



Laboratorio para la Producción  
de Recursos Didácticos



Animación · Edición · Video · Audio · Multimedia · Textos · Presentaciones

# **Sistema de Gestión para el Laboratorio de Producción de Recursos Didácticos de la UCI**

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

**Autores:** Alberto Pantoja Ponce

Leydis Morales Morales

**Tutores:** Ing. Alien Gutiérrez Ledesma

Ing. Addiel Rodríguez Guadarrama

**Cotutor:** Lic. Deymis Tamayo Rueda



## Declaración de autoría

---

### Declaración de autoría

Por este medio declaramos ser los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del autor**

Alberto Pantoja Ponce

---

**Firma del autor**

Leydis Morales Morales

---

**Firma del tutor**

Ing. Alien Gutiérrez Ledesma

---

**Firma del cotutor**

Ing. Addiel Rodríguez Guadarrama

---

**Firma del cotutor**

Lic. Deymis Tamayo Rueda



## Opinión del usuario del trabajo de diploma

El Trabajo de Diploma, titulado **Sistema de Gestión para el Laboratorio de Producción de Recursos Didácticos de la UCI**, fue realizado en la **Facultad 8 de la Universidad de las Ciencias Informáticas (UCI)**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un \_\_\_\_ %

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

- El centro contará con una herramienta que facilitará la rapidez y mejorará la calidad de los procesos internos que se llevan a cabo durante la producción de los Recursos Didácticos en el LAREDI.
- Tendrá una aplicación capaz de gestionar las solicitudes de producción, incluso las que se realicen simultáneamente, sin importar donde se encuentre el solicitante; capaz de almacenar, buscar y recuperar la documentación en un corto período de tiempo, y que permitirá generar de una manera rápida y sencilla, análisis estadísticos para facilitarle al Coordinador Líder la toma de decisiones.
- El LAREDI, contará con un espacio para publicar sus propios cursos, ya sean impartidos por una persona del Laboratorio o fuera de él. Permitiendo la inscripción de los profesores a estos cursos.
- El cliente tendrá un espacio para emitir quejas y sugerencias con la posibilidad de que sean atendidas en el momento que las emita, al contar con un servicio de alerta y mensajería conjuntamente con el servicio de correo de la universidad, podrá realizar un seguimiento de la producción del recurso solicitado y de esta forma mejorará la comunicación con los desarrolladores.

Y para que así conste, se firma la presente a los 31 días del mes de mayo del año 2010.

\_\_\_\_\_  
Representante de la entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño



◆—————▶

**“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica”.**

**Aristóteles**



## Agradecimientos

---

### **Leydis Morales:**

A la Revolución y a Fidel por darme la oportunidad de estudiar en esta Universidad del futuro.

A mi mamá y mi abuela materna por ser mi ejemplo de sacrificio, por haberme guiado por el buen camino del estudio, la dedicación, la responsabilidad, por haberme enseñado a vencer los obstáculos que se presentan en la vida y por haber confiado en mí.

A mi hermanito por ser parte de mi inspiración, al que siempre quise legarle un buen ejemplo y decirle algún día que siguiera mi ejemplo, pero se me adelantó.

A mis tíos y demás familiares por estar pendientes de mí en todos estos años, especialmente a mi tío Pachi y Eslén por ser la familia más cercana que tenía, que me dio su apoyo y con la que siempre he podido contar.

A mi novio y compañero de tesis, por su comprensión, por su cariño, su apoyo en todo momento y por su paciencia.

A mi gran amigo Homero, por estar siempre ahí en las buenas y en las malas, gracias por ser mi amigo.

A mis tutores y cotutora Deymis, por brindarnos su apoyo incondicional y estar ahí cada día.

A mis compañeros de aula, a mis amigos y a todos los que hicieron posible la realización de este trabajo.



## Agradecimientos

---

**Alberto:**

No creo que existan palabras para escribir algo tan sensible en la persona como son los agradecimientos, sensible porque siempre hacen brotar lágrimas. Tendría que comenzar a darle las gracias a los que dieron su brazo para sostenerme cuando aprendía a caminar, a los que me enseñaron a leer y a escribir, a los que me guiaron por los buenos caminos de la vida, a los que permitieron mi crecimiento como persona y como hombre.

Le agradezco a la que me brindó su brazo, a la que me sirvió de faro, a la que me enseñó a ser humilde, honrado, hombre de bien, a no cejar en el empeño, a sacrificarme hasta el último aliento, a ser responsable, dedicado y no alcanzaría el momento para decirlo todo, TE AGRADEZCO A TI MI MAMI, pues eres esa persona, tan especial, tan inspiradora, GRACIAS POR CONFIAR EN MI.

Agradezco a mi abuela materna, que ha sido otra madre para mí, tan trabajadora. A mi hermanita querida, que siempre me hace sonreír, espero verte algún día así. A mi papá por lo mucho o lo poco. A toda mi familia que me apoyó para que este sueño se hiciera realidad.

A mi novia y compañera de Tesis, los agradecimientos son dobles. A Omaidá, Pachi.

A mi amigo y hermano Homero.

Agradezco al padre de los drupaleros de la Facultad 8, Héctor Pérez Hernández, por brindar sus conocimientos incondicionalmente, por ser quien es.

A mis compañeros de aula y amigos.

A los tutores y a la cotutora Deymis por su paciencia y atención.

A la Revolución y a Fidel. A la UCI, y a mis profesores.



## Dedicatoria

---

**Leydis Morales**

Al compañero Fidel, motor impulsor de todas las obras que han hecho posible mi formación.

A tres personas que de conjunto han hecho de mí una mujer de bien, mis abuelos Raúl y Belarmina, y a mi mamá querida.

A mis familiares por su preocupación y su apoyo incondicional.

A mi hermano querido que alegra cada momento de mi vida.

A mi novio Alberto, por aceptarme como soy.

**Alberto Pantoja**

A mi mamá, que no hace falta describirte, tan solo hay que mirarte.

A mi hermanita, tan risueña.

A mi abuela, la más trabajadora.

A mi novia, cuanto hay que agradecerte.

A Pachi y Omaida, por ser parte de mi otra familia.

A Fidel y la Revolución.

A mi facultad, aquí se gradúa un octavito.



## Resumen

---

El presente trabajo que lleva por título “Sistema de Gestión para el Laboratorio de Producción de Recursos Didácticos (LAREDI) de la UCI”, surgió por la necesidad que tenía el LAREDI de obtener mejor control, rapidez y eficiencia de los procesos de producción. Para darle solución a esta problemática se desarrolló una aplicación web que permite la gestión de las solicitudes de una manera más rápida y simultánea, documentaciones, entre otros elementos de producción que se realizan en el laboratorio, ubicado en el Centro de Producción de la Facultad 8 de la Universidad de las Ciencias Informáticas.

Para ello se realizó un estudio sobre: conceptos asociados a los CMS, metodologías de desarrollo de software, herramientas de modelado, lenguajes de programación, sistemas gestores de base de datos, además de otros aspectos asociados a la Ingeniería de Requisitos (IR) y al flujo de trabajo Análisis y Diseño definido en el Proceso Unificado de Desarrollo (RUP). Finalmente se realizaron pruebas a la aplicación web dirigidas a evaluar la calidad de los módulos generados, obteniéndose resultados satisfactorios.





<b>Introducción.....</b>	<b>1</b>
<b>Capítulo I.....</b>	<b>7</b>
<b>1.1 Aplicación de Escritorio vs Aplicación Web. ....</b>	<b>8</b>
1.1.1 Aplicación de Escritorio.....	8
1.1.2 Aplicación Web. ....	9
<b>1.2 Sistemas de Gestión de Contenidos. ....</b>	<b>10</b>
1.2.1 ¿Qué es un CMS? .....	11
1.2.2 Características principales de los CMS.....	12
1.2.3 Infraestructura tecnológica de un CMS .....	12
1.2.4 Criterios de selección .....	15
1.2.5 Drupal.....	17
<b>1.3 Metodologías de desarrollo de Software. ....</b>	<b>21</b>
1.3.1 Extreme Programming (XP).....	21
1.3.2 Proceso Unificado de Desarrollo (RUP).....	22
<b>1.4 Lenguajes de modelación. ....</b>	<b>25</b>
1.4.1 Notación para el Modelado de Procesos de Negocio (BPMN).....	25
1.4.2 Lenguaje de Modelamiento Unificado (UML).....	26
<b>1.5 Lenguajes de programación. ....</b>	<b>27</b>
1.5.1 HTML.....	28
1.5.2 Javascript.....	28
1.5.3 PHP .....	29
<b>1.6 Herramientas para desarrollar un Software. ....</b>	<b>31</b>
1.6.1 Visual Paradigm.....	31
1.6.2 Servidor Web .....	33
1.6.2.1 Apache .....	34
1.6.2.2 Wamp.....	34
1.6.3 Servidor de Bases de Datos MySQL .....	35
1.6.4 Zend Studio .....	36
<b>1.7 Propuesta. ....</b>	<b>37</b>
<b>Capítulo II.....</b>	<b>38</b>
<b>Introducción.....</b>	<b>38</b>
<b>2.1 Objeto de Estudio. ....</b>	<b>39</b>
<b>2.2 Objeto de Automatización. ....</b>	<b>40</b>
<b>2.3 Descripción de los procesos del negocio. ....</b>	<b>40</b>
<b>2.4 Actores del Negocio. ....</b>	<b>41</b>



## Índice

2.5 Trabajadores del Negocio. _____	42
2.6 Casos de Uso del Negocio. _____	43
2.7 Diagrama de Casos de Uso del Negocio. _____	43
2.8 Diagramas de Actividades del Negocio (Ver Anexo 1). _____	43
2.9 Diagrama de Clases del Modelo de objetos del negocio (Ver Anexo 1). _____	44
2.10 Descripción textual del Caso de Uso del Negocio (Ver Anexo 2). _____	44
2.11 Especificación de los requisitos de Software _____	44
2.11.1 Requisitos Funcionales.....	44
2.11.2 Requisitos No Funcionales.....	47
2.12 Descripción textual de los Casos de Uso del Sistema. _____	49
2.12.1 Actores del Sistema.....	49
2.12.2 Listado de los Casos de Uso del Sistema.....	50
2.13 Diagrama de Casos de Uso del Sistema. _____	55
Conclusiones _____	55
<b>Capítulo III.....</b>	<b>56</b>
Introducción _____	56
3.1 Modelo de Análisis. _____	57
3.1.1 Diagrama de clases del análisis. ....	57
3.2 Diseño. _____	57
3.2.1 Diagrama de Clases del Modelo de Diseño.....	58
3.2.2 Módulos.....	59
3.2.2.1 Módulos de Drupal.....	60
3.2.2.2 Módulos de Drupal modificados.....	63
3.2.2.3 Módulos desarrollados.....	63
3.2.3 Diagrama de clases Persistentes.....	66
3.2.4 Diseño de la Base de Datos.....	66
3.2.5 Descripción de las tablas.....	66
Conclusiones _____	66
<b>Capítulo IV.....</b>	<b>67</b>
Introducción _____	67
4.1 Diagrama de Despliegue _____	68
4.2 Diagrama de componentes _____	68
4.3 Arquitectura y Patrones _____	70



## Índice

4.4 Modelo de Prueba	71
Conclusiones	72
<b>Conclusiones Generales</b>	<b>73</b>
<b>Recomendaciones</b>	<b>74</b>
<b>Referencias Bibliográficas</b>	<b>75</b>
<b>Anexos</b>	<b>77</b>
Anexo 1: Diagramas del Negocio	79
Anexo 2: Caso de Uso: "Gestionar RD"	81
Anexo 3: Casos de Uso del Sistema.	84
Anexo 4: Casos de Uso expandidos.	88
Anexo 5: Diagramas de Clases de Análisis.	119
Anexo 6: Diagrama de Clases Persistentes.	126
Anexo 7: Diagrama Entidad-Relación.	127
Anexo 8: Descripción de las tablas de la base de datos.	128
Anexo 9: Diagrama de Despliegue.	133
Anexo 10: Diagrama de Componentes.	134
Anexo 11: Modelo de Prueba.	135



## Introducción

---

### Introducción

Con el desarrollo vertiginoso en las últimas décadas de la Informática, las Tecnologías de la Información y la Comunicación (TIC) han sido utilizadas de manera intrínseca, en todas las esferas de la sociedad actual, incluyendo en las Universidades, para alcanzar un eslabón superior en cuanto al proceso de Enseñanza y Aprendizaje.

Constituye una premisa del Ministerio de Educación Superior de la República de Cuba (MES), llevar a todos sus centros universitarios, los nuevos adelantos de las técnicas más actualizadas de la información y la comunicación. Las mismas están al servicio de su claustro y estudiantes apoyándose para dicho proceso, en un elemento importante dentro de las TIC: los Recursos Didácticos<sup>1</sup> (RD), entre los que se encuentran multimedia, videos, animaciones y otros. Dichos RD se producen en empresas, entidades y en algunas universidades cubanas.

Una de estas universidades, es la Universidad de las Ciencias Informáticas (UCI), creada con el objetivo de informatizar el país y desarrollar la Industria del Software para contribuir al desarrollo económico del mismo. Para lograr la producción de los RD, se decide aprovechar la estructura organizativa de la UCI, donde existía una Dirección de Teleformación (DT), con la misión de “implementar y promover el uso intensivo de las TIC en el modelo de formación de pregrado y postgrado de la UCI en el que se combinan los elementos pedagógicos y tecnológicos más avanzados”<sup>2</sup> que ahora son parte de los objetivos del Centro de Producción de la Facultad 8 (FORTES) de dicha universidad.

La producción de RD en la universidad, anteriormente se realizó de manera independiente y sin ningún control sobre lo que se hacía, lo que trajo como consecuencia repetición de recursos sobre una misma materia y contenido. Además no se lograba la calidad requerida, ni un estándar;

---

<sup>1</sup> Conjunto de elementos que facilitan la realización del proceso de enseñanza y aprendizaje, los cuales contribuyen a que los estudiantes logren el dominio de un conocimiento determinado, al proporcionarles experiencias sensoriales representativas de dicho conocimiento.

<sup>2</sup> Entrevista con la Dirección de Teleformación



## Introducción

---

por la alta complejidad que puede llegar a alcanzar la producción de un RD. Para asumir la UCI el rol de productor, surgió la idea de crear un laboratorio para la producción de RD.

Basada en las experiencias de universidades españolas, la Universidad Agraria de la Habana (ISCA), con sus recursos Polimedia<sup>3</sup>, el CREA en la CUJAE y además con la idea propia de los directivos de la Dirección de Teleformación; se decidió crear el LAREDI. Este laboratorio tiene como objetivo producir RD, para mejorar el proceso de enseñanza de la Universidad de las Ciencias Informáticas, aumentar la cantidad de RD por asignaturas y como premisa apoyar al profesorado universitario en la elaboración de sus materiales para la Web.

La producción de recursos ocurre normalmente de la siguiente manera: un profesor hace personalmente una solicitud de un Recurso Didáctico al Coordinador Líder del Proyecto, utilizando el modelo de solicitud establecido, este a su vez la procesa y analiza la misma. Luego el Coordinador Líder del Proyecto envía el modelo a los Desarrolladores para seleccionar la fecha de la entrevista y este le notifica al profesor lo acordado vía correo. Se procede a la entrevista en fecha acordada y se toman acuerdos quedando plasmados en un acta, se hace el levantamiento de requisitos del producto final. Posteriormente, se hacen los prototipos necesarios y se le presentan al cliente, si este está de acuerdo se pasa a la etapa de producción de lo contrario, se diseñan nuevos prototipos. Terminada la producción se presenta al cliente el recurso y este verifica el cumplimiento de los requerimientos pactados, si no cumple con los mismos se corrigen las no conformidades. El proceso culmina con la firma de un acta de conformidad del producto desarrollado.

Actualmente, el flujo de trabajo no es el más óptimo en cuanto a control, rapidez y eficiencia de los procesos de producción. Dentro de estos, se llevan a cabo solicitudes para la creación de un Recurso Didáctico (RD). Estas se realizan de forma personal, siendo una

---

<sup>3</sup> Es un sistema diseñado para la creación de materiales educativos como apoyo a la docencia presencial, que abarca desde la preparación del material docente hasta la distribución a través de distintos medios a los destinatarios.



## Introducción

---

dificultad para el solicitante. En ocasiones, le resulta incómodo el desplazamiento hacia el Laboratorio, por factores ajenos a su voluntad, la distancia que lo separa de dicho Laboratorio, en situaciones adversas del tiempo, entre otras. En caso de llegar un profesor a realizar una solicitud y existir otro anteriormente, debe esperar a que este termine para ser atendido por el responsable de la recepción de las peticiones de producción. Esto trae como consecuencia que no se realice la solicitud en el tiempo que tenía previsto y con ella el tiempo para obtener el RD.

Luego se pasa a llenar las documentaciones de entrada y salida necesarias para llevar a cabo la producción. Las mismas se realizan manualmente haciendo más complejo el manejo de la información para el responsable de la recepción de la misma; provocando lentitud en los procesos y siendo propensa a errores, a pérdidas de la documentación, omisión de datos importantes, entre otras.

El cliente no cuenta con un espacio para emitir su opinión y sugerencias sobre el producto en cuestión, y en caso de querer hacerlo, debe esperar a que haya concluido la producción para opinar o sugerir de forma personal.

No se posee un sistema de aviso, que sirva de manera rápida y efectiva para notificarle al cliente en qué estado se encuentra el recurso solicitado y el avance de las tareas. Esto trae consigo que la comunicación entre el cliente y el equipo de trabajo, sea lenta y engorrosa.

Las estadísticas de las solicitudes resultan difíciles de obtener, por la cantidad de informaciones y documentos que se llevan durante la creación de los recursos.

Todo eso atenta directamente con el objetivo del Laboratorio, producir con rapidez, calidad y así lograr la satisfacción total del cliente.

A partir de lo analizado, se plantea el siguiente **problema de investigación** surgió el siguiente: ¿Cómo mejorar la calidad, rapidez y eficiencia de los procesos de producción de los RD en el LAREDI de la UCI?



## Introducción

---

Se define como **objeto de estudio** los sistemas de gestión de información.

Se trazó como **objetivo general**:

- Realizar un sistema para la gestión de los procesos que se realizan en el LAREDI de la UCI.

Por lo tanto el **campo de acción** es el sistema de gestión para el LAREDI de la UCI.

Como **objetivos específicos**:

- Estudiar aspectos teóricos referentes a los sistemas de gestión y herramientas que permitan su desarrollo.
- Analizar y diseñar el sistema de gestión para el LAREDI de la UCI.
- Implementar el sistema de gestión para el LAREDI de la UCI.
- Realizar pruebas al sistema.

Como **posibles resultados** se tienen los siguientes:

Con el sistema, el centro contará con una herramienta que facilitará la rapidez y mejorará la calidad de los procesos internos que se llevan a cabo durante la creación de los RD en el LAREDI. Además, tendrá una aplicación capaz de gestionar las solicitudes de producción, incluso las que se realicen simultáneamente y sin importar donde se encuentre el solicitante; capaz de almacenar, buscar y recuperar la documentación en el menor tiempo posible.

Una aplicación que permitirá generar de una manera rápida y sencilla, análisis estadísticos (cantidad de solicitudes en cierto período de tiempo, etapas críticas de producción, etc.), para facilitarle al Coordinador Líder la toma de decisiones.



## Introducción

---

El LAREDI, podrá publicar sus propios cursos, ya sean impartidos por una persona del Laboratorio o fuera de él. Permitiendo la inscripción de los profesores a estos cursos.

El cliente tendrá un espacio para emitir quejas y sugerencias con la posibilidad de que sean atendidas en el propio momento que las emita, al contar con un servicio de alerta y mensajería conjuntamente con el servicio de correo de la universidad, podrá realizar un seguimiento de la producción del recurso solicitado y de esta forma mejorará la comunicación con los desarrolladores.

Para el diseño metodológico de la investigación se utilizaron los siguientes métodos:

### 1- Métodos Teóricos

- **Análisis Histórico Lógico:** Este método permite realizar un análisis de los elementos que se utilizan en la implementación de sistemas Web, dígame origen y evolución de los diferentes lenguajes de programación, Sistemas de Gestión de Contenidos (CMS). Además se utiliza para la investigación de la metodología a seguir y Sistemas Gestores de Base de Datos que permita determinar los más idóneos para desarrollar la aplicación.
- **Analítico Sintético:** Este método permite determinar los principales métodos y algoritmos que son usados a nivel mundial y que pueden servir en la implementación del sistema.
- **Modelación:** Se utiliza para modelar el sistema mediante diferentes diagramas que ayudan a comprender mejor lo que se está desarrollando, utilizando una herramienta de modelado.

### 2- Métodos empíricos

- **Entrevista:** Se utiliza para conocer los servicios que las personas desean que estén presentes en el sistema a realizar. Esta información es de vital importancia para el levantamiento de los requisitos.





## Introducción

---

La estructura del documento es la siguiente:

### **Capítulo 1:** Fundamentación teórica.

Se realiza una comparación entre los lenguajes de programación destinados al desarrollo de aplicaciones web para concluir en las razones por las que se seleccionó PHP en la implementación del sistema, al igual que la argumentación del por qué el uso de una aplicación web y no una de escritorio. Se abordan brevemente los motivos que conllevaron a la selección del Sistema Gestor de Contenidos Drupal.

### ➤ **Capítulo 2:** Características del sistema.

Se hace una propuesta de forma general de cómo debe funcionar el sistema y una descripción de los procesos del negocio necesarios para el diseño del sistema. Se determinan además los requerimientos funcionales y no funcionales del sistema, agrupándolos en casos de uso.

### ➤ **Capítulo 3:** Análisis y diseño.

En este capítulo se muestran los diagramas de clases correspondientes y el diseño de la base de datos.

### ➤ **Capítulo 4:** Implementación y prueba.

A través de este capítulo se describe el modelo de prueba y se incluye el diagrama de despliegue y de componentes.



# Capítulo I

## Fundamentación teórica

En este capítulo se realizará una comparación entre las distintas herramientas y lenguajes posibles a emplearse en el Análisis, Diseño y la Implementación del *Sistema de Gestión para el LAREDI de la UCI* que se propone en el presente trabajo de diploma. Se realiza una caracterización de los Sistemas de Gestión de Contenido, herramienta propuesta por el cliente para crear la aplicación; los entornos de desarrollo como Zend Studio para PHP y los sistemas gestores de base de datos como MySQL, especificando las ventajas y desventajas, así como sus características; concluyendo con una breve explicación del porqué de la selección realizada para el desarrollo definitivo de la aplicación.



## Capítulo 1. Fundamentación teórica

---

### 1.1 Aplicación de Escritorio vs Aplicación Web.

#### 1.1.1 Aplicación de Escritorio.

Por aplicación de escritorio se entiende toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, Linux ó Mac. El desarrollo sobre una plataforma, normalmente, implica que la aplicación "no" pueda ser ejecutada en otras [1].

Entre las ventajas de las aplicaciones de escritorios se encuentran las siguientes [1]:

- Mayor capacidad gráfica visual.
- Menor tiempo de respuesta (aplicación más rápida).
- Mayor personalización.

A pesar de esto las aplicaciones de escritorio poseen las siguientes desventajas [1]:

- Duplicidad de datos por la falta de unificación de los mismos.
- Diseminación de la información y lógica en muchas partes (cada computadora que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada es un caos.
- Dificultad para configurar cada una de las instalaciones dependiendo de las necesidades de cada usuario.



## Capítulo 1. Fundamentación teórica

---

### 1.1.2 Aplicación Web.

En la ingeniería de software se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, ASP, PHP) en la que se confía la ejecución al navegador [2].

Desde el punto de vista de la arquitectura de software se distinguen dos lados: el cliente, donde se encuentra el usuario final utilizando la aplicación por intermedio de un navegador (Internet Explorer, Mozilla Firefox, entre otros), es aquí donde el usuario interactúa con la aplicación localizada al otro lado o servidor en donde residen realmente los datos, reglas y lógica de la aplicación [2].

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Las ventajas que ofrece el desarrollo de aplicaciones web son notables, entre ellas pueden citarse [2]:

- **Instalación:** la principal ventaja de un servicio web es poder acceder a él y a los datos que contiene desde cualquier sitio. Si el usuario accede al sistema es que tiene instalado todo lo necesario para ejecutar la aplicación: el navegador accede desde cualquier equipo.
- **Compatibilidad multiplataforma:** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.



## Capítulo 1. Fundamentación teórica

- **Actualizaciones:** No hay que actualizar nada porque tampoco ha habido que instalar nada. Es decir la administración es nula: no tiene que instalarse, no tiene que configurarse, no se tiene que hacer nada, simplemente empezar a utilizarlo. El usuario tiene todas sus preferencias/configuraciones guardadas en el servidor web y no en su ordenador.
- **Consumo de recursos para terceros:** la mayor parte de consumo de ciclos de procesador, memoria, etcétera, ocurren en el servidor.
- **Múltiples usuarios concurrentes:** Las aplicaciones web pueden ser utilizadas por múltiples usuarios al mismo tiempo.

Entre las desventajas que se destacan en las aplicaciones web se pueden citar [2]:

- Dependen de una conexión a Internet permanente (generalmente) y una conexión promedio para una óptima navegación.
- La seguridad de los datos, depende de la seguridad de la aplicación Web y del servidor donde esté alojado.
- La estabilidad de la aplicación está sujeta al número de visitas en un mismo lapsus de tiempo (sobrecarga del servidor).

Dentro de las aplicaciones web más usadas se encuentran los:

### 1.2 Sistemas de Gestión de Contenidos.

Inicialmente existían herramientas profesionales capaces de publicar información en intranets, tales como Lotus Notes, o herramientas más complicadas de gestión del conocimiento empresarial. Posteriormente surgieron potentes herramientas de gestión de contenidos que son de acceso libre, gracias al Movimiento de Software Libre (Open Source). Una de estas potentes herramientas son los Sistemas de Gestión de Contenidos o Content Management System herramienta que se utiliza para facilitar principalmente la Gestión de Portales Web, ya sea en Internet o en una Intranet.



## Capítulo 1. Fundamentación teórica

---

Los sistemas de gestión de contenidos (CMS) son un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (*Web Content Management* o WCM) [3].

### 1.2.1 ¿Qué es un CMS?

Un CMS es una interfaz que es capaz de controlar una o varias bases de datos donde se encuentra el contenido del sitio. El sistema permite manejar el contenido y el diseño de manera independiente. Así, es posible manejar el contenido y darle en cualquier momento un diseño diferente al sitio sin tener que darle formato al contenido de nuevo, encargándose de los trabajos más pesados que ocupaban el tiempo de los administradores de las Páginas Web. [4]

Dentro de los CMS más conocidos se encuentran Joomla, Typo3, PHP-Fusion, eZ Publish, Drupal entre otros.

**Sistema:** Trata generalmente de un grupo de herramientas, apoyado habitualmente por una base de datos, y que consisten en una serie de programas en un servidor Web, y, opcionalmente, un conjunto de programas cliente que te permitan acceder con facilidad a esos programas en el servidor. [4]

**La gestión de contenidos:** Desde el punto de vista del usuario del sistema, se trata de gestionar, de forma accesible, uniforme y cómoda, un sitio web dinámico, con actualizaciones periódicas, y sobre el que pueden trabajar una o más personas, cada una de las cuales tiene una función determinada. Desde el punto de vista del cliente, se trata de un sitio web dinámico, con apariencia e interfaz uniforme, con un diseño centrado en el usuario, y que permite llevar a cabo fácilmente las tareas para las que ha sido diseñado. Por lo tanto, un CMS tiene dos funciones fundamentales: facilitar la creación y la presentación de contenidos. Con respecto a la primera, suministra una serie de herramientas que hace que la publicación de contenido sea tan fácil como rellenar un formulario y haya, además, una sola fuente para todos ellos; con respecto a la segunda, da facilidades a la publicación de contenidos en formatos múltiples a partir de una única



## Capítulo 1. Fundamentación teórica

---

fuente, y añade metadatos a los mismos, para facilitar la navegación en múltiples facetas. También habría que considerar otras dos fases: gestión de contenidos y mantenimiento de los mismos; aunque estas fases se pueden incluir en la anterior. [4]

Un sistema de gestión de contenidos sirve para que la gestión de un sitio web, por pequeño que sea, no se te vaya de las manos: permite tener una apariencia y navegación uniforme en todo el sitio, y actualizar y gestionar el contenido fácilmente. [4]

### 1.2.2 Características principales de los CMS

Como principales características de los CMS se pueden resaltar: [4]

- Independencia del contenido respecto a la presentación:
  - ✓ Facilita futuros cambios de diseño en la página web.
  - ✓ Posibilita publicar un mismo contenido a través de varios canales a la vez.
  - ✓ Organización de la información en tipos de contenido.
  - ✓ Tratamiento individualizado de cada tipo de contenido.
- Flujo de edición y publicación.
  - ✓ Creación y publicación de contenidos no publicados.
  - ✓ Publicaciones programadas.
  - ✓ Edición de contenidos supervisados por más de un usuario mediante flujos de trabajo.

### 1.2.3 Infraestructura tecnológica de un CMS

En un CMS se podrán ver dos partes bien diferentes, una de ellas es la parte hardware, es decir, la parte física necesaria para poder llegar a la parte lógica, en la que se ubica el software.

[4]



## Capítulo 1. Fundamentación teórica

En la parte hardware se verán las características y requisitos que debe tener un servidor para poder llegar a la parte lógica de la configuración del CMS. En la lógica se verá el software necesario para que el CMS funcione y en este software se incluye el propio software (valga la redundancia) del gestor de contenidos. Se llama hardware al conjunto de elementos, electrónicos en nuestro caso, que interconectados entre sí llegan a configurar un único elemento llamado servidor. [4]

Los CMS están cimentados en el modelo cliente servidor, el cual se divide en tres capas [4]:

- Interfaz de control sobre la base de datos (capa de almacenamiento).
- Procesador de aplicaciones (capa lógica) donde se ubicarán el colectivo de programas (Servidor).
- Interfaz de usuario, es la parte con la que el usuario podrá manejar las anteriores capas. (capa de presentación final)

La parte software se encuentra dividida a su vez en varias partes (Ver Anexos):

A continuación se definen las capas software teniendo en cuenta el organigrama anterior.

[4]

- El Sistema Operativo es la base desde donde los servicios ftp y telnet tomarán protagonismo junto con otros que no se mencionarán para no sobrecargar de contenido:
  - telnet: es utilizado para acceder de forma remota a un servidor o una máquina. Es un protocolo que da la posibilidad de acceder a la línea de comandos del servidor, para realizar cualquier tipo de administración del sistema. Telnet suele escuchar el puerto 23.
  - ftp: Protocolo usado para la transferencia de ficheros.





## Capítulo 1. Fundamentación teórica

- Servidor Web, es el software que se encarga de proporcionar el acceso Web al servidor, este se realizará a través del puerto 8080 (por defecto). Todas las peticiones se harán a través de este aplicativo, pues el software CMS necesita apoyarse en el servidor Web para justificar su existencia.
- Otro elemento que está posicionado al mismo nivel que el Servidor Web es el Servidor de Aplicaciones<sup>4</sup>, este software tiene como premisa la de ejecutar los programas que hayan sido situados en el servidor y que serán solicitados para su ejecución desde el CMS. Existe una gran lista de los servidores de aplicaciones, entre los más utilizados se encuentran: Tecnología de servidor de aplicaciones, Macromedia ColdFusion MX 7, Microsoft IIS 6 con .NET Framework, SUN ONE Active Server Pages, Macromedia JRun, Sun ONE Application Server, IBM WebSphere, BEA WebLogic, Servidor PHP.
- Base de datos, en el mundo revolucionado por las tecnologías el dato figura entre la unidad lógica más importante y que mejor lugar para guardar esos datos que un lugar creado para ello. En la base de datos se almacena todo lo que sea de vital importancia y que posteriormente sea susceptible de volver a ser utilizado. Los sistemas gestores de bases de datos más utilizadas son: Oracle, Informix, PostgreSQL y MySQL, esta última muy extendida en el mundo GNU.
- El software del propio gestor de contenidos que actúa sobre el resto de pasos intermedios mencionados hasta llegar a este, y se encarga de proporcionar una interfaz amigable de cara al usuario con el único fin de ayudar en las tareas cotidianas del usuario en cuestión.

---

<sup>4</sup> Servicios de aplicaciones es una implementación de la especificación del estándar J2EE. Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas. Es el corazón de un gran sistema distribuido.



### 1.2.4 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 al público destinatario, y estableciendo una serie de requerimientos que tendría que poder satisfacer el CMS [4].

- **Código abierto.** El CMS tendría que ser de código fuente abierto.
- **Arquitectura técnica.** Tiene que ser fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos de 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 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 respuesta 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 un 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 no 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 una web, el CMS tendría que cumplir un estándar de accesibilidad.
- **Velocidad de descarga.** Teniendo en cuenta que no todos los usuarios disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente o dar la opción.



## Capítulo 1. Fundamentación teórica

- **Funcionalidades.** No se espera que todas las herramientas ofrezcan todas las funcionalidades, ni que estas sean las únicas que tendrá finalmente la web. Entre otras:
  - ✓ Editor de texto WYSIWYG a través del navegador.
  - ✓ Herramienta de búsqueda.
  - ✓ Comunicación entre los usuarios (foros, correo electrónico, chat).
  - ✓ Noticias.
  - ✓ Artículos.
  - ✓ Ciclo de trabajo (workflow) con diferentes perfiles de usuarios y grupos de trabajo.
  - ✓ Fechas de publicación y caducidad.
  - ✓ Webs personales.
  - ✓ Carga y descarga de documentos y material multimedia.
  - ✓ Avisos de actualización de páginas o mensajes en los foros, y envío automático de avisos por correo electrónico.
  - ✓ Envío de páginas por correo electrónico.
  - ✓ Páginas en versión imprimible.
  - ✓ Personalización según el usuario.
  - ✓ Disponibilidad o posibilidad de traducción al catalán y al castellano.
  - ✓ Soporte de múltiples formatos (HTML, Word, Excel, Acrobat, etc.).
  - ✓ Soporte de múltiples navegadores (Internet Explorer, Netscape, etc.).
  - ✓ Soporte de sindicación (RSS, NewsML, etc.).
  - ✓ Estadísticas de uso e informes.
  - ✓ Control de páginas caducadas y enlaces rotos.

Según estadísticas mundiales, uno de los CMS más difundidos y usados es:



## Capítulo 1. Fundamentación teórica

---

### 1.2.5 Drupal

Drupal es conocido por un sistema de administración de contenido para sitios Web. Dentro de sus características fundamentales se pueden citar las más significativas: permite publicar artículos, imágenes u otros servicios añadidos como foros, encuestas, votaciones y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto. Además, no tiene fronteras con los sistemas operativos, basta que este, soporte un servidor web con librerías PHP para ser instalado Drupal. Puede también trabajar contra cualquier base de datos ya que utiliza una interfaz de abstracción muy potente para programar independientemente de la tecnología utilizada [5].

En Drupal existe también un sistema de menús personalizables que brinda la facilidad de construir la navegación secundaria y primaria del sitio Web. Se pueden añadir jerarquías de enlaces internos y externos y automáticamente podrán ser incluidos como bloques en cualquiera de las áreas definidas por el template o plantilla visual. Otras de sus características para la gestión de sitios Web es la organización de los usuarios en roles que definen conjuntos de permisos de acceso y acciones.

Si bien es cierto que dentro de cualquier módulo puede generarse contenido HTML, los estándares establecidos para desarrolladores motivan al uso de funciones con prefijo “theme\_” para definir la forma en que se mostrará el contenido. De este modo cualquiera de estas funciones podrá ser sobrescrita por el diseñador en su template.

#### Características generales [5]

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



## Capítulo 1. Fundamentación teórica

- Búsqueda: Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- Código abierto: El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Módulos: La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.
- Personalización: Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo con las preferencias definidas por el usuario.
- URLs amigables: Drupal usa el mod\_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

### Gestión de usuarios

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

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

### Gestión de contenido

Control de versiones: El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha,



## Capítulo 1. Fundamentación teórica

---

quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.

Enlaces permanentes (Permalinks): Todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro [4].

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

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

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

### **Blogging**

Agregado de noticias: Drupal incluye un potente Agregado de Noticias para leer y publicar enlaces a noticias de otros sitios Web. Incorpora un sistema de caché en la base de datos, con temporización configurable [4].



## Capítulo 1. Fundamentación teórica

---

Soporte de Blogger: La API<sup>5</sup> de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que puedan ser 'herramientas Web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.

### Plataforma

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

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

Múltiples idiomas y Localización: Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz Web, importando traducciones existentes o integrando otras herramientas de traducción como GNU gettext [4].

### Rendimiento y escalabilidad

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

---

<sup>5</sup> API application programming interface, en español interfaz de programación de aplicaciones. Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.



## Capítulo 1. Fundamentación teórica

Sistema de Caché: El mecanismo de caché elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

### 1.3 Metodologías de desarrollo de Software.

En la actualidad, la necesidad del hombre de informatizar y automatizar el mundo que le rodea, ha llevado a la creación de sistemas más complejos que los anteriores, rompiendo así sus propios paradigmas, esto ha traído como consecuencia que se busquen métodos cada vez más exactos, que sean más fáciles para coordinar el trabajo. Existen varias metodologías, las más conocidas mundialmente son: XP y RUP.

#### 1.3.1 Extreme Programming (XP).

Esta metodología promueve los siguientes valores [6]:

##### **Comunicación.**

El extreme Programming se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación: la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos o más personas se comunican directamente pueden no solo consumir las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas, etc. que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva.

##### **Coraje**

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" estas de ninguna de las maneras. Esto es muy importante ya que





## Capítulo 1. Fundamentación teórica

---

un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave.

### **Simplicidad**

Dado que no se puede predecir cómo va a ser en el futuro, el software que se está desarrollando; un equipo de programación extrema intenta mantener el software lo más sencillo posible. Esto quiere decir que no se va a invertir ningún esfuerzo en hacer un desarrollo que en tiempo futuro pueda llegar a tener valor. En el XP frases como "...en un futuro vamos a necesitar..." o "Haz un sistema genérico de..." no tienen ningún sentido ya que no aportan ningún valor en el momento.

### **Feedback**

La agilidad se define (entre otras cosas) por la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Por este motivo uno de los valores que nos hace más ágiles es el continuo seguimiento o feedback que recibimos a la hora de desarrollar en un entorno ágil de desarrollo. Este feedback se toma del cliente, de los miembros del equipo, en cuestión de todo el entorno en el que se mueve un equipo de desarrollo ágil.

### **1.3.2 Proceso Unificado de Desarrollo (RUP)**

El RUP constituye una metodología muy potente utilizada para llevar a cabo la ingeniería de un software o de cualquier proyecto, además de atribuirle al usuario un previo conocimiento predecible del costo y el tiempo de desarrollo del software, obteniendo así una mayor calidad. RUP hace uso de Lenguaje Unificado de Modelado (UML) para acomodar los esquemas de un sistema software.

Existe algo peculiar y que hace único el Proceso Unificado de Desarrollo, estas son las tres características fundamentales que presenta: Iterativo e Incremental, Dirigido por Casos de Uso y Centrado en la arquitectura. [7]



## Capítulo 1. Fundamentación teórica

---

### Guiado por Casos de Usos

Los desarrolladores se basan en los Casos de Usos para crear una serie de modelos de diseño e implementación que llevan a cabo los Casos de Uso. Es de esta forma que los Casos de Uso no solo inician el proceso de desarrollo, sino que le proporcionan un hilo conductor, avanza a través de un conjunto de flujos de trabajo que parten de los Casos de Usos.

### Centrado en la arquitectura

La arquitectura de un software se describe mediante vistas diferentes del propio sistema en construcción, incluye los aspectos estáticos y dinámicos más importantes del sistema. La arquitectura es una vista del diseño completo con las características fundamentales resaltadas, dejando a un lado los detalles.

### Iterativo e Incremental

Según diversas teorías, es una práctica dividir para vencer, por eso es práctico dividir el esfuerzo de desarrollo de un proyecto en más pequeños o mini proyectos. Cada uno será una iteración que irá en incremento, las iteraciones hacen referencia a los distintos pasos en el flujo de trabajo, y los incrementos en el producto.

Un proyecto realizado según RUP se divide en cuatro fases: [7]

- *Conceptualización (Concepción o Inicio)*: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los Casos de Uso del sistema.
- *Elaboración*: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los Casos de Uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la



## Capítulo 1. Fundamentación teórica

arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo el alcance definido.

- *Construcción:* Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release<sup>6</sup> del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- *Transición:* El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas. RUP define nueve actividades a realizar en cada fase del proyecto. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo: [7]

- **Modelación del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

---

<sup>6</sup> Release: comprende un producto final, preparado para publicarse como versión definitiva a menos que aparezcan errores que lo impidan.



## Capítulo 1. Fundamentación teórica

- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

### 1.4 Lenguajes de modelación.

#### 1.4.1 Notación para el Modelado de Procesos de Negocio (BPMN)

BPMN es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes [8].

Está dirigido a gerentes, directores, dueños de empresas, ingenieros de procesos, analistas de negocios, analistas de sistemas, administradores de proyectos, responsables de calidad y todo aquel que necesita definir, documentar y hacer más eficientes sus procesos de negocio con el estándar más avanzado y aceptado a nivel internacional [8].



### 1.4.2 Lenguaje de Modelamiento Unificado (UML)

El UML fue desarrollado en Rational Software Corporation por Grady Booch, James Rumbaugh e Ivar Jacobson con contribuciones de otros metodólogos líderes, vendedores de software y muchos usuarios; el UML está basado en el uso extensivo del método Booch, OMT y Jacobson; es decir, el UML es la evolución de estas y otras aproximaciones para modelado de procesos de negocios, objetos, y el modelado de componentes [8].

UML es una notación con la cual se construyen sistemas por medio de conceptos orientados a objetos. Esta prescribe un conjunto de notaciones y diagramas estándares, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

El UML establece una serie de diagramas estándar y notaciones para la modelación de los sistemas orientados a objetos, y describe la semántica de lo que significan.

UML se usa para modelar diferentes tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas para modelar sistemas [9]:

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.



## Capítulo 1. Fundamentación teórica

- Diagramas de Implementación para modelar la distribución del sistema.

Las características fundamentales de UML son:

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

UML está pensado para ser utilizado en herramientas iterativas (herramientas-CASE (ISW Asistida por Ordenadores)) de modelado visual que tengan generadores de código así como generadores de informe.

### 1.5 Lenguajes de programación.

Un lenguaje de programación, no es más que un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Permite crear programas mediante un conjunto de instrucciones y operadores. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Por lo tanto, un lenguaje de programación es un modo práctico para que los programadores puedan dar instrucciones a un equipo.

Dentro de los lenguajes de programación web más conocidos se encuentran:

- JSP
- Python
- Ruby
- ASP
- ASP.NET

Además los siguientes:



## Capítulo 1. Fundamentación teórica

### 1.5.1 HTML

HTML<sup>7</sup> es el acrónimo inglés de HyperText Markup Language (Lenguaje de Marcas Hipertextuales), Es un lenguaje estático para el desarrollo de sitios web. Gracias a Internet, el HTML se ha convertido en uno de los lenguajes más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

Este lenguaje tiene como **Ventajas** [10]:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Como **Desventajas** [10]:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

### 1.5.2 Javascript

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos y no dispone de herencias [10].

---

<sup>7</sup> S., Christian Van Der Henst. *Maestros del web*. [En línea] 23 de Abril de 2001. [Citado el: 29 de Enero de 2010.] <http://www.maestrosdelweb.com/editorial/aspintro>.



## Capítulo 1. Fundamentación teórica

---

El código Javascript puede ser integrado dentro de las páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

### **Ventajas [10]:**

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

### **Desventajas [10]:**

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

### **1.5.3 PHP**

PHP (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Es uno de los lenguajes de lado servidor más extendidos en la web. Entre sus principales características se destacan su potencia, facilidad de aprendizaje, alto rendimiento y su escasez de recursos, esto lo ha llevado a una gran aceptación en la comunidad de webmasters. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows [11].

PHP permite incrustar sus fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma sencilla y eficaz sin tener que generar programas





## Capítulo 1. Fundamentación teórica

---

programados íntegramente en un lenguaje distinto al HTML. Por otra parte ofrece una infinidad de funciones para la explotación de bases de datos sin complicaciones. También les permite a los usuarios involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones [11].

Se cuenta además con PHP para el desarrollo del sistema debido a que es utilizado generalmente para la creación de contenido dinámico en aplicaciones Web. Además, es un soporte sólido para la Programación Orientada a Objetos pues contiene gran cantidad de funcionalidades [11]:

- *Enorme eficiencia:* Con escaso mantenimiento y un servidor gratuito (en el caso que se trata es wamp5), puede soportar sin problema millones de visitas diarias.
- *Gratuito:* Debido a que es software libre el mismo puede descargarse y ser utilizado por todo el que quiera.
- *Gran número de funciones predefinidas:* PHP fue diseñado especialmente para el desarrollo de páginas Web dinámicas. Por ello, está dotado de un gran número de funciones que simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con cookies y secciones, etc.
- *Versatilidad:* PHP puede usarse con la mayoría de los sistemas operativos, ya sea basado en UNIX, como con Windows y eso se debe a que es multiplataforma.

PHP ha crecido en soporte y en funciones y ahora viene con decenas de bibliotecas para funciones matemáticas, de bases de datos, etc. En este momento se encuentra en proceso de fortalecimiento tras unos cuantos años de éxitos y lauros, y la fase expansiva ha sido más bien dejada atrás para madurar en aspectos más relacionados con la integración de sus partes entre sí.



## Capítulo 1. Fundamentación teórica

### 1.5.1.1 Ventajas de PHP [11]

Actualmente PHP es uno de los lenguajes más conocidos y usados a nivel mundial en la creación de aplicaciones web debido a las ventajas que posee. Algunas de ellas que lo hacen ser tan utilizado por los programadores son:

- Es un lenguaje multiplataforma.
- PHP posee además la capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

## 1.6 Herramientas para desarrollar un Software.

### 1.6.1 Visual Paradigm

Visual Paradigm (VP) International, es un proveedor líder de software que permite a las organizaciones desarrollar aplicaciones más rápidas, de mejor calidad y más baratos. VP es una herramienta CASE que utiliza UML como lenguaje de modelaje, con el uso del acercamiento orientado a objeto. Se dedica a la prestación continua de software y servicios de manera precisa, transforma los sistemas en soluciones de software de calidad, con un riesgo mínimo y una rentabilidad máxima. Permite diseñar y desarrollar productos que eliminen la complejidad, mejoran la productividad y comprime sus marcos de tiempo de desarrollo de software.



## Capítulo 1. Fundamentación teórica

---

El VP es el único proveedor demostrado, a ofrecer la más amplia cobertura de las soluciones de desarrollo de la interoperabilidad de software que son independientes de la plataforma, en verdad el apoyo a la opción del cliente de Microsoft Visual Studio, Eclipse y otros grandes entornos de desarrollo.

Miles de clientes, incluyendo corporaciones globales, las principales instituciones financieras y agencias gubernamentales de todo el mundo, usan los productos de VP para modelar, construir y desplegar sus aplicaciones empresariales. Y lo eligen además debido a su reputación de excelencia en productos y servicios.

### Características [13]

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversion (nueva característica).
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los Casos de Uso, incluyendo la especificación del modelo general y de las descripciones de los Casos de Uso.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.



## Capítulo 1. Fundamentación teórica

- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

### 1.6.2 Servidor Web

Un Servidor Web se compone de dos elementos:

- El Hardware
- El Software para Servicios Web.

El Hardware normalmente es un computador con alto rendimiento, es decir, velocidades de reloj altas, medidas en Mega Hertz (2.5 Ghz, 3.0 Ghz ó más), Cantidad de Memoria Grande (512 Megas ó más), Gran capacidad de almacenamiento (discos de 80 Gigas ó más) [14].

En términos técnicos, los servidores web soportan el protocolo de transferencia de Hipertexto (HTTP) que es un estándar para comunicación web en Internet. Con este protocolo el servidor web envía páginas Web en HTML y otros tipos de scripts (ASP, ASP.NET, JavaScript, etc) al navegador web cuando estos lo requieren. Cuando un usuario hace clic sobre un enlace a una página web, se envía una solicitud al servidor web para localizar los datos nombrados por ese enlace (o link). El servidor web recibe esta solicitud y suministra los datos que le han solicitado, en forma de una página con formato HTML, o bien, devuelve un mensaje de error.

Como ejemplos de Servidores Web tenemos [15]:

- Internet Information Server (IIS).



## Capítulo 1. Fundamentación teórica

- Xitami.
- Cherokee.
- Tomcat, un servidor basado en Apache para estándares de Java (JSP y Servlets).
- Apache.

### 1.6.2.1 Apache

Software de código abierto que más se utiliza en el mundo, lo que demuestra que es una solución dominante y ampliamente probada, funciona sobre cualquier plataforma, permite que otros ordenadores vean la web mediante un navegador. Es una solución altamente configurable y extensible a través de módulos, se integra perfectamente con varias tecnologías, lenguajes, plataformas, bases de datos, etc. Es considerado el servidor web con la mejor funcionalidad/velocidad [16].

Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información [16].

La versión 2.0, incorpora grandes novedades y mejoras, combina las herramientas libres Apache, PHP, logrando centrar la atención en millones de sitios web dinámicos.

### 1.6.2.2 Wamp

La mejor opción para Windows es WAMP Server. WAMP es la abreviación de Windows, Apache, Mysql y PHP. Es un proyecto desarrollado por los franceses, paquete que instala la versión 1.3 de Apache, PHP5, MySQL, PhpMyAdmin y SQLitemanager. Además ofrece los servicios que se instala como un “tray icon”.<sup>8</sup>

WAMP es un sistema indicado para los usuarios que no tienen instalado en el sistema ninguno de los programas necesarios para programar en PHP (Apache, PHP y MySQL), ya que

---

<sup>8</sup> S., Christian Van Der Henst. Maestros del web. [En línea] 30 de Mayo, 2006. [Citado el: 16 de Enero de 2010.] <http://www.maestrosdelweb.com/editorial/ampfacil>



## Capítulo 1. Fundamentación teórica

---

realiza una instalación de PHP5 completa y desde cero. Los usuarios que disponen de Apache, PHP y/o MySQL en su sistema también pueden utilizar este programa. Y en este caso, simplemente se realizará otra copia de las aplicaciones en un directorio distinto, que en principio, no tiene por qué interferir con las otras instalaciones alojadas en nuestro equipo [17].

Una de las ventajas que tiene este paquete además de ofrecer la versión de PHP5 en el instalador, es que también te ofrece sus constantes actualizaciones. La instalación es bastante sencilla y destaca que los servicios de Apache y de MySQL se instalarán en Windows bajo los nombres de wampapache y wampmysql para no interferir con otras instalaciones previas de estos servidores.

### 1.6.3 Servidor de Bases de Datos MySQL

En la actualidad los servidores de base de datos más usados en software libre son:

- PostgreSQL
- MySQL
- Firebird
- MaxDb
- SQLite
- Sybase

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones [18].

Supera desde hace tiempo a muchas soluciones comerciales en velocidad y dispone de un sistema de permisos elegante y potente. Este sistema de gestión de base de datos usa el lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información. MySQL es muy rápido y capaz de almacenar grandes cantidades de datos.

Existen muchas razones para escoger MySQL [18]:



## Capítulo 1. Fundamentación teórica

- *Costo*: es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- *Asistencia*: ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- *Velocidad*: es mucho más rápido que la mayor parte de sus rivales.
- *Funcionalidad*: dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID9, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL10 e integración con la mayor parte de los entornos de programación. Así mismo, se desarrolla y actualiza de forma mucho más rápida que muchos de sus rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo.
- *Portabilidad*: se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- *Facilidad de uso*: resulta fácil de utilizar y administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

### 1.6.4 Zend Studio

Es un ambiente de desarrollo integrado (IDE) orientado al desarrollo de aplicaciones Web, usando PHP. Incrementa la velocidad de los ciclos de desarrollo y hace más fácil los proyectos de gran complejidad mediante herramientas de edición, análisis, optimización y bases de datos [18].

---

<sup>9</sup> Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

<sup>10</sup> Protocolo de Capa de Conexión Segura.



## Capítulo 1. Fundamentación teórica

---

El diseño cliente/servidor de Zend Studio permite correr múltiples clientes para cada uno de los programadores de un gran proyecto desde un único servidor. Los clientes y el servidor de Zend Studio pueden correr tanto en Linux como en Windows [19].

### 1.7 Propuesta.

Se decidió utilizar como metodología de desarrollo: RUP, porque es más adaptable para proyectos a largo plazo, como lenguaje de modelado a UML porque es el más conocido y utilizado en la actualidad, y como herramienta CASE al Visual Paradigm, porque es de gran utilidad, robusto, multiplataforma; es una considerable opción para la UCI, que está migrando hacia el software no propietario.

Se decidió usar la plataforma Drupal 6.14 que es un CMS integral y fácil de manejar. Para la gestión de la base de datos se seleccionó MySQL y como servidor web Wamp5 1.7.4 a petición de la entidad que solicitó el producto. Para la implementación los lenguajes de programación seleccionados fueron HTML, PHP y Javascript para ganar en rapidez y ahorro de ancho de banda. El Zend Studio 5.1.0 por brindar mayores funcionalidades y una mayor rapidez a la hora de trabajar.





# Capítulo II

## Características del sistema

### Introducción

El desarrollo de la aplicación se centra en el Proceso Unificado de Desarrollo de Software (RUP) y el Lenguaje de Modelado Unificado (UML - Unified Model Language) para la modelación de los artefactos. Para ello ha sido de gran ayuda el uso de la herramienta Visual Paradigm.

En este capítulo se describe el objeto de estudio y de automatización, se hace un análisis de los procesos, actores, trabajadores y Casos de Uso pertenecientes al negocio. Además se analizan y obtienen los actores, Casos de Uso del sistema, los requerimientos funcionales, en los cuales se recogen las características básicas que tendrá el sistema y los requisitos no funcionales, que son necesarios para garantizar el éxito del mismo.



## Capítulo 2. Características del sistema.

---

### 2.1 Objeto de Estudio.

El LAREDI se encuentra inmerso en la búsqueda de un medio más rápido y eficiente para solicitar un RD, esto hace necesario la creación de un sistema de gestión, para de esta manera brindar un servicio en línea al profesorado universitario, contribuyendo a obtener los siguientes resultados:

- Mejor calidad de los procesos internos que se llevan a cabo durante la creación de los RD en el Laboratorio.
- Gestionar las solicitudes de producción, incluso que se realicen simultáneamente y sin importar donde se encuentre el solicitante.
- Almacenar y recuperar la documentación en el menor tiempo posible.
- Realizar búsquedas avanzadas de la documentación.
- Generar de una manera rápida y sencilla, resultados estadísticos (cantidad de solicitudes en cierto período de tiempo, el profesor que más solicitudes ha realizado, etc.), para facilitar la toma de decisiones.
- Un espacio donde el cliente pueda emitir quejas y sugerencias con la posibilidad de que sean atendidas en el propio momento que las emita, al contar con un servicio de alerta y mensajería conjuntamente con el servicio de correo de la universidad.
- Intercambiar ideas entre los desarrolladores y el cliente.
- El cliente podrá hacer un seguimiento de la producción del recurso solicitado.

Actualmente, el flujo de trabajo no es el más óptimo en cuanto a control, rapidez y eficiencia de los procesos de producción. Dentro de estos, se llevan a cabo solicitudes para la creación de un RD. Estas se realizan de forma personal, siendo una dificultad para el solicitante, en ocasiones le es incómodo el desplazamiento hacia el LAREDI. En caso de llegar un profesor a realizar una solicitud y existir otro anteriormente, debe esperar a que este termine para ser atendido por el responsable de la recepción de las peticiones de producción. Esto trae como



## Capítulo 2. Características del sistema.

---

consecuencia que no se realice la solicitud en el tiempo que tenía previsto y con ella el tiempo para obtener el RD.

### **2.2 Objeto de Automatización.**

Basándose en el negocio descrito se propone automatizar esta primera versión del Sistema de Gestión para el LAREDI de la UCI, lo que le permite al usuario solicitar un RD, desde cualquier parte de la universidad de una manera rápida y sin espera alguna. También se desea automatizar la documentación generada en cada etapa por la que pasa un recurso solicitado, mejorando el almacenamiento de la misma y su recuperación en un menor tiempo.

### **2.3 Descripción de los procesos del negocio.**

Este proyecto tendrá su aplicación práctica en el contexto universitario. Los profesores universitarios tendrán la posibilidad de concebir sus RD con ayuda especializada para la producción, se pondrán estos en función de la formación en cualquiera de sus modalidades.

Los Departamentos Docentes Centrales y los profesores en general serán beneficiarios de los servicios que se prestarán.

El proceso permite elaborar un recurso didáctico, a partir de una petición por un profesor, mediante diferentes etapas tales como: Negociación, Preproducción, Producción y Postproducción.

Un profesor realiza una Solicitud al Coordinador Líder mediante el documento establecido y de forma personal, este recepciona la solicitud y determina la fecha de la Negociación y le comunica al cliente la fecha establecida. En la Negociación se determina simplemente si el recurso que solicita el cliente se puede realizar por el equipo de trabajo. En caso de no ser posible, se le comunica al cliente los motivos y concluye la Negociación. En caso de ser posible la Producción, se precisa si se necesita algún Guión para la realización del producto, la entrega de los recursos de los que disponga el cliente (audios, videos, imágenes, textos, etc.), se le



## Capítulo 2. Características del sistema.

---

entrega el documento de Estructura Didáctica(ED) en blanco que debe elaborar y entregar en la siguiente cita de trabajo, se establece el cronograma de trabajo de las distintas etapas por las que pasará su recurso y se procede a firmar el documento que da inicio a la producción del RD.

Vencida esta etapa el Coordinador Líder selecciona el equipo que trabajará en dicho recurso (Desarrollador (es), Asesor Pedagógico, Diseñador Instruccional, Diseñador Gráfico, Montador, Revisor, Editor de sonido-video), selecciona el jefe del equipo y les comunica el cronograma de las etapas. Los seleccionados del equipo de trabajo se preparan para entrevistar al cliente.

Se inicia la segunda entrevista de trabajo con el cliente (inicio de la Preproducción) donde se determinan los prototipos que serán elaborados por el equipo de trabajo, se revisan los recursos entregados por el cliente (medias), se revisa la estructura didáctica del recurso y se le hacen sugerencias. Se comienza la Producción del recurso y se continúa la Preproducción en caso de ser necesaria la elaboración de medias que el cliente aprobará al ser terminadas. Cada media desarrollada es evaluada y aprobada al menos por el revisor al frente.

Durante la producción se realiza el montaje de las distintas medias y la salida en diferentes soportes según las estrategias definidas y necesidades del contexto en la cual se utilizarán, no sin antes ser evaluado por el revisor designado y aprobadas por el cliente.

Al terminar la Producción se procede a la Postproducción donde se realiza la evaluación integral del recurso según los indicadores de calidad previamente definidos y la publicación del producto en el ROA.

### **2.4 Actores del Negocio.**

En los procesos del negocio intervienen el claustro de la universidad que sería el que realiza la solicitud para la producción de un RD. Los profesores al final se benefician al obtener



## Capítulo 2. Características del sistema.

los recursos para apoyar el proceso de Enseñanza y Aprendizaje, proporcionándoles a los estudiantes una mayor motivación y así obtener mejor recepción de los contenidos impartidos.

Este usuario es considerado **Actor del Negocio**. El mismo se encuentra especificado en la Tabla1.

Nombre del Actor	Descripción
Profesor	Representa a los profesores de la UCI que acuden al laboratorio de Teleformación para solicitar que se le realice un RD.

**Tabla 1: Actor del Negocio**

### 2.5 Trabajadores del Negocio.

En los procesos de negocio intervienen trabajadores que participan directamente en la realización de estos procesos, denominados **Trabajadores del Negocio**. Los mismos se encuentran especificados en la Tabla 2.

Nombre del Trabajador	Descripción
Coordinador Líder del Proyecto	Se encarga de recibir la solicitud y entregarla a los Desarrolladores. Envía la fecha de la entrevista al profesor.
Equipo de Trabajo (Desarrollador (es), Asesor Pedagógico, Diseñador Instruccional, Diseñador Gráfico, Montador, Editor de sonido-video)	Se encarga de determinar los prototipos, de revisar los recursos entregados por el profesor (medias), de revisar la estructura didáctica del RD, de hacerle sugerencias al profesor, aprobar las propuestas del profesor y ayuda en la confección de la misma.
Revisor	Se encarga de evaluar y aprobar las medias (audios, videos, imágenes, textos, etc.) entregadas por el



## Capítulo 2. Características del sistema.

	profesor o realizadas durante la Producción. Además de velar porque se cumplan las exigencias del profesor capturadas en la entrevista, y los requisitos de calidad establecidos.
--	---

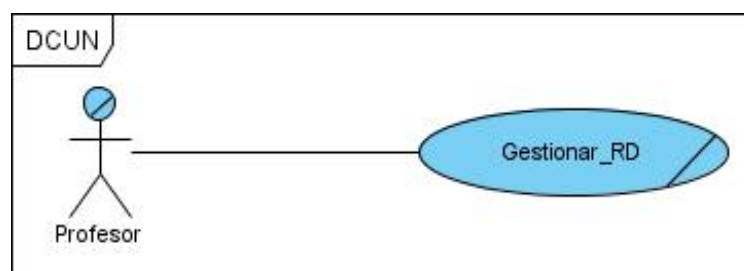
**Tabla 2: Trabajadores del Negocio.**

### 2.6 Casos de Uso del Negocio.

Nombre del Caso de Uso del Negocio	Descripción
Gestionar Recurso Didáctico	En este Caso de Uso el profesor puede solicitar la producción un Recurso Didáctico, puede modificar, eliminar dicha solicitud, así como participar en cada etapa de la producción del mismo.

**Tabla 3: Caso de Uso del Negocio.**

### 2.7 Diagrama de Casos de Uso del Negocio.



**Fig.1: Diagrama del Caso de Uso del Negocio.**

### 2.8 Diagramas de Actividades del Negocio (Ver [Anexo 1](#)).



## Capítulo 2. Características del sistema.

---

**2.9 Diagrama de Clases del Modelo de objetos del negocio (Ver [Anexo 1](#)).**

**2.10 Descripción textual del Caso de Uso del Negocio (Ver [Anexo 2](#)).**

### **2.11 Especificación de los requisitos de Software**

A continuación se presentan los requisitos funcionales y no funcionales del sistema expresados en lenguaje natural. Los mismos serán identificados con las siglas **RF y RNF**, seguidas del **Número del requisito** (Ej. RF 1).

#### **2.11.1 Requisitos Funcionales**

Los requerimientos funcionales no son más que las capacidades o condiciones que el sistema debe de cumplir. Es decir, son aquellos requisitos que desde el punto de vista de las necesidades del usuario, debe cumplir el sistema.

##### **Propios del CMS**

RF 1: Autenticar usuario.

RF 2: Crear rol.

RF 3: Modificar rol.

RF 4: Eliminar rol.

##### **Requisitos Implementados**

RF 5: Registrar nombre y apellidos.

RF 6: Registrar solapín.

RF 7: Registrar usuario.

RF 8: Registrar recurso.



## Capítulo 2. Características del sistema.

---

RF 9: Visitar información.

RF 10: Mostrar enlaces a otros sitios.

RF 11: Agregar documentación.

RF 12: Modificar documentación.

RF 13: Mostrar documentación.

RF 14: Eliminar documentación.

RF 15: Llenar documentación.

RF 16: Último registrado.

RF 17: Cantidad de solicitudes por mes.

RF 18: Cantidad de solicitudes en una fecha determinada.

RF 19: Cantidad de solicitudes por usuario.

RF 20: Insertar Solicitud.

RF 21: Modificar Solicitud.

RF 22: Eliminar Solicitud.

RF 23: Enviar fecha de entrevista.

RF 24: Enviar notificación de aceptación de la solicitud.





## Capítulo 2. Características del sistema.

---

RF 25: EL sistema debe permitir visualizar las noticias promovidas en la página frontal.

RF 26: EL sistema debe permitir visualizar una noticia completa.

RF 27: EL sistema debe permitir visualizar el listado de las noticias titulares.

RF 28: Agregar etapas.

RF 29: Modificar etapas.

RF 30: Eliminar Etapas.

RF 31: Agregar un tipo de recurso.

RF 32: Modificar un tipo de recurso.

RF 33: Eliminar un tipo de recurso.

RF 34: Agregar curso.

RF 35: Modificar curso.

RF 36: Mostrar curso.

RF 37: Eliminar curso.

RF 38: Abandonar un curso.

RF 39: Registrar en un curso.

RF 40: Atender solicitud.



## Capítulo 2. Características del sistema.

---

### 2.11.2 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades no son más que las características que hacen que el producto sea atractivo, usable, rápido y confiable. Estos requerimientos forman una parte significativa de la especificación y son importantes para que tanto los clientes como los usuarios puedan valorar las características no funcionales del producto.

#### ➤ Usabilidad

- Diseño limpio y claro que facilite la navegación. El sistema puede ser utilizado por cualquier persona que se encuentre dentro de la Universidad, además el mismo tiene que estar orientado a usuarios con conocimientos básicos de informática y trabajo con la Web.

#### ➤ Confiabilidad

- Sistema de tracking de errores
- El sistema debe llevar un log donde se registren los eventos del sistema (incluyendo los eventos de errores)
- Replicación de la BD
- Garantizar la replicación de la BD para asegurar la recuperación ante fallos.

#### ➤ Rendimiento

El sistema debe ser eficiente y preciso en la información que le suministra al usuario para evitar cualquier tipo de error. El tiempo de respuesta ante cualquier solicitud del usuario debe ser el mínimo posible por lo que debe implementar varias transacciones por segundo para dar una respuesta rápida al usuario y evitar demoras. Debe estar disponible todo el tiempo para trabajar. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de procesamiento de la información.



## Capítulo 2. Características del sistema.

---

### ➤ Soporte

Para garantizar el soporte de esta aplicación, se requiere MySQL 5.x como servidor de bases de datos, PHP en su versión 5.0, un servidor Web Apache 2.2.x, por parte del cliente se requiere un navegador capaz de interpretar JavaScript y CSS, el sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios, las pruebas realizadas al sistema deben permitir evaluar sus ventajas y funcionalidades, además de detectar los errores que presenta.

### ➤ Restricciones de diseño

- Hacer la página lo más ligera posible (Minimizar el uso de imágenes)

### ➤ Interfaz

- Diseñado para una resolución de 800x600
- Interfaces de usuarios
- Interfaces usables para usuarios de nivel medio.
- Interfaces con otros Software

### ➤ Adquisición de Componentes

- Drupal CMS versión 6.14.
- Módulos desarrollados.

### ➤ Software

- Servidor de BD MySQL.
- Servidor Web Apache 2.2.6.
- PHP 5.
- Sistema de Administración de Contenidos Drupal 6.14.



## Capítulo 2. Características del sistema.

### ➤ Seguridad

- Solicitar al usuario su identificación antes de que pueda ejecutar cualquier acción de modificación sobre el sistema.
- Mostrar a cada usuario solo las funcionalidades del sistema sobre las cuales tiene permiso de acceso.
- El servidor donde se encuentre instalado el sistema debe estar ubicado en un local protegido contra el hurto y los desastres naturales.

### ➤ Disponibilidad

- Diseñado para su funcionamiento constante.

## 2.12 Descripción textual de los Casos de Uso del Sistema.

### 2.12.1 Actores del Sistema

Actores	Justificación
Usuario Anónimo	Usuario que se encuentra accediendo al sistema de forma anónima.
Profesor	Actor que se registra en el sistema encargado de realizar una solicitud, llenar documentos, puede consultar toda la información del sistema y poseer determinados privilegios.
Servidor de Correo	Permite enviar una notificación vía correo a un Profesor que realizó una solicitud.
Desarrolladores	Responsable de gestionar las noticias y los cursos que se estén impartiendo en el laboratorio.



## Capítulo 2. Características del sistema.

Administrador	Administrador del sistema responsable de configurarlo, administrarlo y modificar la información del sistema.
---------------	--

**Tabla 4: Actores del Sistema.**

### 2.12.2 Listado de los Casos de Uso del Sistema

CU-1	Autenticarse.
Actor	Profesor
Descripción	En este caso de uso el usuario accede al formulario de autenticación en el que llenará los campos de usuario y contraseña, el sistema comprobará la identidad y si es correcta podrá acceder al sitio de forma personalizada de lo contrario, en el caso de que el usuario haya puesto mal el usuario o contraseña se le dará la posibilidad de volver al formulario y ponerla correctamente.
Referencia	RF 1

**Tabla 5: Caso de Uso Autenticar Usuario.**

CU-2	Visitar Información.
Actor	Profesor
Descripción	El usuario accede a la página principal. El sistema muestra una bienvenida y enlaces a otros sitios y puede visitar toda la información contenida en el mismo.
Referencia	RF 9, RF 10



## Capítulo 2. Características del sistema.

**Tabla 6: Caso de Uso Visitar información.**

CU-3	Gestionar Documentación
Actor	Administrador
Descripción	El Administrador accede al sistema para publicar, después de realizada la solicitud, los documentos que debe llenar el profesor en cada etapa de producción del RD. El sistema muestra la documentación.
Referencia	RF 12, RF 13, RF 14, RF 15

**Tabla 7: Caso de Uso Gestionar Documentación.**

CU-4	Llenar Documentación
Actor	Profesor
Descripción	El caso de uso inicia cuando el usuario llena los documentos necesarios en cada etapa del recurso solicitado.
Referencia	RF 15

**Tabla 8: Caso de Uso Llenar documentación.**

CU-5	Obtener estadísticas
Actor	Administrador
Descripción	El Administrador obtiene las estadísticas referentes a las solicitudes, en cuanto, a cantidad de solicitudes por profesor, por mes y en un rango de fecha



## Capítulo 2. Características del sistema.

	determinado. El sistema muestra la información.
Referencia	RF 17, RF 18, RF 19

**Tabla 9: Caso de Uso Gestionar estadísticas.**

CU-6	Gestionar solicitud de producción
Actor	Profesor
Descripción	El Profesor accede al sistema para solicitar la producción un recurso didáctico.
Referencia	RF 20, RF 21, RF 22

**Tabla 10: Caso de Uso Gestionar solicitud de producción.**

CU-7	Enviar notificaciones
Actor	Servidor de Correo
Descripción	El caso de uso se inicia cuando se le envía al Profesor un correo con una notificación de que su solicitud fue aceptada o no.
Referencia	RF 23, RF 24

**Tabla 11: Caso de Uso Enviar Notificación.**

CU-8	Gestionar rol
------	---------------



## Capítulo 2. Características del sistema.

Actor	Administrador
Descripción	El caso de uso inicia cuando el Administrador gestiona los roles que permitirán personalizar el acceso al portal.
Referencia	RF 2, RF 3, RF 4

**Tabla 12: Caso de Uso Gestionar rol.**

CU-9	Gestionar Noticia
Actor	Desarrolladores
Descripción	El caso de uso inicia cuando los Desarrolladores acceden al sistema para gestionar todo lo referente a las noticias y finaliza cuando se muestran las noticias en dependencia de su categoría.
Referencia	RF 25, RF 26, RF 27

**Tabla 13: Caso de Uso Gestionar Noticia.**

CU-10	Gestionar Etapas
Actor	Administrador
Descripción	El caso de uso inicia cuando el Administrador accede al sistema para adicionar, modificar o eliminar una etapa de producción.
Referencia	RF 28, RF 29, RF 30

**Tabla 14: Caso de Uso Gestionar Etapas.**





## Capítulo 2. Características del sistema.

CU-11	Gestionar Tipo de Recurso
Actor	Administrador
Descripción	El caso de uso inicia cuando el Administrador accede al sistema para adicionar, modificar o eliminar un tipo de recurso didáctico.
Referencia	RF 31, RF 32, RF 33

**Tabla 15: Caso de Uso Gestionar Tipo de Recurso.**

CU-12	Gestionar Curso
Actor	Desarrolladores
Descripción	El caso de uso inicia cuando los Desarrolladores acceden al sistema para adicionar, modificar o eliminar un curso que se desee impartir a los profesores de cómo hacer un determinado recurso.
Referencia	RF 34, RF 35, RF 36, RF 37

**Tabla 16: Caso de Uso Gestionar curso.**

CU-13	Registrarse en Curso
Actor	Profesor
Descripción	El caso de uso inicia cuando un profesor registrado en el sitio decide registrarse en un curso o si está registrador decide abandonarlo.
Referencia	RF 38, RF 39



**Tabla 17: Caso de Uso Registrarse en Curso.**

CU-14	Atender solicitud
Actor	Administrador
Descripción	El caso de uso inicia cuando el Administrador accede al sistema para aceptar o denegar las solicitudes realizadas.
Referencia	RF 40

**Tabla 18: Caso de Uso Atender solicitud.**

### 2.13 Diagrama de Casos de Uso del Sistema.

Para obtener una mejor claridad se ha organizado el Diagrama de Casos de Uso del Sistema por paquetes y a continuación del diagrama general se representan los paquetes, clasificados de acuerdo al usuario que interactúa con el sistema (Ver [Anexo 3](#)).

Para la descripción detallada de los casos de uso del sistema (Ver [Anexo 4](#)).

### Conclusiones

En el presente capítulo se hizo un análisis de los procesos del negocio y los requisitos funcionales y no funcionales que describen, respectivamente, la funcionalidad y las propiedades que debe cumplir el sistema a desarrollar. Se describieron las acciones de los actores del sistema con los Casos de Uso que se relacionan. Finalmente se hace una descripción más amplia de algunos Casos de Uso críticos del sistema.



# Capítulo III

## Análisis y Diseño

### Introducción

En este capítulo se describen los elementos fundamentales del análisis y diseño del Sistema de gestión para el LAREDI, se modelan los diagramas de clases del análisis, los diagramas de clases y el diagrama entidad relación de la Base de Datos.



### 3.1 Modelo de Análisis.

En la construcción del modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

RUP propone clasificar las clases en:

**Entidad:** Modelan información que es de larga vida y que es a menudo persistente.

**Interfaz:** Modelan la interacción entre el sistema y sus actores.

**Control:** Coordinan la realización de uno o pocos Casos de Uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

#### 3.1.1 Diagrama de clases del análisis.

Un diagrama de clases del análisis es un artefacto donde se representan los conceptos en un dominio del problema. Para cada Caso de Uso del sistema se realizó un diagrama de clases del análisis (Ver [Anexo 5](#)).

### 3.2 Diseño.

El diseño tiene el propósito de adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías de interfaz de usuario. Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.



### 3.2.1 Diagrama de Clases del Modelo de Diseño

En el Diagrama de Clases para las aplicaciones web se definen las pautas para el diseño de la interfaz de la aplicación partiendo de la importancia que representa para la obtención del producto final como para el usuario que interactúa con la misma, además se modelan las páginas y los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de las mismas, una vez que estén en el navegador del cliente. Teniendo en cuenta esto se realizan los diferentes diagramas, los cuales son la representación de los paquetes que en este caso estarán conformados por los módulos de Drupal.

**Themes:** Incluye las distintas plantillas, cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

**Includes:** Contiene un conjunto de ficheros indispensables para el correcto funcionamiento del CMS, como por ejemplo **Database** que provee las funcionalidades de acceso a la base de datos de Drupal.

**Modules:** Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.

**Scripts:** Contiene un conjunto de ficheros indispensables para el funcionamiento. Fundamentalmente orientados al aspecto visual, CSS y java script. Las páginas que genera Drupal puede tener o no formularios, en dependencia de lo que se desee. Es importante señalar que este CMS contiene una única página servidora.

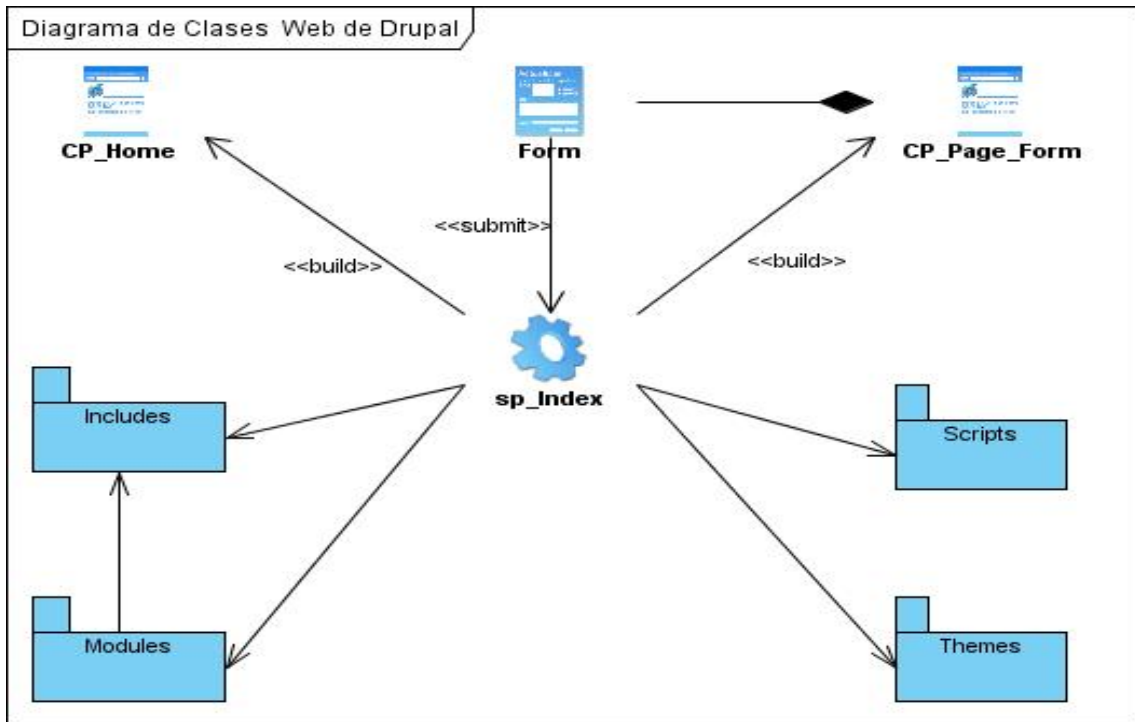


Fig. 2: Diagrama de Clases Web de Drupal.

### 3.2.2 Módulos

Debido a que la aplicación se desarrolla sobre el CMS Drupal, el cual tiene una estructura desarrollada en módulos, para lograr que en un futuro se puedan implementar nuevas funcionalidades e incluso optimizar las existentes se han empleado interfaces, y para mantener una estructura organizada también se implementaron módulos a los cuales le asignamos determinada responsabilidad.

En esta sección se realizará una descripción de cada módulo por separado, debido a que tienen funcionalidades independientes y los diagramas de clases de los módulos desarrollados.



### 3.2.2.1 Módulos de Drupal

#### Módulo 1: Evento (Event)

Este es un módulo simple para hacer un seguimiento de los eventos y mostrar a los usuarios diferentes calendarios. También proporciona una API de eventos para que otros módulos puedan almacenar y trabajar con datos actualizados. El módulo de eventos tiene su propio calendario que se puede utilizar para mostrar los campos de eventos.

#### Módulo 2: Módulo del Área banner (Area Banner Module)

Módulo del Área banner permite observar la imagen y banners Flash basado en la actual "área". Un área se compone de todas las páginas de un elemento de menú determinado. Los banners pueden ser cargados y asignados a un elemento de menú. Todos los hijos de este elemento de menú heredarán la bandera asignada.

El módulo de banner área permite dos tipos de banderas: Imagen y banners Flash. Cuando ambos, un archivo Flash y una imagen se ha subido por el mismo menú, el banner Flash se mostrará con la bandera de la imagen como alternativa en caso de que el navegador no soporte Flash.

#### Módulo 3: LDAP integración (LDAP\_integration)

Permite la conexión con el LDAP de la UCI, de esta forma los usuarios entran al sitio con la cuenta del dominio.

#### Módulo 4: Administración-Módulos-Contenido (Content Construction Kit - CCK)

El módulo CCK permite la creación de nuevos tipos de contenido. Al activar este módulo se activan los módulos Nodereference, Number, Optionswidgets, Text, Userreference los cuales se utilizan a la hora de crear un nuevo tipo de contenido, y permite que los campos sean del tipo



## Capítulo 3. Análisis y Diseño

---

que se necesite. Es de gran importancia porque diversos son los contenidos que se pueden crear a través de Drupal. Este trae instalado por defecto los contenidos: libros, página, una noticia del foro, encuesta, blog y un artículo. Si un diseñador del sitio web necesita crear un contenido que tenga otro estilo diferente al que tienen estos contenidos, necesitaría activar estos módulos o por lo menos alguno de ellos.

### **Módulo 5: Menú de Administración (Admin\_menu)**

El módulo Menú de Administración proporciona un tema de la interfaz de administración independiente para Drupal. Es una ayuda para los usuarios novatos procedentes de CMS Drupal, un verdadero ahorro de tiempo para los administradores del sitio y una necesidad para los desarrolladores de Drupal.

El módulo hace que todos los elementos del menú administrativo por debajo del “administrar” en un menú basado en limpio, atractivo y puramente CCS, en la parte superior del sitio web. Contiene no sólo los elementos de menú regular, las tareas locales también están incluidas, dando un rápido acceso a cualquier recurso administrativo y ofrece la función de su instalación de Drupal.

### **Módulo 6: WYSIWYG un editor de HTML (FCKeditor)**

Este módulo FCKeditor - un editor de HTML visual, a veces llamado el editor WYSIWYG permite a Drupal reemplazar los campos textarea. Este editor de texto HTML proporciona muchas de las potentes funciones de edición WYSIWYG de los editores de escritorio conocido como Word a la web. Es relativamente liviano y no requiere ningún tipo de instalación en el equipo cliente.





### **Módulo 7: SWF Herramientas (SWF Tools)**

SWF Herramientas le permite integrar fácilmente el contenido de Flash en sus páginas. Usted puede utilizar campos de archivo CCK o el uso de PHP y imprimir swf ('mymovie.swf'); El módulo consta de una serie de componentes que usted puede optar por instalar. Por ejemplo, usted puede elegir una variedad de métodos de inserción, como la incrustación de marcado directo, o SWF Objeto 2 JavaScript. También puede activar el soporte para un número de reproductores de medios comunes, tales como Wijering Player o FlowPlayer. SWF Tools también soporta listas de reproducción. Ahora también soporta el flujo de RTMP, por lo que puede permitir a los usuarios saltar a través de los contenidos.

### **Módulo 8: Vistas (Vistas)**

El módulo Vistas permite presentar las listas de contenido de una manera diferente a la tradicional en que se presentan en los bloques. Este módulo permite organizar el contenido dentro de un bloque de la manera que se desee. Por ejemplo, se desea tener en un bloque todas las páginas que pertenezcan a la categoría Linux, con el módulo Vistas se puede realizar, o cuando se quieren organizar alfabéticamente los términos dentro de un bloque.

Este es el módulo que permite personalizar y crear páginas web y bloques según se necesite. Una vez activado el módulo y con los permisos necesarios, se crea una nueva opción en el menú del administrador llamado vistas, a través de este, se puede observar las vistas que ya están creadas, o crear una nueva.

### **Módulo 9: Mensajería Privada (Privatemsg)**

El módulo de Privatemsg permite a los visitantes del sitio web enviar mensajes privados entre sí. Es perfecto para sitios orientados a la comunidad, como los foros y sitios de redes sociales. Entre sus características se encuentra la de enviar y recibir mensajes privados en un sitio (entre varios destinatarios), conversaciones con rosca (que sea más fácil hacer un



seguimiento de los mensajes y respuestas), buscar mensajes privados, etiquetado y filtrado (ayuda a organizar las conversaciones), bloqueo de usuario y notificaciones por e-mail de nuevos mensajes.

### **3.2.2.2 Módulos de Drupal modificados.**

#### **Módulo 10: Usuario (User)**

El módulo Usuario administra el registro de usuarios y el sistema de inicio de sesión. El mismo es modificado con el objetivo de mostrar los datos de los usuarios autenticados por el dominio UCI, al igual que la foto. Se verifica si el usuario es profesor, de serlo se le asigna el rol de Profesor y son almacenados sus datos.

### **3.2.2.3 Módulos desarrollados**

#### **Módulo 11: Tipos de Recursos**

Permite tener un control de los tipos de RD que se desarrollan en el laboratorio, facilitándole a los administradores de manera sencilla agregar un nuevo tipo de recurso, así como eliminar o modificar alguno ya existente, sin necesidad de ir al código, porque el módulo contiene una lista de los recurso hasta el momento y te dará la posibilidad de realizar las acciones antes mencionadas (agregar, eliminar o modificar). Este módulo genera una tabla en la base de datos con el nombre “recurso”. Este módulo resuelve los requisitos funcionales 31, 32 y 33.

#### **Módulo 12: Etapas de Producción**

El mismo muestra cada una de las etapas por la que pasa el desarrollo de un RD, negociación, preproducción, producción y posproducción. En un futuro quizás algunas de las etapas no sea necesaria y el propósito del módulo es poder eliminar alguna o integrarlas en caso de ser necesario. La tabla “etapas” de la base de datos, es generada por este módulo. Este módulo resuelve los requisitos funcionales 28, 29 y 30.



### **Módulo 13: Solicitud de Producción**

El módulo solicitud de producción es el más importante. El mismo posibilita realizar una solicitud de producción de un RD, seleccionando el tipo de recurso. Una vez realizada la petición será enviado un mensaje al correo del cliente, notificándole que la solicitud se ha realizado correctamente. El usuario puede modificar o eliminar la solicitud si aún no ha sido aceptada. Por la parte administrativa, muestra un listado con todas las solicitudes realizadas, permitiéndole aceptarla o denegarla, además de poderla imprimir en formato pdf. En caso de proceder se le envía fecha de entrevista al cliente en caso contrario, se elimina la solicitud y se le notifica al solicitante explicando el por qué no procedió su petición. Crea en la base de datos las tablas con el nombre de “solicitud” y “profesor”. Depende de Tipos de Recursos. Este módulo resuelve los requisitos funcionales 20, 21, 22 y 40.

### **Módulo 14: Mostrar Usuario**

Muestra el último usuario que registró el sistema. Este módulo resuelve el requisito funcional 16.

### **Módulo 15: Cursos**

Brinda la posibilidad de adicionar nuevos cursos, listarlos, modificarlos y eliminarlos, además de exportar a pdf el listado de los profesores inscritos en determinado curso. Le permite al profesor unirse o abandonar el curso, en caso de estar inscrito y al administrador de vincular los profesores a un curso específico. Muestra en un bloque los cursos agregados y le muestra al profesor los cursos en los que se encuentra inscrito. Crea en la base de datos las tablas “curso” y “curso-profesor”. Este módulo resuelve los requisitos funcionales 34, 35, 36, 37, 38 y 39.



## Capítulo 3. Análisis y Diseño

---

### **Módulo 16: Documentación por Etapas**

Gestiona la documentación que lleva cada etapa. Depende de: Upload, Etapas de Producción y genera la tabla “documentación”. Este módulo resuelve los requisitos funcionales 8, 9, 10, 11 y 12.

### **Módulo 17: Contador del Sitio**

Permite llevar algunas estadísticas del sitio como la cantidad de visitas realizadas, número de usuarios registrados y dirección IP del servidor y el cliente. Crea la tabla “contador” en la base de datos.

### **Módulo 18: Documentos por usuario**

Le brinda la posibilidad al usuario, que realizó una solicitud, de subir el documento que llenó, para el sitio. Este módulo genera en la base de datos la tabla “documento”. Depende de: Documentación por Etapas. Este módulo resuelve los requisitos funcionales 8, 9, 10, 11 y 12.

### **Módulo 19: Estadísticas de Solicitudes**

Muestra la cantidad de solicitudes realizadas desde que comenzó a funcionar el sistema hasta el día que se realiza la petición de las estadísticas. Muestra la cantidad de profesores que han realizado al menos una solicitud en forma de tabla y el que más solicitudes tenga. Se pueden obtener dichas solicitudes por meses y con la posibilidad de adquirirlas por rango de fechas. Depende de: Solicitud de Producción. Este módulo resuelve los requisitos funcionales 14, 15 y 16.

### **Módulo 20: Noticias**

Permite la gestión de las noticias que se deseen publicar, además de crear la tabla “noticia” en la base de datos. Este módulo resuelve los requisitos funcionales 25, 26 y 27.



### 3.2.3 Diagrama de clases Persistentes

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado está dado por al almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información (Ver [Anexo 6](#)).

### 3.2.4 Diseño de la Base de Datos

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema. Es una serie de conceptos que pueden utilizarse para describir un conjunto de datos y las operaciones para manipularlos (Ver [Anexo 7](#)).

### 3.2.5 Descripción de las tablas

A continuación se describen cada una de las tablas de la base de datos (Ver [Anexo 8](#)).

### Conclusiones

En este capítulo se ha logrado traducir los requisitos a una especificación que describe cómo implementar el sistema. En la etapa de análisis se ha logrado obtener una visión del sistema que se encarga de mostrar el comportamiento de los casos de uso apoyándose en los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis, el cual tiene en cuenta los requisitos no funcionales, y ver cómo el sistema cumple sus objetivos, lo que se convierte en un elemento más que suficiente para que la aplicación web pueda ser implementada sin ambigüedades.



# Capítulo IV

## Implementación y Prueba

### Introducción

En este capítulo se desarrollan los flujos de trabajo implementación y prueba. La implementación se comenzó con el resultado del diseño y se implementó el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El flujo de trabajo de análisis y diseño se propone crear un plano del modelo de implementación, por lo que sus últimas actividades están vinculadas a la creación del modelo de despliegue.

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. El flujo de implementación está fuertemente determinado por el lenguaje de programación.

La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. El mismo nos da la posibilidad de descubrir las debilidades del sistema y una vez descubiertas trabajar en base a mejorar sus funcionalidades.



### 4.1 Diagrama de Despliegue

El diagrama de despliegue es un diagrama de objeto que indica la situación física de los componentes lógicos desarrollados y muestra las relaciones entre los componentes hardware y software en el sistema final. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o un procesador. Es decir se sitúa el software en el hardware que lo contiene. Cada hardware se representa como un nodo (Ver [Anexo 9](#)).

A continuación se describen cada uno de los nodos presentes en el diagrama anterior y la comunicación entre ellos:

- **Servidor Web:** Representa el servidor sobre el cual se encuentra corriendo la aplicación Web. Este accede al **servidor de base de datos** para el manejo de información mediante el protocolo TCP/IP.
- **PC Cliente:** Representa la computadora del cliente que se conecta al **Servidor Web**. La misma se comunica con el servidor por el protocolo HTTP.
- **Servidor BD:** Es la representación del servidor de base de datos en el cual los datos pertenecientes a la aplicación son almacenados.
- **Impresora:** Permite al cliente que se conectan a la aplicación imprimir cualquier documento que sea de su agrado.

### 4.2 Diagrama de componentes

Un componente es una parte física y reemplazable de un sistema que está conformado por un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usa para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces; poseen relaciones de traza con los elementos del modelo de



## Capítulo 4. Implementación y prueba

---

diseño que implementan. Un diagrama de componentes muestra la organización y las dependencias lógicas entre un conjunto de componentes de software, sean estos de código fuente, librerías, tablas archivos, ejecutables, binarios y documentos que formen parte del sistema. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar el código y organizando los subsistemas de implementación en capas. Es decir sirve para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

A continuación se muestra el diagrama de componentes que genera Drupal para estructurar los subsistemas de implementación que usa este CMS (Ver [Anexo 10](#)).

### Componentes del diagrama:

**Index.php:** Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.

**Modules:** Contiene todos los módulos, que permiten las distintas funcionalidades del CMS.

**Themes:** Incluye las plantillas que se pueden utilizar, nos permite separar el contenido de la presentación.

**Includes:** Contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal.

**DataBase:** Representa la base de datos.

**Misc:** Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js, etc.).



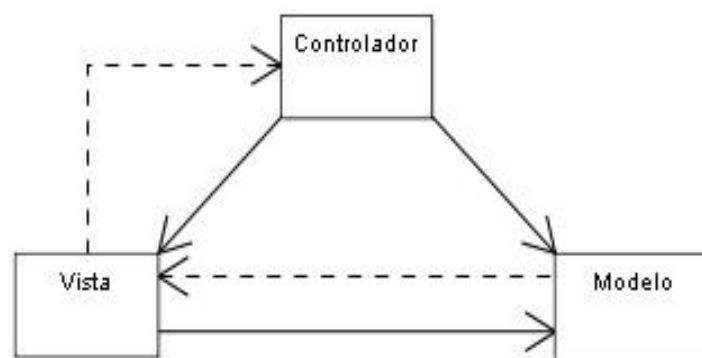


### 4.3 Arquitectura y Patrones

El sistema está implementado usando el CMS Drupal, por lo que la arquitectura y los patrones usados heredan del CMS.

Este CMS se puede enmarcar en los paradigmas de la programación orientada a objetos (POO), aunque no se utilicen a fondo todas las características de PHP para este paradigma. El código de Drupal está separado en módulos, cada uno cumple funciones específicas, que además manejan inclusión de archivos, todo esto simplemente con el propósito de que se cargue la menor cantidad de código por petición para que resulte lo más eficiente posible, las funciones son definidas dentro de otras funciones en tiempo de ejecución.

En cuanto a la arquitectura de Drupal se puede hablar de “Modelo - Vista - Controlador” (MVC) pues es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.



**Fig. 3: Patrón MVC.**



## Capítulo 4. Implementación y prueba

---

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Más allá de clasificar Drupal en uno u otro patrón, se habla en términos de "modelo-vista-controlador" en el sentido de que Drupal hace una excelente separación entre la persistencia de los datos y los formularios (modelo), la lógica (ciclo de respuesta al pedido http, o sea, ciclo de vida de un "request" a un sitio Drupal), y la vista que emerge como la respuesta HTML al explorador del internet (browser). En cada esfera, hay muchas oportunidades para efectuar modificaciones de manera limpia.

### 4.4 Modelo de Prueba

Actualmente el desarrollo de un software implica una serie de actividades en las cuales es común cometer errores. Los errores pueden comenzar a cometerse desde el principio del proceso, con un planteamiento erróneo de los objetivos a alcanzar o durante los procesos de diseño e implementación. Debido a la incapacidad humana de comunicarse y realizar tareas de manera perfecta, es necesario llevar a cabo actividades que garanticen la calidad.

“Las pruebas a una aplicación web son actividades en las cuales un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente”.



## Capítulo 4. Implementación y prueba

---

Las pruebas a una aplicación web son elementos críticos para la garantía de la calidad del software, la importancia de los costos asociados a los fallos de un sistema han motivado la realización de pruebas minuciosas y bien planificadas.

Seguidamente se describen algunos casos de prueba desarrollados por cada caso de uso definido, especificándose los resultados y las condiciones que debe dar el sistema (Ver [Anexo 11](#)).

### Conclusiones

En este capítulo se realizó la modelación de los nodos en los que será distribuida la aplicación, especificando en cada caso el protocolo de comunicación a través del Diagrama de Despliegue, además se describe la arquitectura en la que se basa la solución del software. Se realizó la descripción de los casos de prueba de integración de cada uno de los Casos de Uso del Sistema y se obtuvo el Diagrama de Componentes en el que se representaron las dependencias de los componentes software.



## Conclusiones Generales

Con este trabajo se presenta una aplicación web, donde se han cumplido exitosamente los objetivos trazados, trabajando a la altura de que se esperaba y se necesitaba, cumpliendo estrictamente con los requisitos solicitados por el cliente, con las mejoras y ventajas propuestas; capaz de resolver los problemas en el LAREDI y de mantener informado a todos los usuarios de la UCI sobre los temas del área y específicos de la línea de trabajo, a través de las noticias nacionales e internacionales y otros documentos relacionados con los recursos didácticos y la producción en general.

El sistema se diseñó siguiendo la metodología RUP y se utilizaron representaciones UML para la modelación de todas las fases del proyecto, obteniéndose como resultado un sistema con una interfaz amigable y fácil de navegar por el usuario, permitiendo un acceso rápido y simultáneo.



## Recomendaciones

---

### Recomendaciones

Una vez concluido el desarrollo de este documento en el cual se cumplieron los objetivos propuestos se hacen las siguientes recomendaciones:

- Mantener sobre el sistema un estricto cumplimiento del proceso de soporte y actualización periódica, para procurar la fiabilidad y funcionamiento óptimo del sitio y de la información que se gestiona a través de él.
- Incorporar una mayor colección de módulos ya configurados a la aplicación para el aumento de las funcionalidades que beneficien a los administradores que trabajarán con la aplicación.
- Aprovechar el conocimiento adquirido con esta investigación para seleccionar los módulos más viables y crear perfiles de instalación de Drupal para tareas específicas, como noticias con video incorporado, entre otras.
- Trabajar en la traducción al español de todos los módulos estudiados.
- Desarrollar el sitio en la versión más reciente de Drupal.
- Utilizar como gestor de base de datos PostgreSQL.
- Integrar la aplicación obtenida al sistema de ROA.
- Incrementar las funcionalidades de la aplicación web.



## Referencias Bibliográficas

### Referencias Bibliográficas

- [1] APLICACIONES DE ESCRITORIO. Disponible en:  
<http://www.freebsd.org/doc/es/books/handbook/desktop.html>
- [2] CAPÍTULO III APLICACIÓN WEB. en: *Aplicación Web para el conocimiento y conversión de unidades*. 48. P.
- [3] CUERDA GARCIA XAVIER Y MIGUILLÓN ALFONSO JULIÀ. *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto*, [2008]. Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>.
- [4] CUERDA GARCÍA XAVIER Y MINGUILLÓN ALFONSO JULIÀ. *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto*, 2008. Disponible en: <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto>.
- [5] REYERO, J. A. *Sobre Drupal*, 2006. [2006]. Disponible en:  
<http://www.drupal.org.es/drupal>.
- [6] WELLS, DON 2003. *Extreme Programming: A gentle introduction*. Disponible en:  
<http://www.extremeprogramming.org>.
- [7] JACOBSON, I. *El Proceso Unificado de Desarrollo de software*. La Habana, Editorial Félix Varela, 2004. 438 p.
- [8] ANÓNIMO. *Curso práctico de Modelado de Negocios con UML y BPMN*. Disponible en: <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>
- [9] JAVIER, M. N. *Diseño del Sistema de tarjeta de Crédito con UML*. Disponible en: [http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/mendoza\\_nj/Cap1.pdf](http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/mendoza_nj/Cap1.pdf).



## Referencias Bibliográficas

- [10] PÉREZ VALDÉS, DAMIÁN. *Los diferentes lenguajes de programación para la web*  
Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>
- [11] GALLEGO VÁZQUEZ, J. A. *Desarrollo web con php y mysql* Disponible en:  
<http://bibliodoc.uci.cu/pdf/reg02138.pdf>.
- [12] CASARES, CLAUDIO. *Tutorial de SQL, Introducción a SQL* [2004]. Disponible en:  
<http://www.maestrosdelweb.com/editorial/tutsql1>.
- [13] ANÓNIMO. Disponible en: [www.visual-paradigm.com](http://www.visual-paradigm.com)
- [14] PERALTA FERNÁNDEZ, SAGRARIO. *El servidor Web. Arquitectura y funcionamiento*. Disponible en: <http://observatorio.cnice.mec.es>.
- [15] LÓPEZ ORTEGA, ÁLVARO. *Proyecto Cherokee: Diseño, implementación y aspectos de rendimiento de servidores Web*. Apartado “Tipos de servidores Web”.
- [16] ANTONIO SAORÍN, J. J. G., GINÉS ÁNGEL ESTEBAN, JOSÉ ANTONIO MORGADO, MARÍA DEL MAR LERA, JOSÉ LUIS LÓPEZ-BARAJAS, SARA REDONDO. *Sirviendo web desde la escuela*, 2006]. Disponible en:  
<http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/apache.html>.
- [17] BOURDON, ROMAIN. <http://www.wampserver.com>.
- [18] IAN, GILFILLAN *La Biblia MySQL* [Disponible en:  
<http://bibliodoc.uci.cu/pdf/reg02847.pdf>
- [19] VIVABORG. *Zend Studio 3.0 listo para PHP 5.0*, 2003. [2006]. Disponible en:  
<http://www.vivaphp.com.ar/soft/2444.html>.