UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA"
FACULTAD DE INGENIERÍA INDUSTRIAL
CENTRO DE ESTUDIOS DE INGENIERÍA Y SISTEMAS
INGENIERÍA INFORMÁTICA

MODELAMIENTO DE UN SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA

AUTOR

Yohan Guevara Arias.

TUTOR

Ing. Sergio Enrique Tamayo Bermúdez

Ciudad de la Habana.
Junio 2005

AGRADECIMIENTOS.

A *mamá* y a *papá* por todo lo que han significado para mí. Donde estén gracias.

A *mami* por ser todo. Porque has soñado con esto toda tu vida. Y por mucho más.

A Dios, solo Él sabe porque se lo agradezco.

A Migue que siempre has sido el tipo.

A mi hija que es lo mejor que me ha pasado en la vida.

A mi hermano por estar siempre orgulloso de mí a pesar de los traspiés que he tenido.

A Liana por ser ella, y confiar ciegamente en lo que soy capaz de hacer.

A mi tío lindo por todos esos consejos, cariño y enseñanzas que me has regalado a lo largo de mi vida.

A tía Miriam por ser mi otra madre.

A mi tía Ida que ha sido un ejemplo de intelectualidad.

A mi tía Mary que siempre supiste que lo iba a lograr.

A mi tía Anolan que ya no me tiene que apretar más los cachetes.

A Tito que se merece más que este humilde agradecimiento.

A Silvia por ser una de las personas más buenas que he conocido.

A Rafael mi suegro que a parte de todo es mi amigo.

A mis primas Yamirka, Acra, Migda, Miry, Mabel, Marle, Mirelbis, Daylin, Yumilexys, Blanquita, Daimara.

A mis primos Alberto, Julio, Marinin.

A mi puro Abel.

A mis hermanas Marisol y Nela, a Alexys, Julio y Lachy que han sido como mis hermanos, a mi sobrinito Manuel.

A Edith, a Rafa, a Ole, a Yoney, al Gordo, a Eduard, a Sergio , Osbi, Eric, Yoba, a Ramonsito, a Loannys, a los mellos, a Yadira, a Yusma, a Maikel, a Maritsa, Mariluz, Magdalena, Yadira (la de magda), a Masita, Octavio, Onelio, a mi Prof. Virginia, a Otero, Liecel, Adrian, Andy, Roiky, Osmel, Allan, Lion, Albert, Niuldis, a BroodWarCuba, a Chuchi, Vera, Obdel, Dayner, Yoba, Raymill, Dundee, Yoennis, Karoline, Catherine, Yorgelis, al Chupe, a Noa, al ninja, al motro, a los que he mencionado y a los que no, **muchas gracias a todos**.

RESUMEN.

Debido a la creciente necesidad de favorecer la gestión de la información que se genera de los servicios que prestan intranets como la de La Universidad de las Ciencias Informáticas (UCI) se plantea la tarea de diseñar el modelo de una aplicación capaz de mostrar los beneficios acordes con la problemática antes expuesta. Actualmente toda la gestión de la información se hace a través de sistemas separados que si bien brindan un cúmulo grande de información, se hace difícil y compleja la actualización y mantenimiento de los mismos, presentando la carencia de funcionalidades como la capacidad de obtener datos específicos y puntuales de la información en determinados momentos.

A través de un estudio de la situación actual, con una debida búsqueda bibliográfica y tomando en consideración las nuevas tendencias de las Tecnologías de la Informática y las Comunicaciones se hace un análisis que da pie a los pasos concretos del modelado de un sistema usando una metodología de impacto actual. Todo el proceso de investigación se apoya de un estudio de factibilidad. A pesar de que el presente está sustentado en técnicas de código abierto o más conocido como Open Source, de acuerdo a los lineamientos trazados por la dirección del estado cubano y de la Facultad 10 de la UCI.

ÍNDICE

INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (**UCI**) al igual que otros centros de altos estudios del país tiene como principal objetivo formar profesionales cada vez más integrales tanto intelectual como profesionalmente.

La **UCI** como parte de los nuevos programas de la Revolución y supervisada directamente por la máxima dirección del país ha elaborado un exquisito programa de estudio y tecnología para lograr una formación eficiente de profesionales, utilizando lo mas avanzado en Tecnología de la Información y las Comunicaciones.

Actualmente la (**UCI**) brinda disímiles servicios en su Intranet. p.e. (Correo electrónico, Servicio de noticias, Multimedia, entre otros) todo esto para facilitar el trabajo de alumnos y profesores. Se trabaja en la realización de sistemas para la comercialización con otros países como Venezuela y se desarrollan en conjunto con empresas nacionales Sistemas para la Salud, la Educación etc.

Realizar un Sistema Web puede ser un trabajo complicado y muy laborioso actualmente las herramientas que se usan (en la **UCI** como en otros centros de desarrollo del país) son básicamente editores que generalmente están enfocados más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de Sistema de Gestión de Contenidos (***Content Management System o CMS***). Se trata de herramientas que permiten crear y mantener un sistema Web con facilidad, encargándose de los trabajos mas tediosos que hasta ahora ocupaban el tiempo de los administradores de las Webs.

Este trabajo surge como necesidad de dar solución a las situaciones antes expuestas; por lo que el **problema** a solucionar en él, consiste en: ¿Cómo facilitar la creación y mantenimiento de los servicios y sistemas Web que se brindan y desarrollan en la Universidad de las Ciencias Informáticas?

El desarrollo de una Intranet esta basado en las prestaciones de servicios de uso común para una comunidad de usuarios. Por las facilidades que brindan las Web su uso se ha multiplicado aceleradamente en los últimos años. Para la realización de un servicio o sistema Web necesitamos de tiempo suficiente para diseñar, programar etc., lo que puede traer consigo que los servicios caduquen o pierdan fiabilidad en la comunidad de usuarios.

El uso de un **CMS** es de gran utilidad para gestionar entornos Web, pero se podría pensar que no es necesario para una Web relativamente pequeña o cuando no se necesitan tantas funcionalidades. Eso solo podría suceder en un sistema estático en el que no se prevé crecimiento futuro ni muchas actualizaciones, lo que en realidad nunca sucede.

Con el presente trabajo se propone realizar una herramienta que permita una mejor y más rápida creación y actualización de servicios como los de la Intranet de la **UCI** extensible a otras entidades que requieran su uso.

Por tanto el **objeto de estudio** de este trabajo es la creación de una herramienta que facilite la administración de los servicios de la intranet de La Universidad de las Ciencias Informáticas.

De ello se deriva que el **campo de acción** que abarca este trabajo es la automatización de la creación de portales y servicios webs en la **UCI**.

Como **hipótesis** se parte de la idea de que esta herramienta va a ser esencialmente para la creación de portales, basados en un gestor de Bases de Datos rápido como MySql Server, y un intérprete rápido y potente como PHP. Así será posible resolver todos los problemas de servicios en las intranet.

El **objetivo general** de este trabajo será: desarrollar una propuesta de una herramienta Web que permita la creación y actualización de los servicios webs que

se desarrollan en la **UCI** y de los propios mediante avanzadas técnicas de gestión y creación de contenidos.

Objetivos específicos.

Realizar un estudio del arte sobre los **CMS** existentes para valorar sus ventajas y desventajas obteniendo un diseño adaptado a los requisitos del centro.

Diseñar e implementar una aplicación Web (**CMS**) que permita administrar todos los servicios de la intranet y además sirva como plataforma para los proyectos que se comercialicen.

Diseñar e implementar un **CMS** sencillo y potente.

Para dar cumplimiento a los objetivos anteriormente planteados se definen las siguientes tareas:

Estudio y descripción de los CMS más usados actualmente.

Obtener requisitos del Sistema.

Selección de la metodología de Análisis y Diseño de sistemas informáticos, que facilite la creación y garantice la calidad del sistema.

Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollara la aplicación. Fundamentando su elección.

Diseño de una Base de Datos que soporte la mayoría de las funcionalidades del Sistema.

La propuesta de crear un CMS en la UCI no tiene precedentes ya que la gestión de contenidos para la Web en esta se hace a través de CMS ya existentes o de webs dinámicas.

Capítulo 1 Fundamentación del tema.

Explica brevemente todo sobre los CMS, ventajas, desventajas con respecto a los métodos y herramientas utilizados hoy en día en la UCI y cuales fueron las principales causas que generaron la necesidad del cambio; y como conclusión, se obtienen los objetivos generales y específicos a cumplir.

Capítulo 2 Tendencias y tecnologías actuales a considerar.

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 3 Descripción de la solución propuesta.

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 4 Construcción de la solución propuesta.

Enfoca la construcción de la solución mediante diagramas de clases, de datos, y se plantean los principios para el diseño y la implementación. Aquí se construyen las funcionalidades que se definieron en el capítulo anterior.

Capítulo 5 Estudio de factibilidad.

Es un estudio de factibilidad sobre el sistema, obteniendo los beneficios tangibles e intangibles y analizando los costos del desarrollo de esta propuesta.

Capítulo 1. FUNDAMENTACIÓN DEL TEMA.

INTRODUCCIÓN.

En el presente capítulo se da una visión general de los aspectos relacionados con los Sistemas de Administración de Contenidos (**CMS**). Las características de cada herramienta de desarrollo así como los principales conceptos asociados al dominio del problema y que son necesarios para entender el negocio y la propuesta de solución. [18]

1.1. Content Management System (CMS).

El **CMS** es "un proceso que permite a la gente crear y actualizar más fácilmente, especialmente en sus sitios Web"

- ReUse – permite reutilizar el mismo contenido en diversos lugares del página Web, con diferente presentación.
- Publicación – Las páginas pueden ser creadas dentro del **CMS**, pero no serán visibles en el sitio Web hasta que no hayan sido "publicadas" por un usuario autorizado.

Hoy en día se conoce como **CMS** a las aplicaciones que en la industria de las publicaciones on-line permiten la generación de los sitios Web dinámicos. El objetivo que estos programas persiguen y cumplen con enorme eficacia es la creación y gestión de información on-line, estando esta información compuesta por textos (artículos, informes,...), imágenes, gráficos, videos, sonido, etc. Como se ha comentado el objetivo de los **CMS** es doble, por una parte la generación de la información y por otra su administración y difusión.

El objetivo del **CMS** es permitir a un usuario no entrenado en informática introduzca el contenido en el **CMS** a través de una interfaz sencilla, y permitir que otro usuario (tampoco entrenado) pueda ocuparse exclusivamente de gestionar la

presentación y la estructura de los documentos entre sí a través de otra interfaz distinta. Ver Fig. [22]

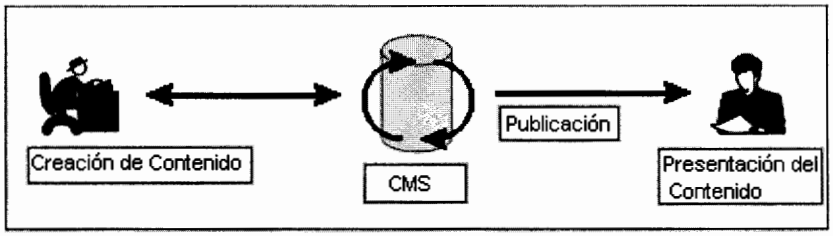


Fig: Requerimientos de un CMS.

1.1.1. Concepto de CMS.

Un sistema de administración de contenidos es un sistema informático (hardware y software) responsable de la colección, gestión y publicación de bloques (chunks) de información, llamados componentes de contenido. [14]

1.2. Historia de los CMS.

A principio de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a medida.

Ya en el año 1994 illustra Information Technology utilizaba una base de datos de objetos como repositorio de los contenidos de una Web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no cuajó entre el publico y la parte de la empresa enfocada a la Web fue comprada por AOL, mientras que Informix adquirió la parte de bases de datos.

RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos en el año 1994. No fue hasta finales del año siguiente que presentaron su **CMS** basado en una base de datos.

Entre los **CMS** de código abierto uno de los primeros fue **Typo3**, que empezó su desarrollo en 1997, palabras dichas por su autor, *Kasper Skarhoj*, "antes de que el término gestión de contenidos fuera conocido sobradamente".

PHPNuke, la herramienta que popularizo el uso de estos sistemas para las comunidades de usuarios en Internet, se comenzó a desarrollar en el año 2000. La primera versión supuso tres semanas de trabajo al creador, rescribiendo el código de otra herramienta, Thatware. Ver **Anexo1** [22]

1.3. Presente y futuro de los CMS.

En la actualidad, aparte de la ampliación de las funcionalidades de los **CMS**, uno de los campos más interesantes es la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al de sistema y aportan calidad y estabilidad.

Algunos de estos estándares son los **CSS,** que permite la creación de hojas de estilo, **XML,** un lenguaje de marcas que permite estructurar un documento, **XHTML,** que es un subconjunto del anterior orientado a la presentación de documentos vía Web, **WAI**, asegura la accesibilidad del sistema y **RSS**, para sindicar contenidos de tipo noticia.

También las aplicaciones que rodean los **CMS** acostumbran a ser estándar, como los servidores Web, ***Apache*** y ***IIS***; los lenguajes **PHP**, **Perl**, **Python**; y las bases de datos **MySQL** y **PostgreSQL**. La disponibilidad para los principales sistemas operativos de estas aplicaciones y módulos, permite que los **CMS** puedan funcionar en diversas plataformas sin muchas modificaciones.[22]

Robertson (2003a) apunta que.

- Los **CMS** se convertirán en un artículo de consumo, cuando los productos se hayan establecido y más soluciones salgan a la luz.
- Muchos proyectos fracasarán por no ajustarse a los estándares y no entender conceptos como, usabilidad, arquitectura de la información, gestión del conocimiento y contenido.
- El campo de los gestores de contenido madurará hasta conseguir un alto grado de consistencia y profesionalismo.
- Se adoptarán estándares en el almacenaje, estructuración y gestión del contenido.
- Se producirá una fusión entre gestión de contenidos, gestión de documentos y gestión de registros.

También se puede añadir la incorporación de sistemas de e-learning y gestión del conocimiento y en los entornos de intranet corporativa, la posibilidad de acceder a otras fuentes de datos como por ejemplo sistemas de soporte de decisiones (***DSS***). El campo de los **CMS** de código abierto tendría que seguir un desarrollo similar.[25]

1.4. Creación de contenido.

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 él usuario ve el resultado final mientras escribe, 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.

Hay otras herramientas como la edición de los documentos en ***XML***, utilización de aplicaciones ofimáticas con las que se integra el **CMS**, importación de documentos existentes y editores que permiten añadir marcas, habitualmente ***HTML***, para indicar el formato y estructura de un documento.

Un **CMS** puede incorporar una o varias de estas herramientas, pero siempre tendría que proporcionar un editor ***WYSIWYG*** por su facilidad de uso.

Para la creación del sitio propiamente dicho, los **CMS** aportan herramientas para definir la estructura, el formato de páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir las funciones previstas originalmente. [22]

1.4.1. Editores WYSIWYG.

En un editor ***HTML*** se pueden colocar imágenes, definir estilos, utilizar negritas o cursivas, etc. sin preocuparse de las etiquetas correspondientes a cada estilo o elemento. Es el editor el que se encarga de usar estas etiquetas y las utiliza convenientemente, solamente se tiene que escribir o diseñar las páginas como si se estuviese utilizando Word.

Este tipo de editores ***HTML*** se denominan habitualmente ***WYSIWYG*** (*What You See Is What You Get*) porque cuando trabajas con ellos lo que ves que estás creando con el editor es lo que obtienes luego cuando grabas la página. [29]

1.4.2. Gestión de contenido.

1.4.2.1. ¿Que es la gestión de contenidos?

Es un complejo proceso de colección, gestión y publicación de contenido informativo en entorno digital.

- Colección: creación o adquisición de información.
- Gestión: creación, gestión y mantenimiento de repositorios.
- Publicación: extracción de componentes del repositorio para la creación de publicaciones, incluye información, funcionalidad, etc. [8]

1.4.2.2. Requerimientos previos.

- Definición de políticas y objetivos de información.
- Auditoria de la información.

- Adopción de estándares.
- Definición de productos de información digital.
- Selección de herramientas informáticas. [8]

1.4.2.3. Sistemas para gestión de contenidos.

- Variedad de productos y de aplicaciones, de precios y de prestaciones.
- Muchos tienen su origen y gran parte de sus prestaciones, en sistemas de gestión documental.
- La principal división se establece entre nominales y dinámicos (database-driven). [8]

1.4.2.3.1. Sistemas nominales.

- Sistemas básicos de edición y gestión de páginas estáticas.
- Proveen *templates*, estado básico del fichero, control de enlaces y sistemas de actualización.
- Útiles para Webs simples a pequeña escala.
- Orientados a la página como unidad de trabajo. [8]

1.4.2.3.2. Sistemas dinámicos.

- Producción de páginas dinámicas, bajo demanda del usuario.
- No todos los Web database-driven son sistemas de gestión de contenidos.
- Integran plantillas (templates) con diferentes datos de diferentes tipos.
- Ofrecen la metáfora de publicación digital interactiva. [8]

1.4.2.4. Subsistemas e interacciones

- Sistema de gestión: puede servir como parte del sistema de colección.
- Sistema de colección: puede servir como parte del sistema de publicación.

- Sistema de publicación: puede servir como parte del sistema de colección. [8]

1.4.2.4.1. Subsistema de colección, 1.

Autoría: proceso de creación de contenidos.

- Entorno de creación.
- Orientación a audiencia.
- Ayuda a aplicación de estándares.
- Ofrecer plantillas (templates).
- Flujo de trabajo, estado y control de versiones.

Adquisición: integración de contenidos externos.

- Sindicación.
- Otros ficheros fuente.

[8]

1.4.2.4.2. Subsistema de colección, 2.

Conversión: convertir información en otros formatos a los estándares del CMS.

- Eliminación de información innecesaria.
- Separación de formato y estructura.
- Crear mapas de estructura.

Servicios de colección: funciones de apoyo a la creación de colección, a la integración de información en el repositorio.

[8]

1.4.2.4.3. Subsistema de colección, 3.

Agregación: proceso de integrar fuentes de información diversas en una estructura específica a través de:

- ***Procesamiento editorial:*** reglas de corrección, de comunicación, de consistencia.

- ***Procesamiento de segmentación:*** división del contenido en parte adecuadas (***shunks***), poner contenido en componentes.
- ***Procesamiento metatorial:*** aplicación consistente y completa de metadatos.

[14]

1.4.2.4.4. Subsistema de gestión.

Repositorio: conjunto de ficheros, bases de datos, etc., que almacenan el contenido y los datos asociados al mismo (control y configuración).

Flujo de trabajo: coordinación, agenda, puntos de control de los procesos, relación con los otros subsistemas.

Administración: definición de parámetros y estructura del **CMS**, relación con los otros subsistemas.

[15]

1.4.2.4.5. Subsistema de publicación, 1.

Encargado de tomar componentes del repositorio y crear automáticamente publicaciones. [14]

1. ***Templates***: ficheros que guían la creación de una publicación con contenido del repositorio, suelen incluir:
    - Elementos estáticos.
    - Llamadas a servicios de publicación.
    - Llamadas a servicios externos al **CMS**.

1.4.2.4.6. Subsistema de publicación, 2.

Servicios de publicación: lógica de aplicación y servicios que ofrece el CMS para ayudar a la creación de publicaciones. [15]

- Carga y ejecuta templates.
- Servicios específicos de publicación.
- Llamadas a servicios externos.

Otras publicaciones: impreso, digital, sindicación.

1.4.2.4.7. Asegurar la calidad del contenido.

Aunque es difícil de cuantificar, el valor de tener la información actualizada en cualquier Sitio Web es indiscutible. Hay muchas características de los sistemas de administración de contenidos que aseguran que solo la información de alta calidad es presentada e intercambiada en línea. Para asegurar que el contenido actual esta disponible los sistemas de administración de contenido notifican a la persona indicada (Administrador del Sistema) cuando el contenido perecedero debe ser cambiado, actualizado o removido.

Si la apariencia consistente de un Sitio Web es crítica, un sistema de administración de contenido ayuda a asegurar dicha consistencia y la coherencia de los mensajes publicados a través de formas o plantillas de contenido. El manejo de versiones y flujos de trabajo en un sistema de administración de contenido también ayuda a reducir el monto de información desactualizada. [15]

1.4.3. Publicación.

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para todo el sitio Web o para la sección concreta donde esta situada, de forma que el resultado final es un sitio Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio Web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas. [22]

.

1.4.4. Presentación.

Un **CMS** puede gestionar automáticamente la accesibilidad de un sitio Web, con soporte de normas internacionales de accesibilidad como ***WAI***, y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (*Windows, Linux, Mac, Palm,* etc.) y su capacidad de internacionalización le permite adaptarse al idioma, sistema de medidas y cultura del visitante.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual del sitio, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otro sitio Web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado. [32]

1.5. Clasificación de los CMS.

De acuerdo a estas funcionalidades los **CMS** pueden clasificarse en Portales, Blogs (Página Web que sirve como sistema de noticias, etc.), Soporte Técnico o ayuda, *e-commerce, Groupware, Forums, e-Learning, Wiki.* De todas formas, dentro de la clasificación anterior, caben muchos tipos diferentes de **CMS** con mayor o menor popularidad. Para dar solución a la problemática descrita es necesario crear un **CMS** de tipo Portal pero que presente funcionalidades capaces de interactuar con los usuarios. [29]

1.6. CMS propietarios y de código abierto.

Se puede hacer una primera división de los **CMS** según el tipo de licencia escogida. Por una parte están los **CMS** comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso. Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del 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.

Algunas empresas también dan acceso al código, pero solo con la adquisición de una licencia especial o después de su desaparición. Generalmente las modificaciones sólo pueden hacerlas los mismos desarrolladores, y siempre según sus prioridades.

Los **CMS** de código abierto son mucho más flexibles en este sentido, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los **CMS** de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales.

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría: su precio. Habitualmente todo el software de código abierto es de acceso libre, es decir, sin ningún costo de licencias. Solo en casos aislados se hacen distinciones entre empresas y entidades sin ánimo de lucro o particulares. En comparación, los productos comerciales pueden llegar a ten un precio que solo una gran empresa puede asumir.

En cuanto al soporte, los **CMS** comerciales acostumbran a dar soporte técnico, con un precio elevado en muchos casos, mientras que los de código abiertos se basan más en las comunidades de usuarios que comparten información y soluciones de problemas. Las formas de soporte se pueden mezclar, y así encontramos **CMS** de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuario. En el caso comercial también sucede, pero el precio de las licencias hace que el gran público escoja otras opciones y por lo tanto las comunidades de soporte son más pequeñas.

Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos o mal redactada. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

Hoy en día existen **CMS** de calidad tanto comerciales como de código abierto. Una característica de algunos de los **CMS** de código abierto como de los **CMS** propietarios es que se encuentran poco elaborados *(aunque en plena evolución)*. En definitiva un **CMS** de código abierto es mucho más económico que su homólogo propietario, con la ventaja de disponer de todo el código fuente y de una extensa comunidad de usuarios.

Por todos estos motivos, y como apuesta por la filosofía del software libre, en este trabajo solo veremos algunos ejemplos de **CMS** de código abierto. Ver **Anexo2**. [30]

1.7. Necesidad de un CMS.

En los epígrafes anteriores se han presentado bastantes motivos para ver la utilidad de un sistema que gestione un entorno Web. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de entorno Web

Estos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un **CMS**: [16]

1.7.1. Inclusión de nuevas funcionalidades en el Sitio Web:

Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un **CMS** eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en el sitio. El sistema puede crecer y adaptarse a las necesidades futuras. [13]

1.7.2. Mantenimiento de gran cantidad de páginas.

En un sitio Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos. [13]

1.7.3. Reutilización de objetos o componentes.

Un **CMS** permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado. [32]

1.7.4. Páginas interactivas.

Las páginas estáticas llegan al usuario exactamente como están en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para

conseguir esta interacción, los **CMS** conectan con una base de datos que hace de repositorio central de todos los datos del sitio Web. [25]

1.7.5. Cambios del aspecto del Sitio Web.

Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede dar como resultado la rescisión de muchas páginas para su adaptación. Los **CMS** facilitan los cambios con la utilización, por ejemplo, del estándar ***CSS*** (Cascading Style Sheets u hojas de estilo en cascada) con lo que se consigue la independencia entre presentación y contenido. [24]

1.7.6. Consistencia del Sito Web.

La consistencia de un sitio Web no quiere decir que todas las páginas sean iguales, sino que existe un orden (visual) en vez de un caos. Un usuario nota enseguida cuándo una página no es igual que las del resto del mismo Sitio Web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el Sitio no lo han diseñado profesionales. Los **CMS** pueden aplicar un mismo estilo en todas las páginas con el mencionado ***CSS,*** y aplicar una misma estructura mediante patrones de páginas. [24]

1.7.7. Control de acceso.

Controlar el acceso a un sitio Web no consiste simplemente al permitir la entrada al Sitio, sino que comporta gestionar los diferentes permisos a cada área del Sitio aplicados a grupos o individuos (Usuarios). [24]

1.7.8. Criterios de selección.

Antes de empezar el proceso de selección de un **CMS** concreto, hay que tener claros los objetivos de la Web, teniendo en cuenta a los usuarios finales, y estableciendo una serie de requerimientos que tendría que poder satisfacer el **CMS**.

La siguiente lista esta basada en las funcionalidades principales de los **CMS** expuestas anteriormente, las indicaciones de *Robertson, J. (2003ª)* y una recopilación de los requerimientos básicos de un sitio Web.

- **Código abierto:** Por los motivos mencionados anteriormente, el **CMS** tendría que ser de código abierto (o libre).
- **Arquitectura técnica:** Tiene que se fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos contenido, presentación y estructura que permita la modificación de uno de ellos sin afectar a los otros. Es recomendable, pues, que se utilicen hojas de estilo (***CSS***) y patrones de páginas.
- **Grado de desarrollo:** Madurez de la aplicación y disponibilidad de los módulos que le añaden funcionalidades.
- **Soporte:** La herramienta tiene que tener soporte tanto por parte de los creadores como por otros desarrolladores. De esta manera se puede asegurar de que en el futuro habrá mejoras de la herramienta y que se podrá encontrar solución a los posibles problemas.
- **Posición en el mercado y opiniones:** Una herramienta poco conocida puede ser muy buena, pero hay que asegurar de que tiene cierto futuro. También son importantes las opiniones de los usuarios y de los expertos.
- **Usabilidad:** La herramienta tiene que ser fácil de utilizar y aprender. Los usuarios siempre serán técnicos, por lo tanto hace falta asegurar que podrán utilizar la herramienta sin muchos esfuerzos y sacarle el máximo rendimiento.

- **Accesibilidad:** para asegurar la accesibilidad de un Sitio Web, el **CMS** tendría que cumplir un estándar de accesibilidad. El más extendido es ***WAI*** (*Web Accessibility Initiative*) del ***World Wide Web Consortium***.
- **Velocidad de descarga:** Teniendo en cuenta que todos los usuarios no disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente o dar la opción. [24]

1.8. Objeto de estudio.

La Universidad de las Ciencias Informáticas es una Institución Universitaria comprometida con su Patria, que se propone, mediante la formación integral y continua de profesionales, la actividad científico-técnica y la extensión universitaria; contribuir de forma significativa al desarrollo sostenible de la sociedad cubana; mantener un liderazgo nacional en el campo de la tecnología, y con soluciones creativas y autóctonas, ser competitivos internacionalmente, para lo cual hace suyas las aspiraciones más legítimas de los trabajadores y estudiantes.

Con este fin, se provee al estudiante a lo largo de sus años de estudio de una gran cantidad de Información; y a la vez, como resultado de procesos investigativos y docentes se crean nuevos recursos. Además, gracias al desarrollo de las Tecnologías de la Información y las Telecomunicaciones, hoy contamos con los medios necesarios para poner a disposición de la comunidad universitaria todo tipo de recursos, tanto físicos como digitales y en los más disímiles formatos.

1.8.1. Descripción del proceso actual.

Actualmente en La Universidad de las Ciencias Informáticas se prestan diversos servicios Webs en su Intranet al mismo tiempo que se trabaja en la realización de proyectos de carácter comercial con países como Venezuela entre otros. Con el propósito de facilitar el trabajo de estudiantes y profesores además de insertarse en el mercado del Software Libre en la región.

Existen grupos de desarrollo integrado por estudiantes y profesores que realizan sistemas Web ya sea para agregar un nuevo servicio a la Intranet de la universidad o par algún proyecto con el objetivo de ser comercializado etc. Estos grupos de trabajo deben dedicar tiempo tanto a la implementación como al diseño del Sistema. Para el diseño de los patrones se utilizan un grupo de diseñadores formado principalmente por estudiantes de las distintas facultades del centro, aunque no suficientes aún. Se trabaja en el aumento de estos para la culminación más rápida y eficiente de los sistemas en construcción.

1.8.2. Situación Problémica.

La creación de patrones de diseño unido a la creación de contenidos para un sistema Web son los principales factores para que el tiempo de creación sea relativamente grande. Los métodos de desarrollo utilizados hasta el momento son eficientes pero su uso conlleva a dedicar demasiado tiempo en la elaboración de patrones de diseño y contenido.

Otro problema, es el hecho de que el mantenimiento a los sistemas que están publicados en la intranet es muy lento y esto trae consigo que la información no este lo suficientemente actualizada. Por lo que el usuario que solo tiene el deseo de informarse no lo puede hacer por la falta de actualidad en los artículos publicados.

1.9. CMS existentes vinculados al campo de acción.

En los proyectos de Software Libre que se realizan en la **UCI** se ha popularizado el uso del **XOOPS**. Existen otros **CMS** como **MAMBO, TYPO3, PLONE** entre otros muchos todos bajo los términos de la (***GPL***). [1]

1.9.1. ¿Que es XOOPS?

XOOPS es un sistema de administración de contenido (**CMS**) poderoso, flexible y fácil de usar, que está basado en el lenguaje script **PHP.** Provisto de **MySql,** permite

manejar sitios Web dinámicos, construir comunidades en línea, gestionar usuarios, modificar a su libre albedrío la maqueta del sitio y alimentar contenido a través de una interfaz sencilla.

XOOPS significa e**X**tensible **O**bject **O**riented **P**ortal **S**ystem, es decir, Sistema extensible de portales orientado a objetos. Esta poderosa herramienta se ofrece bajo los términos de *Licencia Pública General* (***GPL***), lo cual significa que se puede usar y modificar gratuitamente. [1]

1.9.2. MAMBO Open Source.

Sencillísimo de usar y a la vez de nivel profesional, este sistema de administración de Contenidos en línea ***WYSIWYG*** para el contenido, noticias, banners publicitarios, mailing a usuarios, administrador de enlaces, estadísticas, archivo del contenido antiguo(no se borran las noticias antiguas, sino que se archivan, con posibilidad de recuperación), contenido contextual según fechas, idiomas, módulos y componentes. [12] [11]

1.9.3. Drupal.

Contiene ayuda en línea tanto para usuarios como para administradores, código abierto, personalización de las páginas de acuerdo al usuario, excelente autenticación de usuarios que puede integrar con un servidor **LDAP**, permisos basados en roles y diseño basado en plantillas. Independencia de la base de datos, implementada y mantenida para **MySql** y **PostgreSQL**, multiplataforma pues funciona con **Apache** o *Microsoft* **IIS** como servidor Web y en sistemas como *Linux, FreeBSD, Solaris, Windows y Mac OS X.* [17]

1.10. Propuesta de Solución.

Después de realizar un análisis sobre algunos de los tipos de **CMS**, y determinar claramente cual es la situación actual sobre el objeto de estudio que tiene este

trabajo, se concluye que se hace necesario diseñar e implementar un sistema que permita gestionar el contenido de los sistemas Webs que existen en la **UCI**.

Por la complejidad que requiere la realización de un Sistema de Administración esto llevara tiempo y por el momento se realizara el diseño para un **CMS** sencillo y funcional, al cual con el tiempo se le incorporan módulos como e-learning entre otras funcionalidades de gran importancia para el desarrollo de una aplicación de gran escalabilidad.

1.11. Fundamentación de los objetivos que se proponen en el trabajo.

De acuerdo con las nuevas tecnologías en el campo de la gestión y administración de contenido y para darle respuesta a la situación problémica planteada, se propone para este trabajo un conjunto de objetivos para cumplimentar la propuesta de solución planteada en la sección anterior.

1.11.1. Objetivos generales.

Realizar un estudio sobre los **CMS** existentes en la actualidad y proponer una Aplicación Web que permita la Gestión y la Administración de Contenidos.

1.12. Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio a través de los conceptos y definiciones planteadas. Se determinaron las condiciones específicas que rodean al problema y sobre esta base se definieron los objetivos generales y específicos para el trabajo. Aunque en esta etapa solo se habla de ideas, es correcto que estén bien fundamentadas, porque constituyen la base para el posterior desarrollo del trabajo.

Capítulo 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES UTILIZADAS.

INTRODUCCIÓN.

En el presente capítulo, se hace un análisis de las tecnologías que consideramos adecuadas para realizar el sistema propuesto.

Como elemento indispensable se analiza la posible metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Este capítulo persigue el propósito de decidir que tecnología utilizar en el desarrollo de este trabajo, después de hacer un riguroso estudio del arte de las más usadas en el ámbito internacional.

Así mismo se abordaran temas relacionados con la Gestión de Contenidos, el uso de las tecnologías Web, Lenguajes de Programación, Sistemas de Gestión de Bases de Datos (SGBD), etc.

2.1. ¿Qué es INTERNET?

En 1957 ocurre un suceso que influiría posteriormente en el desarrollo de las redes de computadoras. El departamento de Defensa de los EE.UU. crea la Agencia de Proyectos de Investigaciones Avanzadas, ARPA (Advanced Research Projects Agency), como respuesta al lanzamiento por la URSS del primer satélite artificial de la tierra. Esta agencia se encargaría de desarrollar proyectos técnicos e investigativos aplicables en el campo militar.

Es en este contexto que ARPA decide realizar investigaciones con el objetivo de interconectar computadoras utilizando como medio de transmisión la red telefónica existente en el país y experimentar si computadoras ubicadas en lugares geográficos distantes podían comunicarse empleando una nueva tecnología que había surgido recientemente en el viejo continente con el nombre de Conmutación de Paquetes.

Es conveniente recalcar que la meta de ARPA no era la creación de la red internacional de computadoras existente hoy día, sino como dijimos antes, la

conformación y desarrollo de una red de computadoras que pudiera sobrevivir a un ataque nuclear soviético. Estados Unidos estaba envuelto en la escalada militar y su principal preocupación era poseer un sistema de comunicaciones que sobreviviera a dicho ataque. La solución era crear una red de computadoras sin centro, en la que los mensajes no tuvieran una única ruta para transitar de un punto a otro del territorio norteamericano.

Para ello en 1966 un grupo de investigadores de ARPA inició el primer esfuerzo serio por "enseñarle" a las computadoras a "hablar" unas con otras. Como resultado ese esfuerzo aparece en diciembre de 1969 la red bautizada con el nombre de ARPANet, germen de la actual INTERNET.

La red ARPANET conectaba las Universidades de California en Santa Bárbara y en los Ángeles, la de Utah en Salt Lake City y la Universidad de Stanford. Esta tecnología posibilitaba que los científicos de esas Universidades compartieran la información y los recursos de la red independientemente de la distancia que los separara.

Coincidentemente, en el propio año 1969 Ken Thompson creaba la primera versión de UNIX, sistema operativo que luego representaría un papel principal en la comunicación entre computadoras.

Los científicos descubrieron enseguida que podían usar esa red para comunicarse. A la red primitiva se comenzaron a enlazar Universidades que empezaron a usarla entre otras cosas para transmitir conferencias en línea. Pero hasta esos momentos sólo podían interconectarse computadoras con similares Sistemas Operativos.

En 1972 ya había 37 sitios diferentes conectados a ARPANet. Estos sitios intercambiaban mensajes entre usuarios individuales, lo que se llamo correo electrónico (e-mail), tenían la posibilidad de controlar computadoras de manera

remota, proceso conocido como login remoto (login o telnet) y permitía el envío de largos ficheros de textos o de datos entre computadoras utilizando el FTP (File Transfer Protocol).

Durante 1973 ARPANet desborda las fronteras de los EE.UU. al establecer conexiones internacionales con la "University Collage of London" de Inglaterra y el "Royal Radar Establishment" de Noruega. En este propio año se presentan las ideas básicas de Internet en el Grupo de Trabajo de Interconexión de Redes (INWG, Internet Working Group).

Ya más avanzada la década del 70, DARPA (Nuevo nombre tomado por ARPA a partir de 1972) le encarga a la Universidad de Stanford la elaboración de protocolos que permitieran la transferencia de datos entre diferentes tipos de redes de computadoras. Es el momento en el que Bob Kahn, Vinton Cerf y un grupo de sus estudiantes desarrollan los protocolos TCP/IP que hicieron posible el surgimiento de la red universal de Computadoras que existe en la actualidad bajo el nombre de Internet.

En la década de 1980 esta red de redes conocida como la Internet fue creciendo y desarrollándose debido a que con el paso del tiempo cientos y miles de escuelas, Universidades, centros de investigación y agencias del gobierno fueron conectando sus computadoras a la red. Algunas de estas instituciones comenzaron a permitir acceso al público, es decir, que a partir de esos momentos toda persona con una computadora y un MODEM podía colocarse en la red.

Como es natural, en la actualidad la Red Internet esta soportada por una sofisticada y continuamente cambiante plataforma tecnológica, la cual permite a esta "Red de Redes" crecer exponencialmente.

2.1.1. ¿Cómo funciona Internet?

El uso de la tecnología de Conmutación de Paquetes propicia que todas las computadoras estén al mismo nivel en la red, lo que descentraliza su control, cumpliéndose así el objetivo de evitar la existencia de una computadora central.

Para que todas estas computadoras puedan comunicarse entre sí, debe existir un medio físico que las una (pares cobre, enlaces satelitales, infrarrojos, microondas, fibra óptica, etc.). Además deben "ponerse de acuerdo" en cuanto al lenguaje que utilizarán para comunicarse, lo que técnicamente se conoce como Protocolo de Comunicación.

Los protocolos de comunicación son conjuntos de reglas que rigen la forma de comunicación entre máquinas. Al igual que los humanos usamos ciertas reglas para comunicarnos, el mundo de los unos y los ceros de las computadoras también las tienen. Cada una de esas formas es llamada protocolo de comunicación. Un conjunto de esos protocolos está encapsulado en TCP/IP (Transmision Control Protocol/ Internet Protocol), los que fueron adaptados por decisión de DARPA en 1982 como el conjunto de protocolos estándar para todas las computadoras conectadas a ARPANet. Decisión gracias a la cual se convirtieron en el lenguaje de comunicación de Internet.

2.1.2. La información a través de Internet. El World Wide Web.

World Wide Web (WWW), o simplemente Web, es el universo de información accesible a través de Internet, una recopilación universal del conocimiento humano. Es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en Hipertexto, con plataforma de enlaces cruzados, que se ejecuta en Internet.

El componente más usado en Internet es definitivamente la Web. Su característica sobresaliente es el texto de remarcado, un método para referencias

cruzadas instantáneas. Usando la Web, se tiene acceso a millones de páginas de información en todo el mundo. La exploración se realiza por medio de un software especial denominado" Browser" o navegador. La apariencia de un Sitio Web puede variar ligeramente dependiendo del navegador que se utilice. Las versiones más recientes de estos navegadores incorporan funcionalidades adicionales par permitir la reproducción de animaciones, realidad virtual, sonido y video en formato digital. El protocolo que se utiliza para la comunicación en la Web es el HTTP (Hypertext Transfer Protocol) y el formato que se utiliza para la transferencia es el HTML (Hypertext Markup Language).

2.1.3. Aplicaciones Web. Aplicaciones Web vs. Sitios Web.

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En términos más simples, una Aplicación Web es un Sistema Web que permite a los usuarios ejecutar lógica de negocio a través de un navegador (Browser), o lo que es lo mismo, modificar el estado del negocio.

Las aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocios en el servidor, el sistema no puede ser considerado una aplicación Web, en ese caso se considera como un sitio Web. En esencia una aplicación Web usa un sitio Web como entrada (Front-end) a una aplicación típica.

La arquitectura de un Sitio Web es simple. Contiene como componentes principales: el Servidor Web, una Red y uno o más navegadores o clientes. El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se utiliza el protocolo HTTP.

El cliente o navegador (Browser) es el responsable de mostrar la información al usuario y de hacer validaciones sencillas en la entrada de datos para que la información sea mostrada al usuario.

2.1.4. Entornos distribuidos. Modelo Cliente Servidor.

La arquitectura Cliente-Servidor, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos se realiza se efectúe con la mayor eficiencia y permita simplificar las actualizaciones y mantenimiento del sistema. En una arquitectura monolítica no hay distribución, los tres niveles tienen lugar tienen lugar en el mismo equipo; en el modelo Cliente-Servidor, en cambio, el trabajo se reparte entre dos Ordenadores.

Se puede decir que todas las aplicaciones tienen la misma arquitectura básica y se pueden subdividir e tres partes:

1. Interfaz de usuario: La presentación al usuario, con las entradas de datos y las consultas.
2. Reglas de Negocio: El procesamiento de la Información.
3. Acceso a Datos: El control del almacén de Datos.

Ventajas del modo Cliente servidor:

El servidor no necesita una elevada capacidad de procesamiento, parte del proceso se reparte con los clientes.

Se reduce considerablemente el tráfico en la red. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

El sistema es fácilmente escalable.

2.2. Desarrollo basado en RUP bajo la herramienta Rational Rose.

El proceso unificado de Rational (**RUP**) es una metodología guiada por casos de uso, centrados en la arquitectura iterativa e incremental. Su desarrollo está basado en componentes. **RUP** contiene un Proceso Integrado y propone un modelo de referencia organizacional para la organización de personal. Utiliza ***UML*** como único lenguaje de modelado para el desarrollo de todos los modelos.

UML (Unified Modeling Language) se ha convertido en el estándar de factores para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Rational Rose es la herramienta **CASE** desarrollada por los creadores de ***UML*** (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

El navegador ***UML*** de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Rational Rose es una herramienta para "modelado visual", que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software.

Rational Rose permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (***RUP***), en concreto: [18]

Sus características principales son:

1. Guiado/Manejado por casos de uso.
2. Centrado en arquitectura.
3. Iterativo e Incremental.
4. Desarrollo basado en componentes.

5. Utilización de un único lenguaje de modelado (***UML***).
6. Proceso Integrado.

2.2.1. UML.

UML (Unified Modeling Languaje) o Lenguaje de Modelado Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos.

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

De forma general las principales características son:

- Lenguaje unificado para el modelado de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational. [18]. Ver **Anexo3**

2.2.2. Rational Rose.

Existen herramientas Case de trabajo visuales como el Analise, el Designe y el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

2.2.3. Características principales

Admite como notaciones: **UML, COM, OMT y Booch**

Realiza Chequeo semántico de los modelos

Ingeniería "de ida y vuelta": Rose permite generar código a partir de modelos y viceversa

Desarrollo multiusuario

Integración con modelado de datos

Generación de documentación

Tiene un lenguaje de script para poder ampliar su funcionalidad

Disponible en múltiples plataformas [18]

2.3. Características de la Plataforma.

El uso de los sistemas operativos en el mundo de la informática es algo vital desde su surgimiento. Los avances en las distintas ramas de la informática han dado la posibilidad de que estos vallan siendo cada vez más orientado a los usuarios normales. Compañías como Microsoft han desarrollado sistemas pero con características como que son sistemas propietarios de código cerrado. Desde hace

ya casi una década se viene manejando el termino Open Source, lo que en un principio fue Software Libre.

2.3.1. ¿Qué es Software Libre?

El software libre supone una forma de autogestión en un campo que tradicionalmente no ha formado parte de los movimientos antagonistas en general: la tecnología y el conocimiento. Así el software libre es parte de una revolución aún mayor, una revolución cognitiva donde se lucha donde se lucha contra el burocratismo tecnológico al que quieren empujarnos muchos gobiernos y entidades.

Se podría decir que el software libre es una forma de autogestión cognitiva, un cuestionamiento radical de la lógica polarizada que ha gobernado la tecnología los últimos 20 años. Muchos movimientos sociales no lo ven así. [2]

2.3.2. El Sistema Operativo Linux.

Linux es una versión de **UNIX** de libre distribución, inicialmente desarrollada por Linus Torvalds en la Universidad de Helsinki, en Finlandia con tal de hacer una mejor versión de Minix, sistema escrito por el profesor Andrew Tanenbaum. Fue desarrollado con la ayuda de muchos programadores y expertos de **UNIX** a lo largo y ancho del mundo, gracias a la presencia de Internet. Cualquier habitante del planeta puede acceder a Linux y desarrollar nuevos módulos o cambiarlo a su antojo.

Buena parte del software para Linux se desarrolla bajo las reglas del proyecto de GNU de la Free Software Foundation, Cambridge, Massachusets. Las comúnmente llamadas distribuciones están constituidas por el núcleo del sistema y un conjunto de aplicaciones que compilan diversas compañías. Las distribuciones más conocidas son Red Hat, Debian, Mandrake y Suse.

Linux puede ser portado a la mayoría de las arquitecturas de computadoras existentes tales como **Intel, 68k, PowerPC, Alpha, Sparc, SMP** entre otras. Estas características unidas a su robustez y al gran soporte existente en la red fueron los motivos que influyeron en la determinación de construir una herramienta de administración de Contenidos sobre esta plataforma. [3]

2.4. Servidor Web Apache.

Es el servidor Web más utilizado en el mundo. Su coste gratuito, gran fiabilidad y extensibilidad le convierte en una herramienta potente y muy fiable.

Dentro de sus fuertes se encuentran:

- Tiene interfaz con todos los sistemas de autenticación.
- Facilita la integración como "Plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Tiene integración en estándar del protocolo de seguridad SSL.
- Provee interfaz a todas las bases de datos.
- Posee Virtual Host.

Apache fue diseñado para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Está ligado a la plataforma Linux (aunque existen versiones para las plataformas más usadas) sobre la cual los individuos o instituciones pueden construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización. Apache por sus características es OpenSource. [4]

2.5. Arquitectura basada en componentes.

La arquitectura basada en componentes tiene como objetivo construir aplicaciones complejas mediante el ensamblado de módulos (**componentes**), que han sido previamente diseñados y que pueden ser reutilizados en múltiples

aplicaciones. Cada componente debe describir de forma completa las interfaces que ofrece, así como las interfaces que requiere para su operación. Debe operar correctamente con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz. Es actualmente una de las técnicas más prometedoras para incrementar la calidad del software, abreviar los tiempos de acceso al mercado y manejar adecuadamente el incremento continuo de su complejidad.

2.6. Lenguajes de programación para la Web.

Desde el surgimiento de Internet han existido disímiles Lenguajes y Herramientas para crear páginas Webs. Se nos hace necesario hacer un análisis de los más usados para determinar cual usar en la creación de esta herramienta (**CMS**). [4]

2.6.1. Hypertext Preprocessor (PHP).

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el **HTML**.

PHP, en el caso de estar montado sobre un servidor Linux o Unix, es más rápido que ***ASP***, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes **COM** que se realizan entre todas las tecnologías implicadas en una página ***ASP***.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como ***MSSQL Server***, ***MySQL***, ***Oracle***, ***Informix***, y ***ODBC***, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de

archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Dos de sus mayores ventajas, que le da la acreditación de multiplataforma, es que puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actualmente, es soportado por la mayoría de los servidores Web, incluyendo los de Microsoft, y además, soporta una gran cantidad de Sistemas de Gestión de Bases de Datos, entre ellas las más potentes.

Dentro de las principales características de este lenguaje tenemos. [4]

- Simplicidad. Su sintaxis está inspirada en C, ligeramente modificada para adaptarla al entorno en el que trabaja, de modo que si está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP.
- Si bien es cierto que hay ciertas características avanzadas que presentan las plataformas J2EE o .NET y que PHP no las tiene, no todas las aplicaciones Internet ameritan tal grado de complejidad. PHP fácilmente puede cubrir más del 75% de las necesidades del mercado.
- Es multiplataforma, es decir, puede ser utilizado en cualquiera de los principales sistemas operativos del mercado actual y es soportado por la mayoría de los servidores Web.
- Es software libre, lo que implica menos costos y servidores más baratos, por lo que podemos usarlo en proyectos comerciales si queremos, sin tener que pagar por su licencia. El tiempo, es uno de los costos más altos que hay que tener en cuenta antes de empezar un proyecto. Para empezar, el tiempo de aprendizaje de PHP es muy corto gracias a su simplicidad. Luego, el tiempo de desarrollo es también corto. Podríamos hacer más de un proyecto Web con PHP en el mismo tiempo que tomaría hacer un solo proyecto con Java o .NET. Otro aspecto que hay que tener en cuenta es el del hardware. Para desarrollar en PHP no se requiere tener grandes capacidades de hardware, como sí lo requieren los pesados IDEs para programar en Java o .Net. Luego, en el caso de los servidores, una aplicación en PHP no requiere tanta memoria de máquina como

podría requerir una aplicación en Java con sus servidores de aplicaciones que podrían requerir hasta varios procesadores y varios Gigas de memoria RAM.

2.6.2. JSP.

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es pues, una tecnología orientada a crear páginas Web con programación en Java.

Con ***JSP*** podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas ***JSP*** están compuestas de código ***HTML/XML*** mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las ***JSP*** podremos escribirlas con nuestro editor ***HTML/XML*** habitual.

2.6.3. ASP.

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ***ASP*** se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (de Microsoft).

La mayor desventaja que presenta este lenguaje es que solo se puede implementar en los Servidores Web de su desarrollador: Microsoft. Actualmente se ha presentado ya la segunda versión de ***ASP***: el ***ASP.NET***, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ***ASP.NET*** tiene algunas diferencias en cuanto a sintaxis con el **ASP**, de modo que se ha de tratar de distinta manera uno de otro. Para implementarlo es necesario montar en el Servidor la Plataforma ***.NET***.

2.6.4. Perl.

Es un lenguaje de programación muy utilizado para construir aplicaciones ***CGI*** para el Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contendido de los ficheros.

Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como ***JavaScript*** o ***ASP***. [5]

2.7. Sistemas Gestores de Bases de Datos (SGBD).

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

2.7.1. MsSql.

Microsoft SQL Server 2000 es uno de los mejores SGDB base de datos para Windows, es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

Desventajas

Esta diseñado solo para la plataforma Windows.

2.8. MySql.

Es un SGBD ***Open Source*** (Código abierto) diseñado para los sistemas Unix, aunque existen versiones para Windows. [6], [7]

Ventajas:

- Diseñado en vistas a la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (phpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos.
- Se comercializa bajo los términos de la GPL.
- Fue creado como SGBD para Portales.

Inconvenientes:

No soporta subconsultas.
No soporta transacciones.
No soporta triggers ni procedimientos en el servidor.
No soporta claves ajenas. Ignora la integridad referencial.
No soporta vistas.

2.9. POSTGRESQL.

PostgreSQL posee una amplia licencia BSD (esta licencia básicamente consiste en que el código puede ser redistribuido y modificado. La FSF lo considera, junto con la licencia GPL, software libre) es ampliamente utilizado. Posee una gran estabilidad

y confiabilidad legendaria nunca a presentado caídas en varios años de operación de alta actividad. Fue diseñado para ambientes de alto volumen intentando estar a la altura de Oracle, Sybase o Interbase. Escala muy bien al aumentar el número de CPU y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. Sin embargo consume muchos recursos y no escala bien en la plataforma Windows. [6]

Ventajas:

Soporta transacciones y desde la versión 7.0, llaves foráneas (integridad referencial).

Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.

Inconvenientes

Consume bastantes recursos y carga más el sistema.

2.10. Otras herramientas necesarias.

Como se pretende implementar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Webs, y una herramienta para el trabajo con las imágenes.

Para estas funciones la elección no ha sido muy difícil, debido a que nuestro país se esta insertando en el mercado del Software libre por las incontables ventajas que brinda hemos decidido que la herramienta que se utilice en el diseño y creación de sitios Webs sea el Kdevelop en el cual se puede realizar un diseño flexible y consistente de cualquier sitio Web.

Además de contar con un soporte para aplicaciones PHP y utilización de bases de Datos MySQL. También cuenta con un amplio soporte para la creación y utilización de CSS (Cascading Style Sheets), para lograr un diseño fácil y óptimo.

A pesar de todo lo que hemos dicho sobre el Open Source no hemos encontrado una herramienta que pueda sustituir al Adobe Photoshop como herramienta principal para crear las imágenes del Sistema, ya que se considera la aplicación estándar para el tratamiento digital de imágenes aunque bien vale destacar el Gimp, herramienta profesional distribuida libremente con todas las distribuciones de Linux que si bien no tiene una interfaz como Photoshop permite realizar casi todas sus funcionalidades. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y creación de imágenes. [4]

2.11. Conclusiones.

Al observar todas las posibles opciones que se ven enmarcadas dentro de las aplicaciones Web, fue necesario hacer un estudio de que utilizar para lograr vencer los objetivos de manera más óptima.

Se decide optar por utilizar el PHP, por sus características antes expuestas, principalmente, porque nos permitirá tener una aplicación multiplataforma, fácilmente transportable de un sistema operativo a otro, y por tener tiempos de respuestas muy rápidos, algo imprescindible en la Web.

Como SGBD tenemos utilizaremos MySql, para colaborar en la creación de una estructura basada en el Open Source en la Universidad de Ciencias Informáticas que es donde se implantará el sistema. Además por ser un gestor que se concibió para la gestión y creación de portales Web.

Para el modelado del sistema se va a utilizar la metodología de Rational Unified Process (RUP), ya que es un estándar universal para el diseño orientado a objetos, apoyándonos en la herramienta Case de Rational (Rational Rose).

Capítulo 3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

INTRODUCCIÓN.

En este capítulo se hace una breve descripción de la propuesta que planteamos con 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 el que se emplaza el sistema necesitamos definir conceptos que podamos agrupar en un modelo de Dominio, para capturar correctamente los requisitos y poder diseñar un sistema correctamente y funcional.

3.1. Descripción de los procesos de Negocio.

Para describir los procesos de negocio que se relacionan en el campo de acción de este trabajo, es necesario centrar la atención en los servicios que presta un **CMS** y los servicios que existen en la Intranet de la Universidad (**UCI**).

Para entender mejor como se realiza la creación, gestión y administración de contenidos en un **CMS** es mejor utilizar las técnicas de modelado que propone el **UML**.

3.2. Modelo de Dominio.

El sistema esta dirigido a cualquier tipo de usuario, no hay una clasificación o restricción en cuanto a que tipo de personas pueden utilizarlo, por tanto no es posible identificar una estructura o una dinámica organizacional, ya que se trata solo de usuarios accediendo a una aplicación. No es factible elaborar un modelo del negocio, y como esta herramienta se hará basándose en la que se está usando hoy en día en la universidad, se llega a la conclusión que para poder entender completamente el contexto en que se sitúa el sistema se necesita definir conceptos que podamos

agrupar en un Modelo conceptual, para capturar correctamente los requisitos y poder construir un sistema correcto.

3.3. Glosario de Términos.

Administrador: Persona que interactúa con un sistema o aplicación de software con el fin de administrar dicho sistema.

CMS: Núcleo de un sistema de Administración de contenidos. Utiliza diferentes clases para su funcionamiento.

Plantilla: Plataforma para crear interfaces graficas para la interacción con el usuario.

Usuario: Persona que utiliza un Sistema o Aplicación.

Grupo de Usuarios: Dominio de identificación de Usuarios, se utiliza mayormente para establecer la seguridad de los usuarios, también son importantes para la organización de dicha seguridad.

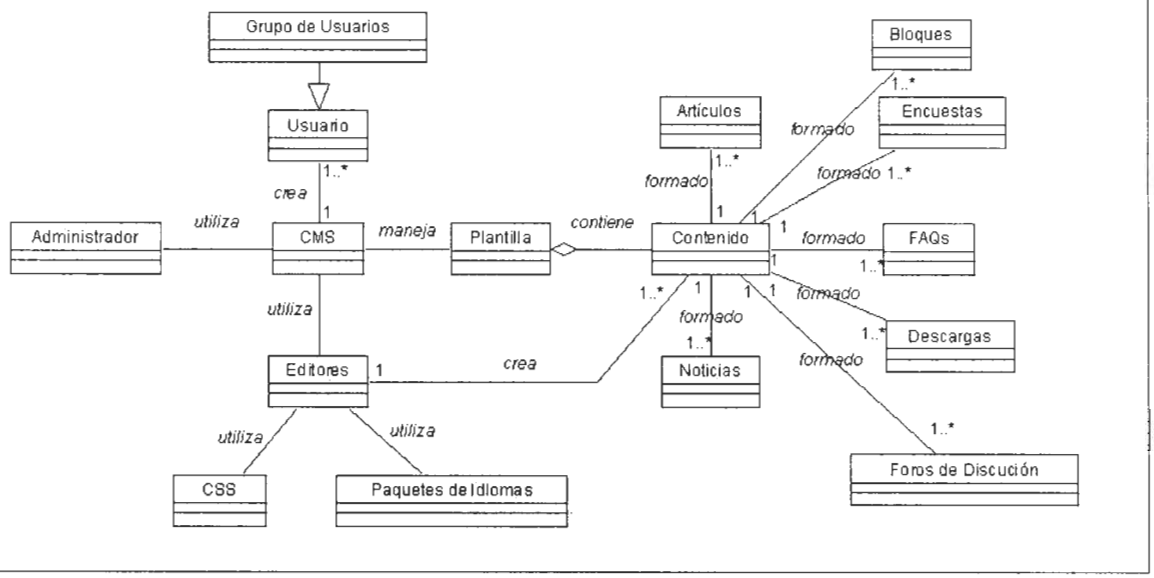
Editores: Software que se emplea para la creación de contenido digital.

Estilos: Se utilizan para crear estilos de diseños que se aplican a interfaces gráficas.

Paquetes de Idiomas: Paquetes que contienen reglas principalmente de Idiomas predefinidos en ellos, se utiliza para internacionalizar el Idioma de una aplicación.

Contenido: Información digital. Documentos, todo lo que se pueda considerar información o incluso, componentes, imágenes, banners, etc.

Modelo de Dominio



3.4. Requisitos Funcionales del Sistema.

Conocidos los conceptos que rodean al objeto de estudio, se debe analizar 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. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

RF1- Autenticar usuarios. (Usuario, Contraseña).

RF2- Registrar un usuario nuevo en el sistema. (Usuario, Grupo).

RF3- Actualizar usuarios. (Usuario, Grupo).

RF4- Eliminar usuarios. (Usuario).

RF5- Crear un nuevo grupo. (Nombre).

RF6- Actualizar grupo. (Nombre).

RF7- Eliminar grupo. (Nombre).

RF8- Asignar usuarios a un grupo. (Grupo, Usuario).

RF9- Asignar permisos de acceso a un grupo. (Grupo, Página).

RF10-Agregar módulos al Sistema.

RF11-Crear Bloques personalizados.

RF12-Administrar Bloques.

RF12.1-Asignar ubicación horizontal a los bloques.

RF12.2-Asignar orden vertical a los bloques.

RF13-Crear paquetes de Idioma adicionales para los módulos.

RF14-Agregar módulo de Noticias.

RF14.1 Decidir si un artículo específico aparece o no en la página principal del Sitio.

RF15- Publicar noticias.

RF16-Agregar módulo de Foros de Discusión.

RF17- Definir foros privados, disponibles sólo para miembros registrados.

RF18- Tener una vista preliminar de cada comentario antes de publicarlo.

RF19-Notificar a los usuarios sobre la mayor parte de los eventos que ocurran en los foros.

RF20-Agregar módulo de Descargas. (Download).

RF21-Proponer y calificar descargas.

RF22-Mostrar bloques para Descargas recientes y Descargas más populares.

RF23-Agregar módulo de Preguntas Frecuentes. (FAQ).

RF24-Clasificar las preguntas en categorías.

RF25-Administrar Artículos en el Sitio.

RF25.1-Publicar Artículos.

RF25.2-Remover Artículos del sitio.

RF26-Notificar a los usuarios cuando se publiquen nuevos Temas o Artículos.

RF27-Aprobar Artículos que propongan los visitantes antes de aparecer en el sitio.

RF28-Permitir que los administradores del sitio puedan ver una vista preliminar de sus notas antes de publicarlas.

RF29- Agregar URLs y direcciones de correo electrónico.

RF30-Agregar módulo de Encuestas.

RF31-Permitir un sistema de votaciones.

RF31.1-Permitir que los usuarios anónimos en los votos tengan los mismos derechos que los usuarios registrados.

RF32-Crear varias opciones de Voto.

RF33-Asignar fecha de expiración para cada encuesta. (Pero una encuesta puede ser reiniciada después de expirar).

3.5. Requerimientos no funcionales.

Apariencia o Interfaz Interna.

La interfaz debe ser sencilla, intuitiva, amigable y mantener el diseño de páginas similares. En general, fácil de usar.

Rendimiento.

Se necesita gran rapidez en el flujo de trabajo(Workflow), para asegurar mayor calidad en la gestión de contenidos ya que es necesaria para esta actividad.

Soporte.

Extensibilidad: se debe garantizar la inserción de módulos nuevos, sin afectar lo realizado hasta el momento o el buen funcionamiento.

Mantenimiento.

El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse. Además se harán salvas de los datos cada cierto tiempo.

Portabilidad.

El sistema se pondrá en función sobre plataforma Linux.

Seguridad.

Integridad:

La información será protegida contra corrupción y estados inconsistentes.

Disponibilidad:

Los usuarios autorizados tendrán acceso a la información en todo momento

Software.

Navegador compatible o superior con Internet Explorer 4, o Netscape Navegator, Mozilla, Mozilla FireFox.

HTML, PHP, MySQL.

Legales:

La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

Confiabilidad:

La herramienta de implementación a utilizar tiene soporte para la recuperación ante fallos y errores.

3.6. Descripción del Sistema propuesto.

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone va estar formado por seis módulos fundamentales: Administración del Sistema; Administración de módulos de noticias, Artículos, download, Encuestas, FAQ.

Se considera, dada la situación, que solo se tendrá en cuenta la existencia de un solo rol; o sea, un usuario se puede comportar solo como Administrador del Sistema.

En el SGBD se controlaran los datos y el rol de cada usuario que exista, también se almacenará todos los metadatos utilizados en los Sitios.

El módulo de administración de noticias puede ser utilizado por el administrador del sistema, y se utilizara para formar un sistema de noticias.

El módulo administración de Artículos forma parte de un mismo propósito junto con el de noticias. Crear un sistema de noticias en el Sitio.

Módulo descargas, es más bien un módulo estadístico el cual brinda la posibilidad de que al agregarlo al sitio se tendrán estadísticas de cuales son las descargas más populares.

Módulo FAQs como sus siglas en ingles lo indican es un modulo de Información para los usuarios que comiencen a usar esta herramienta. En él se ofrecerán respuesta a preguntas sobre todo del tipo técnicas pe. ¿Como se agrega un modulo de noticias a un sitio?
Entre muchas más según las dudas del usuario que utilice la herramienta.

Módulo encuestas se puede considerar un módulo del tipo estadístico pues su objetivo principal es definir encuestas y sistemas de votaciones para estas, obteniendo un resultado final de las diferentes encuestas que se hallan realizado.

El módulo administración de usuarios controlara todo lo referente a la configuración de usuarios y grupos de usuarios.

En resumen, con este sistema se podrá crear, gestionar y administrar cualquier Sitio Web sin reparar en cuanto a su uso y objetivo. De esta forma desde una misma aplicación se podrá gestionar y administrar todo el universo de contenido digital que posee la UCI, y a medida que se popularice el uso de esta herramienta entre los creadores (diseñadores, programadores) de aplicaciones Web en la universidad se

hará mas amplia una comunidad de usuarios que permitirá dar consejos para futuras versiones, agregar módulos implementados por ellos mismos entre otras cosas.

3.7. Modelo de Casos de Uso 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, cual o cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

ACTORES	JUSTIFICACIÓN
Administrador	Representa a una persona que configura y controla el comportamiento del sistema. Puede utilizar el sistema para crear nuevos Sitios, módulos, gestionar contenido, crear contenido, controlar los usuarios, etc.

3.7.1. Casos de uso del Sistema.

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son "fragmentos" de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. [Jacobson, 2000].

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos del sistema.

Nombre del Caso de uso: *Autenticar usuario.* CU-1.
Actores: Administrador del Sistema.
Propósito: Establecer un nivel de seguridad confiable en el sistema.
Descripción: El caso de uso inicia cuando un Administrador desea iniciar sección en el Sistema. El Admistrador debe introducir los datos que le pide el sistema para hacer esta operación. (Datos Nombre de usuario y contraseña.)
Referencias: RF1.
Precondiciones: El administrador se autentifica en el sistema.

Nombre del Caso de uso: *Administrar Usuarios*. CU-2.
Actores: Administrador del Sistema.
Propósito: Crear Usuarios y Grupos de Usuarios con el objetivo de hacer acl para una mejor configuración del Sistema. Esencial para establecer la seguridad el sistema.
Descripción: El caso de uso inicia cuando un Administrador desea adicionar, eliminar o actualizar un Usuario o Grupo de usuarios y darles el acceso correspondiente al sistema. Para adicionar o actualizar un usuario al sistema el Administrador introduce los datos (*Usuario, Grupo*), los cuáles pasan para la base de datos del sistema.
Referencias: RF2, RF3, RF4.
Precondiciones: Él Administrador selecciona la opción administrar usuarios.

Nombre del Caso de uso: *Administrar Grupo de Usuarios.* **CU-3.**
Actores: *Administrador del Sistema.*
Propósito: Administrar Grupos de Usuarios con el objetivo de tener una organización de accesos al sistema.
Descripción: El caso de uso inicia cuando un Administrador desea adicionar, eliminar o actualizar un Grupo de usuarios necesitando el dato (*Nombre del grupo*), luego el Administrador escoge un grupo y le asigna varios usuarios a él, además el Administrador puede escoger un grupo y asignarle diferentes permisos de acceso al Sistema.
Referencias: RF5, RF6, RF7, RF8, RF9.
Precondiciones: El administrador selecciona la opción administrar Grupos de Usuarios.

Nombre del Caso de uso: *Administrar Bloques* **CU-4.**
Actores: *Administrador del Sistema.*
Propósito: El objetivo de este caso de uso es administrar bloques de noticias, artículos para una mejor vista en el Sitio.
Descripción: El caso de uso inicia cuando un Administrador desea adicionar, eliminar o insertar un Bloque a un determinado Módulo del Sistema que forme parte de alguno de los sitios o portales que se administran. Para esto tiene que consultar en el menú Módulos ver bloques y adicionar, eliminar o insertar un bloque. El administrador tendrá otras opciones como alinear hacia la izquierda, hacia la derecha, vertical y horizontal. Tendrá opciones para personalizar Bloques.
Referencias: RF11, RF12, RF12.1, RF12.2.
Precondiciones: El administrador del Sistema Selecciona la opción Administrar bloques en el Menú Módulos.

Nombre del Caso de uso: *Administrar paquetes de Idiomas.* **CU-5.**
Actores: *Administrador del Sistema.*
Propósito: Surge con el propósito de dar soporte a gran parte de los idiomas del mundo principalmente Ingles, Español, etc.
Descripción: El caso de uso inicia cuando un Administrador accediendo a través del menú Módulos puede adicionar o actualizar un determinado paquete de Idiomas.
Referencias: RF13
Precondiciones: El administrador del Sistema selecciona la opción administrar paquetes de idioma en el Menú configuración.

Nombre del Caso de uso: *Administrar Módulo de Noticias.* **CU-6.**
Actores: *Administrador del Sistema.*
Propósito: Con el propósito de prestar servicios de noticias o sencillamente hacer un diario personal.
Descripción: El caso de uso inicia cuando un Administrador accediendo a través del menú Módulos puede adicionar, eliminar o actualizar un determinado módulo de noticias. El administrador tendrá la opción de ver una vista preliminar del modulo antes de publicarlo
Referencias: RF14, RF14.1, RF15.
Precondiciones: El Administrador del Sistema selecciona la opción administrar módulo de noticias en el Menú Administrar Módulos.

Nombre del Caso de uso: *Administrar Módulo de Foros de discusión* **CU-7.**
Actores: *Administrador del Sistema.*
Propósito: Con el objetivo de crear y administrar foros donde los usuarios de los distintos sitios puedan opinar sobre temas de interés.
Descripción: El caso de uso ocurre cuando el administrador del Sistema agrega un nuevo foro de noticias o lo actualiza teniendo la posibilidad de agregar nuevos temas o algunos propuestos por los visitantes (usuarios anónimos), darles prioridad a usuarios registrados y notificar a estos sobre nuevos foros.
Referencias: RF16, RF17, RF18, RF19.
Precondiciones: El Administrador del Sistema selecciona la opción Administrar Foros en el Menú Módulos.

Nombre del Caso de uso: *Administrar Módulo de descargas.* **CU-8.**
Actores: *Administrador del Sistema.*
Propósito: Con el objetivo de llevar de forma estadística las descargas más populares en los distintos Sitios.
Descripción: El caso de uso inicia cuando un Administrador del Sistema propone y califica módulos de descargas, estos se pueden calificar en recientes y populares.
Referencias: RF20, RF21, RF22.
Precondiciones: Cuando el Administrador del Sistema selecciona la opción administrar módulo de descargas en el Menú Módulos.

Nombre del Caso de uso: *Administrar Módulo de preguntas frecuentes.* **CU-9.**
Actores: *Administrador del Sistema.*
Propósito: El uso de los CMS es algo todavía en el mundo muy nuevo, se hace necesario tener módulos de FAQs para que los usuarios de los distintos sitios agilicen su aprendizaje a la hora de usar esta herramienta.
Descripción: El caso de uso inicia cuando un Administrador del Sistema utiliza el módulo de FAQs para clasificar las preguntas en categorías.
Referencias: RF23, RF24.
Precondiciones: El Administrador del Sistema selecciona la opción Administrar Módulo de FAQs en el Menú Módulos.

Nombre del Caso de uso: *Administrar Artículos en el Sitio.* **CU-10.**
Actores: *Administrador del Sistema.*
Propósito: Con el propósito de publicar Artículos de interés.
Descripción: El caso de uso inicia cuando un Administrador del Sistema actualiza, crea o elimina un Artículo de algún sitio.
Referencias: RF25, RF25.1, RF25.2, RF26, RF27, RF28.
Precondiciones: El Administrador del Sistema selecciona la opción Administrar Artículos en el Sitio, en el Menú Módulos.

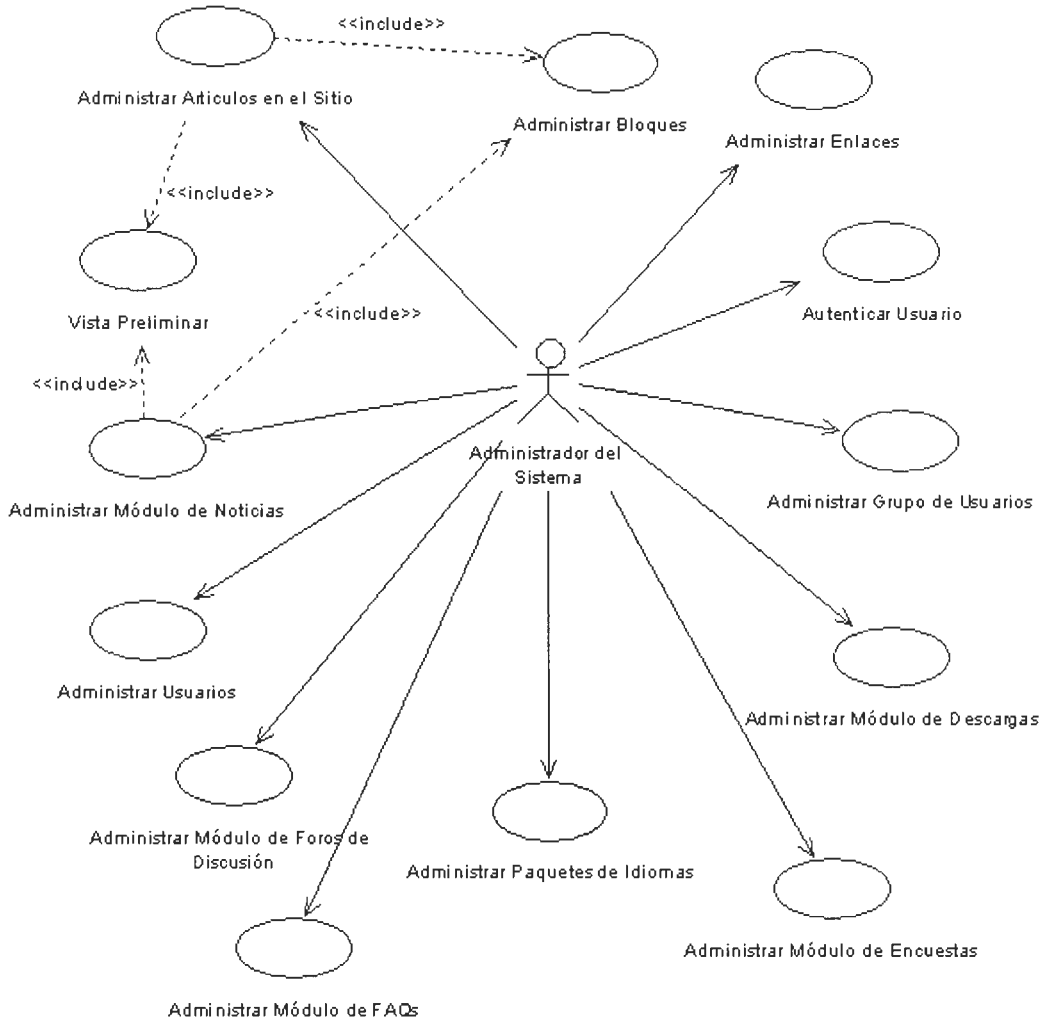
Nombre del Caso de uso: *Dar vista preliminar.* **CU-11.**
Actores: *Administrador del Sistema.*
Propósito: Cuando se edita un artículo o una noticia, antes de publicarlo se hace necesario tener una vista preliminar de ese artículo o noticia.
Descripción: El caso de uso inicia cuando un Administrador del Sistema desea ver una vista preliminar de un Artículo o noticia antes de publicarlo.
Referencias: RF28.
Precondiciones: El Administrador del Sistema selecciona la opción dar vista preliminar, antes de publicar algún artículo o noticia.

Nombre del Caso de uso: *Administrar Módulo de Encuestas.* **CU-12.**
Actores: *Administrador del Sistema.*
Propósito: Con el propósito de saber gustos y preferencias de los usuarios de los distintos sitios. Medir nivel de aceptación de algo en la comunidad de usuarios.
Descripción: El caso de uso inicia cuando un Administrador del Sistema agrega un sistema de votaciones donde pueda tener varias opciones de voto, posibilidad que voten los usuarios anónimos, fijar una fecha de expiración para cada encuesta además de poder reiniciar una encuesta.
Referencias: RF30, RF31, RF31.1, RF32, RF33.
Precondiciones: El Administrador del Sistema puede inicia una encuesta.

Nombre del Caso de uso: *Administrar Enlace.* **CU-13.**
Actores: *Administrador del Sistema.*
Propósito: Brindar la posibilidad de que visiten otros sitios dentro y fuera del CMS, que sea de interés para los usuarios.
Descripción: El caso de uso inicia cuando un Administrador del Sistema actualiza, agrega o elimina algún enlace o email del sistema.
Referencias: RF29.
Precondiciones: El Administrador del Sistema selecciona la opción agregar enlaces (email, URLs) en el Menú Módulos.

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

Diagrama de Casos de Uso de Sistema



Conclusiones.

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las funciones que debe tener el sistema, que se representaron mediante un Diagrama de Casos de Uso.

Capítulo 4. CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.

4.1. Introducción.

En este capítulo se modelan los artefactos que ayudan a manejar las complicaciones que implican la construcción de aplicaciones Web. Para ello los componentes de la aplicación se tratan como clases, y utilizando las extensiones del UML, se pueden presentar a través de diagramas de clases Web. Además se presenta el modelo de datos que es la base para construir finalmente la base de datos que soportará el trabajo del sistema. Finalmente después de modelar la lógica del negocio a través de las clases Web, se tratan los principios del diseño de la aplicación.

4.2. Diagrama de clases.

En el diagrama de clases para la aplicación, es donde hemos modelado los artefactos del sistema, es decir, páginas, enlaces entre estas, 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 o los desarrolladores los implementen y luego obtener así el producto final.

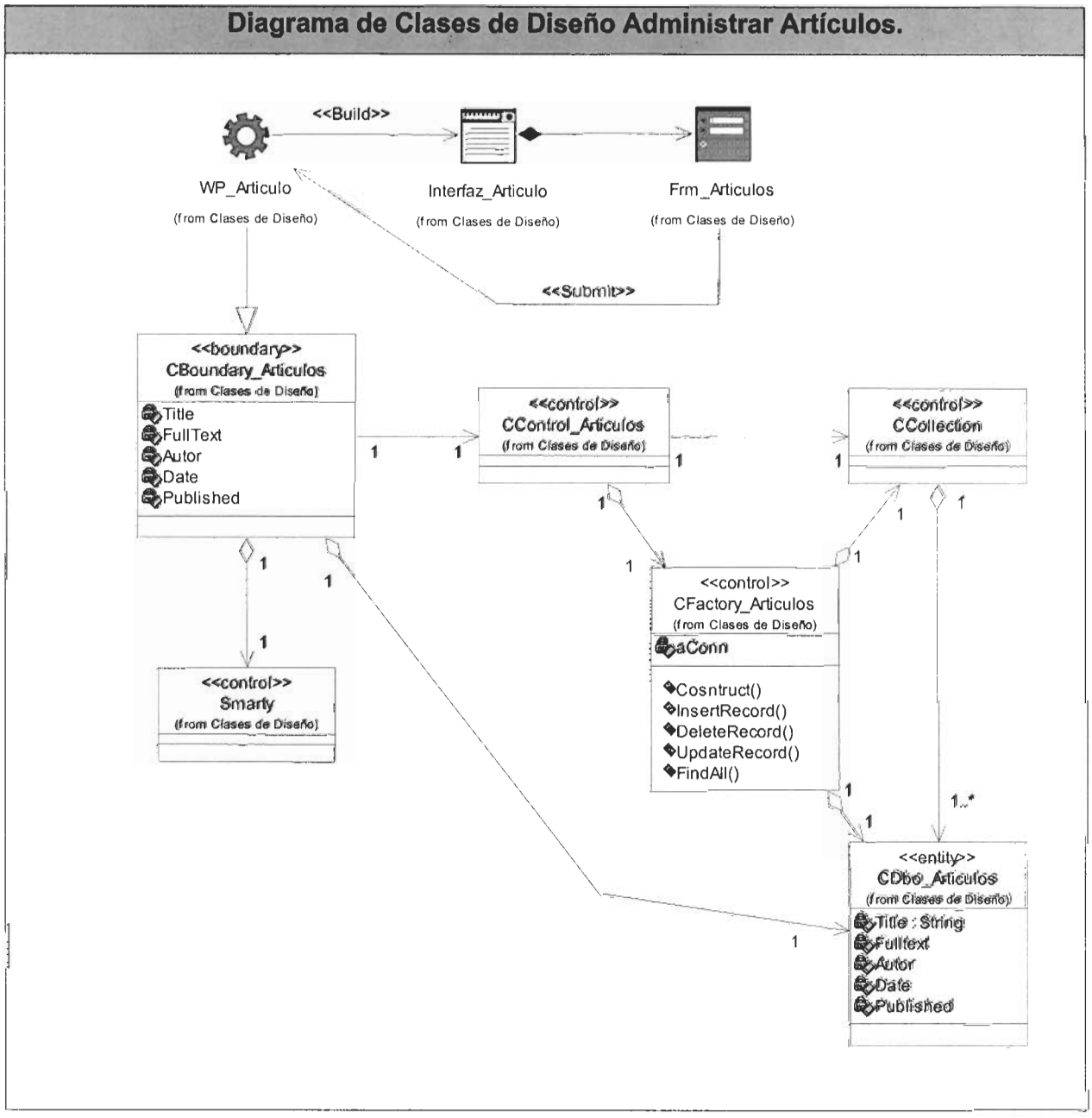
Casi todas las páginas del servidor tienen relación con algunos componentes, como pueden ser componentes de acceso a la base de datos, de sesión, etc. a continuación se presentan los diagramas de las clases que representan los procesos más importantes, se ocultan las operaciones de las clases para facilitar la comprensión de las mismas.

4.3. Diagramas de Clases de Diseño.

Aquí se muestran los diagramas de las clases de diseño que intervienen para dar solución a los casos de uso del sistema.

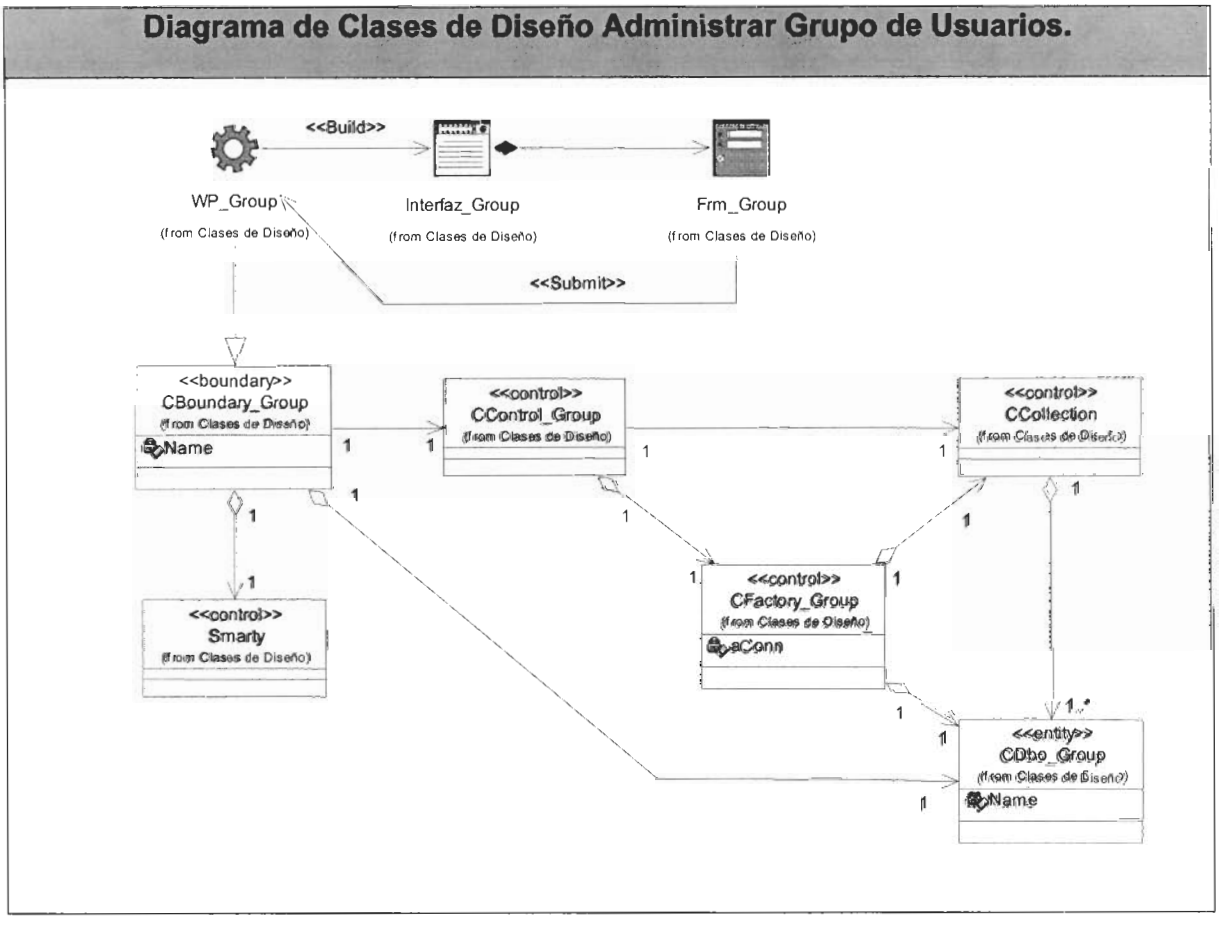
4.3.1. Administrar Artículos.

Diagrama de Clases de Diseño Administrar Artículos.



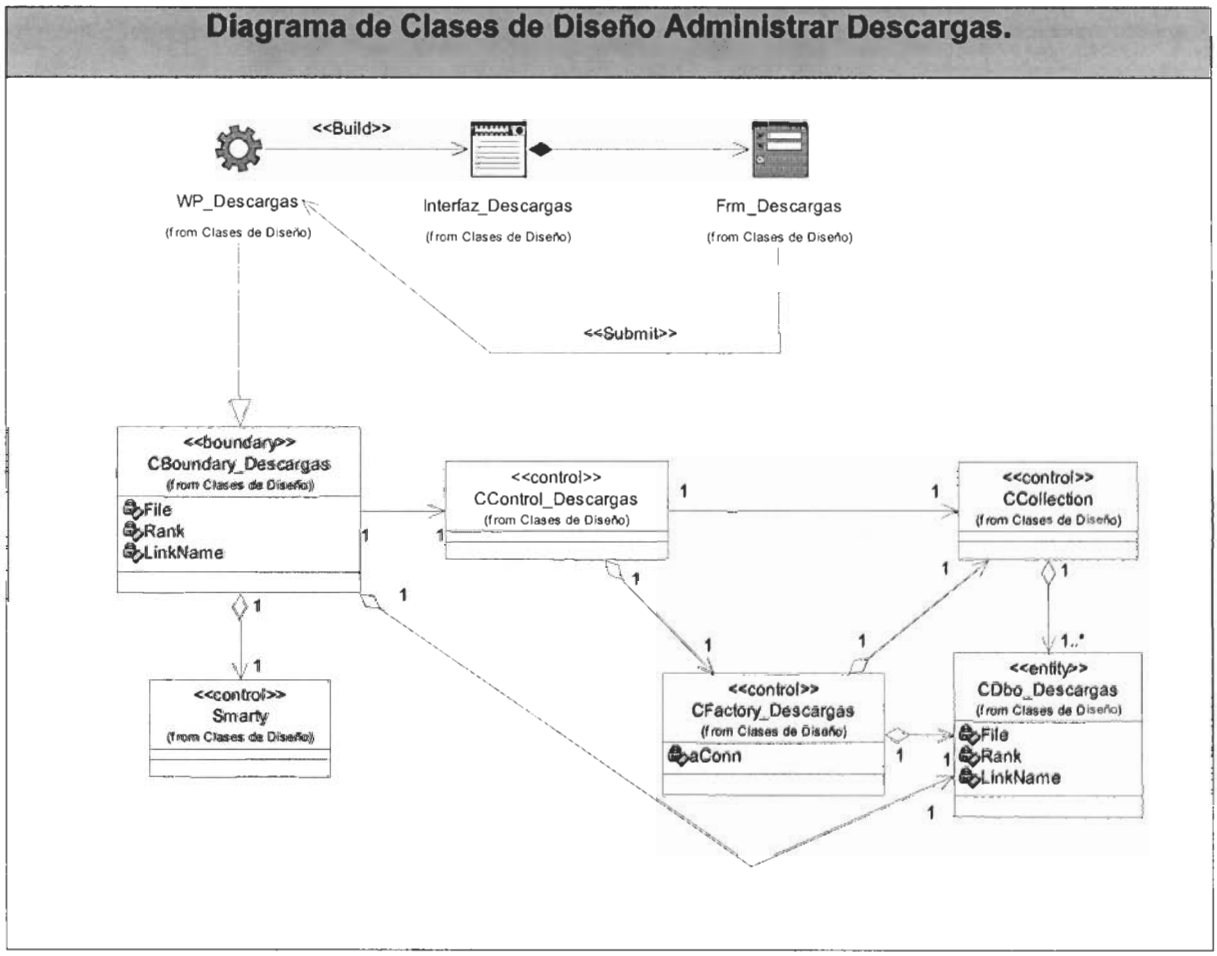
4.3.2. Administrar Grupo de Usuarios.

Diagrama de Clases de Diseño Administrar Grupo de Usuarios.



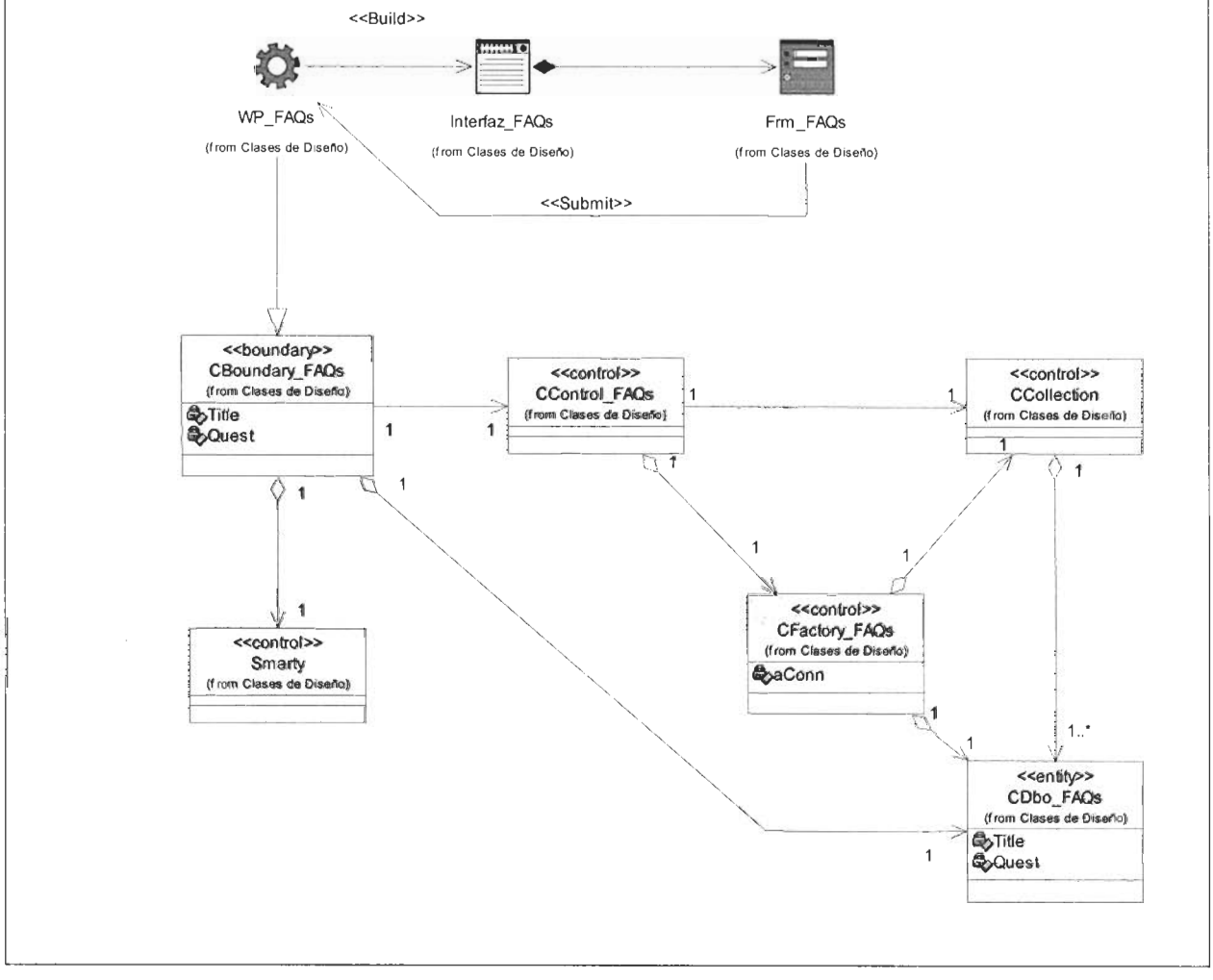
4.3.3. Administrar Descargas.

Diagrama de Clases de Diseño Administrar Descargas.



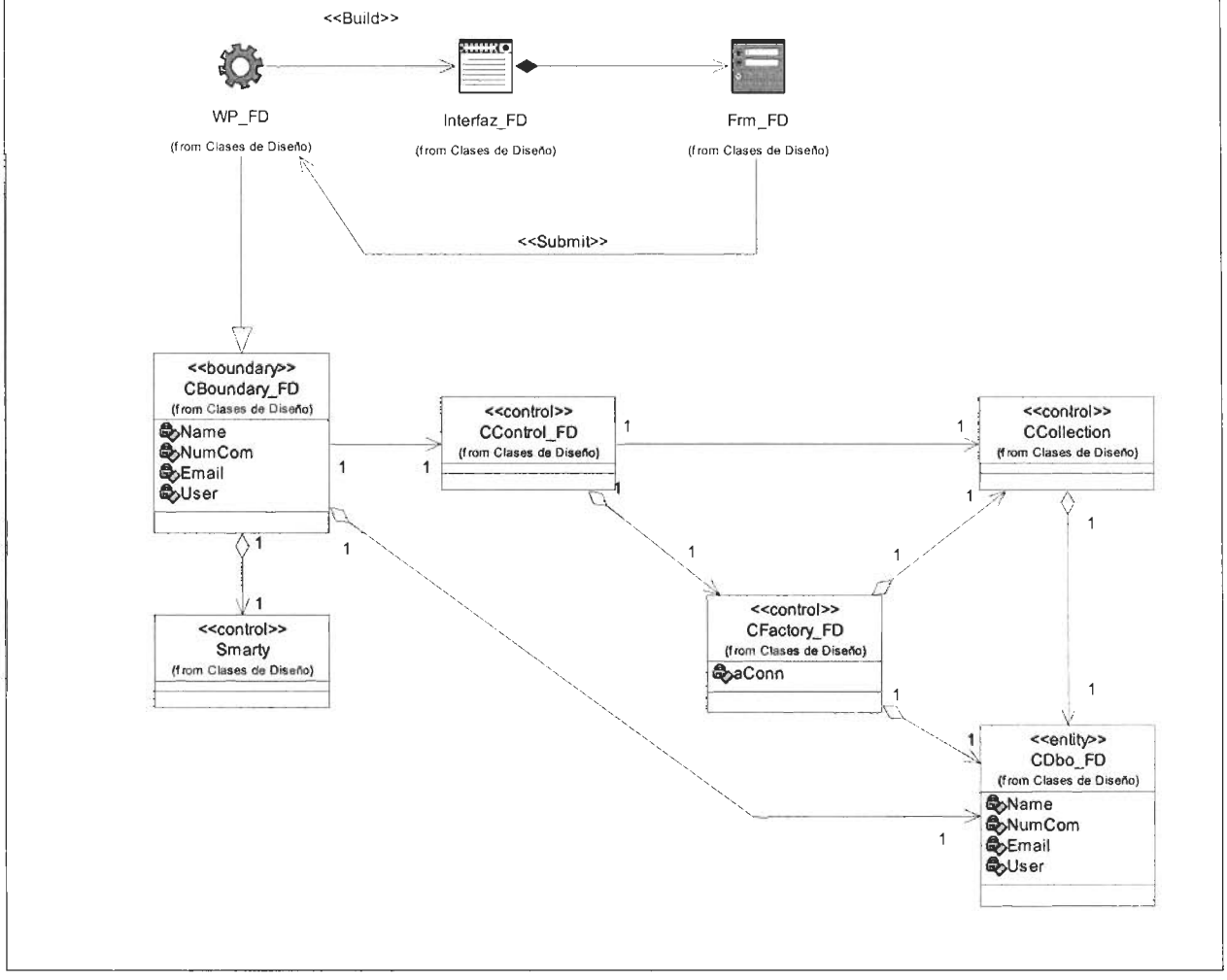
4.3.4. Administrar FAQs.

Diagrama de Clases de Diseño Administrar FAQs.



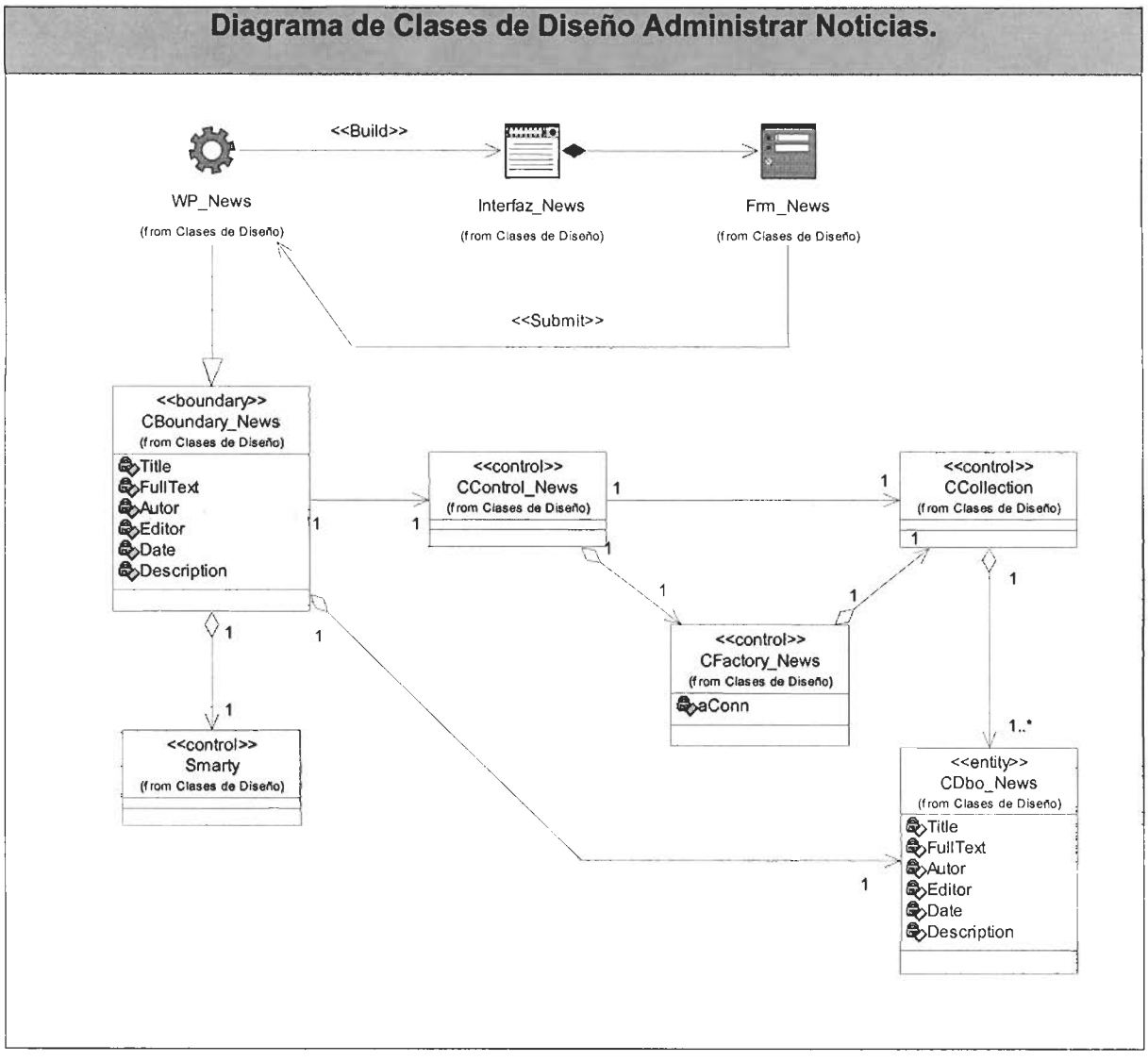
4.3.5. Administrar Foros de Discusión.

Diagrama de Clases de Diseño Administrar Foros de Discusión.



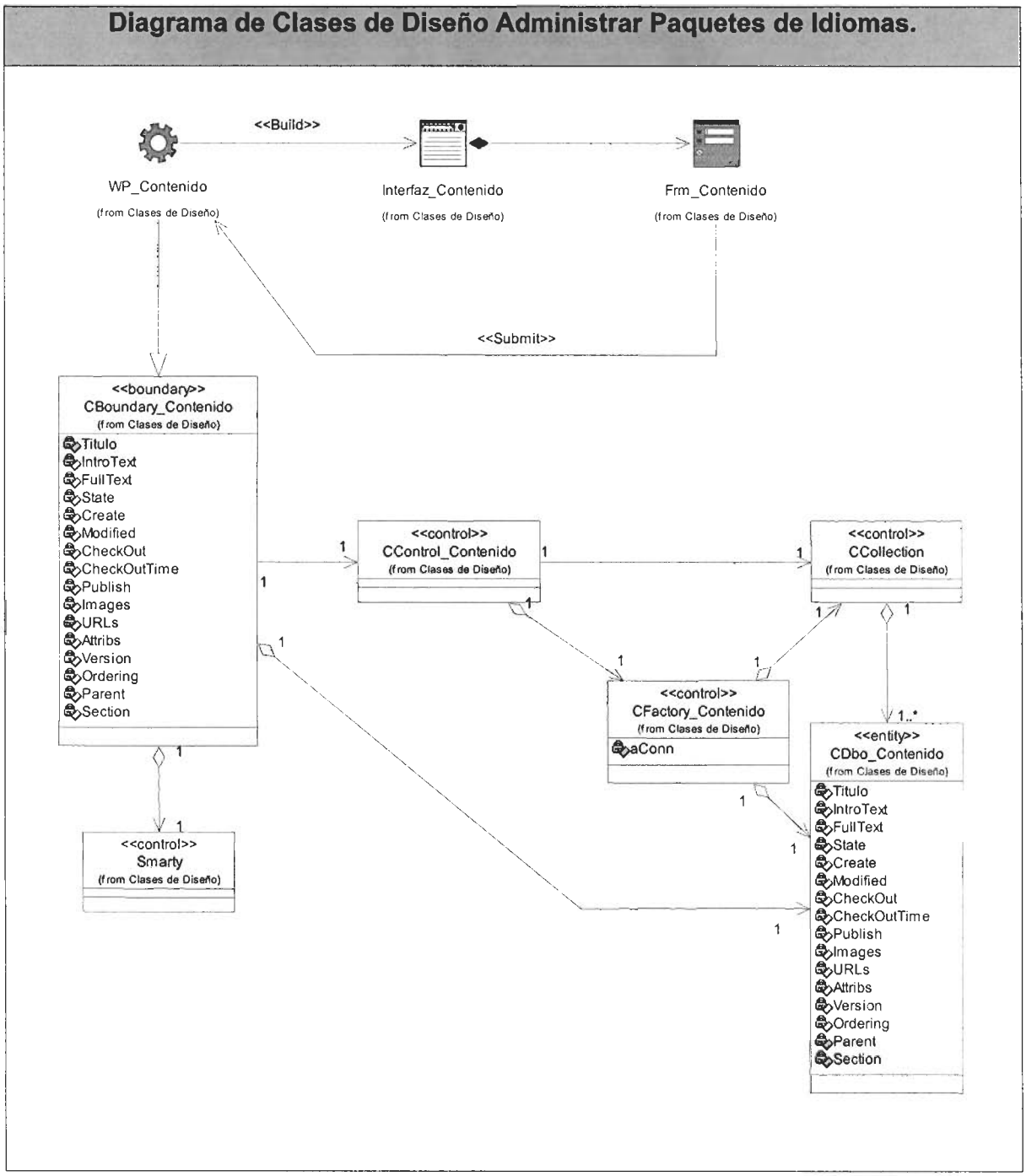
4.3.6. Administrar Noticias.

Diagrama de Clases de Diseño Administrar Noticias.



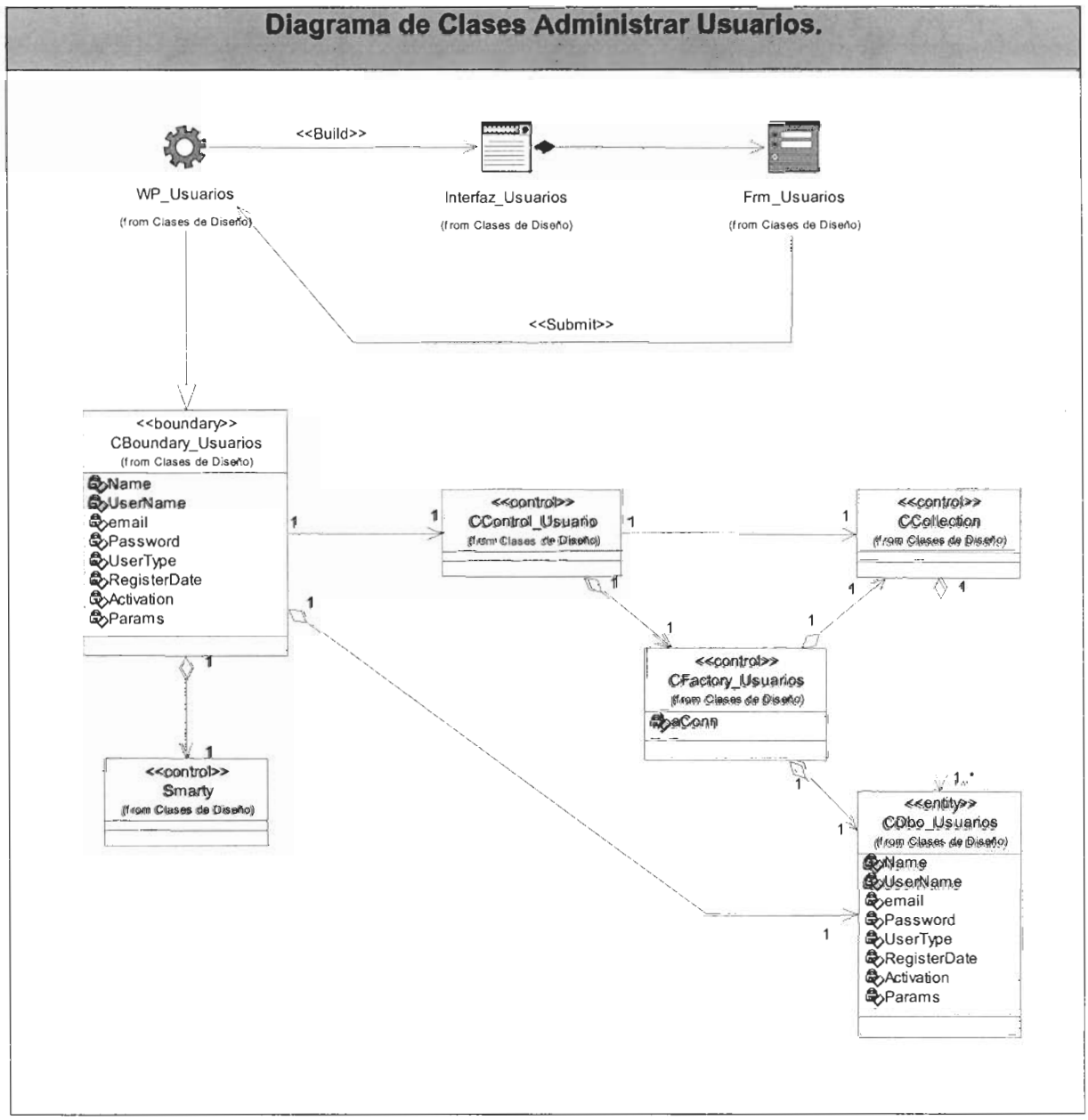
4.3.7. Administrar Paquetes de Idiomas.

Diagrama de Clases de Diseño Administrar Paquetes de Idiomas.



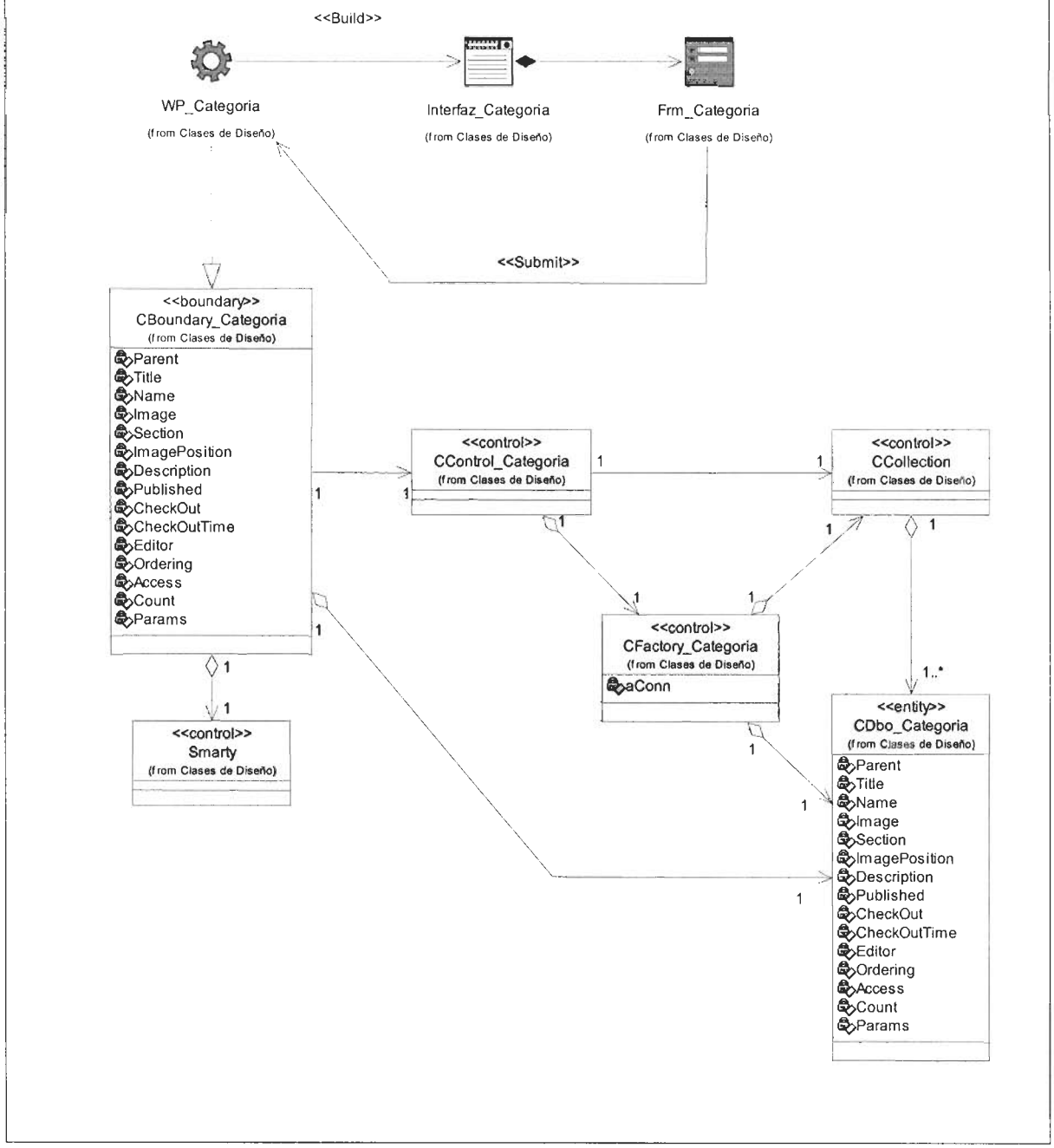
4.3.8. Administrar Usuarios.

Diagrama de Clases Administrar Usuarios.



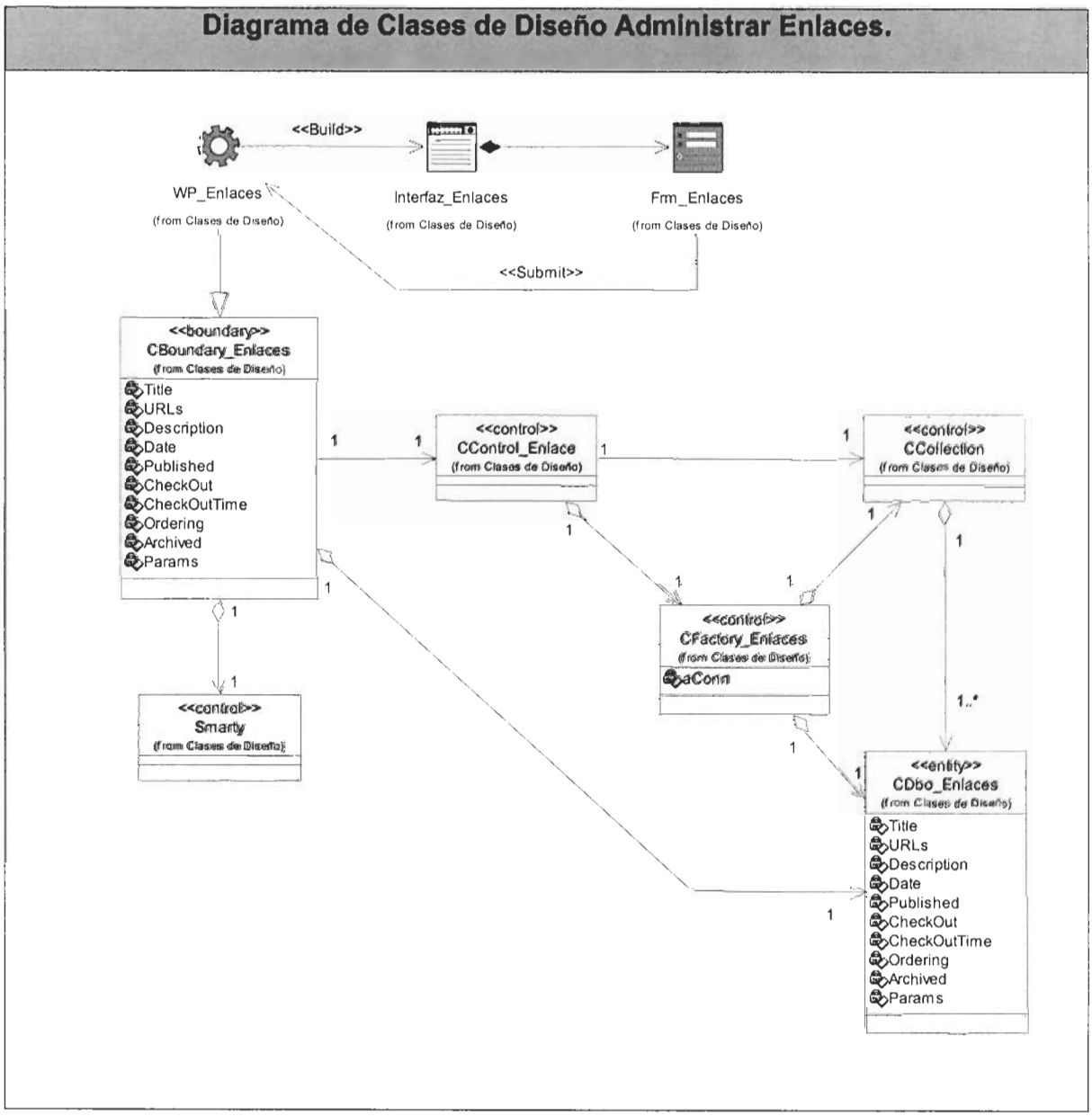
4.3.9. Administrar Bloques.

Diagrama de Clases Administrar Encuestas.



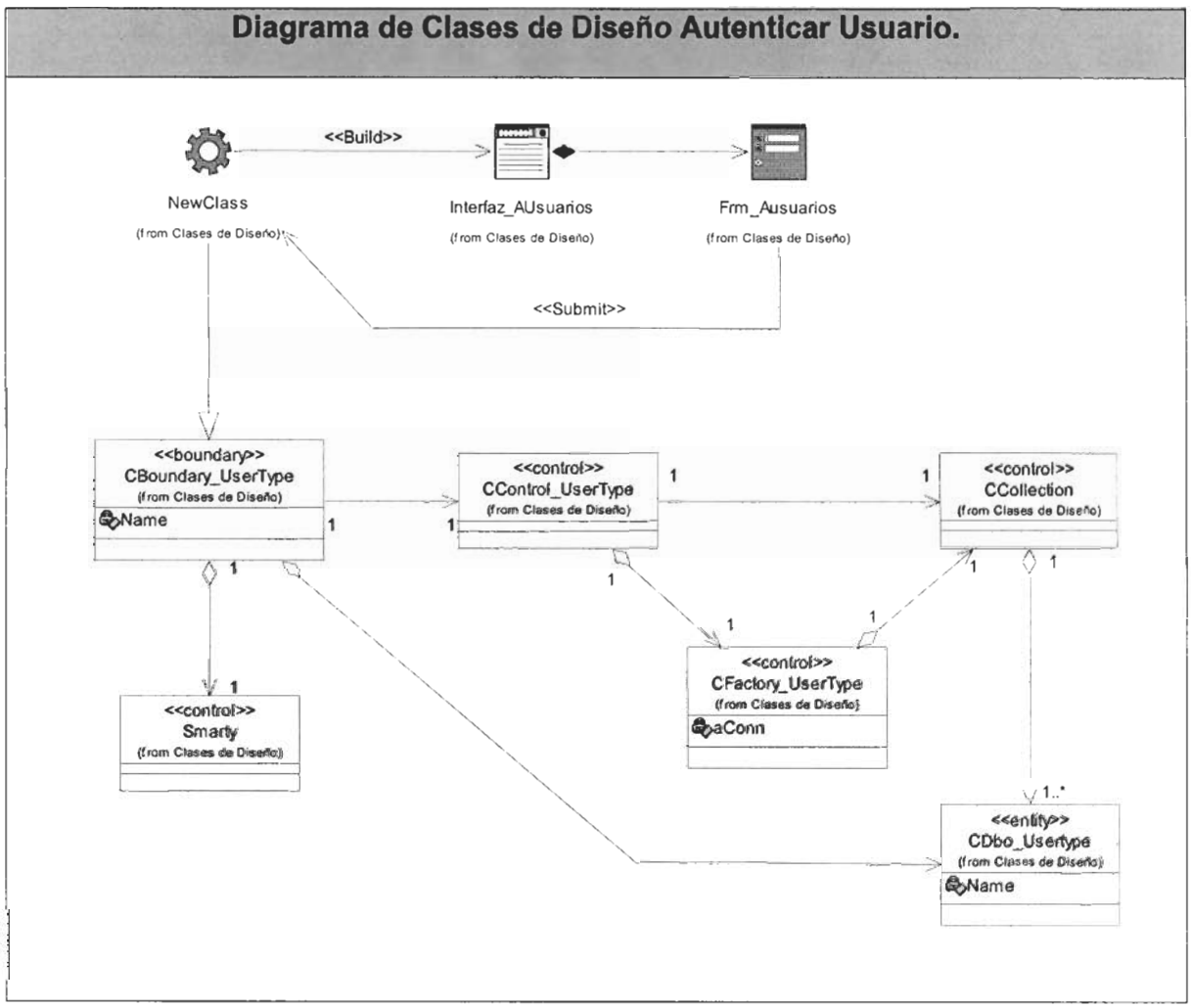
4.3.10. Administrar Enlaces.

Diagrama de Clases de Diseño Administrar Enlaces.



4.3.11. Autenticar Usuario.

Diagrama de Clases de Diseño Autenticar Usuario.



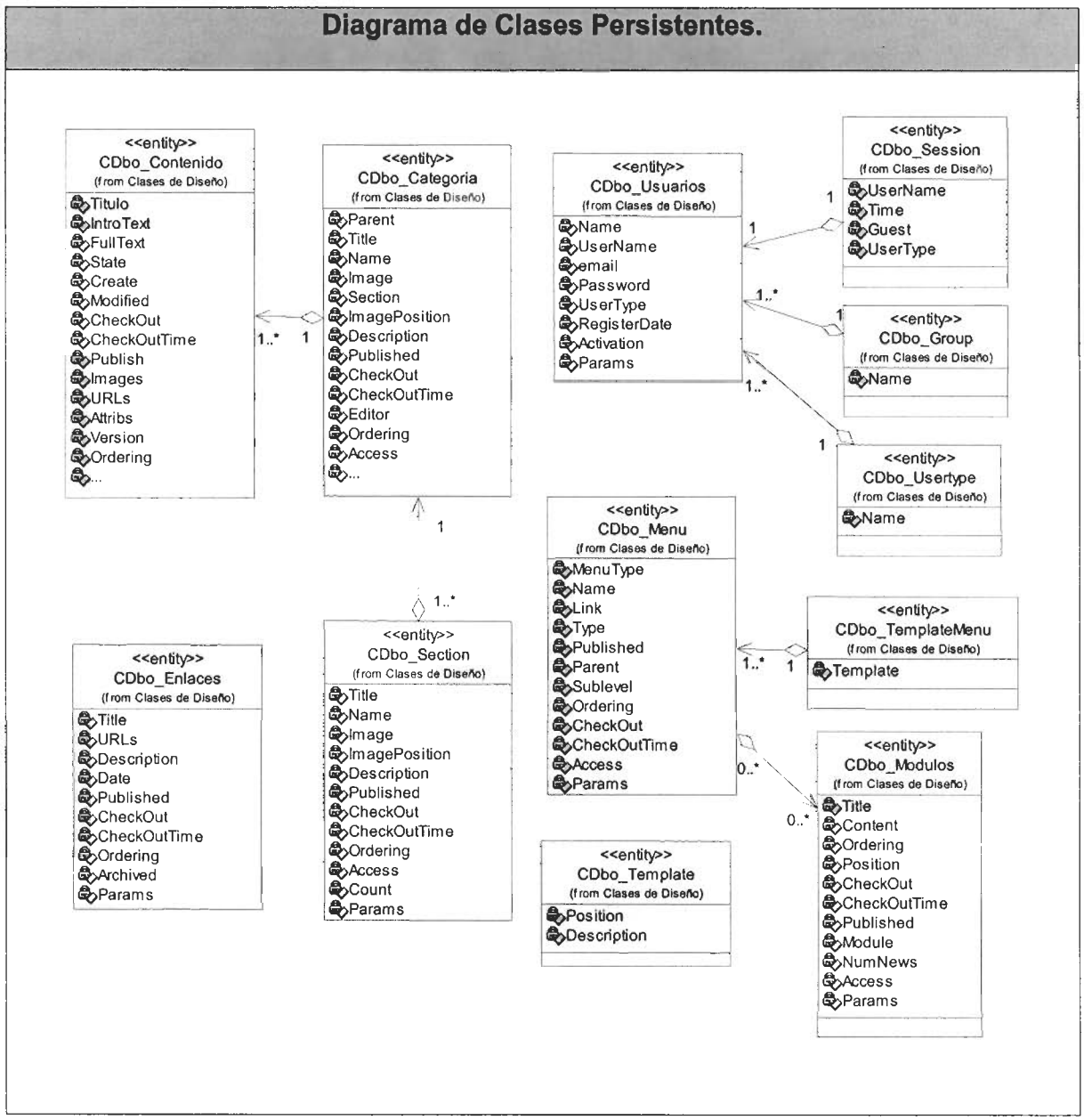
4.4. Diseño de la Base de Datos.

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

relación entre los datos, y completará el modelado de la lógica de negocio de la aplicación.

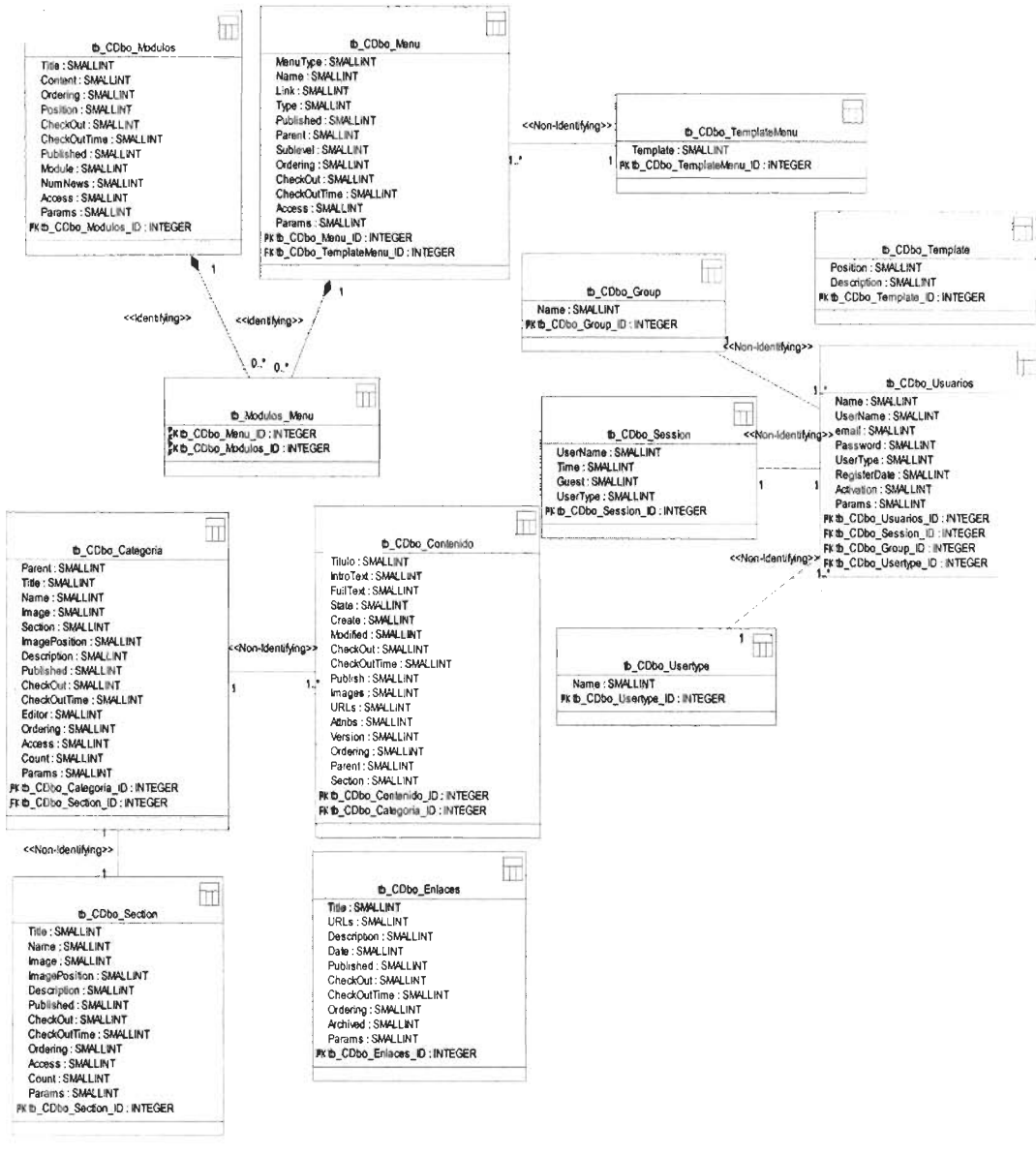
4.4.1. Diagrama de Clases Persistentes.

Diagrama de Clases Persistentes.



4.4.2. Modelo de Datos.

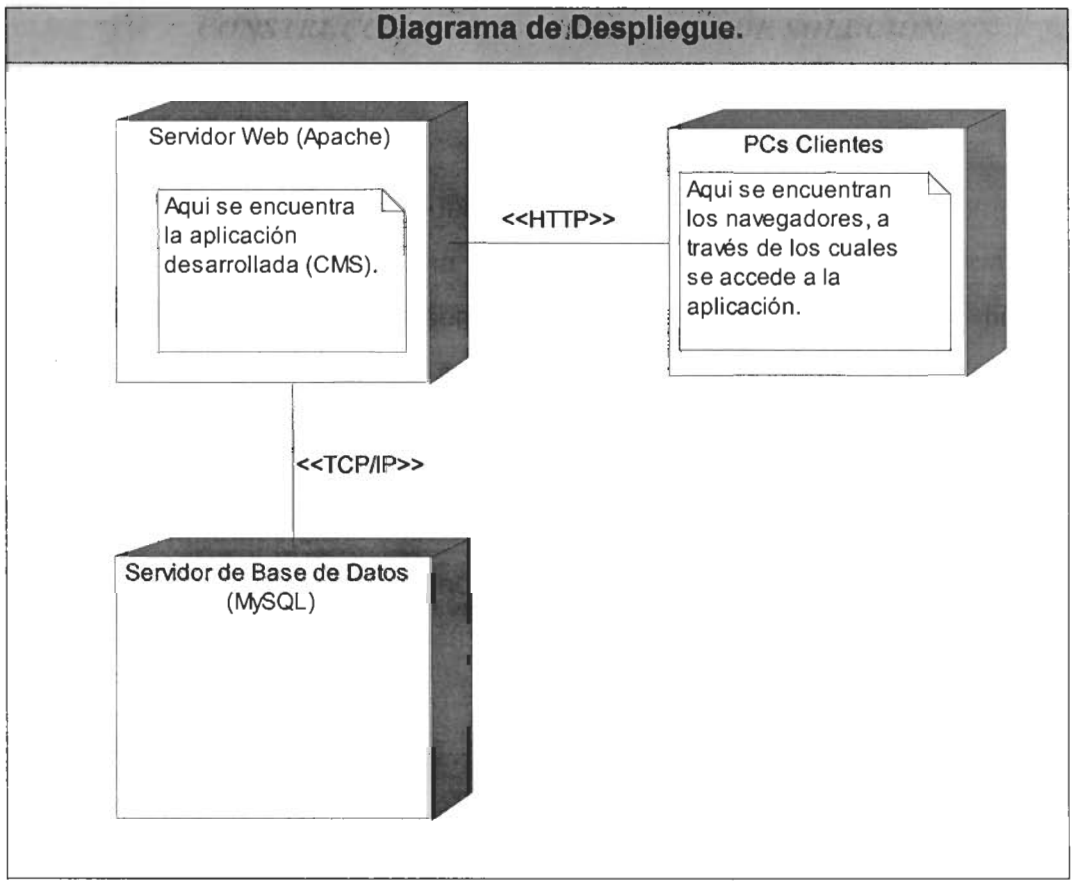
Modelo de Datos.



4.5. Diagrama de Despliegue.

Un diagrama de despliegue muestra las relaciones finales de los componentes hardware y software en el sistema final. Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes de software, objetos, procesos. En general un nodo será una unidad de computación de algún tipo, desde un sensor hasta un Mainframe.



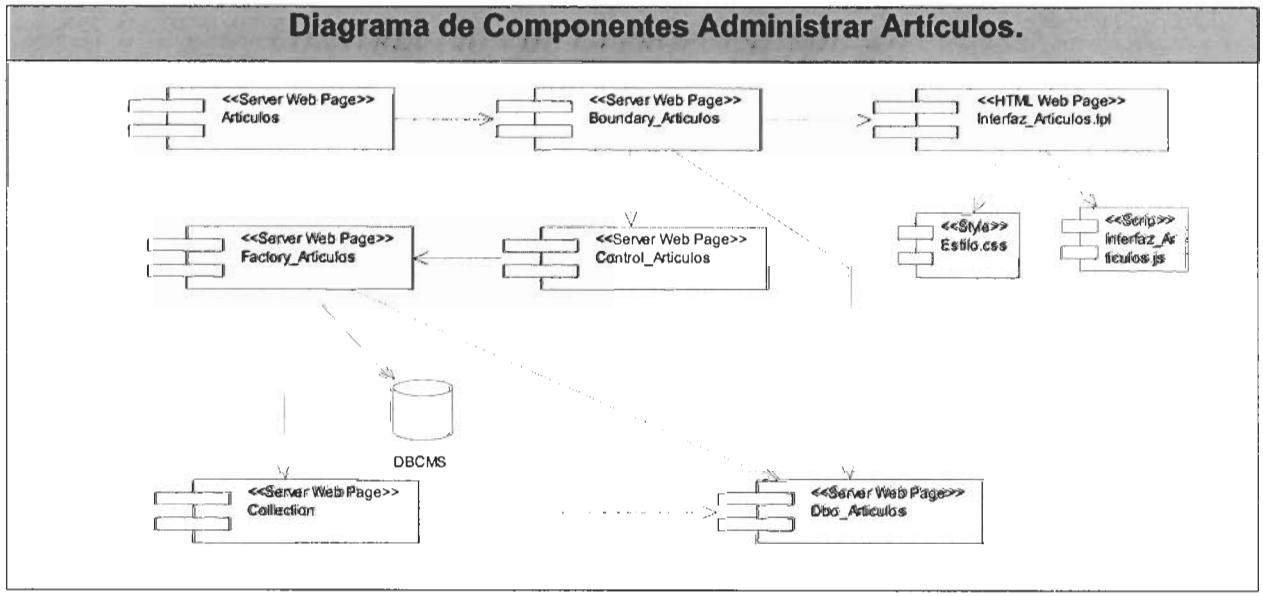
4.6. Diagrama de componentes.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes sólo aparecen tipos de componentes ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

En este epígrafe se mostraran de forma gráfica los diagramas de componentes por casos de uso.

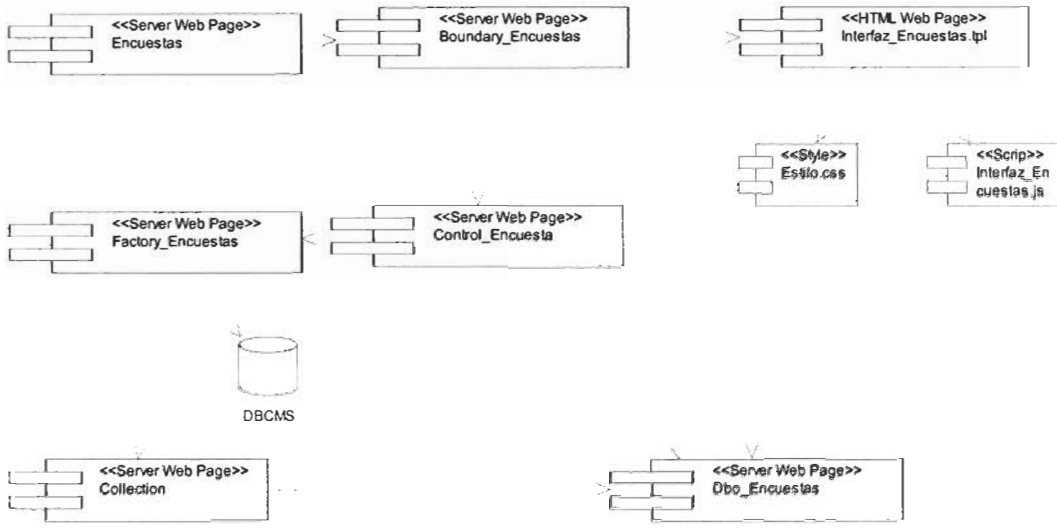
4.6.1. Administrar Artículos.

Diagrama de Componentes Administrar Artículos.



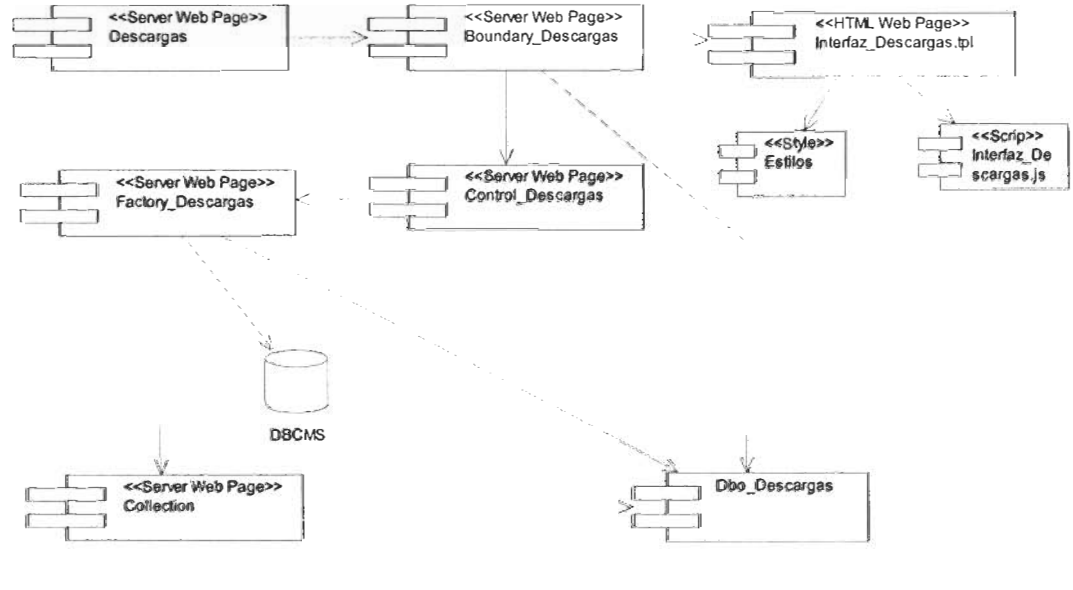
4.6.2. Administrar Encuestas.

Diagrama de Componentes Administrar Encuestas

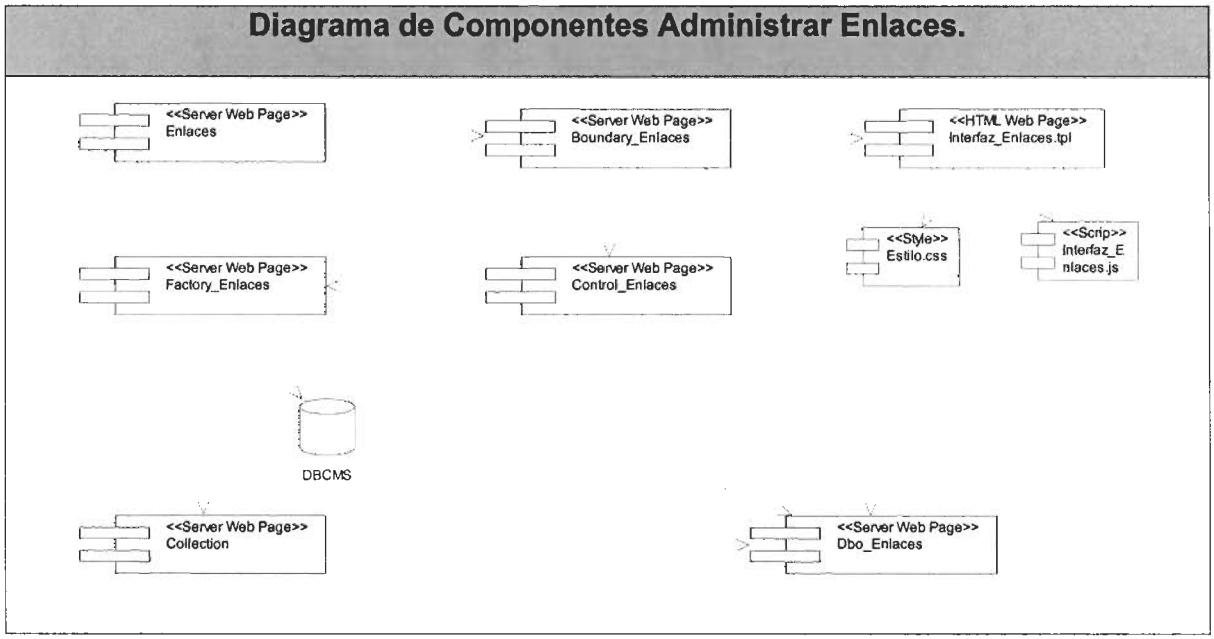


4.6.3. Administrar Descargas.

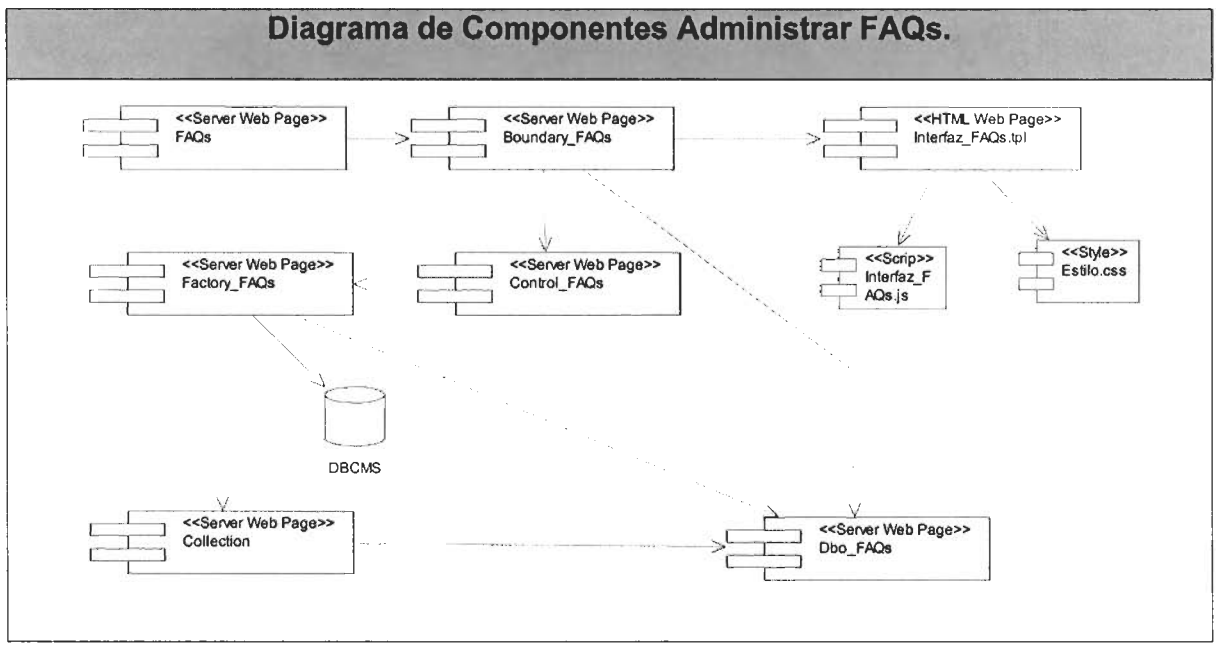
Diagrama de Componentes Administrar Descargas.



4.6.4. Administrar Enlaces.

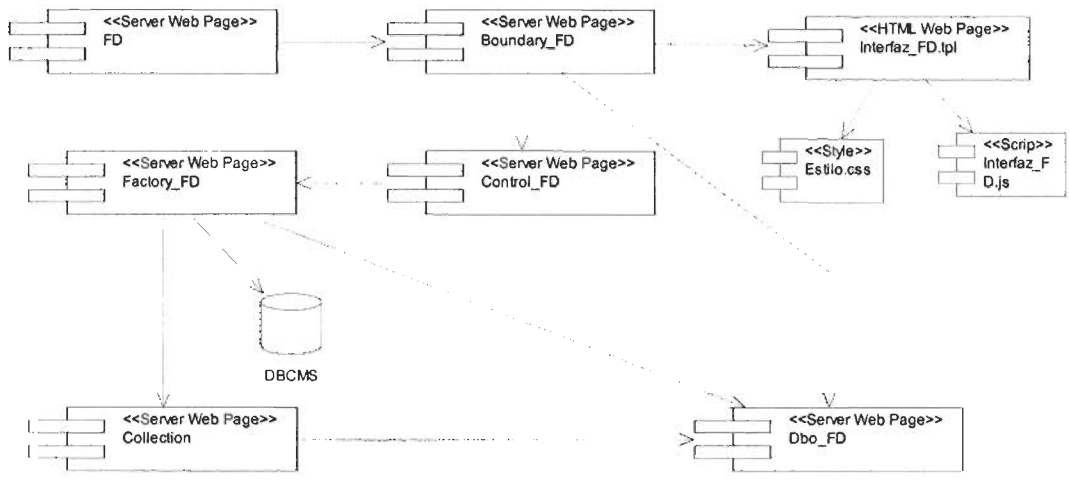


4.6.5. Administrar FAQs.



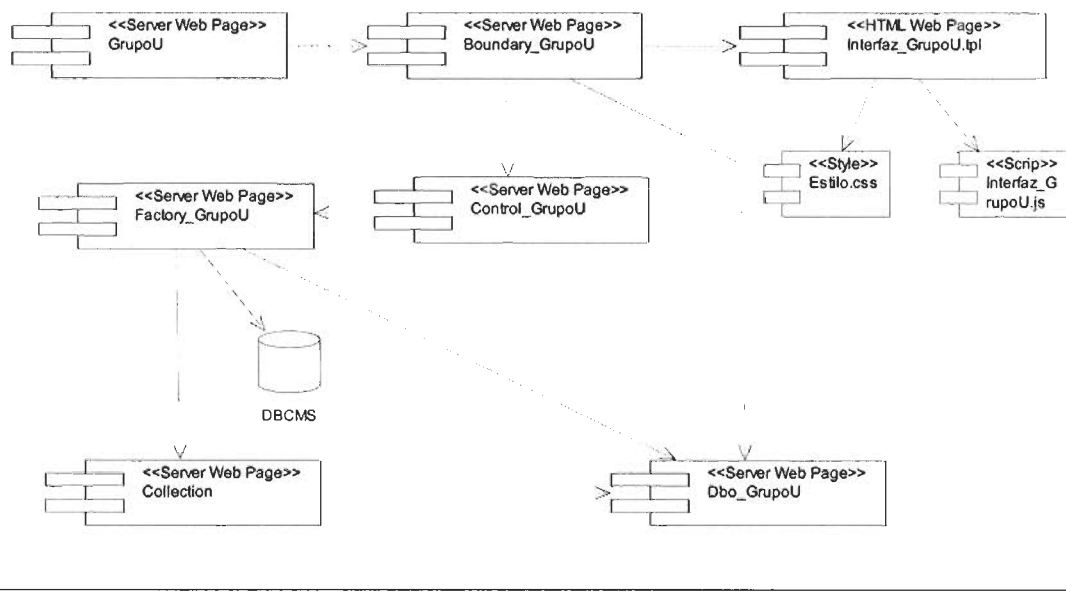
4.6.6. Administrar Foros de Discusión.

Diagrama de Componentes Administrar Foros de Discusión.



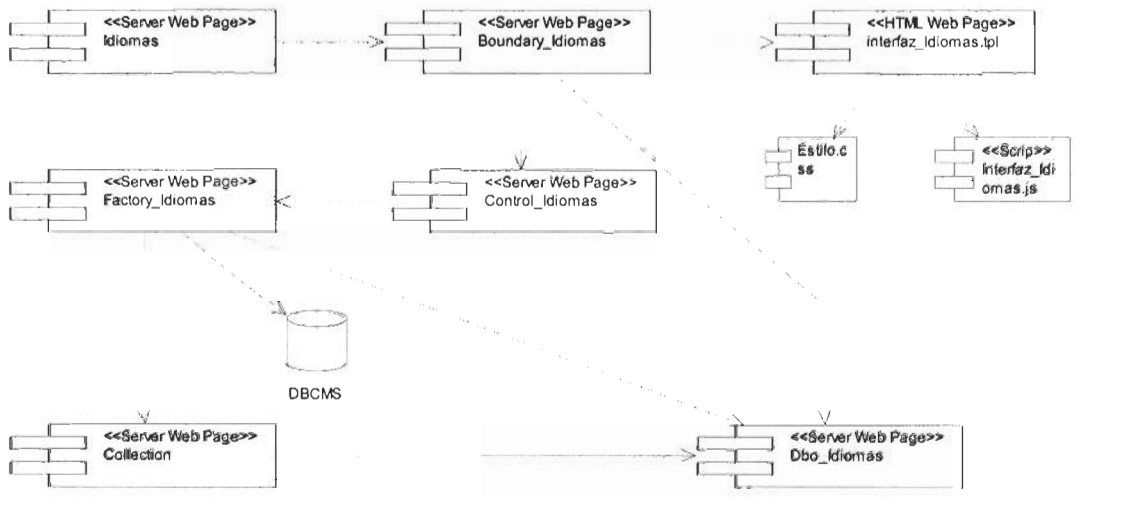
4.6.7. Administrar Grupo de Usuarios.

Diagrama de Componentes Administrar Grupo de Usuarios.



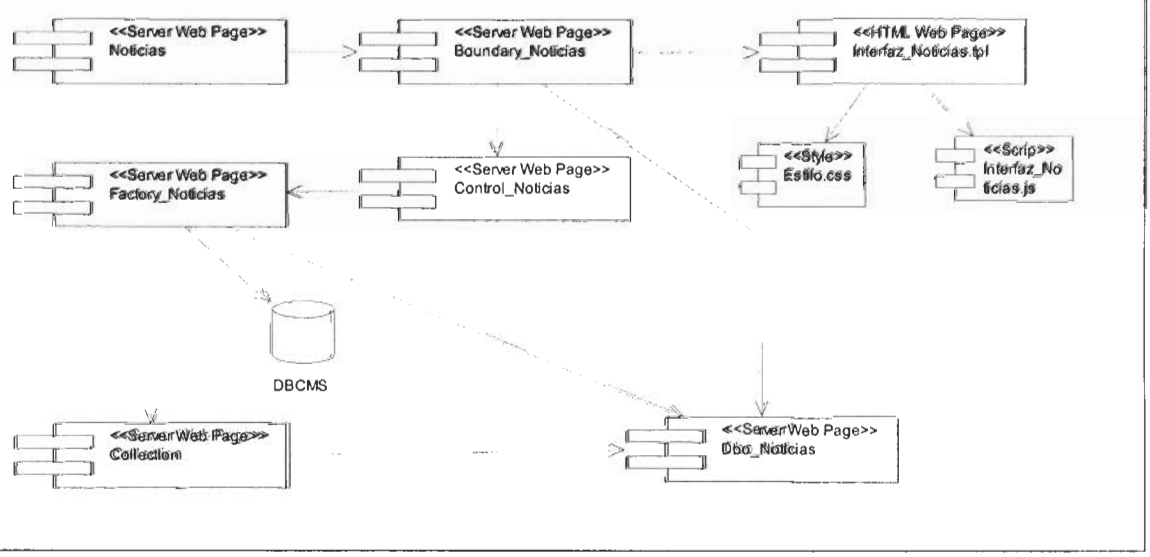
4.6.8. Administrar Paquetes de Idioma.

Diagrama de Componentes Administrar Paquetes de Idioma.



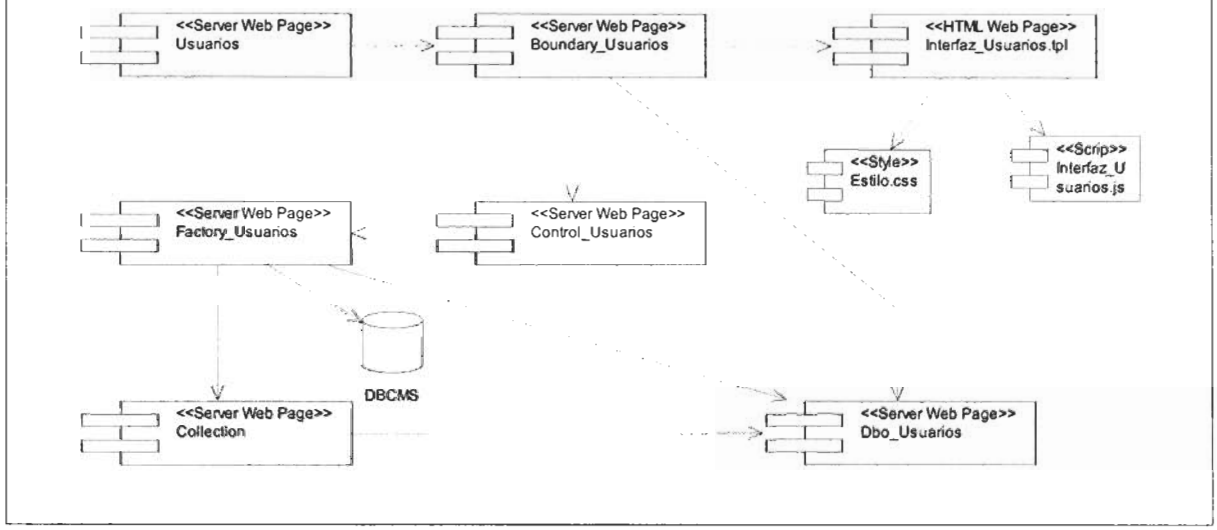
4.6.9. Administrar Noticias.

Diagrama de Componentes Administrar Noticias.



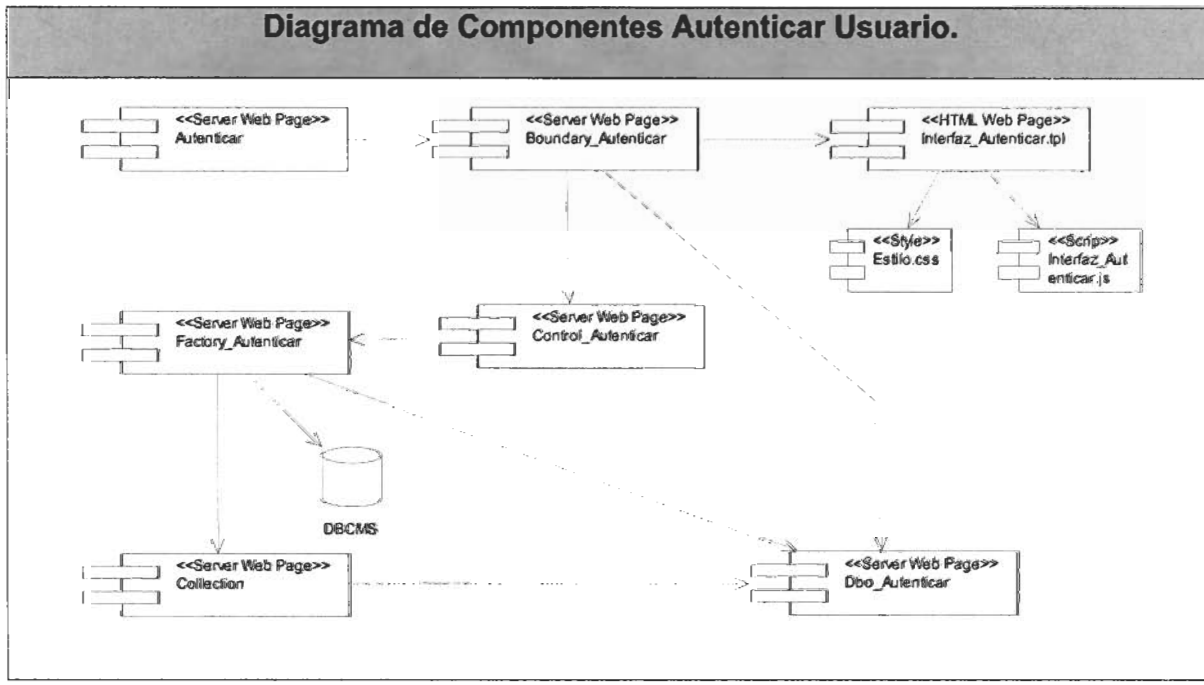
4.6.10. Administrar Usuarios.

Diagrama de Componentes Administrar Usuarios.



4.6.11. Autenticar Usuario.

Diagrama de Componentes Autenticar Usuario.



Componente	Propósito	Contenido
Articulo	Genera el código HTML necesario par administrar un artículo.	
Boundary_Articulo	Genera la interfaz del módulo Administrar Artículo	
Interfaz_Articulo.tpl	Resultado de la Clase Boundary_Articulos.	
Interfaz_Articulo.js	Interfaz en java script utilizada para la creación de interfaz gráfica.	
Estilo.css	Hojas de estilo, se utilizan para dar estilos a diseños Web.	
Control_Articulos	Se encarga de manipular todos los métodos para administrar un Artículo.	Contiene Métodos, como modificar, eliminar, agregar, etc.
Factory_Articulo	Se encarga de la conexión entre las clases y la BD.	Métodos de acceso a datos.
Dbo_Articulo	Se encarga de manipular los métodos propios de un Articulo.	
Collection	Clase Controladora.	
DBCMS.	DB de el CMS.	Todo el contenido que se gestiona en la aplicación.
Encuestas	Genera el código HTML necesario par administrar una Encuesta.	
Boundary_Encuestas	Genera la interfaz del módulo	

	Administrar una Encuesta.	
Interfaz_Articulo.tpl	Resultado de la Clase Boundary_Encuestas.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_Encuestas	Se encarga de manipular todos los métodos para administrar una Encuesta.	Contiene Métodos, como modificar, eliminar, agregar, etc.
Factory_Encuestas	Se encarga de la conexión entre las clases y la BD.	
Dbo_Encuestas	Se encarga de manipular los métodos propios de un Bloque.	
Descargas	Genera el código HTML necesario par administrar una Descarga.	
Boundary_Descargas	Genera la interfaz del módulo Administrar Descargas.	
Interfaz_Descargas.tpl	Resultado de la Clase Boundary_Descargas.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_Descargas	Se encarga de manipular todos los métodos para administrar una Descarga.	
Factory_Descarga	Se encarga de la conexión entre las clases y la BD.	
Dbo_Descargas	Se encarga de manipular los métodos propios de un Bloque	
Enlaces	Genera el código HTML	

	necesario par administrar un Enlace.	
Boundary_Enlaces	Genera la interfaz del módulo Administrar Enlaces.	
Interfaz_Enlaces.tpl	Resultado de la Clase Boundary_Enlaces.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_Enlaces	Se encarga de manipular todos los métodos para administrar un Enlace.	
Factory_Enlaces	Se encarga de la conexión entre las clases y la BD.	
Dbo_Enlaces	Se encarga de manipular los métodos propios de un Enlace.	
FAQs	Genera el código HTML necesario par administrar un módulo de FAQs.	
Boundary_FAQs	Genera la interfaz del módulo Administrar FAQs.	
Interfaz_FAQs.tpl	Se generan a partir de la Boundary_FAQs.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_FAQs	Se encarga de manipular todos los métodos para administrar el módulo FAQs.	
Factory_FAQs	Se encarga de la conexión entre las clases y la BD.	
Dbo_FAQs	Se encarga de manipular los	

	métodos propios de un módulo de FAQs.	
FD	Genera el código HTML necesario par administrar un Módulo FD.	
Boundary_FD	Genera la interfaz del módulo Administrar FD.	
Interfaz_FD.tpl	Se generan a partir de la Boundary_FAQs.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_FD	Se encarga de manipular todos los métodos para administrar el módulo FD.	
Factory_FD	Se encarga de la conexión entre las clases y la BD.	
Dbo_FD	Se encarga de manipular los métodos propios del módulo FD.	
GrupoU	Genera el código HTML necesario par administrar un grupo de usuarios.	
Boundary_GrupoU	Genera la interfaz del módulo Administrar Grupo de Usuarios.	
Interfaz_GrupoU.tpl	Se generan a partir de la Boundary_GrupoU.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_GrupoU	Se encarga de manipular todos los métodos para administrar los grupos de	

	usuarios.	
Factory_GrupoU	Se encarga de la conexión entre las clases y la BD.	
Dbo_GrupoU	Se encarga de manipular los métodos propios del módulo Grupo de usuarios.	
Idiomas	Genera el código HTML necesario par administrar el módulo de Paquetes de Idiomas.	
Boundary_Idiomas	Genera la interfaz del módulo Administrar Paquetes de Idiomas.	
Interfaz_Idioma.tpl	Se generan a partir de la Boundary_Idiomas.	Contienen imágenes, editores, objetos gráficos que forman una interfaz.
Control_Idiomas	Se encarga de manipular todos los métodos para administrar paquetes de Idiomas.	
Factory_Idiomas	Se encarga de la conexión entre las clases y la BD.	
Dbo_Idiomas	Se encarga de manipular los métodos propios del módulo Paquetes de Idiomas.	
Noticias	Genera el código HTML necesario par administrar el módulo de Noticias.	
Boundary_Noticias	Genera la interfaz del módulo	

	Administrar Noticias.	
Interfaz_Noticias.tpl	Se generan a partir de la Boundary_Idiomas.	Contienen imágenes, editores, objetos gráficos que forman una interfaz
Control_Noticias	Se encarga de manipular todos los métodos para administrar Noticias.	
Factory_Noticias	Se encarga de la conexión entre las clases y la BD.	
Dbo_Noticias	Se encarga de manipular los métodos propios para administrar, gestionar, publicar una Noticia.	
Usuarios	Genera el código HTML necesario par administrar Usuarios en el Sistema.	
Boundary_Usuarios	Genera la interfaz del módulo Administrar Usuarios.	
Interfaz_Usuarios.tpl	Se generan a partir de la Boundary_Usuarios.	Contienen imágenes, editores, objetos gráficos que forman una interfaz
Control_Usuarios	Se encarga de manipular todos los métodos para administrar Usuarios en el sistema.	

Factory_Usuarios	Se encarga de la conexión entre las clases y la BD.	
Dbo_Usuarios	Se encarga de manipular los métodos propios para administrar Usuarios en el Sistema.	
Autenticar	Genera el código HTML necesario para que un usuario se autentifique correctamente en el Sistema.	
Boundary_Autenticar	Genera la interfaz del módulo Autenticar Usuario.	
Interfaz_Autenticar.tpl	Se generan a partir de la Boundary_Autenticar.	Contienen imágenes, editores, objetos gráficos que forman una interfaz
Control_Autenticar	Se encarga de manipular todos los métodos para Autenticar a un Usuario.	
Factory_Autenticar	Se encarga de la conexión entre las clases y la BD.	
Dbo_Autenticar	Se encarga de manipular los métodos propios para Autenticar Usuarios en el Sistema.	
Clases Persistentes		
CDbo_Contenido	*Clase persistente.*	
CDbo_Categoria	*Clase persistente.*	
CDbo_Section	*Clase persistente.*	

CDbo_Enlaces	*Clase persistente.*	
CDbo_Usuarios	*Clase persistente.*	
CDbo_Session	*Clase persistente.*	
CDbo_Group	*Clase persistente.*	
CDbo_UserType	*Clase persistente.*	
CDbo_Menu	*Clase persistente.*	
CDbo_Modulos	*Clase persistente.*	
CDbo_TemplateMenu	*Clase persistente.*	
CDbo_Template	*Clase persistente.*	

4.7. Conclusiones.

En este capítulo se mostraron varias vistas para llevar acabo el proceso de implementación del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema, y se diseñaron las clases persistentes que permiten crear el diagrama entidad-relación, en el sistema de gestión de base de datos que se utilizará. En este momento ya se ha completado la propuesta que trae este trabajo.

Capítulo 5. ESTUDIO DE FACTIBILIDAD.

5.1. Introducción.

Hoy en día el software es el elemento más caro de la mayoría de los sistemas informáticos, un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas. Es necesario y prudente evaluar la viabilidad de un proyecto cuanto antes. Se pueden evitar meses o años de esfuerzo, miles o millones de dólares si se reconoce un sistema mal concebido en la pronta fase de definición.

En el presente capítulo se abordara el tema de estudio y factibilidad del sistema donde se brinda una descripción de la planificación del proyecto así como los costos asociados al mismo y los beneficios tangibles e intangibles que reportaría su elaboración. Finalmente se realiza un análisis entre los costos y beneficios que aporta el sistema.

Para realizar estas estimaciones utilizamos el modelo de COCOMO. COCOMO es una jerarquía de modelos de estimación de costes, que incluye submodelos básicos, intermedio y detallado.

5.2. Planificación

Tabla Entradas externas

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Administrar Artículo	1	5	Simple
Administrar Encuestas	1	15	Simple
Autentificar Usuario	1	1	Simple
Administrar Usuario	1	8	Simple
Administrar Enlaces	1	10	Simple
Administrar Paquetes de Idioma	1	16	Simple

Administrar Grupo de Usuarios	1	1	Simple
Administrar Descargas	1	3	Simple
Administrar Noticias	1	6	Simple
Administrar FAQs	1	2	Simple
Administrar Foros de Discusión	1	4	Simple

Tabla Peticiones.

Nombre de la petición	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Actualizar Contenido	1	1	Simple
Actualizar usuario	1	1	Simple
Listado de Artículos	1	1	Simple
Listado de Noticias	1	1	Simple
Autenticación	1	1	Simple

Tabla Ficheros Internos.

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
tb_Categoría	1	17	Simple
tb_Contenido	1	18	Simple
tb_Enlaces	1	11	Simple
tb_Group	1	2	Simple
tb_Menu	1	14	Simple
tb_Modulos	1	12	Simple
tb_Section	1	13	Simple
tb_Session	1	5	Simple
tb_Template	1	3	Simple
tb_TemplateMenu	1	2	Simple
tb_UserType	1	2	Simple
tb_Usuarios	1	12	Simple
tb_Modulos_Menu	1	2	Simple

Tabla Interfaces Externas.

Elementos	Simples		Medios		Complejos		Subtotal de puntos de función
	No.	X Peso	No.	X Peso	No.	X Peso	
Ficheros lógicos internos	13	7	0	10	0	15	91
Entradas externas	11	3	0	4	0	6	33
Salidas externas	0	4	0	5	0	7	0
Peticiones	5	3	0	4	0	6	15
Total							139

5.3. Costos

Tabla 1Cantidad de Instrucciones Fuentes.

Características	Valor
Puntos de función desajustados	139
Lenguaje	PHP (85%) JavaScripts (3 %) HTML (12 %)
Instrucciones fuentes por puntos de función	(33) (58) (47)
Instrucciones fuentes por lenguaje (miles de instrucciones)	(3,89895) (0,24186) (0,78396)
Instrucciones fuentes (miles de instrucciones)	4,92477

Se utilizo el ratio del lenguaje C++ pues el PHP no se encontró y este es un lenguaje muy parecido.

Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

SFi	Valor
PREC	1.24
FLEX	1.01
RESL	7.07

TEAM	1.10
PMAT	7.80
ΣSF_i	**18,22**

EMj	Valor	Justificación
RCPX	1.00	BD moderada, con exigencia de documentación básica y complejidad nominal moderada.
RUSE	1.00	A lo largo del proyecto se reutiliza el código.
PDIF	1.00	Plataforma estable con requerimientos bajos de memoria y tiempo.
PERS	0.50	Todos a un mismo nivel, sin movimiento de personal.
PREX	0.87	Se tiene experiencia en la plataforma y herramientas de desarrollo. Aproximadamente unos 2 años y medio.
FCIL	1.00	Se utilizan CASEs para la documentación referida a la Ingeniería de Software.
SCED	1.00	Se asume nominal por no conocer el tiempo de desarrollo.
ΠME_j	0.43	

Valores Calibrados:

A = 2,94; **B** = 0,91; **C** = 3,67; **D** = 0,24

5.3.1. Cálculo del esfuerzo (PM)

E = B + 0,01 * ΣSFi = 0,91 + 0,01 * 18,22 = **1,09.**

MSLOC = **23,9397.**

PM = $A \cdot (MSLOC)^{E} \cdot \Pi ME_j$ = 2,94 * $(4.92477)^{1,09} \cdot 0.43$ = 2.94 * 5.68 * 0.43 = 7.18 **hombres/mes.**

5.3.2. Tiempo de desarrollo (TDEV)

F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.09 – 0.91) = **0.276.**

TDEV = $C \cdot PM^{F}$ = $3.67 + 7.18^{0,276}$ = 3,67 + 1.72 = **5.39 meses**

CH = PM/TDEV = 7.18 / 5.39 = **1.33 hombres**

Ajustando para 1 hombre en el proyecto:

CH* = 1 **hombres.**

TEDV = PM/CH* = 7.18 /1 = 7.18 **meses.**

5.3.3. Costo del proyecto

Salario = $ 90.00

CHM = 1 * Salario = **$ 90,00.**

C = CHM * PM = $ 90.00 * 7.18 hombres/mes = **$ 646.2.**

Cálculo de:	Valor
Esfuerzo(PM: Hombres - mes)	7.18
Tiempo estimado de desarrollo (meses)	7.18
Cantidad estimada de hombres	1
Costo real (pesos)	$646.2
Salario medio real (pesos)	$90.00

5.4. Beneficios Tangibles e Intangibles.

El sistema de gestión de contenidos no es un producto con fines comerciales, aunque puede adjuntarse por sus características a cualquier sistema que lo necesite por sus características. Su principal objetivo es resolver uno de los problemas actuales en la UCI que es la gestión y administración de contenidos. Por tanto, los beneficios inmediatos son generalmente intangibles.

5.5. Análisis de costos y beneficios.

El desarrollo de este sistema no supone grandes gastos de recursos, ni tampoco de tiempo; la base de datos que contiene el contenido a gestionar en la aplicación, puede ser alojada en los Sistemas de Gestión existentes en la Universidad, ya que los mismos tienen buenas prestaciones y acceso rápido. La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios.

5.6. Conclusiones.

No caben dudas de que una óptima planificación del proyecto contribuye a un desarrollo organizado y como consecuencia a una mejor calidad del producto obtenido.

No solo se comprendió el ámbito del trabajo a realizar, los recursos que se necesitan y las tareas específicas a ejecutar, sino que además se tuvo en cuenta el costo y la factibilidad ya que esto puede influir grandemente en la aceptación y desarrollo del sistema deseado.

CONCLUSIONES.

La informatización de la sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.

Actualmente La Universidad de las Ciencias Informáticas, se encuentra en un proceso de continua mejora de los servicios que se prestan en su intranet o en la realización de sistemas para la comercialización. Identificar la situación problémica de estos servicios, brinda la posibilidad de trabajar más concretamente en las dificultades que aún existen.

El sistema que se ha diseñado posibilitará la interacción con los usuarios y facilitará la gestión y administración de contenido. Proporcionará un cúmulo grande de información la cual será utilizada por los usuarios de la Universidad así como toda la comunidad de Open Source del país.

Este sistema de administración de contenidos permitirá además una estrecha vigilancia acerca de la seguridad informática con respecto al uso de los servicios informáticos de la red que se brindan en la UCI.

Atendiendo a la Hipótesis planteada al inicio de este trabajo puede decirse que se ha cumplido y aprobado la misma pues el modelado de este sistema se llevo hasta la fase final cumpliendo todas las etapas de modelado del software que plantea la metodología de trabajo en grupo RUP.

Esta herramienta ha de formar parte de un paquete de herramientas Open Source que se desarrollan en la Universidad. Para impulsar el uso de Sistemas Operativos y Software Libre.

RECOMENDACIONES.

El sistema que se ha diseñado es un eslabón de la cadena de transformaciones en que se encuentra inmersa la Universidad de las Ciencias Informáticas. Muchas son las expectativas de mejoras en los diferentes servicios de red que se brindan. El estudio de la situación problémica arrojo otras dificultades que por diferentes aspectos no fueron posibles culminar, y las que se reflejan a continuación como parte de las recomendaciones de este trabajo.

1. Continuar con el diseño de lo módulos funcionales.
2. Estudiar la posibilidad de añadir a la herramienta un módulo I-Learning.
3. Continuar trabajando para hacer cada día mejor esta herramienta.
4. Lograr crear una comunidad de usuarios que ayuden en el desarrollo de otras funcionalidades.
5. Se recomienda a los usuarios de la red de la UCI aprovechar las potencialidades de esta herramienta ya sea para elevar su cultura informática, así como el de fieles guardianes de la utilización de cada uno de los servicios que se presten.

BIBLIOGRAFÍA.

[1] Official XOOPS Website http://www.xoops.org/ (abril, 2005).

[2] Linux Online. http://www.linux.org/ (abril, 2005).

[3] Linux Standard Base Project. http://www.linuxbase.org/ (abril, 2005).

[4] Álvarez, Miguel Angel. Introducción a los lenguajes del Web. http://www.desarrolloweb.com/ (abril, 2005).

[5] Tutorial de Perl en español. www.merelo.net/tutoperl/ (abril, 2005).

[6] Aguilar, Vicente y Suau, Pablo. MySQL vs. PostgreSQL (18 de Agosto de 2000).

[7] MySQL, The world's most popular open source database. www.mysql.com (abril, 2005).

[8] OpenSourceCMS (Ingl,s). http://www.opensourcecms.com/, (diciembre 2004).

[9] Portal de Proyectos Open Source. http://sourceforge.net, (Enero 2005).

[10] Comparaciones CMS (Ingl,s). http://www.cmsmatrix.org/matrix, (diciembre 2004).

[11] Portal de Mambo para Hispanohablantes. www.mambohispano.org, (Enero, 2005).

[12] Portal de Mambo (Ingl,s). http://www.mambosolutions.com/, (abril, 2005).

[13] cmsInfo. http://www.cmsinfo.org (abril, 2005).

[14] OpensourceCMS, http://www.opensourcecms.com (abril, 2005).

[15] Open source content management, The international association for Open Source Content
Management. http://www.oscom.org (abril, 2005).

[16] Robertson, J., How to evaluate a content management system. Step Two, 23 enero 2002
http://www.steptwo.com.au/papers/kmc_evaluate/index.html (abril, 2005).

[17] Drupal.org. http://drupal.org/ (abril, 2005).

[18] JACOBSON, Ivar; BOOCH, Grady, RUMBAUGH, James, "El Proceso Unificado de
Desarrollo de Software".2000. Addison Wesley.

[19] OpenSourceCMS (Ingl,s). http://www.opensourcecms.com/, diciembre 2004.

[20] Portal de Proyectos Open Source. http://sourceforge.net, Enero 2005.

[21] M s All de Linux from Scratch. http://es.tldp.org/Manuales-LuCAS/blfs-es/blfs-es-5.0/index.html, mayo 2005.

[22] http://www.uoc.edu/mosaic/articulos/cms1204.html, mayo 2005.

[23] http://www.drupal.org

[24] Robertson, J., How to evaluate a content management system [en línea]. Step Two, 23 enero 2002 <http://www.steptwo.com.au/papers/kmc_evaluate/index.html>

[25] ¿Robertson, J., So, what is a content management system? [En línea]. Step Two, 3 junio 2003 <http://www.steptwo.com.au/papers/kmc_what/index.html>

[26] Robertson, J., Looking towards the future of CM [en línea]. Step Two, 14 enero 2003 <http://www.steptwo.com.au/papers/cmb_future/index.html>

[27] Rhyno, A. The Ten Commandments of Content Management [en línea] usr/lib/info, 18 Feb 2003 <http://usrlib.info/story/2003/2/17/82354/8716

[28] Papers "Case Studies [en línea] Step Two Designs <http://www.steptwo.com.au/papers/index.php>

[29] Suh, P., [et al.], Content Management Systems.

CmsInfo [en línea], cmsInfo.org, <http://www.cmsinfo.org>

[30] OpensourceCMS [en línea], opensourceCMS, <http://www.opensourcecms.com>

[31] Open source content management [en línea], The international association for Open Source Content Management, <http://www.oscom.org>

[32] portalZine [en línea], portalZine, <http://www.portalzine.de>
[33] http://www.oscom.org/matrix/index.html

UCI Universidad de las Ciencias informáticas

GLOSARIO DE TERMINOS.

Apache: Servidor de páginas Web de código abierto para diferentes plataformas (UNIX, WINDOWS, etc.)

Blog (*Web log*): Diario en formato Web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

CMF (*Content Management Framework*): Entorno a programación de aplicaciones enfocado al desarrollo de CMS.

CMS (Content Management System): Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como *Web Content Management* (WCM) sistema de gestión de contenido Web.

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

CSS (*Cascading Style Sheets*): las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llama "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en un Sitio Web.

HTML (*HyperText Markup Laguage*): Lenguaje basado en marcas que indican las características del texto utilizado para definir documentos de hipertexto en Webs.

HTTP (*HyperText Transfer Protocol*): Protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML).

LAMP (*Linux, Apache, MySQL y PHP, Perl o Python*): Arquitectura formada por el sistema operativo Linux, el servidor Apache, el SGBD MySQL y uno o más de los lenguajes de programación PHP, Perl o Python.

LCMS (*Learning Content Management System*): Software para la gestión automatizada de cursos en línea, que incluye, gestión de usuarios, de resultados y de recursos. Es un sistema de gestión de cursos con las capacidades de un CMS y por tanto gestionar también los contenidos de los recursos.

LGPL (*Lesser General Public License*): Licencia que permite el uso y modificación de librerías de código para desarrollar software libre o propietario. Antes conocida como *Library GPL*.

LMS (*Learning Management System*): Se diferencia de los LCMS en que no hay gestión de los contenidos, sino simplemente administración del curso, pero acostumbra a utilizarse como sinónimo. También conocido como Course Management System (CMS) o Virtual Learning Enviroment.

Metadatos: Datos sobre los datos. Información que describe el contenido de los datos. Por ejemplo de un documento serian metadatos, entre otros, su título, el nombre del autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.

MySQL: Base de Datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

Open Source: Código abierto 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.

Perl: Lenguaje de programación de alto nivel que hereda de diversos lenguajes, muy utilizado para el desarrollo de Webs dinámicas.

PHP (*PHP Hypertext Preprocessor*): Lenguaje de programación para el desarrollo de Webs dinámicas, con sintaxis parecida a la de C. Originalmente se conocía como Personal Hombre Page Tools, herramienta para páginas personales (en Internet).

Python: Lenguaje interpretado de alto nivel orientado a objetos.

URL (Uniform Resource Locator): Dirección de recurso en la Web. Tiene el formato protocol: //máquina.dominio: port/ruta/recurso, por ejemplo http://intranet.uci.cu donde no se indica el puerto pues el protocolo http tiene uno por defecto (80).

WAI (*Web Accessibility Initiative*): Iniciativa del Consorcio de la World Wide Web para asegurar que las Webs están diseñadas pensando en el acceso de personas con discapacidades.

Web: Sistema para presentar información en Internet basado en hipertexto. Cuándo se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

WebDAV (Web-based Distributed Authoring and Versioning): es una extensión del protocolo HTTP que permite a los usuarios editar y administrar ficheros de forma colaborativa en servidores Web remotos.

Web log: ver Blog.

WYSIWYG: Traducido- lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor *WYSIWYG*, implica trabajar con los códigos que indican el formato que tendrá el texto sin ver el

Universidad de las Ciencias Informáticas

Anexo 1.

Comparación entre los CMS más usados.

	Drupal	EZ Publish	Geeklog	Mambo	Midgard	Moodle	OpenCMS	PHPNuke	Plone	Postnuke	Slash	TWiki	Typo3	WebGUI	Xoops
Versión	4.3.2	3.3	1.3.9	4.5		1.1.1	5.0.1	7.2	2.0	0.726	2.2.6	1.8.1	3.6.0	6.0	2.0.6
Requerimientos															
Servidor Web	Apache		Apache / IIS	Apache	Apache		Tomcat Servlet	Apache / IIS	Apache / IIS / Zope	Apache / IIS	Apache	Apache / IIS	Apache / IIS	Apache / IIS	Apache / Otros
Base de datos	MySQL		MySQL / otros	MySQL	MySQL	MySQL / postgresql	MySQL / otros	MySQL / otros	Zope	MySQL	MySQL	MySQL / otros	MySQL	MySQL	MySQL
Lenguaje	PHP		PHP	PHP	PHP	Perl	Java JSP / PHPML	PHP	Python	PHP	Perl	PHP	PHP	Perl	PHP
SO	Unix	Unix	Unix	Unix / Win	Unix		Unix / Win	Unix / Win	Unix / Win	Unix / Win	Unix	Unix / Win	Unix / Win	Unix / Win	
Soporte															
Ayuda contextual							SI	No	No	SI		Limitado	SI	SI	
CMF / API		SI			SI		SI	SI	SI	SI		No	SI	SI	
Foro y/o lista de correo							SI	SI	SI	SI		SI	SI	SI	
Trabajo en grupo															
Aprobación contenido				SI	SI		SI	No	SI	SI		Limitado	SI	No	
Control sesión							No	No	No	No		No	SI	SI	
Permisos por recurso							SI	Limitado	SI	SI		SI	SI	SI	
Versiones				SI	SI		SI	No	SI	No		SI	Limitado	SI	
Ciclo de trabajo (Workflow)	SI			SI			SI	No	SI	No		SI	Limitado	No	
Seguimiento proyectos							No	No	SI	SI		No	No	SI	
Usuarios/autores															
Deshacer									SI	No		Limitado	No	SI	
Editor WYSIWYG				SI	SI		IE	No	SI	SI		SI	SI	SI	
Ficheros upload/load							No	SI	SI	SI		SI	SI	SI	SI
Página personalizada							No	No	Limitado	SI		SI	SI	No	
Características															
Accesibilidad AA									W3C AA	No		Limitado	SI	SI	
Area de test							SI	No	SI	No		SI	SI	SI	
Auditoría							SI	No	SI	No		Limitado	SI	SI	
Backup base datos				SI								SI			
Cache	SI						SI					SI			
Búsqueda		SI	SI	SI	SI		SI	SI				SI			
Contenido programado				SI	SI		SI	No	SI	SI		SI	SI	SI	
Correo (email) a foro							No	No	SI	No		SI	SI	No	
Estadísticas				SI			No	SI	No	SI		SI	SI	Limitado	
Gestión centralizada de ficheros							SI	No	SI	SI		SI	SI	SI	
Gestión publicidad				SI			No	SI	No	SI		SI	SI	No	
Gestión patrones							SI	Limitado	SI	No		Limitado	SI	SI	
Grupos de usuarios					SI										SI
Informes de bases de datos							No	No	No	SI		No	SI	SI	
Internacionalización			SI			SI	SI	SI	SI	SI		SI	SI	SI	
Español			SI			SI							SI		
Links externos							No	No	No	SI		No	SI	No	
Lenguaje de macros							No	No	Limitado	No		Limitado	SI	SI	
Lenguaje de patrones							No	No	SI	No		SI	SI	SI	
Metadatos	SI			SI		SI									
Módulos externos	SI		SI	SI			SI	SI	SI	SI					SI
Niveles de interfície según usuario							No	No	No	No		SI	SI	SI	
Sindicación	RSS		RSS	RSS	RSS			RSS	RSS	RSS		RSS	RSS	RSS	

UCI Universidad de las Ciencias Informáticas

	Mambo	Plone	Typo3
Arquitectura			
API O / CMF	Sí	Sí	Sí
Módulos externos	Sí	Sí	Sí
Separación contenido - presentación	Sí	Sí	Sí
Grado desarrollo	Alto	Alto	Alto
Módulos de terceros disponibles	Muchos. Fáciles de instalar	Sí, no están disponibles en un área de descarga.	Muchos
Aspecto profesional	Sí	Sí	Sí
Soporte	Alto	Alto	Alto
Documentación	Suficiente	Abundante	Abundante
Comunidad soporte	Sí. Grande	Sí	Sí
Posición en el mercado y opinión usuarios	Buena	Buena	Buena
Usabilidad	Muy buena	Muy buena	Muy buena
Accesibilidad	-	WAI	WAI
Funcionalidades			
Editor WIS WYG	Sí	Sí con Epoz, y cambiando las preferencias. Incluye edición con formularios, HTML y texto plano.	Sí. También se puede editar en HTML.
Inserción de imágenes	Seleccionando de las carpetas de imágenes. Fácil	Se tiene que escribir la ruta donde se encuentra la imagen	Permite cargar imágenes de cualquier directorio local. Fácil
Herramienta de busca	Sí	Sí	Sí
Foros	Sí	Sí	Sí
correo electrónico	Entre miembros del grupo.	Módulo groupware.	Entre miembros del grupo
Chat	-	Sí	Sí
Noticias	Sí	Sí	
Artículos	Sí	Sí	
Comentarios de los usuarios	-	Sí. Desactivable.	
Workflow	Permite envío de notas	Sí. Se pueden crear estados de un objeto y roles de las personas.	Permite envío de notas a miembros de un grupo
Fechas de publicación y caducidad	Sí	Sí	
Webs personales	-	Sí. Hay herramientas para administrar la propia web personal. Se puede compartir una página para su edición. Todos los miembros se pueden encontrar con una herramienta de busqueda. Quizás demasiado complicado.	-
Avisos actualización por correo electrónico	-	-	-
Envío páginas por correo electrónico	Sí	Sí	-
Páginas en versión imprimible	Sí	Sí	-
Páginas en versión pdf	Sí	-	-
Personalización según usuario	-	Sí. Limitado a algunas opciones. Según el manual se puede cambiar el tema, pero la opción no aparece en la página.	Sí
Internacionalización	Sí	Sí. Interfaz en catalán y castellano	Sí
Ficheros en diferentes formatos (Word, PDF, etc.)	Sí		Sí
Navegadores soportados	IE, Netscape, Mozilla	IE, Netscape, Mozilla, Opera, Konkeror, Safari y otros	IE, Netscape, Mozilla
Soporte sindicación	Sí	Sí	Sí
Estadísticas	Sí	-	Sí
Temas para personalizar presentación	Sí	Sí. Editor CPSSkins.	Sí
Otros			
Similitud webs	No. Buenas capacidades gráficas	Alta. Quizás por dificultades en la edición.	No. Buenas capacidades gráficas
Direcciones legibles	Sí	Sí	-
Deshacer	-	Sí	Sí
Herramienta de administración	Gráfica y potente	Básica. Hay que utilizar la herramienta del CMF para muchas tareas	Gráfica y potente

UCI Universidad de las Ciencias Informáticas

Anexo2.

Listado de CMS y licencias. [34]

Content Management Systems

Projects	License
Aegir CMS	LGPL
Apache Lenya	Apache Software License
Ariadne	GNU GPL
Back-End	GPL
Bitflux	GNU General Public License (GPL)
Bricolage	BSD License
Callisto	GPL (General Public License)
Campsite	GNU General Public License
CocoBlog	Apache Software License, 1.1
Cofax	Cofax Software License (based on apache)
DBPrism CMS	Apache Software License
Drupal	GNU General Public License
elevateIT	GPL
eNvolution	GPL
eZ publish	GPL
Jahia CMS and Portal Server	Jahia Collaborative Open Source License
Komplete Lite	GPL
Kontentor	GPL (GNU Public License)
Krang	BSD
LifeCMS	GPL
Magnolia	LGPL
Mambo	GNU/GPL License
MMBase	Mozilla Public License
MySource	Apache License
Nukes	LGPL
Nuxeo CPS	GPL

OmegaCMS	BSD-style License
OpenCms	GNU Lesser General Public License (LGPL)
phpCMS	GNU General Public License (GPL)
PHP-Nuke	GNU General Public License (GPL)
phpSlash	GNU General Public License (GPL)
Plone	BSD License
Red Hat CCM	CCM resp. IBM Public License
Silva	Apache like Software License
Tiki	GNU LGPL License
Typo3	GPL
webEditor	???
WebGUI	GNU GPL
XIMS	GNU GPL
XOOPS	GNU GPL
Zeit CMS	GPL

UCI Universidad de las Ciencias Informáticas

Anexo3.

ANEXO. EVOLUCIÓN Y ORÍGENES DEL PROCESO DE DESARROLLO DEL SOFTWARE.