Se modifica el estudiante correctamente	Se debe mostrar un listado de los estudiantes asignados a proyectos productivos en la Facultad 1

R5.3. Eliminar estudiante

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un estudiante que se encuentra en la base de datos.	Inserción de un estudiante que no se encuentra en la base de datos.	Se elimina el estudiante correctamente	Se elimina el estudiante correctamente	Se debe mostrar un listado de los estudiantes asignados a un proyecto productivo en la Facultad 1

R6. Gestionar profesor en proyecto productivo

R6.1 Asignar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que no se encuentra en la base de datos.	Inserción de un profesor que se encuentra en la base de datos.	Se asigna el profesor correctamente	Se asigna el profesor correctamente	<Observaciones>

R6.2 Modificar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que se encuentra en la base de datos.	Inserción de un profesor que no se encuentra en la base de datos.	Se modifica el profesor correctamente	Se modifica el profesor correctamente	Se debe mostrar un listado de los profesores asignados a un proyecto productivo en la Facultad 1

R6.3 Eliminar profesor

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un profesor que se encuentra en la base de datos.	Inserción de un profesor que no se encuentra en la base de datos.	Se elimina el profesor correctamente	Se elimina el profesor correctamente	Se debe mostrar un listado de los profesores asignados a un proyecto productivo en la Facultad 1

R7. Gestionar especialista externo en proyecto productivo

R7.1 Asignar especialista externo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un especialista que no se encuentra en la base de datos.	Inserción de un especialista que se encuentra en la base de datos.	Se asigna el especialista correctamente	Se asigna el especialista correctamente	<Observaciones>

R7.2. Modificar especialista externo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un especialista que se encuentra en la base de datos.	Inserción de un especialista que no se encuentra en la base de datos.	Se modifica el especialista correctamente	Se modifica el especialista correctamente	Se debe mostrar un listado de los especialistas asignados a un proyecto productivo en la Facultad 1

R7.3. Eliminar especialista externo

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Inserción de un especialista que se encuentra en la base de datos.	Inserción de un especialista que no se encuentra en la base de datos.	Se elimina el especialista correctamente	Se elimina el especialista correctamente	Se debe mostrar un listado de los especialistas asignados a un proyecto productivo en la Facultad 1

3.4 Propuesta de despliegue.

El módulo formará parte de un sistema (Drupal) ya desarrollado, aunque está en mantenimiento, que utiliza una base de datos con motor PostgreSQL, que puede estar o no físicamente en el mismo ordenador que el propio servidor web. De esta misma forma puede ser opcional el uso de un servidor de aplicaciones. Como bien lo recomiendan la mayoría de los manuales de Drupal, el módulo en cuestión estará localizado en la ruta `sites/all/modules`, y no en la carpeta `modules` que se encuentra en la raíz del sitio, ya que una modificación del módulo puede dejar fuera de servicio todo el sistema. De la integración lógica del módulo se encarga el propio mecanismo interno de Drupal, de la que se deriva su dependencia con los otros 8 módulos ya desarrollados que han sido rehusados por el presente trabajo: *node*, *upload*, *block*, *system*, *page*, *views*, *cck*, *taxonomy*.

3.5 Propuesta de soporte técnico.

La intranet de la Facultad 1 se encuentra hosteado en un laboratorio de desarrollo, por lo que se le podrá dar soporte técnico y mantenimiento especializados con bastante frecuencia. Aunque el sistema es multiplataforma, se recomienda trabajar sobre plataforma libre (GNU/Linux), ya que sus potencialidades desde el punto de vista del rendimiento minimiza cualquier problema de esta índole.

Conclusiones

En este capítulo se hizo una descripción de los casos de prueba diseñados para validar los requisitos funcionales de la aplicación. Se realizó además una propuesta de despliegue del sistema, teniendo en cuenta los lineamientos de arquitectura definidos, por el proyecto Intranet de la Facultad 1.

Conclusiones generales:

Primero que nada cabe señalar que de la necesidad antes expuesta de un software como este en su propio campo de acción, se deriva el impacto social que tendrá. El mismo elevará radicalmente el nivel de calidad y rapidez, en el desempeño de sus responsabilidades, a los trabajadores de la esfera de la producción en la Facultad 1. Se multiplicará el grado de integridad y fidelidad de la información, dado que se va a controlar el acceso a la misma a través de mecanismos tecnológicos, según el grado de implicación que tenga cada usuario con el negocio; y se asegurará la completa disponibilidad de la información, teniendo en cuenta que estará contenida y publicada en la aplicación que prestará servicios en todo momento, contando además con la facilidad que brinda, desde el punto de vista de la reproducción y distribución, la información en formato digital. De manera general, con el desarrollo de este trabajo se alcanzó satisfactoriamente el objetivo propuesto, pues se desarrolló una aplicación que permite gestionar la información que generan las actividades productivas de la Facultad 1, y los recursos humanos correspondientes, cumpliendo así con los requisitos funcionales y no funcionales del cliente. Siendo así se puede concluir que:

- ✓ Se abordaron los principios básicos de la programación, además de las tareas fundamentales que debe desarrollar el programador, según la metodología seleccionada para el desarrollo del sistema.
- ✓ Se realizó un estudio sobre las herramientas definidas para el desarrollo del Sistema.
- ✓ Se expuso la estructura modular de Drupal, así como sus principales funciones, para una mayor comprensión del proceso de implementación.

- ✓ Se expone el resultado del trabajo del programador, acorde con la planificación, para el cumplimiento de la implementación y prueba de cada funcionalidad.

Recomendaciones:

- ✓ Mantener sobre el sistema el proceso de mantenimiento y actualización adecuado, logrando de esta manera una mayor fiabilidad del mismo y de la información que con este se gestiona.
- ✓ Para próximas versiones integrar al sistema funcionalidades relacionadas al proceso de gestión de la información relacionada con el control no solo de los polos y proyectos de manera general sino también de informaciones específicas de cada uno de los integrantes del personal vinculado a las actividades productivas de la facultad, como la evaluación del desempeño, las competencias etc.
- ✓ Someter el sistema a pruebas de calidad para la validación del mismo.
- ✓ Elevar el alcance de la aplicación de forma tal que se pueda desplegar en toda la universidad.
- ✓ Establecer los permisos para acceder a las distintas secciones de la aplicación, según el grado de implicación que tenga cada usuario con el negocio.
- ✓ Desplegar la aplicación en plataforma libre, así (aunque es multiplataforma) sería mucho más fácil darle soporte técnico, por el propio concepto de libre de la misma, además del alto rendimiento que le proporciona a las aplicaciones que corren en ella.
- ✓ Realizar un manual de usuario, con el objetivo de de capacitar al personal necesario para trabajar con la aplicación.

Referencias Bibliográficas

Antes que nada. (2007). Obtenido de <http://lefaltaazucaramicafe.tzhost.org/>

GestProject. (2007). Obtenido de Nuevas Tecnologías aplicadas a la calidad de Gestión de los Estudios de la Arquitectura:

http://www.fidas.es/aases/actividades/jornada_AASES012.html

JavaConGanas. (2008). Obtenido de Metodologías:

<http://www.javaconganas.com/space/path/metodologias>

Programación en Castellano. (2004). Obtenido de

<http://www.programacion.net/noticia/1018/>

Ramón, Y. (2007). *Sistema para el control de proyectos productivos.* UCI.

Sanidad. (2007). Obtenido de ¿Qué son los servicios Web?:

<http://blogs.creamoseelfuturo.com/sanidad/2007/02/20/%C2%BFque-son-los-servicios-web/>

Web, M.d. (2007). Obtenido de Los diferentes lenguajes de programación para la

web: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

UCI(Universidad de las Ciencias Informáticas). (2003). *Portal de la Universidad de las Ciencias Informáticas.* Recuperado el 22 de Abril de 2009, de:

<http://www.uci.cu/?q=node>

Fernández, G. (30 de Abril de 2005). *Estándar codificación DOTNET*. Recuperado el 04 de Junio de 2009, de

http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.htm

Drupal Coding standards. (3 de 10 de 2008). Recuperado el 15 de 3 de 2009, de:

<http://drupal.org/coding-standards>

Oficina de Informática y Telecomunicaciones. (s.f.). *Sistemas de Información Institucional*. Recuperado el 12 de Junio de 2009, de

<http://oitel.univalle.edu.co/sistemas-de-informacion/investigacion.html>

Proyectos Agiles. (s.f.). *proyectosagiles.org*. Recuperado el 2 de Junio de 2009, de

<http://www.proyectosagiles.org/que-es-scrum>

Palacio, J. (2007). *Flexibilidad con Scrum, Principios de diseño e implantación de campos de Scrum*. Octubre: Safe Creative.

Ansueta. (2 de Abril de 2008). *Pyme Crunch*. Recuperado el 18 de Mayo de 2009, de

<http://pymecrunch.com/scrum-metodologia-agil-para-tus-proyectos>

Pozo, J. R. (16 de Junio de 2001). *HTML con Clase*. Recuperado el 21 de Abril de 2009, de

<http://html.conclase.net/tutorial/html/1/3>

Microsoft Corporation. (2009). *Microsoft Office Online, Microsoft Office Project 2007*. Recuperado el 10 de Junio de 2009, de

<http://office.microsoft.com/es-es/project/FX100487773082.aspx?ofcresset=1>

Abartia Team. (2008). *Abartia Team*. Recuperado el 25 de Mayo de 2009, de

<http://www.abartiateam.com/dotproject>

González Hernández, J. D., Reyes-Arias, M., Carbonell-de la Fé, J. S., Pérez González, J. Y., & Ramos-Musibay, M. (2007-2008). *Portal UCI, Universidad de las Ciencias Informáticas*. Recuperado el 1 de Junio de 2009, de

<http://www.uci.cu/files/investigaciones/vol1/SC%20Num%207.pdf>

Visual paradigm, Official Homepage. (2008). Recuperado el 16 de Mayo de 2009, de

<http://www.visual-paradigm.com/news/vpsuite40/vpuml70.jsp>

Linux Para Todos. (2009). *Linux Para Todos*. Recuperado el 20 de Abril de 2009, de

<http://www.linuxparatodos.net/portal/>

Veillard, D. (2008). *RPM resource postgresql8.2*. Recuperado el 21 de Marzo de 2009, de

<http://rpmfind.net/linux/rpm2html/search.php?query=postgresql8.2>

Chin, R. (7 de Febrero de 2007). *Does Drupal 5 has a MVC architecture*. Recuperado el 15 de Mayo de 2009, de

<http://drupal.org/node/116677>

Alvarez, M. A. (16 de Noviembre de 2004). *Desarrolloweb.com*. Recuperado el 12 de Junio de 2009, de

<http://www.desarrolloweb.com/articulos/1696.php>

Rojas, J., & Barrios, E. (2007). *Métodos de prueba de caja negra*. Recuperado el 16 de Mayo de 2009, de Grupo ARQUISOFT:

<http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>

Glosario de términos

COOKIES: Fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. Principalmente se usa para llevar el control de usuarios, ofrecer opciones de diseño (colores, fondos, etc.) o de contenidos al visitante, y/o para conseguir información sobre los hábitos de navegación del usuario.

GPL: Es una licencia creada por la *Free Software Foundation*, está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

INTRANET: es una red de computadoras dentro de una red de área local (LAN) privada, empresarial o educativa que proporciona herramientas de Internet. Tiene como función principal proveer lógica de negocios para aplicaciones de captura, informes y consultas con el fin de facilitar la producción de dichos grupos de trabajo; es también un importante medio de difusión de información interna a nivel de grupo de trabajo.

ONLINE: En general, se dice que algo está en línea, on-line u online si está conectado a una red o sistema mayor.

SOFTWARE: Se refiere al equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica. Incluyen, entre otras, aplicaciones informáticas y software de sistema.

THEMES: en Drupal y otros CMS se emplea este término para llamar al conjunto de estilos CCS, imágenes, scripts, que conforman una interfaz de usuario.

HTTP (*HyperText Transfer Protocol*): protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.

TICs: Las tecnologías de la información y la comunicación son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

CMS: Un Sistema de gestión de contenidos, es un programa que permite crear una estructura de soporte (*framework*) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

FRAMEWORK: Es un marco de trabajo definido en el cual otro software puede ser construido, puede incluir soporte de programas, bibliotecas, entre otros software para ayudar a desarrollar o unir componentes de un proyecto. También se le suele llamar plataforma.

HOOKs: Son funciones de nombre `foo_bar ()`, donde "foo" es el nombre del módulo (cuyo nombre de archivo es, pues, `foo.module`) y "bar" es el nombre del hook. Cada hook tiene un conjunto definido de parámetros y un resultado determinado.

APIs: Una API representa una interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente entre los niveles o capas inferiores y los superiores del software.

CONTENIDO: La carga del producto que se guarda en una infraestructura de almacenaje de datos, o se traslada a través de una de infraestructura de telecomunicación.

CÓDIGO FUENTE: Es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

Anexos



Figura #1. MGAP, página principal.

Principal > MGAP

Polo [Lista](#) [Add](#)

Listado de polos de la Facultad

Nombre	Fecha de creacion		Jefe de polo	Personal		
gestion	2009-06-03	Listado de proyectos	aagutierrez	Editar	Eliminar	RRHH
Informatización	2009-06-03	Listado de proyectos	mag	Editar	Eliminar	RRHH

Figura #2. Listado de polos.

Principal > MGAP > Polo

Datos [Ver](#) [Editar](#) [Eliminar](#)

▼ Datos del polo

Id:
1

Nombre:
Informatización

Lineas de trabajo:
lnkclasnklscas

Fecha de creacion:
2009-06-03

Jefe de polo:
mag

Figura #3. Datos de polo.

Principal > MGAP > Polo

Datos Ver **Editar** Eliminar

▼ Datos del polo

Los campos con * son requeridos

Nombre: *

Líneas de Trabajo: *

Fecha de creación: *
Jun ▼ 3 ▼ 2009 ▼

Usuario del líder: *

Figura #4. Editar polo.

Principal > MGAP > Polo

Datos Ver Editar **Eliminar**

Confirmación

Test

Desea eliminar el polo seleccionado?

Figura #5. Eliminar polo.

Principal > MGAP

Proyecto [Lista](#) [Add](#)

Listado de proyectos(gestion)

Nombre	Entidad cliente	Jefe de proyecto	Personal		
akademos	UCI	ygonzalezp	Editar	Eliminar	RRHH

Figura #6. Listado de proyectos

Principal > MGAP > Proyecto

Datos Ver Editar Eliminar

▼ Datos del proyecto

Código:
6td64t76d

Nombre:
akademos

Descripción:
descripcion

Clasificación:
clasificacion

Entidad cliente:
UCI

Polo:
gestion

jefe del proyecto:
ygonzalezp

Fecha de inicio:
2009-06-03

Fecha de culminación:
2013-06-03

Figura #7. Datos de proyecto.

▼ Datos del proyecto

Los campos con * son obligatorios

Código: *

Nombre: *

Descripción: *

Clasificación: *

Entidad cliente: *

Polo: *

Seleccione un polo

jefe del proyecto: *

Fecha de inicio: *

Figura #8.Editar proyecto.

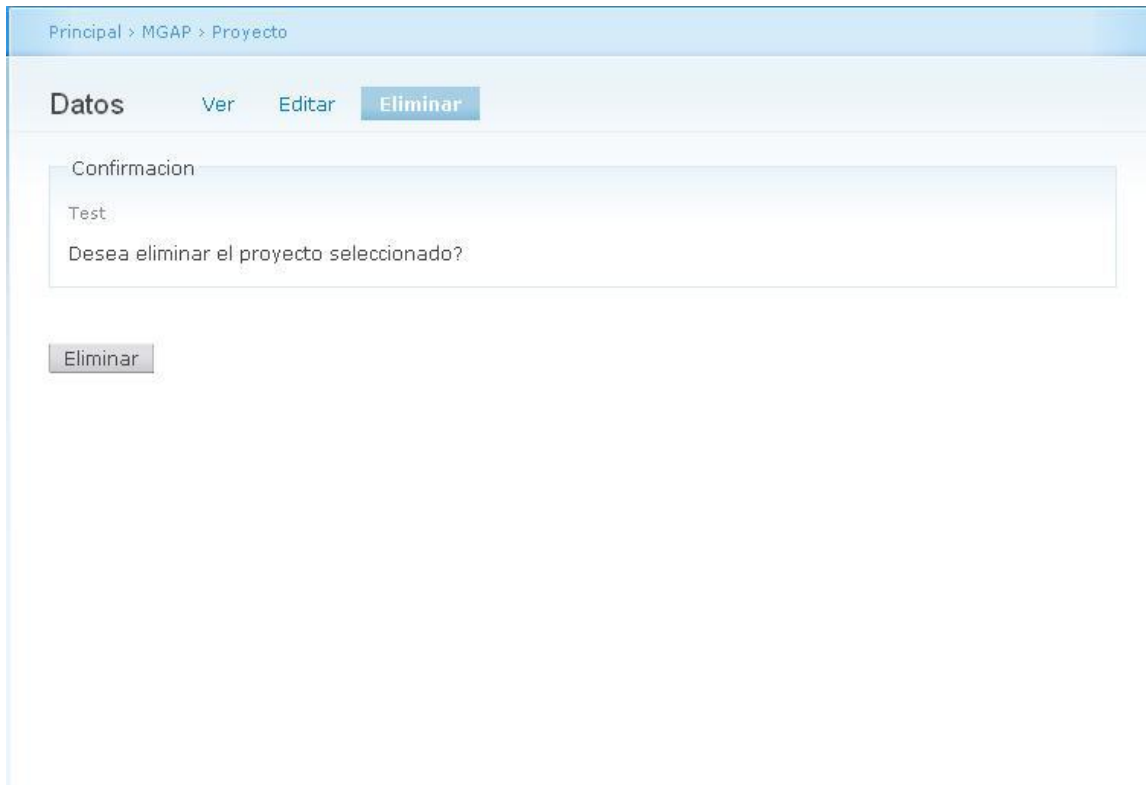


Figura #9. Eliminar proyecto.



Figura #10 Listado de actas de reuniones.

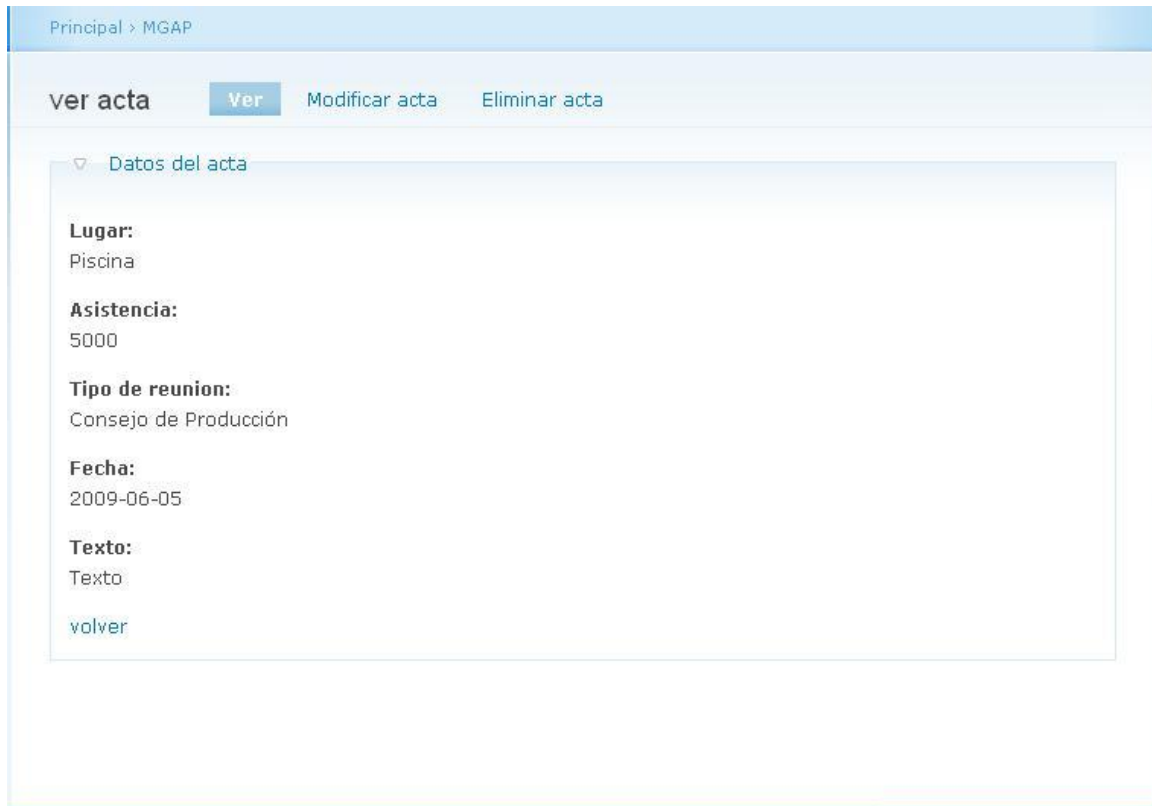


Figura #11. Datos de acta.

ver acta Ver **Modificar acta** Eliminar acta

▼ Datos del acta

Los campos con * son requeridos

Lugar: *

Asistencia: *

Tipo de reunion: *
 ▼
Seleccione un tipo

Fecha: *
 ▼ ▼ ▼

Texto: *

Figura #12.Editar acta.

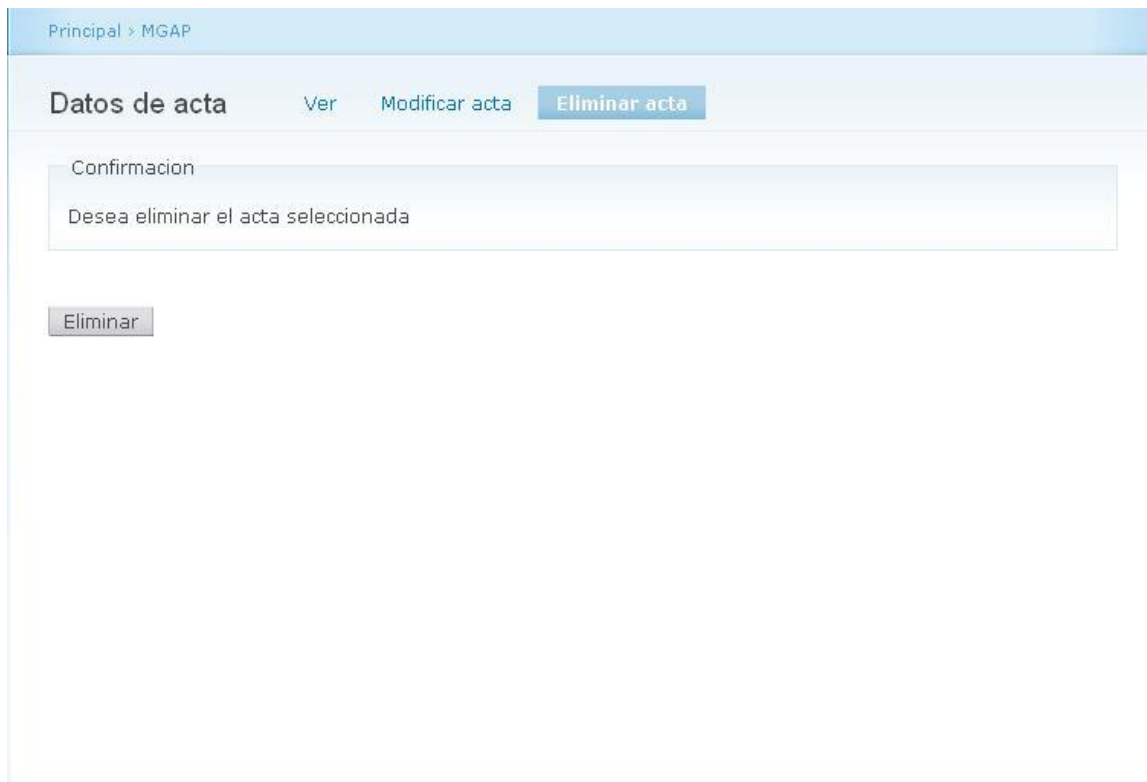


Figura #13. Eliminar acta.



Figura #14. Listado de grupos.

Principal > MGAP

Buscar persona

▼ Buscar persona

Login: *

Insera el usuario UCI

Enviar

Figura #15. Buscar persona.


Datos **Lista** Editar Eliminar

Datos mostrados

El usuario ya esta asignado a un polo

▼ Foto de la Persona

Foto de la persona:



ROSALIDA MORENO ARIAS

Nombre:
Rosalida Moreno Arias

CI:
87011125098

Solapin:
50217

Categoria:
Estudiante

Correo:
rmoreno@estudiantes.uci.cu

-

Figura #16. Datos de persona.

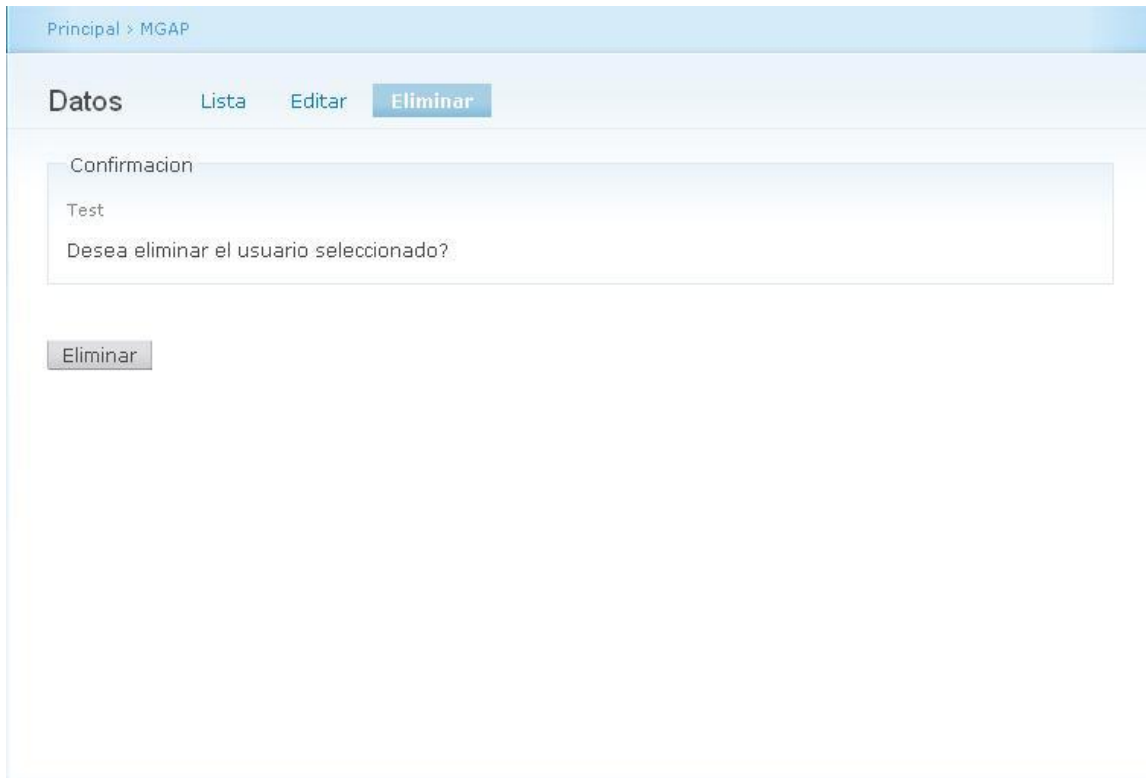


Figura #17. Eliminar persona.



Figura #18. Datos de grupo.

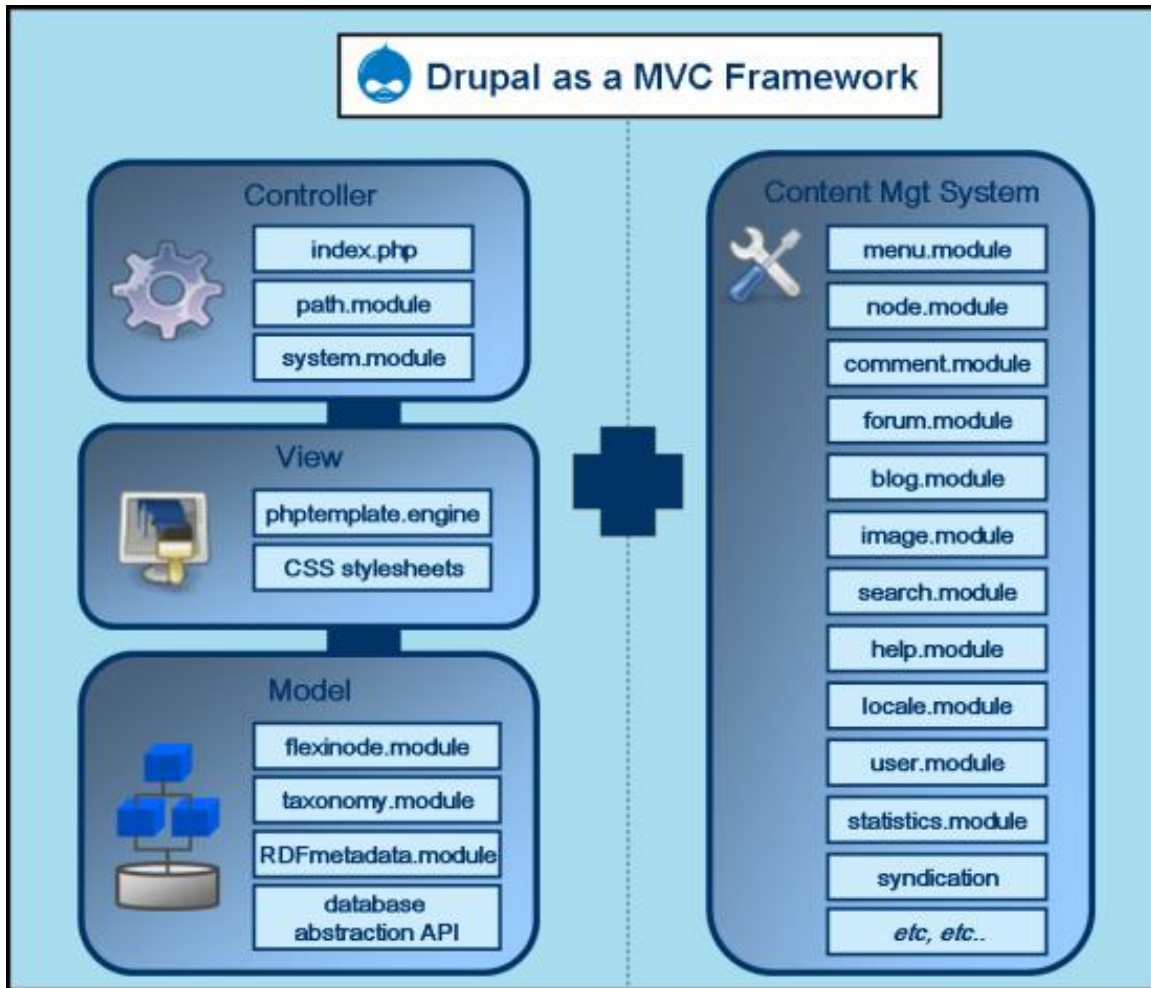


Figura #19. Representación de la arquitectura de Drupal.