



Universidad de las Ciencias Informáticas
Facultad 10

Migración de contenidos de la Intranet Actual hacia Intranet temporal.

Trabajo para optar por el Título de Ingeniería en Informática

Autores: Annalie González Rodríguez
Ernesto Delgado Clavero

Tutor: Ing. Abel Santana Díaz

Ciudad de la Habana, julio del 2007



“Nadie sabe de lo que es capaz de hacer, hasta que trata de hacerlo”...

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

<nombre autor> <nombre tutor>

Firma del Autor Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado, Migración de contenidos de la Intranet actual hacia Intranet temporal, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a

_____.

Y para que así conste, se firma la presente a los ____ días del mes de __ del año 2007.

Representante de la entidad

Firma

Cargo

cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título:

Autores:

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad >

< Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos y propongo que se le otorgue al Trabajo de Diploma la calificación de ____ puntos. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Firma

Fecha

Agradecimientos

...A la Revolución, a la UCI, y a nuestro Comandante Fidel Castro, por habernos dado la posibilidad de realizar nuestros estudios y obtener los conocimientos necesarios para realizar este trabajo.

...A nuestros padres y demás familiares, por el apoyo, el cariño y la confianza que han depositado en nosotros.

...A nuestros compañeros de cinco años de estudio y esfuerzo, por compartir tantos momentos buenos y malos, que durarán en nuestra memoria para siempre, en especial a “el flaco”.

...A Maidely, Susel, Chony, a nuestro tutor Abel, por dedicarnos su tiempo para aclarar nuestras dudas y revisar la tesis.

...A Israel, Deybis, el Rafa, Dionisio, José Ramón, Yanedi, Yusliel, por haber puesto sus granitos de arena en la construcción de la aplicación.

...A Darien por tener tanta paciencia y ayudarnos con sus diseños.

...A Nesty, por darme su amor y comprensión. **Annalie**

...A mi tía, que aunque esté lejos, siempre estará con nosotros. **Annalie**

... A mi hermanita y a mi cuñado Nano, que han sido siempre un ejemplo para mí. **Annalie**

...A Nalita, por su amor y por haber sido mi apoyo en todos estos años. **Ernesto**

...A Jorge Félix por sus buenos consejos. **Ernesto**

... A todos los que no pusimos, pero que de una forma u otra han contribuido a nuestra formación profesional y personal, no se nos olvidan, pero no habría espacio, a todos, muchas gracias.

Dedicatoria

Annalie

*A **mami y a pipo**, que siempre están pensando en mí, dándome apoyo y mucho amor.*

*A **mis abuelas Nelsa y Celia**, por todo su cariño.*

*Y en especial **a mi Taita**, que tiene un pedazo bien grande en mi corazón.*

Ernesto

*A **mis padres** que siempre están a mi lado, por su cariño y por inculcarme las ideas de buenos profesionales.*

*A **mis abuelos** que siempre me apoyan, por todo el amor que me han brindado.*

*A **mi hermana** por seguirme los pasos.*

Resumen

El problema de la gestión y recuperación de la información, ocupa a todos hoy en día. Los grandes volúmenes de información acumulados en la actualidad representan un problema a la hora de escoger cual es la más adecuada para resolver la necesidad de un usuario. La Universidad es un ejemplo claro de lo que está ocurriendo en este sentido.

Las intranets en las empresas son de gran importancia, ya que en la actualidad constituyen una red de servicios casi indispensable, sin ellas no tendríamos un medio de difusión de información interno tan óptimo.

Los contenidos de la intranet de la Universidad de las Ciencias Informáticas, tanto noticias como efemérides, sus ficheros, se manejan actualmente a mano, lo que trae como consecuencia ineficiencia y vulnerabilidades en la seguridad de estos.

Así surge la aplicación para automatizar la migración de contenidos, mediante el uso de un Sistema de Gestión de Contenidos, Drupal, que permite la creación y administración de información en un entorno Web, para poder lograr el manejo de los contenidos de la forma más eficiente.

En este documento se representan los resultados para la construcción del sistema, se explican los conceptos relacionados con el mismo, se hace un razonamiento de la propuesta del sistema, y se dejan algunas observaciones para el mejoramiento futuro del mismo.

Introducción	4
Fundamentación teórica.	8
1.1 Introducción	8
1.2 Bases de Datos	8
1.2.1 Tipos de Base Datos.	9
1.3 Sistemas Gestores de Bases de Datos (SGBD).	13
1.3.1 Propósito	13
1.3.2 Objetivos	13
1.3.3 Ventajas	14
1.3.4 Características principales de algunos Gestores de Base de Datos para Software libre.	15
1.4 Herramientas utilizadas	16
1.4.1 Zend Studio	16
1.4.2 Lenguajes de programación usados en la Web.	16
1.4.3 Otras Herramientas utilizadas	18
1.4.4 Tecnologías usadas.	19
1.5 CMS	19
1.5.1 Tipos de CMS.....	20
1.6 Metodologías de desarrollo de software	22
1.6.1 Metodologías de desarrollo Rational Unified Process (RUP).	22
1.6.2 Extreme Programming (XP)	23
1.6.3 Microsoft Solution Framework (MSF)	24
1.6.4 Comparación	24
1.7 Propuesta	24
1.8 Situación actual de la gestión de la información y de los contenidos a nivel internacional y nacional.	25
1.8.1 Gestión de la información en el mundo.	25
1.8.2 Gestión de la información en Cuba.	25
1.8.3 Gestión de la información y de los contenidos en la UCI.	26

1.9 Descripción del proceso de negocio actual	26
1.10 Conclusiones	26
Características del Sistema.	27
2.1 Introducción	27
2.2 Objeto de estudio.....	27
2.2.1 Problema y Situación problemática	27
2.3 Objeto de automatización	28
2.4 Propuesta del sistema.....	28
2.5 Información que se maneja.....	28
2.6 Modelo de Dominio.....	28
2.7 Especificación de los requisitos de software.	31
2.7.1 Requisitos funcionales	31
2.7.2 Requisitos no funcionales	32
2.8 Definición de los casos de uso	34
2.8.1 Definición de los actores del sistema.	34
2.8.2 Casos de uso del sistema	35
2.8.3 Diagrama de casos de uso del sistema.....	37
2.8.4 Descripción de los Casos de Uso del Sistema.....	38
2.9 Conclusiones	42
Análisis y diseño del sistema.....	43
3.1 Introducción	43
3.2 Análisis.	43
3.2.1 Diagrama de clases de análisis.....	43
3.2.2 Diagrama de Interacción	45
3.3 Diseño.....	46
3.3.1 Diagrama de clases de diseño Web.....	47
3.3.3 Diseño de la Base de Datos.....	48
3.4 Tratamiento de errores.....	61
3.5 Conclusiones	62

Implementación y Prueba	63
4.1 Introducción	63
4.2 Diagrama de despliegue	63
4.3 Diagrama de componentes	64
4.4 Modelo de prueba	66
4.5 Conclusiones	69
Conclusiones	70
Recomendaciones	71
Bibliografía	72
Referencias Bibliográficas	73
Glosario de términos	74

Introducción

Con la evolución que cada día sufren los sistemas de informática, su fácil manejo e innumerables funciones que nos ofrece, se puede decir que igualmente se ha incrementado el número de usuarios que trabajan con computadoras, esto ha constituido una revolución en las operaciones de todas las empresas, otro paso importante ha sido la incorporación de la Internet; una vía de comunicación efectiva y eficaz, donde nos une a todos por medio de una computadora.

También empezó el crecimiento de la intranet, la cual constituye una red de ordenadores, un sistema de administración empresarial o corporativo computarizado, donde los usuarios pueden acceder en busca de soluciones específicas. Cuando se produce esta interacción, se convierte en un vehículo comunicacional. Puede ser una red privada empresarial o educativa.

Es un importante medio de difusión de información interna a nivel de grupo de trabajo. Proporciona una infraestructura de comunicación que permite que la organización se comporte como un grupo. Fue creada para poder compartir archivos, carpetas y recursos. Es una excelente opción de bajo costo para las empresas.

La intranet de la Universidad de las Ciencias Informáticas cumple un papel importantísimo ya que en ella se apoyan soluciones para todo tipo de usuario del centro, dígase reservación de transporte, correo electrónico, directorio. Es decir el usuario acude a la Intranet en busca de una solución.

A pesar de brindarnos resultados favorables tiene una serie de problemas a solucionar. El Servidor generalmente esta congestionado y una Intranet debe de ser rápida, tanto el servidor Web como el de base de datos, deben estar capacitado para soportar los diferentes “clicks” que contra él se estén llevando.

Es decir, debe adaptarse a las necesidades de la organización y a los hábitos de uso de la Intranet que en la misma se crean.

Existe un aumento del número de usuarios en el ámbito universitario. Por tanto ha conllevado a problemas de integridad de la información y de la autogestión de la intranet. No existe personal disponible con

conocimientos del código de la Intranet actual, tampoco existe documentación de la misma. Además hay decadencia natural de un software que no se mantiene. Plantillas diferentes para cada página, no siguen los estándares de HTML y CSS.

Debido a estos problemas ha surgido la necesidad de desarrollar una intranet nueva que se guiará por las características de la UCI, con un diseño diferente, tanto visual como también en cuanto a la arquitectura de información y de desarrollo. Debe de guiarse por las normas de una intranet corporativa, orientada a servicios.

Se decidió hacer la 1ra versión de la intranet en Drupal, porque es flexible y seguro, además de estable. En Drupal aparecen muchos sitios que la UCI ha realizado para proyectos importantes en Venezuela.

Debido a que los ficheros en los que están almacenados los contenidos de la intranet actual, se manipulan a mano, es decir no se encuentran almacenados en una base de datos, lo que se hace muy complejo a la hora de trabajar con estos. La presente tesis se propone dar solución al problema existente mediante el desarrollo de una aplicación con el objetivo de que estos ficheros sean más maniobrables, mediante la migración hacia el gestor de contenido Drupal.

Por tanto surge el siguiente **Problema científico**: ¿Cómo solucionar la organización y el manejo de los ficheros para la nueva intranet mediante una aplicación?

El **Objeto de estudio** es la organización y el manejo de los ficheros para la nueva intranet de la Universidad de las Ciencias Informáticas mediante una aplicación con CMS Drupal.

Delimitando así el **Campo de acción**, siendo este, el manejo de los ficheros para la nueva intranet de la Universidad de las Ciencias Informáticas.

El **Objetivo de la investigación** es realizar una aplicación para responder al problema de la organización y el manejo de los contenidos de la intranet actual de la UCI, permitiendo que el periodista pueda migrar el contenido de las noticias y las efemérides, sin necesidad de poseer conocimientos avanzados de HTML

De este objetivo se derivan los siguientes **objetivos específicos**:

- Facilitar la organización de los ficheros mediante una Base de Datos.
- Posibilitar a los responsables de la aplicación migrar el contenido a través de una interfaz Web.

Se plantearon un grupo de **tareas** que permitiesen realizar una implementación satisfactoria del sistema, ellas son:

- Elaborar la documentación del software siguiendo la metodología del RUP.
- Estudiar el funcionamiento de un Sistema de Gestión de Contenidos fundamentalmente del CMS Drupal.
- Declarar los requisitos que debe cumplir el sistema.
- Análisis de cómo se encuentran en la área internacional las tecnologías que se utilizan para llevar a cabo sistemas como el que se pretende desarrollar.
- Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación. Fundamentando su elección.
- Realizar el diseño y la implementación de una Base de Datos.
- Elaborar una aplicación que satisfaga las necesidades y deseos del cliente.

Cada uno de los capítulos aporta información relevante para lograr el objetivo propuesto. Este documento está estructurado en cuatro capítulos:

- Capítulo I denominado “Fundamentación Teórica” trata la situación de las tecnologías a utilizar en el desarrollo de la aplicación, se comparan y seleccionan las mejores propuestas para el trabajo, y se explican los conceptos principales que se van a tratar.

- Capítulo II denominado “Características del sistema” describe el negocio a través de un modelo de Dominio, y se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, utilizando herramientas de modelación.
- Capítulo III denominado “Análisis y diseño del sistema”, construcción de la solución mediante diagramas de clases, de datos, y se plantean los principios para el diseño. Aquí se construyen las funcionalidades que se definieron en el capítulo anterior.
- Capítulo IV denominado “Implementación y prueba” donde se define la implementación a partir del diagrama de despliegue y el diagrama de componentes, además se determinan las pruebas a realizar en el sistema para verificar su integridad y si se ajusta a los requerimientos planteados.

Por último se plantean posibles mejoras al sistema, bibliografía, referencias bibliográficas, y un glosario de términos.



CAPITULO

Fundamentación teórica.

1.1 Introducción

En el presente capítulo se hace un análisis de las clasificaciones básicas de las bases de datos. Se abordan aspectos relacionados con el uso de las tecnologías y herramientas utilizadas en la informática, y en algunos casos son necesarias comparaciones que fundamentarán la propuesta final.

Son mencionados brevemente algunos gestores de bases de datos, caracterizándolos de forma sencilla y clara, además se toca el tema de las metodologías de desarrollo de software más conocidas, un tema muy común en el mundo de la informática. Se presenta además el concepto de CMS y se caracteriza el CMS Drupal.

1.2 Bases de Datos

Los recursos de información eran uno de los objetivos fundamentales en los sistemas de información que debían hacer frente a grandes cantidades de documentos e información en diferentes formatos. Uno de los componentes principales de estos sistemas son las bases de datos, más concretamente, los sistemas de bases de datos. Por otra parte, la gestión del gran volumen de datos obligará a buscar una teoría sobre la organización de esos datos para alcanzar la máxima efectividad posible.

En el entorno informático, la gestión de bases de datos ocupa un lugar fundamental en los sistemas de información. En la actualidad, un sistema de información será más valioso mientras mayor sea la calidad de la base de datos que lo soporta, la cual resulta a su vez un componente fundamental del mismo, de tal

forma que puede llegarse a afirmar que es imposible la existencia de un sistema de información sin una base de datos, que cumple la función de memoria.

Las bases de datos almacenan, como su nombre dice, datos. Estos datos son representaciones de sucesos y objetos, a diferente nivel, existentes en el mundo real: en su conjunto, representan algún tipo de entidad existente. En el mundo real se tiene percepción sobre las entidades u objetos y sobre los atributos de esos objetos; en el mundo de los datos, hay registros de eventos y datos de eventos. Además, en ambos escenarios se puede incluso distinguir una tercera faceta: aquella que comprende las definiciones de las entidades externas, o bien las definiciones de los registros y de los datos. (ANÓNIMO 2006a)

1.2.1 Tipos de Base Datos.

Las bases de datos pueden clasificarse de acuerdo al criterio elegido para su clasificación.

1.2.1.1 Según la variabilidad de los datos almacenados:

Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones. (ANÓNIMO 2005a)

Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc. (ANÓNIMO 2005a)

1.2.1.2 Según el contenido

Bases de datos bibliográficas

Un registro característico de una base de datos bibliográfica sujeta información sobre el autor, fecha de publicación, editorial, título, edición, de una establecida publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo. Como su nombre lo muestra, el contenido son cifras o números.

Bases de datos de texto completo

Acumulan las fuentes primarias, como por ejemplo, el contenido de todas las ediciones de una colección de revistas informáticas.

1.2.1.3 De acuerdo a su modelo de administración de datos.

Un modelo de datos es un lugar donde se guarda la información, así como de los métodos para almacenar y recuperar la información. Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos.

Bases de datos jerárquicas

La implementación del modelo Jerárquico en los productos se lleva a cabo en base a punteros; estructura física que varía según los productos, e incluso un mismo producto proporciona distintas organizaciones físicas a fin de que el usuario pueda conseguir una mayor eficiencia en el diseño físico de la base de datos.

Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación PADRE/HIJO, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo.

Esta estricta relación PADRE/HIJO implica que no puedan establecerse relaciones entre segmentos dentro de un mismo nivel. (RUIZ 2001)

Las bases de datos jerárquicas son ventajosas en el caso de aplicaciones que manejan un gran número de información y datos muy compartidos permitiendo establecer estructuras estables y de gran rendimiento.

Base de datos de red

El modelo de datos en red general representa las entidades en forma de nodos de un grafo, y las interrelaciones entre estas mediante arcos que unen dichos nodos. En principio esta representación no impone restricción alguna acerca del tipo y el número de arcos que puede haber, con lo que se pueden modelar estructuras de datos tan complejas como sea necesario.

En este modelo, se compone de una componente estática y otra dinámica. La estática estaría compuesta por los objetos (entidades o nodos y atributos), las interrelaciones o arcos y las restricciones, que a su vez pueden ser inherentes (no tenemos en este modelo) y de usuario (pueden ser reconocidas por el modelo de datos o de responsabilidad exclusiva del usuario). Dentro de la componente estática podemos citar un elemento más que atendería a la representación, y son los grafos. (GARCÍA 2001)

Fue un gran avance con respecto al modelo jerárquico, ya que brindaba una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha considerado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Base de datos relacional

En informática, tipo de base de datos o sistema de administración de bases de datos, que almacena información en tablas (filas y columnas de datos) y realiza búsquedas utilizando los datos de columnas especificadas de una tabla para encontrar datos adicionales en otra tabla. En una base de datos relacional, las filas representan registros (conjuntos de datos acerca de elementos separados) y las columnas representan campos (atributos particulares de un registro). Al realizar las búsquedas, una base de datos relacional hace coincidir la información de un campo de una tabla con información en el campo correspondiente de otra tabla y con ello produce una tercera tabla que combina los datos solicitados de ambas tablas. (ANÓNIMO 2001)

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicos. El concepto fundamental es el uso de "relaciones". Estas relaciones podrían ser en forma lógica como conjuntos de datos llamados "tuplas". Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

Bases de datos orientadas a objetos

El modelo orientado a objetos soporta relaciones de muchos a muchos, siendo el primer modelo que lo permite. Aún así se debe ser muy cuidadoso cuando se diseñan estas relaciones para evitar pérdidas de información.

Las bases de datos orientadas a objetos son navegacionales: el acceso a los datos es a través de las relaciones, que se almacenan con los mismos datos. Esto se considera un paso atrás.

La naturaleza navegacional de las bases de datos orientadas a objetos implica que las consultas deben seguir relaciones predefinidas. (MARQUÉS 2002)

Bases de datos documentales

Las bases de datos documentales están concebidas para el procesamiento, captura, almacenamiento, distribución y recuperación de información vinculada con la representación del conocimiento registrado en los documentos. (ANÓNIMO 2004b)

Base de datos deductivas.

Es un sistema de base de datos pero con la discrepancia de que accede a realizar deducciones a través de inferencias. Se basa especialmente en pautas y hechos que son almacenados en la base de datos. También son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

1.3 Sistemas Gestores de Bases de Datos (SGBD).

El software de bases de datos ha experimentado un auge extraordinario a raíz de la progresiva informatización de casi la totalidad de las empresas de hoy día, rapidez, efectividad en los procesos y los grandes flujos de información son la necesidad más apremiante a la hora de optimizar servicios y productos. Ante esta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, estos son programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos.

Para comunicarse con un SGBD, tanto para definir datos y estructuras como para hacer consultas sobre los datos se puede utilizar SQL, que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones.

1.3.1 Propósito

Los sistemas de gestión de base de datos manejan de manera clara, sencilla y ordenada un conjunto de información.

1.3.2 Objetivos

Existen distintos objetivos que deben cumplir los SGBD:

- ❖ **Abstracción de la información.** Los SGBD ahorran a los usuarios pormenores acerca del almacenamiento físico de los datos. Es lo mismo si una base de datos ocupa uno o miles de archivos, este hecho se crea transparente al usuario.
- ❖ **Independencia.** La independencia de los datos consiste en la capacidad de cambiar el esquema (físico o lógico) de una base de datos sin tener que efectuar permutas en las aplicaciones que se sirven de ella.

- ❖ Redundancia mínima. Un buen diseño de una base de datos conseguirá evitar la visión de información repetida o redundante. Lo ideal es alcanzar una redundancia nula; pero, en algunos casos la complicación de los cálculos hace necesaria la visión de redundancias.
- ❖ Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben certificar que esta información se encuentra asegurada frente a usuarios malignos, que pretendan leer información privilegiada; frente a ataques que deseen manejar o destruir la información; o ante las torpezas de algún usuario autorizado pero confundido. Normalmente, los SGBD disponen de un complicado sistema de permisos a usuarios y grupos de usuarios, que permiten consentir diversas categorías de permisos.
- ❖ Integridad. Se trata de acoger las medidas necesarias para responder la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios negligentes, o cualquier otra circunstancia capaz de depravar la información almacenada.
- ❖ Tiempo de respuesta. Lógicamente, es anhelado minimizar el tiempo que el SGBD tarda en darnos la información pedida y en almacenar los cambios efectuados.

1.3.3 Ventajas

- ❖ Habilidad de manejo de grandes volúmenes de información.
- ❖ Gran rapidez en muy poco tiempo.
- ❖ Libertad del tratamiento de información.
- ❖ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, introducciones, consulta.
- ❖ No hay simulación de información, comprobación de información en el momento de implantar la misma.
- ❖ Integridad referencial al concluir los registros.

1.3.4 Características principales de algunos Gestores de Base de Datos para Software libre.

La necesaria introducción del software libre en nuestras vidas cotidianas, ya sea por razones técnicas, económicas o políticas, se ha convertido en una revolución global más allá del mundo de la informática. Para que esta revolución se haga realidad se imponen numerosos retos, entre ellos encontrar alternativas libres que nos garanticen igual o mejor calidad, fiabilidad y eficiencia que las soluciones propietarias.

En la actualidad existen numerosos sistemas gestores de bases de datos, entre ellos: softwares como Oracle y Microsoft SQLServer brindan desde el lado propietario un amplio conjunto de facilidades y potencialidades, pero fuera del alcance de muchos, no solo por sus precios sino también por la falta total de libertades básicas que imponen sus fabricantes.

PostgreSQL, es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS, siglas en inglés) que ha sido desarrollado de varias formas desde 1977. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. (JOHN WORSLEY 2001)

Es un magnífico gestor de bases de datos, es multiplataforma, está disponible en casi cualquier Unix

y una versión nativa de Windows está actualmente en estado beta de pruebas. Permite una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby, etc. El número de base de datos que puede contener es ilimitado.

Tiene una gran Estabilidad y confiabilidad, nunca ha presentado caídas en varios años de operación de alta actividad.

MySQL tal como define propiamente su parte de su nombre SQL, es el servidor de bases de datos relacionales más comúnmente utilizado en GNU/Linux. Es multiplataforma, cualquier persona puede

usarlo y modificarlo para satisfacer sus necesidades, puede ser utilizado gratuitamente. Se encuentra bajo la licencia GPL.

Una de las razones para el rápido crecimiento de popularidad de MySQL, es que se trata de un producto Open Source. Consume muy pocos recursos, se usa tanto en aplicaciones sencillas como complejas. Es utilizado en aplicaciones Web como Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python).

MySQL vs. PostgreSQL

MySQL tiene gran velocidad, flexibilidad y mayor rendimiento. Su principal objetivo de diseño fue la velocidad. Tiene mejor integración con PHP. Consume muy pocos recursos, tanto de CPU como de memoria. Su conectividad y robustez hacen de él un buen sistema gestor de bases de datos.

En el caso de PostgreSQL escala mejor siendo más rápido que MySQL ante gran volumen de datos y conexiones simultáneas, intenta ser un sistema de bases de datos de mayor nivel que MySQL, pero consume más recursos y carga más el sistema, además tiene menos funciones en PHP.

Podemos llegar a la conclusión de que MySQL es indudablemente más rápido, y junto con Apache y PHP forman un buen equipo para servir páginas Web con contenido dinámico, noticias, etc.

1.4 Herramientas utilizadas

1.4.1 Zend Studio

Zend Studio es uno de los ambientes de desarrollo integrado (IDE, siglas en inglés) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

1.4.2 Lenguajes de programación usados en la Web.

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

Existen varios tipos de lenguajes de programación en la actualidad.

PHP es un pre-procesador de páginas HTML, su principal objetivo está encaminado a la construcción de páginas Web.

Es un lenguaje de alta potencia y fácil de usar. Incluye la programación orientada a objetos. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre los que se pueden mencionar, InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. Con PHP se puede hacer cualquier cosa que podemos realizar con un script, como el procesamiento de información en formularios, páginas dinámicas.

¿Por que utilizar PHP y no otras opciones?:

Es multiplataforma, funciona en todas las plataformas que soporten apache. El código PHP es mucho más legible que el de PERL. Al poderse encapsular dentro de código html se puede recoger el trabajo del diseñador gráfico e incrustar el código php posteriormente. Actualmente esta siendo utilizado con éxito en varios millones de sitios Web y en grandes empresas usan PHP como lenguaje de programación Web, entre ellas Cisco, NTT DoCoMo, CMG, Vodafone, Motorola, Siemens, Ericsson, CBS, Unilever, Philips, BMC, NTT, Air Canadá, JAL, Lufthansa, OnVista, Lycos Europe y Deutsche Bank.

PERL ("Processing Extraction Report Language") es un lenguaje dirigido principalmente al tratamiento de archivo de texto. Se utiliza con fuerza para el desarrollo de procesos de mantenimiento de las actividades de un servidor. (ANÓNIMO 2006b)

Es fácil de usar, soporta la programación estructurada, la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas Web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. (ANÓNIMO 2006d)

PHP vs. Python

En PHP podemos hacer todo, trabajar directamente con Java, con .Net con XML con Oracle, PostgreSQL. PHP es muy amigable para desarrollo Web, Python es un lenguaje más general, tipo Perl.

El hecho de que Python sea más limpio no implica que sea el indicado para todo tipo de aplicaciones, el 85% de la documentación actual no trata del desarrollo Web.

PHP es un lenguaje exageradamente fácil de aprender, y que permite crear aplicaciones en un dos por tres, la curva de aprendizaje con respecto a Python y Perl es sin duda la más corta.

1.4.3 Otras Herramientas utilizadas

- **Macromedia Dreamweaver** para el diseño de las páginas Web. El Dreamweaver ha sido históricamente una de las herramientas más poderosas y flexibles para el diseño de sitios Web, el mismo permite gran facilidad para trabajar con los aspectos del diseño de las páginas.
- **Visual Paradigm** para UML, es una de las herramientas considerada como muy completa y fácil de usar, con soporte multiplataforma. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

1.4.4 Tecnologías usadas.

HTML es el lenguaje con el que se definen las páginas Web.

Es el lenguaje más utilizado para la presentación de textos estructurados en formato hipertexto, estándar de las páginas Web. Es muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida pueda enfrentarse a la tarea de crear una Web.

Un documento HTML puede ser mostrado en cualquier tipo de computadora. Esto significa, que sólo se necesita crear un documento HTML y editarlo, a fin de que pueda ser observado, por el usuario que así lo desee.

CSS (Cascading Style Sheets, Hojas de estilos)

Las hojas de estilos son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. Es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación, lo que hace que sea más limpio, y se pueda acceder mucho más fácil. (BARCIA 2003)

1.5 CMS

Un Sistema de Gestión de Contenidos (CMS) es una herramienta que permite la creación y administración de contenidos principalmente en páginas Web.

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas Web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG, en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio Web. (ANÓNIMO 2005b)

Consiste en una interfaz que controla una o varias bases de datos donde se alberga el contenido del sitio. El sistema permite maniobrar de manera independiente el contenido por una parte y el diseño por otra.

Así, es posible manejar el contenido y darle en cualquier momento un diseño diferente al sitio sin tener que darle formato al contenido de nuevo, además de permitir la factible y controlada publicación en el sitio a varios editores.

1.5.1 Tipos de CMS

Según el tipo de licencia escogido los CMS presentan dos clasificaciones: los CMS de código abierto y los CMS comerciales.

1.5.1.1 CMS de código abierto para la Web

Los CMS de código fuente abierto son desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación de dicho código fuente.

La disponibilidad del código fuente posibilita que se hagan personalizaciones del producto, correcciones de errores y desarrollo de nuevas funciones. Este hecho es una garantía de que el producto podrá evolucionar incluso después de la desaparición del grupo o empresa creadora. (ANÓNIMO 2004a)

Existen varios CMS para Web basados en código abierto como: ASP Nuke, Drupal, Plone.

CMS Drupal

Drupal es un marco de administración de contenidos, sistema de administración de contenidos y motor de blogging basado en PHP, que fue originalmente escrito por Dries Buytaert.

Esto significa que es un sistema para administrar el contenido de sitios Web, tal como artículos, fotos, u otros archivos. Drupal es sistema “dinámico”; en lugar de ser archivos pre-generados (estáticos), el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos.

Drupal se distribuye bajo la licencia GNU GPL, y por lo tanto es software libre.

Ventajas

Consta de un núcleo central de funcionalidades que pueden ampliarse mediante módulos programables. Existen módulos casi para cualquier cosa que queramos imaginar. Posee un potente módulo de

Búsqueda, ya que todo el contenido en Drupal es indexado en tiempo real y se puede consultar en cualquier momento. Es fácilmente traducible a otros idiomas y soporte multilingüe. Nos da la posibilidad de adaptar y modificar el sitio como uno quiera, ya sea en el plano de diseño como la forma de gestionar módulos, usuarios y de gestionar los contenidos.

CMS Plone

Plone es un CMS muy fácil de utilizar, fue diseñado principalmente para la publicación de documentos de manera dinámica en grupos de trabajo, por lo que cuenta con un excelente manejador de flujo de trabajo. También cuenta con módulos para blog, foros, álbumes fotográficos, mensajería instantánea y otras.

Usa Python como lenguaje subyacente y Zope como servidor de aplicaciones. Su uso puede ser como servidor para intranet o extranet, como sistema de publicación de contenidos o como herramienta de groupware (trabajo en grupo)

Drupal vs. Plone

Plone, tiene su propio servidor Web y bases de datos; es muy amplio y hay que dedicarle mucho tiempo para exprimir sus funciones; se consumen muchos recursos personales formando a los usuarios; tiene altos requisitos de sistema, y un desarrollo costoso, además Python es un lenguaje de poco crecimiento en Web.

Mientras que Drupal soporta estándares Web, es bien sencillo a la hora de manejar el código. Se ha convertido en más que un simple portal para noticias gracias a su flexible estructura.

1.5.1.2 CMS comerciales para sitios Web

Los CMS comerciales son aquellos desarrollados por empresas para la venta. Esos no permiten el acceso al código fuente para modificaciones.

En cuanto a soporte, los CMS comerciales acostumbran a dar soporte profesional, con un coste elevado en muchos casos.

Existen varios CMS comerciales para Web como: CoreMedia CMS, PipePS, NUKE ET.

1.6 Metodologías de desarrollo de software

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

El uso de una metodología para la elaboración de un producto informático, garantiza determinadas características en el mismo, dentro de ellas la calidad, factor clave tanto para el cliente como para el productor.

El tiempo es uno de los factores críticos que afecta todo producto, más si no se ha hecho un buen estudio de la aplicación que se va a realizar. La cantidad de personal; algunas veces en exceso, otras en déficit, los sistemas de organización, los métodos de control, el dominio sobre el tema y sobre las herramientas de desarrollo por parte de los analistas y programadores, la falta de conocimientos sobre asuntos informáticos por el lado de los clientes, son otros de los factores que afectan todo el ciclo de desarrollo de una aplicación.

Sin duda alguna las metodologías que existen actualmente se adecuan al desarrollo de la mayoría de las aplicaciones, puesto que han surgido de la experiencia en la producción acumulada por varios años, incluso por varias décadas; muchas de ellas son el resultado de la evolución y desarrollo de otras unido al cambio producido en el mundo informático, influenciado por el cambio de paradigmas en la programación.

1.6.1 Metodologías de desarrollo Rational Unified Process (RUP).

Basado en UML para la ingeniería de sistemas y de software. Esta metodología proporciona al equipo del proyecto procedimientos y herramientas.

Según la metodología RUP el ciclo de vida de un proyecto se divide en las siguientes fases:

- Fase de concepción
- Fase de elaboración
- Fase de construcción

- Fase de transición

El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

Ingeniería de Negocios: Entendiendo las necesidades del negocio.

Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.

Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.

Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

Configuración y administración del cambio: Guardando todas las versiones del proyecto.

Administrando el proyecto: Administrando horarios y recursos.

Ambiente: Administrando el ambiente de desarrollo.

Distribución: Hacer todo lo necesario para la salida del proyecto

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.6.2 Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (MENDOZA 2004)

La metodología se basa en las pruebas realizadas a los principales procesos. En la reutilización de código, siendo más flexible al cambio. Además propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

1.6.3 Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (MENDOZA 2004)

Es flexible, adaptable, usado en cualquier parte. Además de escalable, puede organizar equipos, así como también, proyectos que requieren 50 personas a más.

1.6.4 Comparación

La Metodología RUP es más adaptable para proyectos de largo plazo. La Metodología XP en cambio, se recomienda para proyectos de corto plazo. La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología. Podemos concluir además, que lo más importante antes de elegir la metodología que usarás para la implementación de tu software, es determinar el alcance que tendrá y luego de ahí ver cual es la que más se acomoda en tu aplicación.

1.7 Propuesta

Basándose en los elementos expuestos anteriormente se propone usar el Zend Studio como herramienta de desarrollo para construir una aplicación Web, sobre la plataforma Drupal, que es un sistema de Gestión de Contenidos potente, flexible, y a la vez sencillo. Como gestor de Bases de datos se propone el MySQL por sus características y por la completa integración de PHP con el mismo.

1.8 Situación actual de la gestión de la información y de los contenidos a nivel internacional y nacional.

1.8.1 Gestión de la información en el mundo.

En muchas organizaciones la gestión de la información se ha entendido tradicionalmente como la gestión de los datos. En el mundo de las grandes organizaciones el mantenimiento y explotación de estas bases de datos es uno de los pilares de la gestión de la información y probablemente el que mayor inversión requiere. En este sentido tienen una gran popularidad los sistemas ERP (Enterprise Resource Management), que pretenden unificar en un solo sistema toda la información que se maneja, y que se han implantado en casi el 100% de las grandes organizaciones. Actualmente, también tienen una gran popularidad los sistemas CRM (Customer Relationship Management), que recogen en bases de datos toda la información relacionada con los clientes.

1.8.2 Gestión de la información en Cuba.

Cuba, como otros muchos países, está apostando por la informatización de la sociedad a partir del uso del software libre en un esfuerzo por socializar y masificar el uso de estas tecnologías.

Frente a la alternativa del Software Propietario están ganando popularidad y relevancia las herramientas Open Source, la habilidad para modificar y distribuir de manera gratuita los programas informáticos la convierten en una buena opción para los desarrolladores y los usuarios finales.

Están destinadas a facilitar el flujo de trabajo en red y la gestión de contenidos hacia La Biblioteca Virtual en Salud, perteneciente al Portal en Salud de Cuba (INFOMED). La herramienta Greenstone Digital Library ha dado paso a una nueva filosofía de trabajo para el tratamiento y recuperación de la información digital en la organización y con la incorporación del Scout Software Portal Toolkit, se ha optimizado la manera de gestionar los catálogos de fuentes de información en ambiente digital de los que dispone la organización.

1.8.3 Gestión de la información y de los contenidos en la UCI.

En La Universidad de las Ciencias de las Informáticas, los contenidos de la intranet actual, en los ficheros que se guardan, se manipulan a mano, por lo que se hace muy complejo a la hora de trabajar con estos. No se permite la publicación de nuevas noticias, se debe eliminar y poner la otra nueva. Se pierden los títulos de estas, quedándose el texto de la noticia en los archivos, por tanto no se conoce el texto a que noticia pertenece, y debido a esto pueden que hayan textos de noticias repetidas.

1.9 Descripción del proceso de negocio actual

Actualmente en la intranet de la Universidad de las Ciencias Informáticas la gestión de los contenidos, los ficheros de noticias y efemérides son manejados a mano, por lo que resulta muy difícil a la hora de trabajar con estos.

1.10 Conclusiones

En este capítulo se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión de este trabajo. Además se realizó un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar. Una vez conocidas las herramientas óptimas, y los conceptos a utilizar se puede empezar a desarrollar la propuesta del sistema.



CAPITULO

Características del Sistema.

2.1 Introducción

En el presente capítulo se hace la descripción de la propuesta que trae este trabajo, para ello se describen los procesos del negocio que tiene que ver con el objeto de estudio, de acuerdo a esto se llega a la conclusión que debido a la poca estructuración de esos procesos, para poder entender el contexto en que se emplaza el sistema necesitamos definir conceptos que podemos agrupar en un Modelo de Dominio, para capturar correctamente los requisitos y poder construir un sistema correcto.

Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que proponemos, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.2 Objeto de estudio

2.2.1 Problema y Situación problemática

La información que se maneja en la intranet de la Universidad de las Ciencias Informáticas, tanto las noticias como las efemérides, se encuentra en ficheros, no existe una base de datos donde se almacena todo el contenido, esto conlleva a pérdidas de imágenes, información, repetición de los artículos de las noticias y las efemérides. Incurrir también en la seguridad, dígase integridad, pues la información puede ser manipulada o puede perderse, disponibilidad, ya que es de difícil acceso. Se necesita de una

aplicación que permita manipular los contenidos de estos ficheros de manera que estos no se expongan a riesgos de ser cambiado o extraviados.

Actualmente no se cuenta con ningún sistema en la UCI que automatice la gestión de los contenidos en la intranet, son manejados manualmente.

2.3 Objeto de automatización

El proceso que será automatizado es la organización de los contenidos de los ficheros de las noticias y las efemérides referentes a la intranet de la Universidad de las Ciencias Informáticas, el cual servirá como herramienta para el personal que trabaje con el mismo.

2.4 Propuesta del sistema

Con el fin de cumplir con los objetivos y requerimientos planteados en este trabajo, el sistema propuesto debe poseer 1 módulo, encargado de la administración de el mismo y un menú que se encargue de migrar los contenidos de la intranet hacia la base de datos de la intranet temporal.

2.5 Información que se maneja

La información que se maneja en la Intranet actual sobre los contenidos de las noticias y las efemérides, es que están almacenados en ficheros propietarios, los cuales tienen registrados la fecha, título, y la imagen. Particularmente las noticias se encuentran agrupadas en categorías (universitarias, internacionales, nacionales, deportivas, relevantes e interesantes).

En el caso de los artículos y las imágenes de las noticias y las efemérides, están guardados en otras carpetas que están dentro de estos contenidos respectivamente.

2.6 Modelo de Dominio

El primer paso dentro del modelado del negocio es la identificación de los diferentes procesos del negocio de la organización. La obtención de un adecuado conjunto de procesos del negocio es una cuestión crucial puesto que establece los límites del modelado.

Después de haber realizado un estudio de los procesos que se van a efectuar, se llega a la conclusión de que el negocio que se está estudiando, tiene muy bajo nivel de estructuración.

Para ello se utilizará un modelo del dominio, ya que permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados; a utilizar un vocabulario común para poder entender el contexto en que se ubica el sistema.

Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita conocimiento de cómo debe funcionar el proceso en cuestión. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Primeramente hay que identificar todos los **conceptos** que se utilizarán en el diagrama, mediante un glosario de términos sobre los nombres:

UCI: Universidad de las Ciencias Informáticas, en donde se desarrollan los procesos.

Proyecto Intranet2: Entidad dentro de la UCI que se encarga de dirigir todas las actividades.

Sitio Intranet2: Sitio Web que contiene las publicaciones de la Universidad.

Aplicación: Aplicación para migrar contenido de las efemérides y las noticias.

Noticia: Los acontecimientos, que estén ocurriendo tanto en la Universidad de las Ciencias Informáticas, como en Cuba, y en el mundo, que por su contenido sea de interés para un usuario.

Efeméride: Es el nombre dado a una fecha importante, es decir, sucesos notables ocurridos en la fecha en que se está, pero en años anteriores.

Responsable: La persona encargada de insertar el contenido de las noticias y las efemérides en la base de datos.

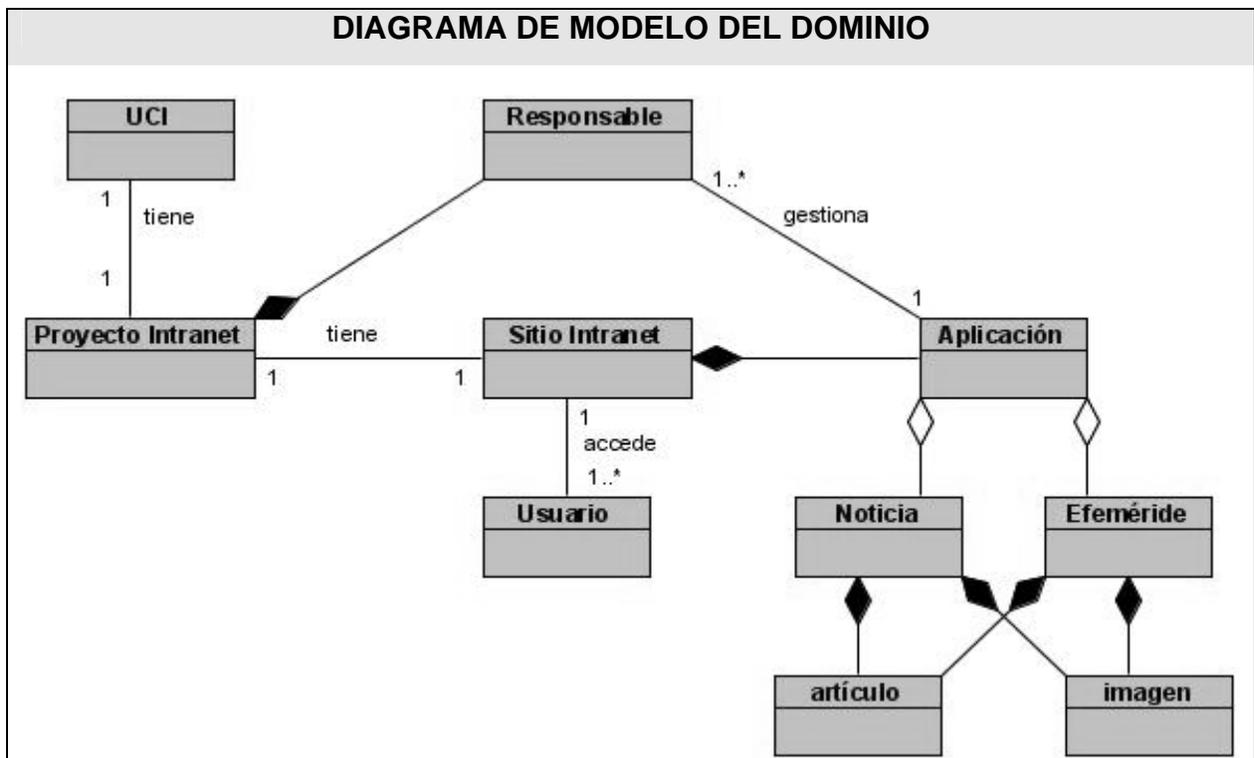
Usuario: Es la persona que cuando desee entra al sitio para buscar alguna información.

Artículo: Tratan de un tema en específico con detalles relevantes.

Imagen: Es una representación visual de un objeto mediante técnicas distintas de diseño, pintura, fotografía.

Diagrama del Modelo de Dominio:

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.



2.7 Especificación de los requisitos de software.

2.7.1 Requisitos funcionales

Una vez conocidos los conceptos que rodean al objeto de estudio, se debe analizar: ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello se enumeran, a través de requerimientos funcionales, las acciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

- **R1** Autenticar usuario: Controlar que cada usuario acceda a la información que le corresponda
 - **R1.1** Comparar Usuario y contraseña con los usuarios del sistema.
 - **R1.2** Definir niveles de acceso.
- **R2** Migrar contenido.
 - **R2.1** El sistema debe ser capaz de migrar el contenido de las noticias y las efemérides.
 - **R2.2** Migrar contenido de las efemérides.
 - **R2.2.1** El sistema debe ser capaz de leer el fichero de los contenidos de las efemérides.
 - **R2.2.2** El sistema debe ser capaz de guardar los contenidos los ficheros en la base de datos.

- **R2.3** Migrar contenido de las noticias.
 - **R2.3.1** El sistema debe ser capaz de leer el fichero de los contenidos de las noticias.
 - **R2.3.2** El sistema debe ser capaz de guardar los contenidos de los ficheros en la base de datos.
- **R3** Administrar los usuarios del sistema.
 - **R3.1** Registrar los datos del usuario.
 - **R3.2** Actualizar los datos del usuario.
 - **R3.3** Eliminar los datos del usuario.

2.7.2 Requisitos no funcionales

Los requisitos no funcionales del sistema son propiedades o cualidades que el producto debe presentar. Son las características que hacen al producto atractivo, usable, rápido o confiable. La razón fundamental de que estos requisitos formen parte del producto es brindarle a este las características deseadas.

Usabilidad:

- Para utilizar el sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente Web en sentido general.
- El sistema podrá ser usado por personas que estén autorizadas.

Apariencia o interfaz externa:

- La interfaz de usuario debe ser sencilla, intuitiva, amigable y mantener el formato en páginas similares. En general, debe ser fácil de usar.

Rendimiento:

- El tiempo de respuesta debe ser corto. El acceso a la información debe ser rápido y de la manera más fácil posible.
- La base de datos debe estar normalizada, para garantizar el rendimiento óptimo de la misma.

Soporte:

- Se requiere MySQL 4.0.13 como servidor de base datos.
- Servidor Web Apache 1.3.x o superior.
- El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.
- Las pruebas realizadas al sistema deben permitir evaluar sus ventajas y funcionalidades, además de detectar los errores que presenta.
- El sistema deberá ser probado, instalado y configurado por los especialistas que también se ocuparán de su mantenimiento.
- La carpeta con los contenidos de los ficheros de las noticias y las efemérides, deben estar guardados en el servidor del sitio Intranet2.

Portabilidad:

- Necesidad de que el sistema sea multiplataforma.

Seguridad:

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del portal.

- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

Políticos-culturales:

- El sistema debe estar completamente identificado con las principales características políticas-culturales de este país.

Legales:

- El empleo de este producto no deberá violar ninguna ley o licencia asociada a este o a terceros.

Confiabilidad:

- La información contenida en el sistema debe ser totalmente confiable.
- Deben implementarse mecanismos para garantizar la respuesta ante posibles fallos lo más rápido posible.

2.8 Definición de los casos de uso

2.8.1 Definición de los actores del sistema.

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello hay que definir de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

Un actor no es parte del sistema, sino un rol que se juega dentro del sistema, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. En este caso interactúan tres actores que se definen a continuación:

Nombre del actor	Justificación
Usuario	Es un usuario que generaliza el rol de autenticación al sistema.
Periodista	Es la persona encargada de migrar los contenidos mediante la aplicación.
Administrador del sistema	Es el encargado de la configuración de la aplicación, así como asignar permisos a los demás usuarios.

2.8.2 Casos de uso del sistema

Los casos de uso del sistema que aparecen a continuación tienen como objetivo satisfacer los requisitos funcionales descritos con anterioridad.

CU-1	Autenticar usuario
------	--------------------

Actor	Usuario
Descripción	El usuario se identifica ante el sistema para recibir los permisos que hayan sido definidos para él.
Referencia	R1

CU-3	Migrar contenido de noticias.
Actor	Periodista
Descripción	El periodista accede al sistema para migrar el contenido de las noticias.
Referencia	R2, CU-1

CU-4	Migrar contenido de efemérides.
Actor	Periodista
Descripción	El periodista accede al sistema para migrar el contenido de las efemérides.
Referencia	R2, CU-1

CU-5	Administrar sistema
Actor	Administrador
Descripción	El administrador accede al sistema para definir la configuración de la aplicación, dar permisos a los usuarios.
Referencia	R3, CU-1

2.8.3 Diagrama de casos de uso del sistema

El diagrama donde se representa la relación existente entre los actores y los casos de uso se representa a continuación:



2.8.4 Descripción de los Casos de Uso del Sistema

Caso de uso	
CU-1	Autenticar usuario
Propósito	Permitir autenticarse.
Actores:	Usuario
Resumen:	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.
Referencias	R1
Acción del actor	Respuesta del sistema
1. El usuario accede al sistema insertando los datos de usuario y contraseña.	1.1 El sistema encripta la contraseña. 1.2 Busca el usuario y compara la contraseña. 1.3 En caso de ser correcto se le asignan los permisos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1 Si los datos son incorrectos el sistema devuelve el mensaje "Disculpe. No se reconoce el nombre del usuario o la contraseña".

Caso de uso	
CU-2	Migrar contenido de efemérides
Propósito	Migrar los contenidos de las noticias para la base de datos.
Actores:	Periodista
Resumen:	El caso de uso se inicia cuando el periodista accede al sistema para migrar los contenidos de los ficheros en la base datos.
Referencias	R2, CU-1
Acción del actor	Respuesta del sistema
1. El periodista accede a la opción “migrar contenido de efemérides”.	<p>1.1 El sistema lee los ficheros de las noticias.</p> <p>1.2 El sistema migra los contenidos en la base de datos y así finaliza el caso de uso.</p>
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Caso de uso	
CU-3	Migrar contenido de noticias
Propósito	Migrar los contenidos de las noticias para la base datos.
Actores:	Periodista
Resumen:	El caso de uso se inicia cuando el periodista accede al sistema para migrar los contenidos de los ficheros en la base datos.
Referencias	R2, CU-1
Acción del actor	Respuesta del sistema
1. El periodista accede a la opción “migrar contenido de noticias”.	<p>1.1 El sistema lee los ficheros de las noticias.</p> <p>1.2 El sistema migra los contenidos en la base de datos y así finaliza el caso de uso.</p>
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Caso de uso	
CU-4	Administrar sistema.
Propósito	Permitir la administración del sistema.
Actores:	Administrador del sistema
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema para definir la configuración del sitio, dar permisos a los usuarios y roles.
Referencias	R3, CU-1
Acción del actor	Respuesta del sistema
1. El administrador del sistema accede a la opción "administrar".	1.2 El sistema muestra el módulo de administración.
2. El administrador del sistema accede a la opción que desea modificar.	2.1 El sistema muestra la información correspondiente a la opción seleccionada.
3. El administrador del sistema realiza las operaciones deseadas.	3.1 El sistema guarda los cambios realizados y finaliza el caso de uso.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

2.9 Conclusiones

A partir del análisis de los procesos del negocio comenzó a desarrollarse la propuesta de solución del sistema, obteniéndose las funciones que debe tener el mismo, las cuales se representaron mediante un Diagrama de Casos de Uso, y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Una vez realizado esto es posible comenzar a construir el sistema teniendo en cuenta el cumplimiento de los requerimientos especificados en el capítulo.



CAPITULO

Análisis y diseño del sistema.

3.1 Introducción

El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica: cómo el sistema satisface los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo.

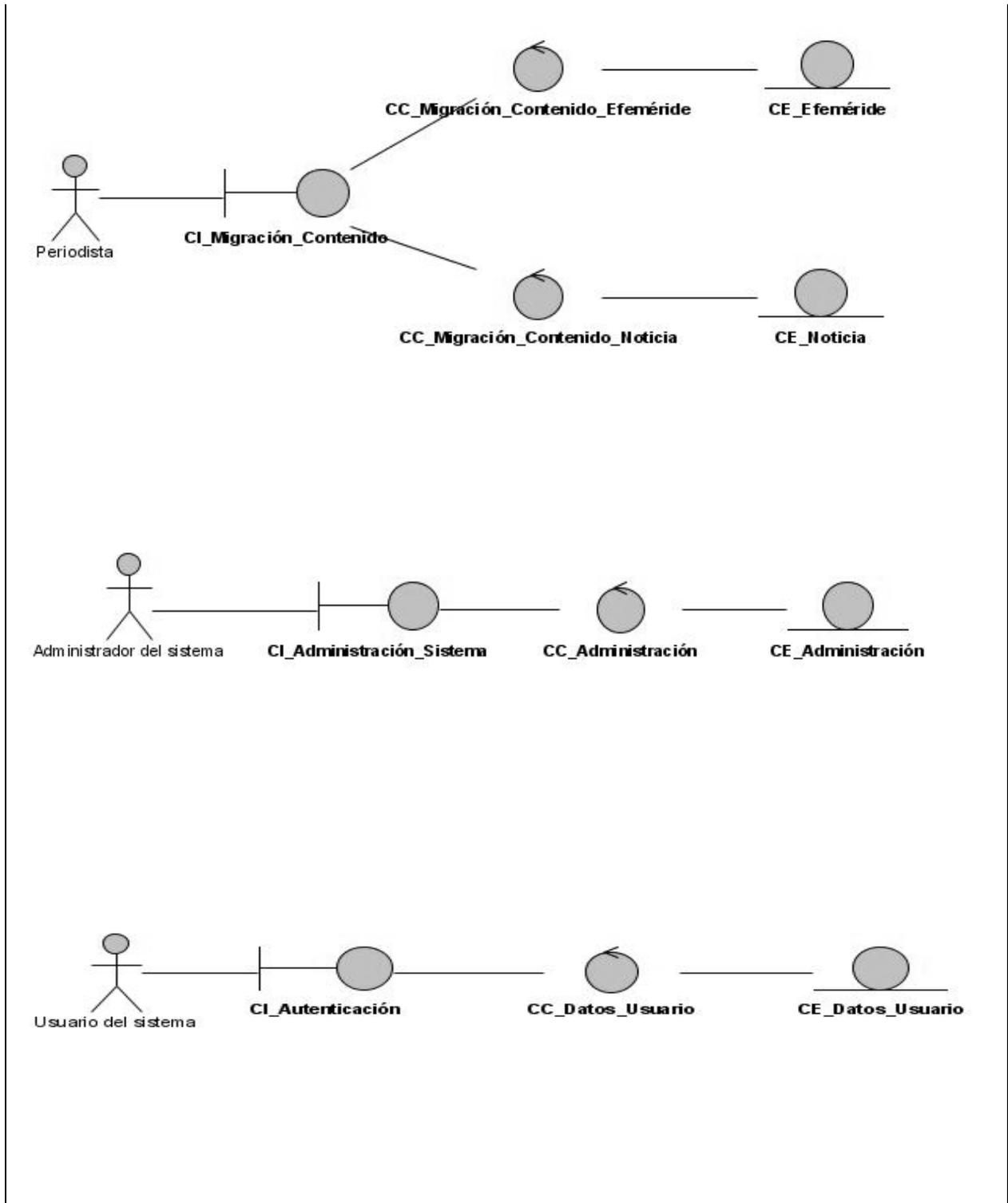
En este capítulo se describen los elementos más importantes correspondientes a la etapa de análisis y diseño del sistema, entre los que se encuentran los diagramas de interacción del sistema por cada realización de casos de uso, así como el diagrama de clases del diseño. Además se realiza la descripción de las clases, se diseña la base de datos y se describen las tablas.

3.2 Análisis.

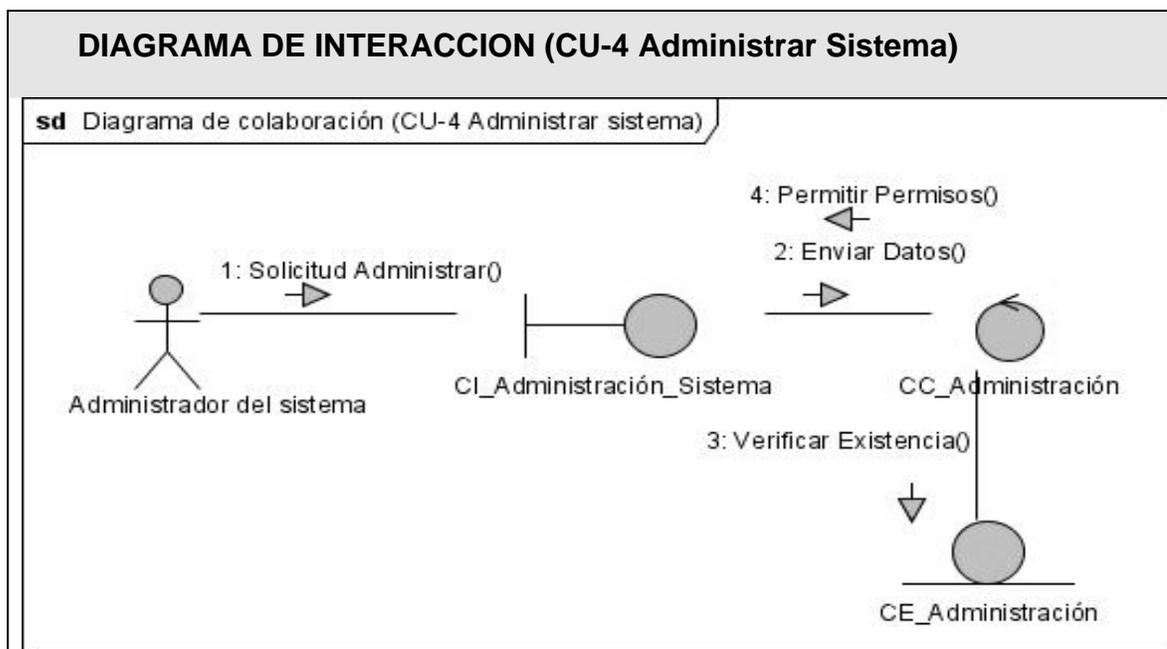
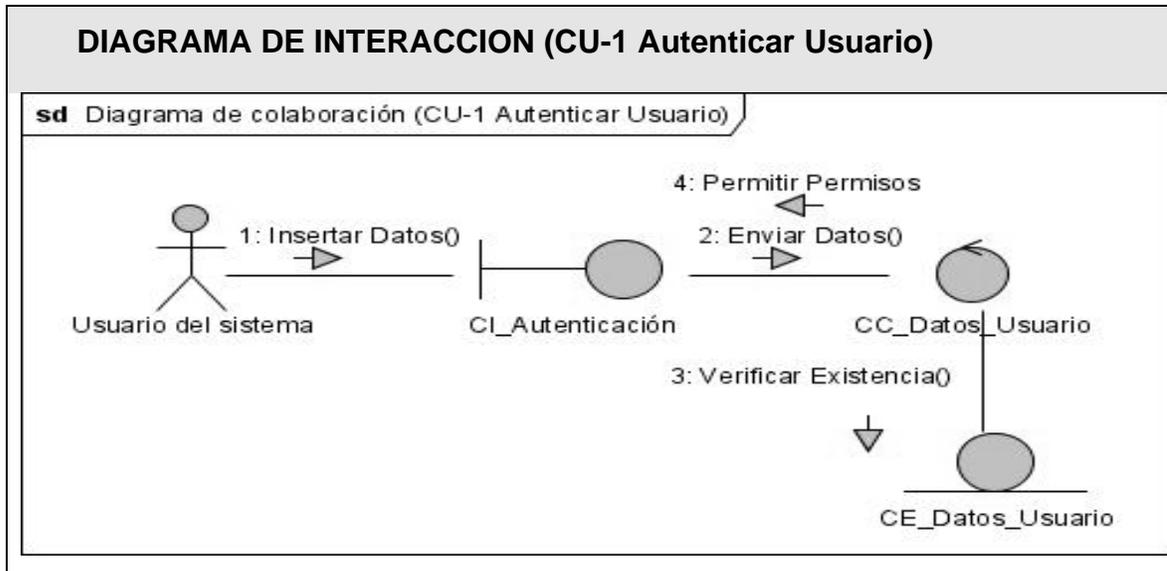
El modelo de análisis es una aproximación al modelo del diseño, es el resultado de la actividad de analizar los casos de uso. A continuación se refleja el modelo de análisis correspondiente a los casos de usos descritos.

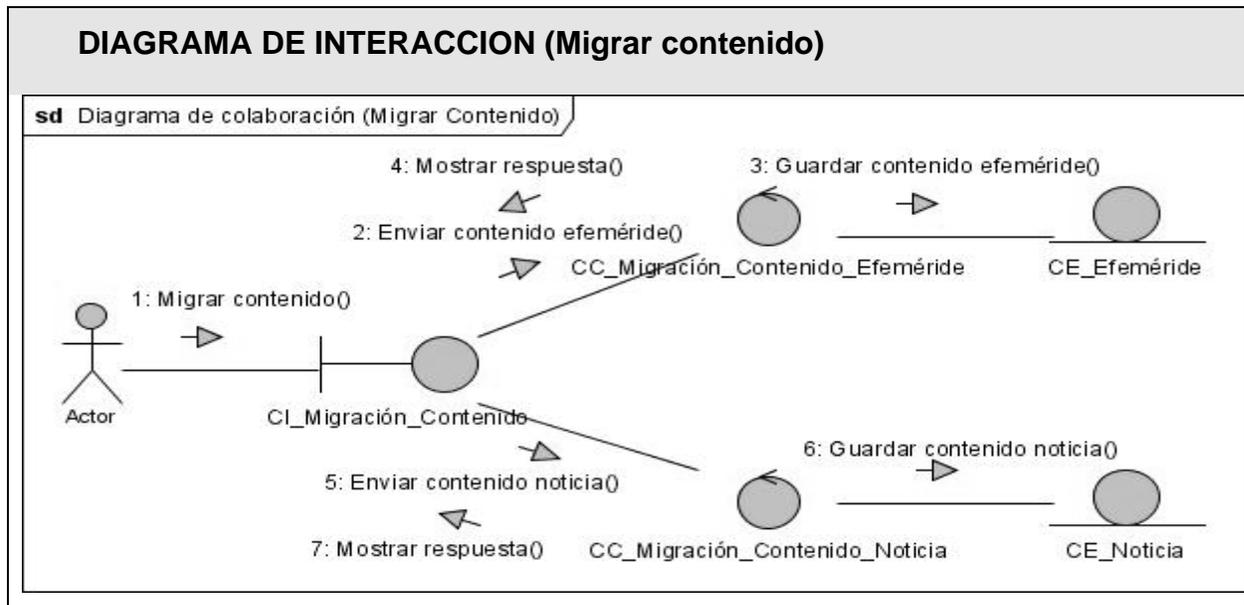
3.2.1 Diagrama de clases de análisis

DIAGRAMA DE CLASES DE ANÁLISIS



3.2.2 Diagrama de Interacción





3.3 Diseño

El diseño es un refinamiento del análisis, que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

El diagrama de clases para las Aplicaciones Web, difiere un poco del resto de las aplicaciones que se construyen, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente; estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así el producto final.

Para la comprensión del diagrama de clases de diseño Web, de la aplicación en conjunto con las clases del CMS Drupal, se explica brevemente el funcionamiento de Drupal como plataforma de publicación.

Provee un tipo de contenido genérico llamado “node”, y permite la creación, edición y publicación de este tipo de contenido. Los documentos creados se depositan en una base de datos central, donde también se

guardan el resto de datos de la Web, cómo son los datos relativos a los documentos (autor, fecha de publicación, título, etc.), datos y preferencias de los usuarios, la estructura de la Web, etc.

Contiene un potente sistema de seguridad, basado en la creación de usuarios y roles, en dependencia de los permisos que se necesiten para administrar el sitio y acceder a los módulos.

Para que se comprenda el diagrama de clases de diseño, se decidió describir por los módulos de drupal.

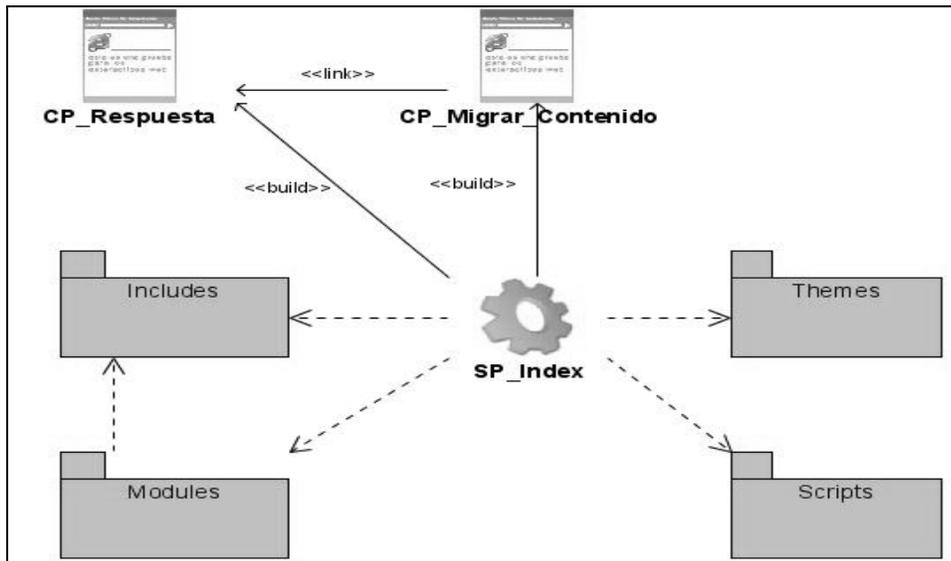
- Themes: Incluye las distintas plantillas, cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.
- Includes: Contiene un conjunto de ficheros indispensables para el funcionamiento.
- Modules: Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo solo se tiene que copiar en dicha carpeta.
- Scripts: Contiene un conjunto de ficheros indispensables para el funcionamiento.

Las páginas que genera Drupal puede tener o no formularios, en dependencia de lo que se desee. Es importante señalar que este CMS contiene una única pagina servidora.

A continuación se refleja el diagrama de clases del diseño Web de la aplicación.

3.3.1 Diagrama de clases de diseño Web.

DIAGRAMA DE CLASES DEL DISEÑO WEB



3.3.3 Diseño de la Base de Datos

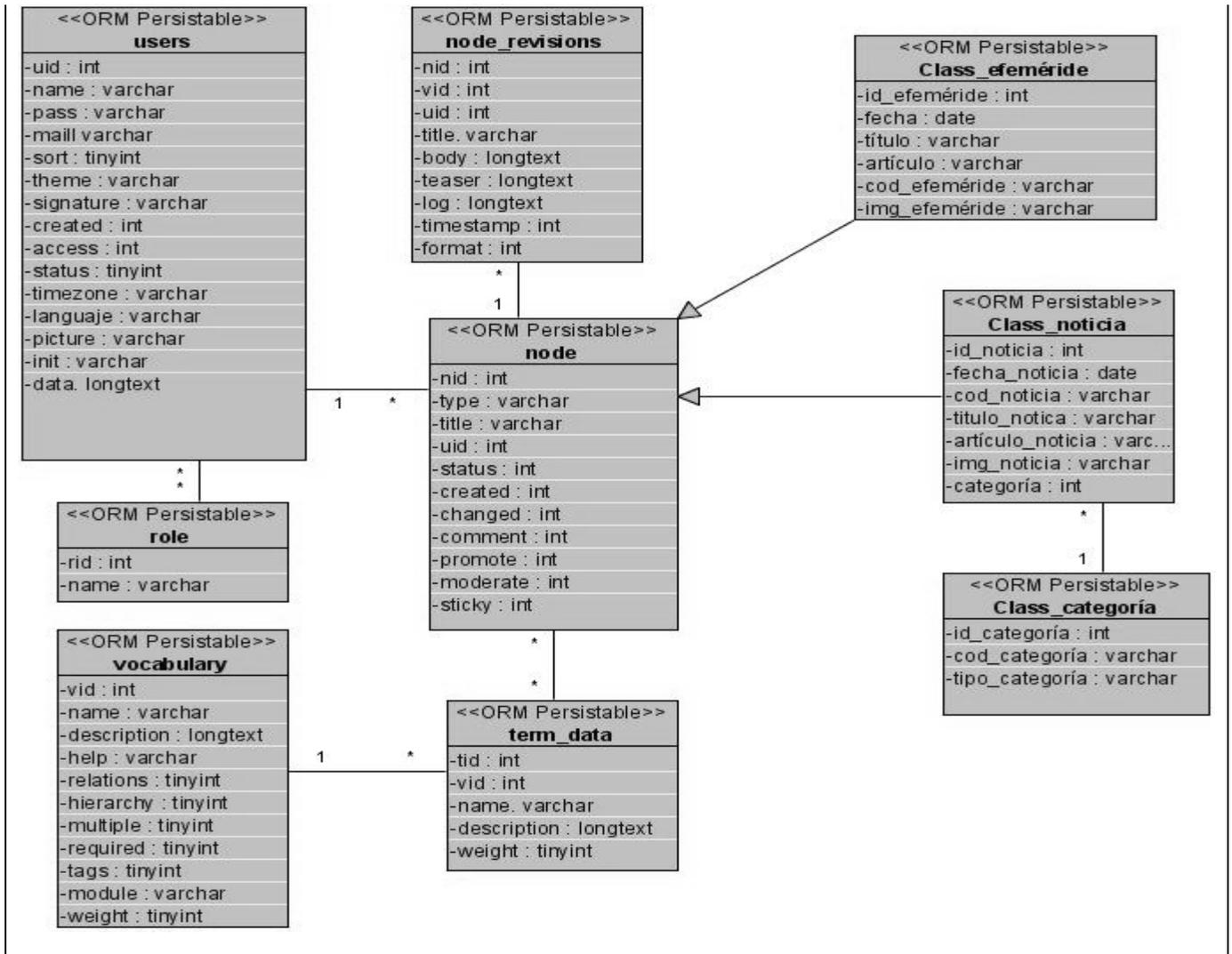
Para diseñar la base de datos del sistema, se utiliza el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior. Algunas de las clases representaban los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son lo que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará la modelación de la lógica de negocio de la aplicación.

El modelo de datos de la aplicación en cuestión tiene un nivel de complejidad muy bajo, debido a que la misma esta desarrollada en el Sistema de Gestión de Contenidos Drupal, la mayoría de las entidades son manejadas por el CMS por lo que no se contemplaran en el modelo.

Para lograr diseñar la Base de Datos se parte del diagrama de clases persistentes, que es un subconjunto del diagrama de clases del sistema.

3.3.3.1 Diagrama de clases persistentes

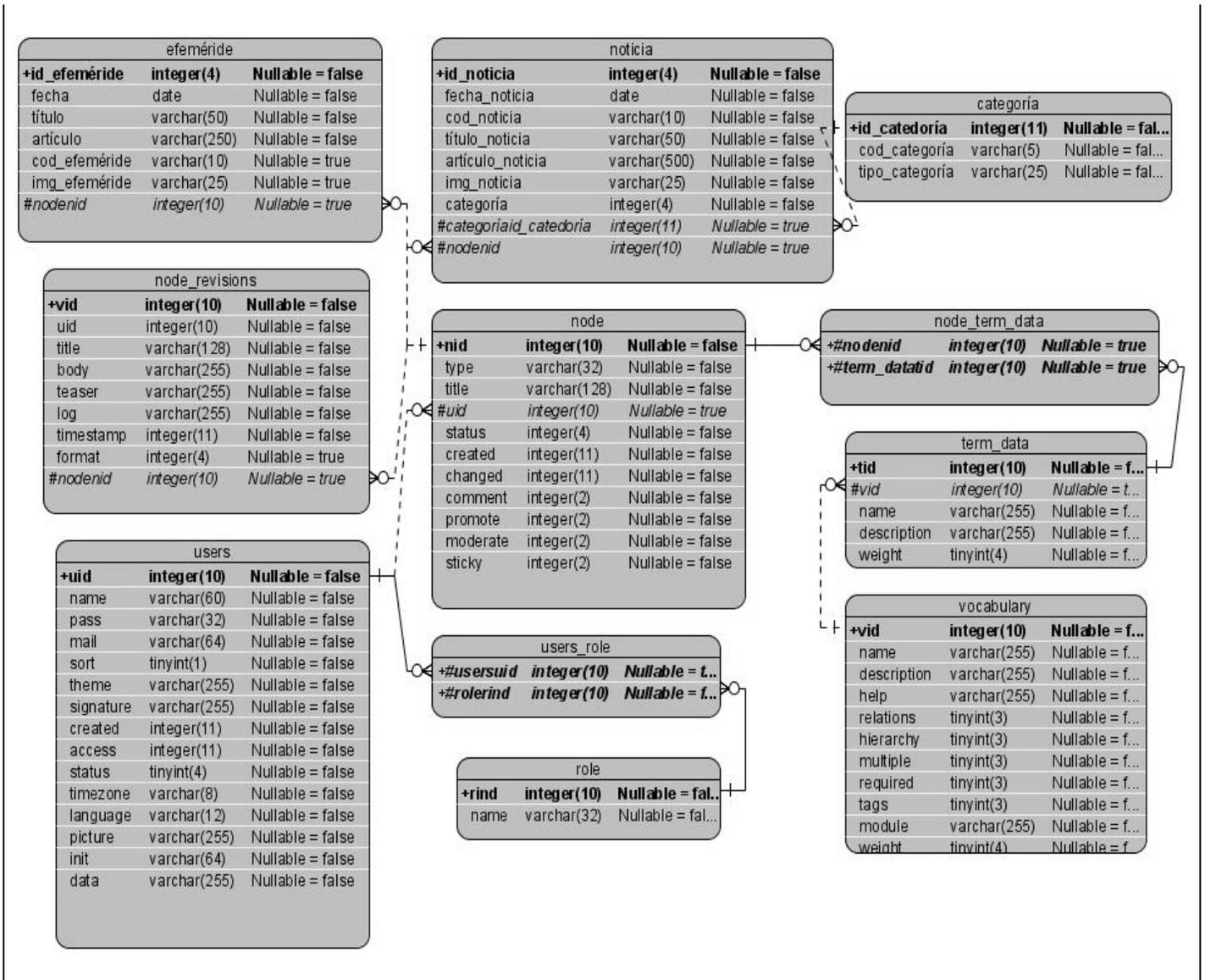
DIAGRAMA DE CLASES PERSISTENTES



3.3.3.2 Modelo de datos

A partir del diagrama de clases persistentes obtuvimos el modelo de datos que se muestra a continuación.

MODELO DE DATOS



3.3.3.3 Descripción de las tablas.

A continuación se describen cada una de de las tablas de la base de datos.

Nombre: node		
Descripción: Es una generalización de las tablas: Class_noticia, Class_efeméride.		
Atributo	Tipo	Descripción
nid	int	Identificador del node.
type	varchar	El tipo de contenido.
title	varchar	Título del contenido.
uid	int	Identificador de la tabla users.
status	int	Indica el estado de publicación de un contenido.
created	int	Fecha en la que se creó el contenido.
changed	int	Fecha en la que se modificó el contenido.
comment	int	Si permite o no el envío de comentarios referente a un contenido
promote	int	Indica el grado de promoción de un contenido, en dependencia del valor que tenga el contenido,

		aparece o no en la página principal.
moderate	int	Indica si el contenido se encuentra en la cola de moderación.
sticky	int	Indica si el contenido tiene prioridad sobre otros contenidos.

Nombre: noticia		
Descripción: Esta tabla contiene todas las noticias que han sido migradas, es una especialización de la tabla node.		
Atributo	Tipo	Descripción
Id_noticia	Int	Identificador de la noticia Valor auto numérico
Fecha _ noticia	Date	Fecha de la noticia
Código _ noticia	Varchar	Con este código se identifica la imagen y el artículo que le corresponde a la noticia.
Título _ noticia	Varchar	Título de la noticia
Artículo_noticia	Varchar	Texto, artículo de la noticia
Imagen _ noticia	Varchar	Imagen de la noticia
Categoría	Int	Categoría a la que pertenece la noticia

Nombre: efemérides		
Descripción: Esta tabla contiene todas las efemérides que han sido migradas en, es una especialización de la tabla node.		
Atributo	Tipo	Descripción
Id_ efeméride	Int	Identificador de la efeméride Valor auto numérico
Fecha_ efeméride	Date	Fecha de la efeméride
Código_ efeméride	Varchar	Con este código se identifica la imagen y el artículo que le corresponde a la efeméride.
Título_ efeméride	Varchar	Título de la efeméride
Artículo_ efeméride	Varchar	Texto, artículo de la efeméride
Imagen_ efeméride	Varchar	Imagen de la efeméride. Valor que puede ser nulo.

Nombre: categoría		
Descripción: Esta tabla contiene las categorías de las noticias.		
Atributo	Tipo	Descripción
Id _ categoría	Int	Identificador de la categoría Valor auto numérico
Código_ categoría	Varchar	Con este código se identifica la categoría que le corresponde a la noticia.
Nombre_ categoría	Varchar	Tipo de categoría (universitaria, nacional, interesante, internacional, deportivas, culturales.)

Nombre: node_revisions		
Descripción: Esta tabla almacena los datos completos de los contenidos.		
Atributo	Tipo	Descripción
nid	int	Identificador de la tabla

		node.
<u>vid</u>	int	Identificador de la tabla node_revisions.
title	varchar	Título del contenido.
body	longtext	Información que aparece en el cuerpo de las páginas.
teaser	longtext	Resumen del contenido.
log	longtext	Registro de la acciones que se realizan en el contenido.
timestamp	int	Fecha/hora de creación
format	int	Formato del contenido, puede ser en HTML o PHP.

Nombre: term_node		
Descripción: Esta tabla surge a partir de la relación de mucho a mucho de la tabla node y term_data.		
Atributo	Tipo	Descripción
<u>nid</u>	int	Identificador de la tabla node.
<u>tid</u>	int	Identificador de la tabla term_data.

Nombre: term_data		
Descripción: Nombre de las categorías.		
Atributo	Tipo	Descripción
<u>tid</u>	int	Identificador de la tabla term_data.
vid	int	Identificador de la tabla vocabulary.
name	varchar	Nombre de la categoría.
description	longtext	Descripción de la categoría.
weight	tinyint	Peso de la categoría

Nombre: vocabulary		
Descripción: Esta tabla contiene un listado de los términos de los vocabularios.		
Atributo	Tipo	Descripción
<u>vid</u>	int	Identificador de la tabla vocabulary.
name	varchar	Nombre del vocabulario.

description	longtext	Descripción del vocabulario.
help	varchar	Instrucciones para presentar al usuario cuando vaya a elegir un término.
relations	tinyint	Permite términos relacionados en este vocabulario.
hierarchy	tinyint	Permite una jerarquía de árboles entre los términos de un vocabulario.
multiple	tinyint	Permite a los nodos tener más de un término del vocabulario (siempre que estén activadas).
required	tinyint	Requerido o no.
tags	tinyint	Etiquetas.
module	varchar	Módulo de una categoría.
weight	tinyint	Peso del vocabulario.

Nombre: users

Descripción: Esta tabla contiene información de los usuarios.		
Atributo	Tipo	Descripción
<u>uid</u>	int	Identificador de la tabla users.
name	varchar	Nombre de usuario.
pass	varchar	Contraseña de usuario.
mail	varchar	Correo electrónico del usuario.
sort	tinyint	Rol del usuario
theme	varchar	Tema especificado para el usuario
signature	varchar	Firma del usuario.
created	int	Fecha en que se creó el usuario.
access	int	Fecha de ultimo acceso
status	tinyint	Estado del usuario (activo o bloqueado).
timezone	varchar	Zona horaria.
language	varchar	El idioma de la interfaz del sitio que se le presenta al usuario.

picture	varchar	Avatar del usuario
init	varchar	Correo electrónico del usuario.
data	longtext	Información sobre la cuenta(registro)

Nombre: users_roles		
Descripción: Esta tabla surge a partir de la relación de mucho a mucho de la tabla users y role.		
Atributo	Tipo	Descripción
<u>uid</u>	int	Identificador de la tabla users.
<u>rid</u>	int	Identificador de la tabla role.

Nombre: role		
Descripción: Esta tabla guarda los roles definidos en el sistema.		
Atributo	Tipo	Descripción
<u>rid</u>	int	Identificador de la tabla role.
name	varchar	Rol del sistema.

3.4 Tratamiento de errores

El tratamiento de errores se realiza con el sistema de captura de errores de Drupal, una vez que ocurre una excepción el cliente es redireccionado a una página de error.

Cada formulario se encarga de la validación de sus datos para evitar errores de concepto. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

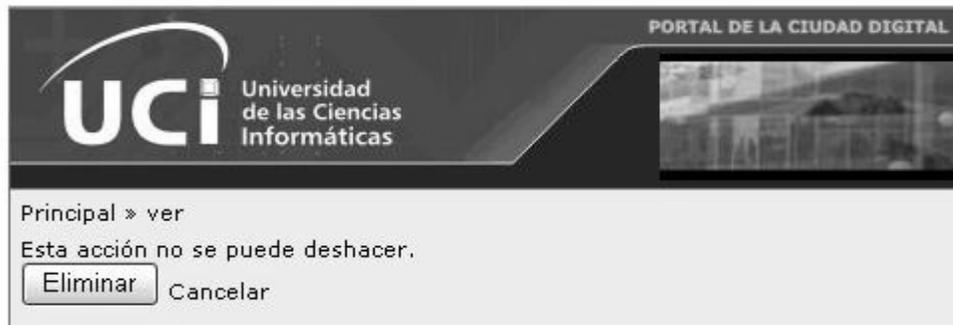


Fig. 1. Mensaje de confirmación

Presencia de errores en forma de mensaje de texto en la misma página donde se ejecuta la acción, de forma que el usuario pueda corregir el error fácilmente.



Fig. 2. Mensaje de Error en la misma página

3.5 Conclusiones

En este capítulo se ha llevado a cabo los diagramas de interacción, la descripción de las clases y demás elementos necesarios para la implementación. Se definieron, cuáles son las clases que serán persistentes, luego, a partir de esto, se construyó el modelo de datos. Se expusieron las pautas seguidas para el diseño de la interfaz, y se explicó cómo se llevó a cabo el tratamiento de errores.

4

CAPITULO

Implementación y Prueba.

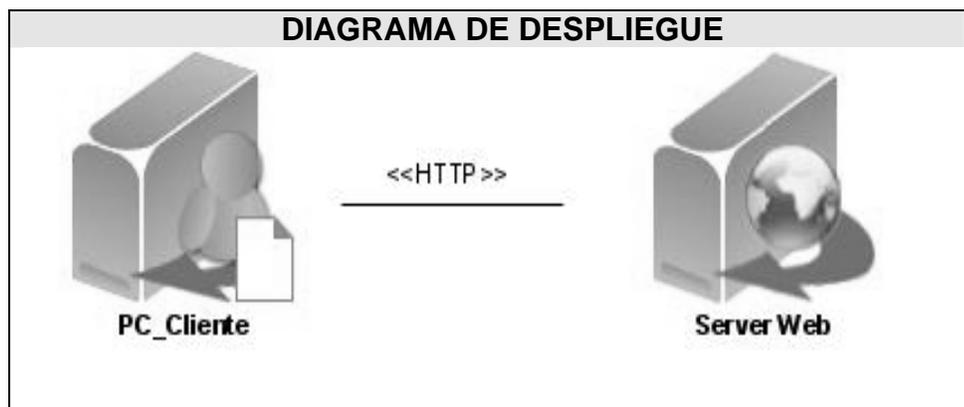
4.1 Introducción.

En el presente capítulo se desarrollan los flujos de trabajo de implementación y prueba. Se muestra la situación física de los componentes lógicos desarrollados a través del modelo de despliegue. Mediante el modelo de componentes se presenta la visa estática del sistema, además se muestra el modelo de prueba, en el cual se describen los casos de prueba de integración por cada caso de uso.

4.2 Diagrama de despliegue.

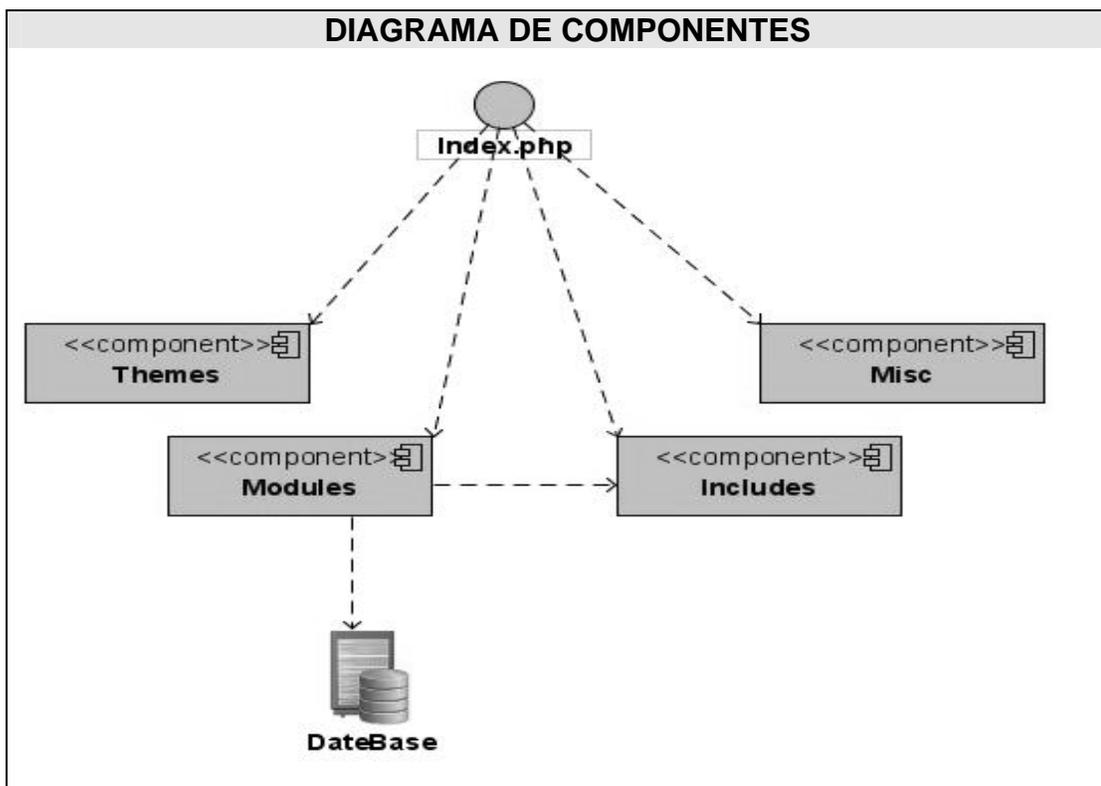
Un diagrama de despliegue muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. En este caso la aplicación se encuentra hospedada en un servidor Web. En la misma tenemos un sistema de gestión de base de datos (MySQL) que se encuentra dentro de este servidor.

Se emplea como protocolo de comunicación entre el Cliente y el Servidor Web el HTTP debido a que está basado en el modelo cliente-servidor.



4.3 Diagrama de componentes.

Durante la fase de implementación, entre los artefactos que se construyen tenemos los modelos de implementación, estos describen los componentes y la organización de acuerdo a los nodos; así como las dependencias de compilación entre ellos.



A continuación una tabla que describe cada uno de los componentes representado en el diagrama anterior:

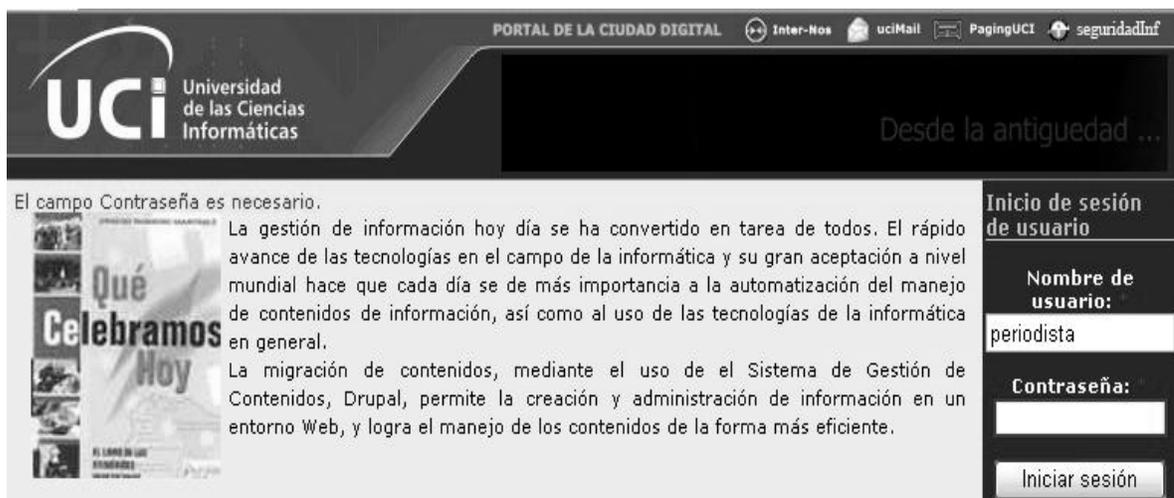
Componente	Propósito
Index.php	Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.
Themes	Este componente incluye las plantillas que se pueden utilizar, nos permite separar el contenido de la presentación.
Modules	Este componente contiene todos los módulos, que permiten las distintas funcionalidades del CMS.
Includes	Este componte contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal.
Misc	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js, etc).
DataBase	Representa la base de datos.

4.4 Modelo de prueba

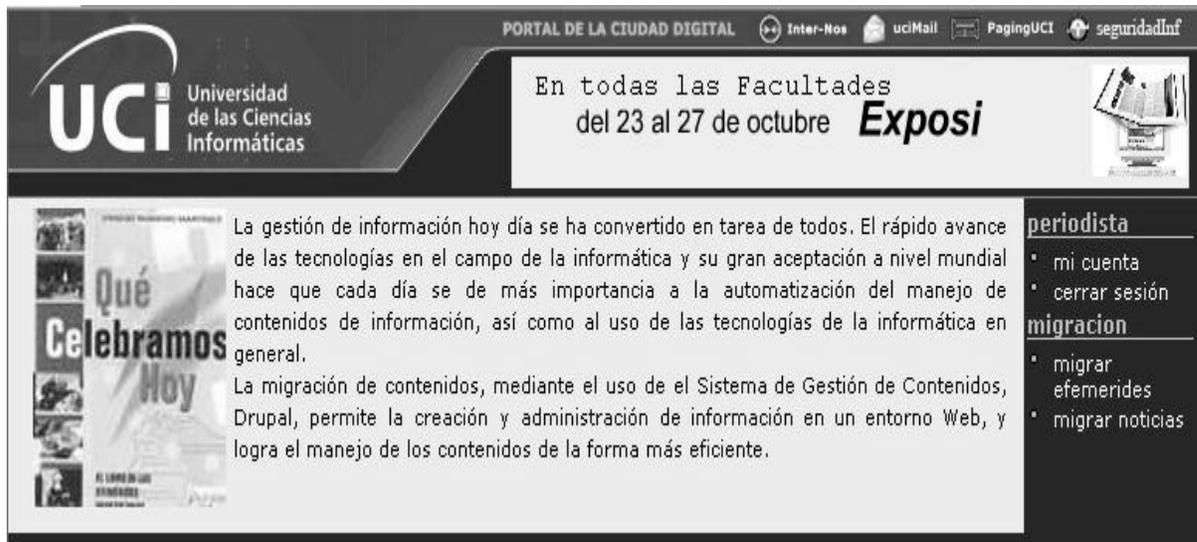
Nombre del caso de uso: Autenticar usuario

Entrada	Resultados	Condiciones
El usuario introduce un usuario y/o contraseña incorrecta.	El sistema emite un mensaje indicando que no se reconoce el nombre de usuario o contraseña.	
El usuario no registrado trata de iniciar sesión pero deja algún campo del formulario vacío.	El sistema emite un mensaje indicando que los campos dejados vacíos son requeridos.	
El usuario se autentica correctamente.	El sistema autentica al usuario registrado y determina en que nivel de acceso se encuentra.	

El sistema emite un mensaje indicando que los campos dejados vacíos son requeridos.



El sistema autentica al usuario no registrado y determina en que nivel de acceso se encuentra.



Nombre del caso de uso: Administrar Sistema

Entrada	Resultados	Condiciones
El administrador del sistema cuando va a modificar o crear una cuenta de un usuario deja campos requeridos en blanco.	El sistema devuelve el siguiente mensaje: “El campo contraseña, o usuario, o dirección de e_mail, es necesario”.	

El campo Contraseña es necesario.

Nombre de usuario: *

 Su nombre completo y el nombre de usuario que prefiera. Sólo se admiten letras, números y espacios.

Dirección de correo-e: *

 A esta dirección de correo-e se enviarán la contraseña y las instrucciones. Por tanto, asegúrese de que es correcta.

Contraseña: *

 Proporcione una contraseña para la cuenta nueva.

Notificar el usuario de la cuenta nueva

Nombre del caso de uso: Migrar contenido de efemérides

Entrada	Resultados	Condiciones
El usuario hace clic en el vínculo “migrar efemérides”.	El sistema emite un mensaje indicando la cantidad de contenidos migrados.	



Nombre del caso de uso: Migrar contenido de noticias

Entrada	Resultados	Condiciones
El usuario hace clic en el vínculo “migrar noticias”.	El sistema emite un mensaje indicando la cantidad de contenidos migrados.	



4.5 Conclusiones

En este capítulo se mostraron varias vistas para llevar a cabo el proceso de implementación del sistema. Igualmente se identificaron otras funcionalidades que se pueden tener en cuenta para futuras versiones del sistema. Se especificaron los protocolos de comunicación, mediante el diagrama de despliegue. Se representaron las dependencias entre los componentes software a través del diagrama de componentes y por último se realizó la descripción de los casos de prueba de integración de cada uno de los casos de uso del sistema.

Conclusiones

La gestión de información hoy día se ha convertido en tarea de todos. El rápido avance de las tecnologías en el campo de la informática y su gran aceptación a nivel mundial hace que cada día se de más importancia a la automatización del manejo de contenidos de información, así como al uso de las tecnologías de la informática en general.

Con este trabajo se propone una solución al problema del manejo de los contenidos de la intranet de la Universidad de las Ciencias Informáticas. Se hacen comparaciones de las tecnologías existentes actualmente. Se presenta una aplicación capaz de migrar los contenidos de la intranet actual para la intranet temporal mediante el CMS Drupal.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones UML para la modelación de todas las fases del proyecto.

El sistema resultante está equipado de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación.

Por tanto, se concluye que los objetivos propuestos para el presente proyecto, han sido cumplidos satisfactoriamente. Se dan una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

Recomendaciones

De forma general los objetivos trazados en la apertura de este trabajo han sido alcanzados, pero, a lo largo del proceso de desarrollo, ha quedado claro que la propuesta es sólo la primera etapa de un proyecto que puede ser mucho más ambicioso. Por tanto hacemos las siguientes recomendaciones:

- Perfeccionar el mecanismo de migración de forma más flexible para que la aplicación tenga un manejo más fácil.
- Realizar un módulo en Drupal con el objetivo de poder ubicarlo donde se desee realizar la migración.
- Extender el sistema de manera que pueda ser utilizado no sólo en la Universidad, sino en cualquier empresa que requiera de migración de contenidos hacia una base de datos.
- Extender el desarrollo de esta aplicación, adicionándole nuevas funcionalidades, ajustándolo más a las demandas de la intranet de La Universidad de las Ciencias Informáticas, haciéndola más útil y provechosa.

Bibliografía

- ANÓNIMO. Bases de Datos Lógicas, 2000. [Disponible en: <http://www.fismat.umich.mx/~elizalde/tesis/node6.html>]
- ---. The comparison of SQL Server 2000 with MySQL v4.1 2005. [Disponible en: <http://www.databasejournal.com/features/mssql/article.php/3087841>]
- ---. Funciones de los sistemas de gestión de bases de datos, 2003. [Disponible en: <http://www3.uji.es/~mmarques/f47/apun/node39.html>]
- ---. Historia de Drupal 2001. [Disponible en: http://www.jsnat.com/historia_de_drupal]
- ---. Intranet, 2006. [Disponible en: <http://www.monografias.com/trabajos12/intrants/intrants.shtml#VISION>]
- ---. Intranet, 2006. [Disponible en: <http://winred.com/EP/articulos/intranet/0020060100100011.html>]
- ---. Introducción a la documática, 2000. [Disponible en: <http://www.tramullas.com/documatica/2-4.html>]
- EDUARDO. ¿Qué es MySQL?, 2003. [Disponible en: <http://www.mysql-hispano.org/page.php?id=2>]
- HENST. ¿Qué es el PHP? , 2003. [Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro/>]
- QUESADA, Greenstone y Scout 2006. [Disponible en: <http://www.congreso-info.cu/UserFiles/File/Info/Info2006/Ponencias/116.pdf>]
- PRESSMAN., R. S. Tomado de la biblioteca UCI. Ingeniería del software. Un enfoque práctico <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
- SCHMULLER, J. Tomado de la biblioteca UCI. Aprendiendo UML en 24 horas. <http://bibliodoc.uci.cu/pdf/reg00004.pdf>, 2000.

Referencias Bibliográficas

- ANÓNIMO. *Base de Datos.*, 2001. [Disponible en: http://html.rincondelvago.com/base-de-datos_3.html]
- ---. *CMS de código abierto para la Web*, 2004a. [Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>].
- ---. *Concepto y características de las bd documentales*, 2004b. [Disponible en: <http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema9/tema9.html>]
- ---. *Glosario*, 2005a. [Disponible en: <http://www.angelfire.com/hiphop/gregory.16/glosario.htm>]
- ---. *Los sistemas de base datos y los SGBD*, 2006a. [Disponible en: <http://tramullas.com/documatica/2-1.html>]
- ---. *Qué es el Perl*, 2006b. [Disponible en: http://www.htmlpoint.com/perl/perl_02.htm]
- ---. *Qué es Javascript*, 2006c. [Disponible en: <http://www.desarrolloweb.com/articulos/25.php>]
- ---. *Qué es Python*, 2006d. [Disponible en: <http://www.desarrolloweb.com/articulos/1325.php>]
- ---. *¿Qué es un CMS?*, 2005b. [Disponible en: <http://cms-hispano.org/index.php?s=content&p=cms>]
- BARCIA, D. *¿Qué es CSS?*, 2003. [Disponible en: <http://www.maestrosdelweb.com/editorial/introcscs/>]
- GARCÍA, I. *Base de datos Modelo en red general*, 2001. [Disponible en: http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T0001_Igarcia.pdf]
- JOHN WORSLEY, J. D. *PostgreSQL Práctico*, 2001. [Disponible en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>]
- MARQUÉS, M. *Tema 2. Bases de datos orientadas a objetos*, 2002. [Disponible en: <http://www3.uji.es/~mmarques/e16/teoria/cap2.pdf>]
- MENDOZA, M. *Metodologías de desarrollo de software*, 2004. [Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html]
- RUIZ, D. F. *Bases de Datos*, 2001. [Disponible en: http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T0001_MAMoraga.pdf]
- SEGAN. *¿Qué es Drupal?*, 2006. [Disponible en: <http://drupaleros.com.ar/node/28>]

Glosario de términos

GPL: (General Public License) - Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

SQL: (Structured Query Language), lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

Código Abierto: es el término con el que se conoce al software distribuido y desarrollado libremente.

CU: Caso de Uso. Se emplea en el Capítulo 2 en las descripciones de los Casos de Uso.

CMS: Content Management System (Sistema de Gestión de Contenidos), son sistemas usados para la construcción de aplicaciones que gestionan contenido.

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

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

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

Zend: Compañía líder de infraestructuras para Web; está reconocida internacionalmente como la autoridad actualmente en PHP. Sus fundadores son los diseñadores del PHP v.4 en adelante, actualmente es una compañía líder dentro de la comunidad Open Source.

Open source - Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como *free software*.

WYSIWYG: What You See Is What You Get (en español: “lo que ves es lo que obtienes”). Para describir un editor de texto que permite escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.