



Universidad de las Ciencias Informáticas
Facultad 8

Herramienta Web para la Autonomía
en el Aprendizaje y la Consulta de
contenidos de Informática.

**TRABAJO DE DIPLOMA
PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

AUTORES

Deiler Sevilla Fernández

Yanirys Silva Lemus

TUTOR

M.Sc. Ing-Lic. Orlando Farray Álvarez

Ciudad de la Habana

Junio 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los __ días del mes de Junio del 2008.

Autores:

Yanirys Silva Lemus

Deiler Sevilla Fernández

Tutor:

M.Sc. Ing-Lic Orlando Farray Álvarez

“No vamos a sentarnos y hacer un alto en el camino para pensar cuales son nuestros próximos pasos. Vamos a pensar caminando, vamos a aprender creando y también, por qué no decirlo, equivocándonos”

A handwritten signature in black ink, appearing to be the initials 'JL' in a cursive style.

Dedicatoria

A mi abuelo Sergio, por lo orgulloso que estaría hoy, y la vida no le dio la oportunidad de verme graduada.

A mi mamá y a mi abuela materna por su amor, apoyo y confianza plena. Por enseñarme a vencer los obstáculos que se presentan en el camino de la vida. Por constituir este trabajo uno de sus más anhelados sueños.

A mi papá, por todo el amor que siento por él.

A mis hermanos, por ser parte de mi inspiración. A ustedes lego mi ejemplo.

A mi novio porque sin su cariño, ayuda y comprensión no lo hubiese logrado.

A todos los que me ayudaron, me apoyaron y estuvieron siempre a mi lado aún cuando la cima parecía estar más lejos, a aquellos que siempre creyeron en mí y no me abandonaron ni un momento, a todos ustedes dedico mi triunfo.

Yaní

A mi mamá por todo el amor y la confianza depositada, por estar en cada momento cuando la necesité, por lo orgullosa que se siente hoy.

A mis hermanos y a mi familia, porque siempre estuvieron pendientes de mi.

Deifer

Agradecimientos Comunes

A nuestro Comandante en Jefe y a la Revolución por proporcionarnos la oportunidad de estudiar en un centro como este. Por ser el paradigma de todo joven revolucionario.

A la Universidad de las Ciencias Informáticas por formarnos como profesionales.

A nuestro tutor M.Sc. Ing-Lic Orlando Farray Álvarez por su paciencia y dedicación.

A todos los profesores que de una forma u otra nos enseñaron a transitar en el camino por la búsqueda del conocimiento.

A nuestros amigos, a todos los que nos ayudaron...

Gracias

Agradecimientos

A mami y a mima, por estar siempre en el momento que las necesité, por depositar toda su confianza en mí.

A Osnielito, por estar siempre a mi lado y ayudarme a ver la luz cuando sentía que había desaparecido.

A mis compañeros de 5 años de estudio, por compartir tantos momentos que quedarán en mi memoria...a mis grandes amigos, gracias por todo: Dany, Emilio, Alex, Deiler, Helen... por disfrutar juntos inolvidables momentos.

A mi entrañable compañero de tesis, por su paciencia.

A mi familia, por estar siempre pendiente de mis estudios y a todos los que de una forma u otra han hecho realidad este sueño...

Yaní

A mi mamá, porque sin su contante esfuerzo no hubiese logrado este sueño.

A mi papá y mis tíos todos, por brindarme su apoyo y la inspiración de un nivel superior.

A mis hermanos, unos por ser el ejemplo a seguir y a los otros lego el mío.

A mi novia por su comprensión y ayuda en cada momento, por su cariño y dedicación.

A mis amigos, socios...por compartir juntos buenos y malos momentos que nunca olvidaré. A Alex, Emilio, Dany, Yanirys, porque juntos logramos ser una nueva familia.

A mi familia de manera general por estar pendientes de mis pasos.

Deiler

RESUMEN

En la Universidad de las Ciencias Informáticas (UCI), al igual que en muchos centros educacionales, el proceso de enseñanza autodidacta permite al estudiantado prepararse en diferentes temáticas que no son parte del plan de estudio. El desarrollo científico alcanzado por la humanidad en las Tecnologías de la Informática y las Comunicaciones es realmente vertiginoso, lo que posibilita que la docencia apoyándose en estas ventajas cobre fuerza. El presente trabajo que lleva por título “Herramienta Web para la Autonomía en el Aprendizaje y la Consulta de contenidos de Informática” pretende brindar cursos de contenidos informáticos para la elección en cualquiera de las modalidades de aprendizaje. El estudiante a través de la aplicación será capaz de escoger dentro de los diferentes bloques de contenidos de cada nivel, cual es aquel, o aquellos, que son de su interés, planificándose lo que desea cursar, y sobre esa base recibirá una certificación acreditativa como constancia de su aprendizaje. El sistema constará con materiales que ayudarán a la preparación, y se podrá ir evaluando las habilidades alcanzadas en cada uno de los contenidos. Este documento recoge el resultado de todo el trabajo realizado para la elaboración del sistema antes mencionado, incluyendo el estudio del estado del arte de aplicaciones con similares objetivos existentes en el mundo, el estudio y definición de las características del sistema, así como la planificación, implementación, prueba y factibilidad del producto.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN..... 1

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA 6

1.1 INTRODUCCIÓN..... 6

1.2 FUNDAMENTACIÓN DEL TEMA 6

1.2.1 Procesos de Gestión de Contenidos6

1.2.2 Autonomía sobre los contenidos.....8

1.2.3 Sistemas vinculados al campo de acción.8

1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES..... 10

1.3.1 Tendencias y Desarrollo del Software libre10

1.3.2 Lenguajes de programación para la Web y Servidores Web11

1.3.3 Gestores de Base de Datos16

1.3.4 Metodologías de Desarrollo de Software.....20

1.3.5 Sistemas de Gestión de Contenidos23

1.4 CONCLUSIONES..... 27

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA 28

2.1 INTRODUCCIÓN..... 28

2.2 DESCRIPCIÓN DE LOS PROCESOS VINCULADOS AL CAMPO DE ACCIÓN..... 28

2.2.1 Flujo actual del proceso28

2.2.1 Objeto de automatización29

2.3 PROPUESTA DEL SISTEMA..... 29

2.3.1 Personas relacionadas con el sistema30

2.3.2 Requerimientos funcionales del sistema31

2.3.3 Requerimientos no funcionales del sistema35

2.4 CONCLUSIONES..... 37

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN 38

3.1 INTRODUCCIÓN..... 38

3.2 FASE DE EXPLORACIÓN 38

3.2.1 Historias de usuario38

3.3 FASE DE PLANIFICACIÓN 45

3.3.1 Estimación de esfuerzos por historias de usuario	46
3.3.2 Plan de iteraciones	46
3.3.3 Plan de duración de las iteraciones	48
3.3.4 Plan de entregas	48
3.4 CONCLUSIONES.....	50
CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS	51
4.1 INTRODUCCIÓN.....	51
4.2 DISEÑO DEL SISTEMA.....	51
4.2.1 Módulos de Drupal	52
4.2.2 Paquete 1: Drupal	53
4.2.3 Sub paquete 1.1: Modules	54
4.3 DISEÑO DE LA BASE DE DATOS	56
4.4 FASE DE IMPLEMENTACIÓN.....	58
4.4.1 Iteración 1	59
4.4.2 Iteración 2	63
4.4.3 Iteración 3	67
4.4.4 Diagrama de Componentes	69
4.4.5 Diagrama de Despliegue	70
4.5 PRUEBAS.....	70
4.5.1 Pruebas de aceptación	70
4.6 CONCLUSIONES.....	80
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.....	81
5.1 INTRODUCCIÓN.....	81
5.2 CARACTERÍSTICAS DEL PROYECTO.....	81
5.2.1 Estimación inicial	82
5.3 CÁLCULO DE INSTRUCCIONES FUERTES, ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.....	83
5.3.1 Cálculo del esfuerzo nominal.....	83
5.3.2 Cálculo del esfuerzo ajustado.....	84
5.3.3 Beneficios tangibles e intangibles.....	86
5.3.4 Análisis de Costo	86
5.4 CONCLUSIONES.....	87

CONCLUSIONES.....	88
RECOMENDACIONES.....	89
REFERENCIAS BIBLIOGRÁFICAS	90
BIBLIOGRAFÍA.....	91
GLOSARIO DE TÉRMINOS.....	94
ANEXOS.....	97

ÍNDICE DE TABLAS

TABLA 2.1 PERSONAS RELACIONADAS CON EL SISTEMA.....	31
TABLA 3.1 HU GESTIONAR CUENTA DE USUARIO EN EL SISTEMA.....	39
TABLA 3.2 HU MOSTRAR PERFIL DE USUARIO.....	39
TABLA 3.3 HU AUTENTICAR.....	40
TABLA 3.4 HU CREAR MATERIALES.....	40
TABLA 3.5 HU MOSTRAR MATERIALES.....	41
TABLA 3.6 HU ELIMINAR MATERIALES.....	41
TABLA 3.7 HU GESTIONAR BIBLIOGRAFÍA.....	42
TABLA 3.8 HU MOSTRAR BIBLIOGRAFÍA.....	42
TABLA 3.9 HU GESTIONAR PROGRAMAS DE CURSOS.....	43
TABLA 3.10 HU MOSTRAR PROGRAMAS DE CURSOS.....	43
TABLA 3.11 HU MATRICULAR EN CURSOS.....	44
TABLA 3.12 HU BUSCAR INFORMACIÓN.....	44
TABLA 3.13 HU MOSTRAR LISTADO DE ESTUDIANTES.....	45
TABLA 3.14 HU MOSTRAR LISTADO DE PROFESORES.....	45
TABLA 3.15 ESTIMACIÓN DE ESFUERZOS POR HU.....	46
TABLA 3.16 PLAN DE DURACIÓN DE LAS ITERACIONES (EQUIPO DE DESARROLLO).....	48
TABLA 3.17 MÓDULOS Y HU ABARCADAS.....	49
TABLA 3.18 PLAN DE DURACIÓN ENTREGA.....	49
TABLA 4.1 TARJETA CRC MÓDULO BIBLIOGRAFÍA.....	55
TABLA 4.2 TARJETA CRC MÓDULO CURSOS.....	56
TABLA 4.3 TARJETA CRC MÓDULO MATRÍCULA.....	56
TABLA 4.4 MÓDULOS ABORDADOS EN LA PRIMERA ITERACIÓN.....	59
TABLA 4.5 TAREA 1 DEL MÓDULO USER.....	59
TABLA 4.6 TAREA 2 DEL MÓDULO USER.....	60
TABLA 4.7 TAREA 3 DEL MÓDULO USER.....	60
TABLA 4.8 TAREA 1 DEL MÓDULO BIBLIOGRAFÍA.....	61
TABLA 4.9 TAREA 2 DEL MÓDULO BIBLIOGRAFÍA.....	61
TABLA 4.10 TAREA 3 DEL MÓDULO BIBLIOGRAFÍA.....	62
TABLA 4.11 TAREA 1 DEL MÓDULO CURSO.....	62
TABLA 4.12 TAREA 2 DEL MÓDULO CURSO.....	63
TABLA 4.13 TAREA 3 DEL MÓDULO CURSO.....	63
TABLA 4.14 MÓDULOS ABORDADOS EN LA SEGUNDA ITERACIÓN.....	64

TABLA 4.15 TAREA 4 DEL MÓDULO CURSO	64
TABLA 4.16 TAREA 4 DEL MÓDULO BIBLIOGRAFÍA.....	65
TABLA 4.17 TAREA 5 DEL MÓDULO BIBLIOGRAFÍA.....	65
TABLA 4.18 TAREA 1 DEL MÓDULO MATRÍCULA	66
TABLA 4.19 TAREA 2 DEL MÓDULO MATRÍCULA	66
TABLA 4.20 TAREA 3 DEL MÓDULO MATRÍCULA	66
TABLA 4.21 TAREA 4 DEL MÓDULO MATRÍCULA	67
TABLA 4.22 MÓDULOS ABORDADOS EN LA TERCERA ITERACIÓN.....	67
TABLA 4.23 TAREA 5 DEL MÓDULO MATRÍCULA	68
TABLA 4.24 TAREA 6 DEL MÓDULO BIBLIOGRAFÍA.....	68
TABLA 4.25 TAREA 1 DEL MÓDULO PERFIL.	69
TABLA 4.26 TAREA 1 DEL MÓDULO SEARCH.	69
TABLA 4.27 PRUEBA 1 AL MÓDULO CURSOS	71
TABLA 4.28 PRUEBA 2 AL MÓDULO CURSOS	71
TABLA 4.29 PRUEBA 3 AL MÓDULO CURSOS	72
TABLA 4.30 PRUEBA 4 AL MÓDULO CURSOS	72
TABLA 4.31 PRUEBA 1 AL MÓDULO BIBLIOGRAFÍA.....	73
TABLA 4.32 PRUEBA 2 AL MÓDULO BIBLIOGRAFÍA.....	73
TABLA 4.33 PRUEBA 3 AL MÓDULO BIBLIOGRAFÍA.....	74
TABLA 4.34 PRUEBA 4 AL MÓDULO BIBLIOGRAFÍA.....	74
TABLA 4.35 PRUEBA 5 AL MÓDULO BIBLIOGRAFÍA.....	75
TABLA 4.36 PRUEBA 6 AL MÓDULO BIBLIOGRAFÍA.....	75
TABLA 4.37 PRUEBA 7 AL MÓDULO BIBLIOGRAFÍA.....	76
TABLA 4.38 PRUEBA 8 AL MÓDULO BIBLIOGRAFÍA.....	76
TABLA 4.39 PRUEBA 1 AL MÓDULO MATRÍCULA	77
TABLA 4.40 PRUEBA 2 AL MÓDULO MATRÍCULA	77
TABLA 4.41 PRUEBA 3 AL MÓDULO MATRÍCULA	78
TABLA 4.42 PRUEBA 4 AL MÓDULO MATRÍCULA	78
TABLA 4.43 PRUEBA 5 AL MÓDULO MATRÍCULA	79
TABLA 4.44 PRUEBA 6 AL MÓDULO MATRÍCULA	79
TABLA 4.45 PRUEBA 7 AL MÓDULO MATRÍCULA	80
TABLA 5.1 ENTRADAS EXTERNAS	81
TABLA 5.2 SALIDAS EXTERNAS	81
TABLA 5.3 CONSULTAS EXTERNAS	82
TABLA 5.4 ARCHIVOS LÓGICOS	82

TABLA 5.5 ARCHIVOS DE INTERFAZ EXTERNA.....	82
TABLA 5.6 PUNTOS DE FUNCIÓN DESAJUSTADOS	83
TABLA 5.7 FACTOR ESCALAR (SF)	84
TABLA 5.8 MULTIPLICADORES DE ESFUERZO (EM).....	85
TABLA 5.9 RESULTADOS ALCANZADOS	86

ÍNDICE DE FIGURAS

FIGURA 1 DIAGRAMA DE CLASES DEL DISEÑO: PAQUETE DRUPAL 54

FIGURA 2 DIAGRAMA DE CLASES DEL DISEÑO: SUB PAQUETE MODULES 55

FIGURA 3 DIAGRAMA DE CLASES PERSISTENTES..... 57

FIGURA 4 MODELO DE DATOS..... 58

FIGURA 5 A.1 DIAGRAMA DE COMPONENTES DEL SISTEMA. 97

FIGURA 6 A.2 DIAGRAMA DE COMPONENTES: MÓDULO BIBLIOGRAFÍA..... 98

FIGURA 7 A.3 DIAGRAMA DE COMPONENTES: MÓDULO CURSOS. 98

FIGURA 8 A.4 DIAGRAMA DE COMPONENTES: MÓDULO MATRÍCULA. 99

FIGURA 9 A.5 DIAGRAMA DE DESPLIEGUE. 99

INTRODUCCIÓN

Desde tiempos remotos, la humanidad ha sentido la necesidad de aprender y enseñar los conocimientos adquiridos. Con el propósito de lograr sus objetivos se apropia de métodos y medios auxiliares que le permitan trasladar estas experiencias de una forma más clara y precisa, propiciándole al oyente una vía de fácil entendimiento.

La formación del ciudadano es uno de los problemas más acuciantes de las sociedades democráticas contemporáneas. Tanto en países con una larga tradición democrática que gozan de una estabilidad institucional ya consolidada, como en aquellos con democracias incipientes que están luchando por el afianzamiento de sus instituciones sociales y políticas, la educación de los ciudadanos con profundas bases éticas, es considerada como la principal herramienta para construir una sociedad más justa y solidaria. Para poder afianzarse y sobrevivir, la democracia necesita alimentarse de las acciones individuales y colectivas que promuevan la recuperación ética de sus principios constitutivos y dentro de este contexto la educación aparece como una condición ineludible para la consolidación y continuidad democrática.

La enseñanza del actual siglo ha estado matizada por el uso de los medios técnicos auxiliares, dentro de los cuales la computadora ha desempeñado una función preponderante por las ventajas que incorporó, tanto para la explicación de los conceptos como para su aprobación. En la medida que ha ido avanzando la tecnología se han buscado métodos que resulten efectivos para el proceso. Se puede afirmar, que a cada paradigma de la información ha estado asociada la didáctica y la metodología que apoyen a la docencia en los contenidos más diversos.

La Universidad de las Ciencias Informáticas, nunca ajena al desarrollo y a la vanguardia de la informatización de los programas docentes, se ha propuesto expandir los conocimientos informáticos no sólo a sus estudiantes, sino a todo el personal interesado, sin importar el nivel cultural que posea. La enorme aceptación de este transformador proyecto lleva implícito un aumento considerable de los cursos a impartir, debido a la diversidad de usuarios que serán matriculados, así como de los profesores capacitados para desarrollar dicha labor.

Muchos son los sistemas implementados en gran parte del mundo que posibilitan el aprendizaje de forma autodidacta y la consolidación de una variedad de contenidos, haciendo uso de las facilidades brindadas por las nuevas tecnologías utilizadas en el proceso docente, así como la informatización de la sociedad, que se podrían poner a prueba en correspondencia con lo que se desea establecer como una nueva vía de aprendizaje.

En estudios realizados se ha percibido que los sistemas usados con este fin posibilitan un amplio catálogo donde muestran las asignaturas ofertadas, así como el plan de estudio de cada una, con el objetivo de que, según el interés de las personas, puedan escoger y matricularse en las que deseen. Sin embargo, no se ha encontrado en ninguno de los cursos, la posibilidad de que el interesado pueda tener autonomía sobre la planificación de los contenidos, es decir, que pueda dentro de un mismo curso ofertado, escoger cuales contenidos son los que desea estudiar.

Esta posibilidad de autonomía sobre cada uno de los contenidos, brinda a los cursistas una vía más factible de aprovechamiento del tiempo, pues lo convierte en el principal protagonista a la hora de tomar decisiones acerca de los contenidos que le son de interés. Con este beneficio se logra hacer un uso racional de planificación en la formación del estudiante sin afectar la calidad de los mismos.

Debido a la situación expuesta anteriormente se plantea el siguiente **problema**: ¿Cómo contribuir a la publicación de cursos con autonomía total en la planificación de los contenidos para el estudio de temas de informática?

El **objeto de estudio** consiste en los procesos de gestión del aprendizaje autónomo, y de éste se deriva que el **campo de acción** sean los procesos de gestión del aprendizaje autónomo para el estudio de contenidos de informática.

Para guiar el desarrollo de este proceso se plantea como **idea a defender**: la implementación del sistema "Herramienta Web para la autonomía en el Aprendizaje y la Consulta de contenidos de informática", brindará a los cursistas un sitio Web que posibilite la autonomía total de los mismos.

Con el fin de dar solución a la problemática antes mencionada se define como **objetivo general** realizar la implementación de un sistema con autonomía total sobre la planificación para el estudio de contenidos informáticos, del cual se derivan los siguientes **objetivos específicos**:

1. Realizar una búsqueda de los diferentes Sistemas de Gestión de Información en el ámbito nacional e internacional, que posibiliten el estudio de contenidos informáticos.
2. Realizar el proceso de desarrollo del software “Herramienta Web para la autonomía en el Aprendizaje y la Consulta de contenidos de Informática”.
3. Validar el sistema propuesto.

Para cumplir con los objetivos precedentes se proponen las siguientes tareas:

1. Realizar la selección y revisión bibliográfica sobre temas relacionados con la publicación de cursos mediante sistemas automatizados.
2. Realizar un estudio sobre la existencia de aplicaciones o soluciones similares con las características del sistema propuesto.
3. Valorar las nuevas tendencias de las tecnologías de la información relacionadas con las aplicaciones Web.
4. Seleccionar las herramientas adecuadas para la implementación del sistema.
5. Definir los procesos vinculados al campo de acción.
6. Realizar la planificación para la posterior implementación del sistema.
7. Realizar el diseño del sistema.
8. Implementar el sistema “Herramienta Web para la autonomía en el Aprendizaje y la Consulta de contenidos de Informática”.
9. Seleccionar y ubicar los contenidos que se ofrecerán a los cursistas con la aplicación.
10. Implementar el mapa del sitio como herramienta de navegabilidad.
11. Realizar pruebas al sistema.
12. Estructurar la investigación realizada en el formato requerido para obtener una buena documentación.

Después de realizar las tareas anteriores se espera:

1. Intercambio docente – cursistas constante, posibilitando un sistema de aprendizaje que estará en función de las capacidades de los cursistas.
2. Facilidad de un mayor servicio a más cantidad de usuarios en menos tiempo.
3. La novedad en el ámbito nacional y/o internacional que brinda el producto es la autonomía que tiene el usuario sobre la planificación del contenido que desea, quiera o necesita adquirir.

Para la realización del trabajo se tienen en cuenta algunos métodos tradicionales investigativos. A continuación se mencionarán cada uno de ellos y de qué forma se ponen de manifiesto en la investigación.

Los métodos teóricos aplicados en la investigación son los métodos histórico-lógico puesto que se realizó un estudio de las tecnologías que existen actualmente para poder realizar la selección de las que se van a utilizar de acuerdo a las características propias del sistema a desarrollar y el método de análisis y síntesis, pues se realiza un análisis de la bibliografía que se utilizó para el estudio del tema.

Los métodos empíricos son los que describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

El método empírico que se utiliza es el método de la observación, con la aplicación del mismo se puede conocer la realidad mediante la percepción directa de los objetos y fenómenos; a través de este método se pudo conocer la esencia de la problemática definida, lo que ayudó al planteamiento del problema científico, además de permitir conocer el proceso definido como objeto de estudio, lo cual influye a la hora de tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.

El trabajo está estructurado en 5 capítulos de los que se realiza una breve descripción a continuación:

El **Capítulo 1** contiene la fundamentación teórica del trabajo, abordando todos los elementos teóricos que sustentan el problema científico y los objetivos del trabajo, se describen aspectos importantes sobre el proceso de gestión y se realiza un análisis de las tendencias de las tecnologías y metodologías actuales en el desarrollo de software. Se fundamentan las tecnologías y herramientas

con las cuales se debe desarrollar el sistema, así como la metodología más adecuada para guiar el proceso.

En el **Capítulo 2** se describe una propuesta de la investigación y los procesos que están vinculados al campo de acción, se plantean los requisitos funcionales y no funcionales que debe tener el sistema para lograr óptimos resultados, además de hacer referencia a las personas vinculadas a éste.

En el **Capítulo 3** se detallan los artefactos generados en la fase de Exploración y Planificación, teniendo especial atención en las Historias de Usuario.

En el **Capítulo 4** se exponen los aspectos relacionados con el diseño del sistema. Se describen las tareas relacionadas a la fase de implementación y las pruebas de aceptación realizadas a solicitud del cliente.

En el **Capítulo 5** se lleva a cabo un estudio sobre la factibilidad del sistema, utilizando COCOMO II como método de estimación.

1

CAPÍTULO FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

En este capítulo se brinda un panorama general de los aspectos relacionados con los sistemas de gestión de contenidos como su definición y características, enfatizando en los sistemas de aprendizaje de contenidos informáticos. Se aborda el significado de la autonomía sobre la planificación de los contenidos en el proceso de aprendizaje. Además se pretende hacer referencia a cómo se encuentran en la actualidad las tecnologías para el desarrollo de aplicaciones Web, y el estudio de los lenguajes y herramientas que se utilizan en el ámbito internacional, permitiendo realizar una selección correcta que se adapte a las necesidades del sistema propuesto, lográndose identificar los lenguajes de programación a utilizar, el sistema gestor de base de datos, el sistema gestor de contenidos y las herramientas con las cuales se debe llevar adelante el desarrollo del sistema.

1.2 FUNDAMENTACIÓN DEL TEMA

1.2.1 *Procesos de Gestión de Contenidos*

Definiciones de Gestión de Contenidos

Definición 1

Una definición aceptada sobre gestión de contenidos es la relacionada con todos los procedimientos y procesos involucrados en la agregación, transformación, catalogación, agrupación, autorización, presentación y distribución de información útil para nuestros propósitos.

Definición 2

La gestión de contenidos puede vincularse a un planteamiento más amplio enfocado a la gestión global de los recursos de información de una institución o empresa mediante tecnologías Web (Internet e

Intranet). Este enfoque da a la tecnología un papel facilitado, y el mayor peso recae en los aspectos relacionados con la identificación de recursos de información internos y externos, gestión y tratamiento eficiente.

Definición 3

La gestión de contenidos en el ámbito de la gestión de publicaciones y documentos electrónicos, se refiere a la aplicación de una serie de técnicas y herramientas para la codificación, almacenamiento y distribución de publicaciones en formato digital.

En los casos anteriores se pueden encontrar puntos comunes. El más importante de ellos es la necesidad de utilizar tecnologías de la información y sistemas informáticos para el almacenamiento y distribución de información.

Características de los Sistemas de Gestión de Contenidos

Un Sistema de Gestión de Contenido debe ofrecer como mínimo, una aplicación que soporte los procesos de publicación, flujo de trabajo y repositorios de información. Un CMS (*Content Management System* o en español Sistema de Gestión de Contenido) se compone de varios subsistemas que interaccionan entre ellos:

- **Colección:** subsistema encargado de la creación y/o adquisición de información. Debe dar soporte a los procesos de creación de contenidos, soporte a flujos de trabajo, sindicación e integración de fuentes externas. Además, debe ofrecer soporte a procesos de conversión entre formatos diversos, y a la agregación de contenidos de fuentes diversas en estructuras específicas.
- **Gestión:** subsistema encargado de la gestión y control de los repositorios de información, de los grupos de usuarios, y de los procesos de soporte para los otros subsistemas. Se encarga de definir y controlar los flujos de trabajo que son utilizados por los otros subsistemas, y de la definición de parámetros para el funcionamiento del sistema.
- **Publicación:** subsistema encargado de la producción final de publicaciones o productos de información digital, de manera automática o casi automática. Utiliza un modelo basado en

plantillas (*templates*) y deberá ofrecer posibilidades de personalización para usuarios y la posibilidad de producir para diferentes tipos de plataformas y/o clientes.

1.2.2 Autonomía sobre los contenidos

La autonomía es aquella facultad que le permite al estudiante tomar decisiones que le conduzcan a regular su propio aprendizaje en función de una determinada meta y a un contexto o condiciones específicas de aprendizaje. Por tanto una persona autónoma es aquella cuyo sistema de autorregulación funciona de modo que le permite satisfacer exitosamente tanto las demandas internas como externas que se le plantean.

En la base de la definición de autonomía se halla la posibilidad del estudiante de aprender a aprender, que resulta de ser cada vez más consciente de su proceso de cognición, es decir, de la metacognición. La metacognición es un proceso que se refiere al conocimiento o conciencia que tiene la persona de sus propios procesos mentales (sobre cómo aprende) y al control del dominio cognitivo (sobre su forma de aprender). Ambos se orientan al servicio de una mejora del estudio personal que le conduzca a resultados satisfactorios de aprendizaje.

La autonomía sobre los contenidos brindará la posibilidad que el estudiante, sabiendo sus necesidades y las perspectivas que persigue, sea capaz de seleccionar los contenidos que le son de su interés y planificar cómo desea adquirirlos.

1.2.3 Sistemas vinculados al campo de acción.

Existen en el mundo disímiles sistemas que gestionan los contenidos de diversos temas. Muchos se encargan de la publicación de cursos, para que los interesados escojan según las expectativas, otros sólo brindan el tema y la capacitación es autodidacta ofreciendo sólo la posibilidad de presentarse al examen y así acreditar el curso. Dentro de estos sistemas podemos mencionar algunos:

- Pergamo PUCP (<http://pergamo.pucp.edu.pe/>): Es un servicio especializado en la publicación de diferentes cursos, revistas. Éste pone a disposición de la comunidad académica una plataforma informática conformada por hardware, software y ambientes de almacenamientos de datos, que facilitan el desarrollo académico del personal asociado a la institución. Se muestran

varios cursos referentes al periodismo digital, propiciando el conocimiento y la capacitación a todo aquel que se sienta atraído.

- <http://www.uvfajardo.sld.cu/> : Esta aplicación pertenece a la Universidad Virtual Cátedra Manuel Fajardo, se encarga de la publicación de cursos de temas como Redes, Idiomas y demás, que ayuda a consolidar conocimientos.
- <http://cursos.universia.net/> : Este sitio se encarga de la publicación de cursos referentes a Lenguajes y Sistemas Informáticos. Brinda la opción de aprender y está conectado con las de 985 Universidades a nivel mundial.
- En la UCI existe un sistema para la publicación de los contenidos (conferencias, clases prácticas y laboratorios) de las diferentes asignaturas: Entorno Virtual de Aprendizaje (<http://teleformacion/>). En sitio se imparten cursos asociados a todas las asignaturas del plan de estudio de pregrado, asignaturas optativas y demandadas por la producción, se imparten cursos de postgrado (cursos independientes, maestrías, especialidades y doctorados), y cursos para la capacitación y formación profesional.
- <http://aulacliic.es/> : Este sitio permite cursar diferentes asignaturas de temas informáticos. Oferta navegabilidad por niveles en los contenidos, además de ofertar la posibilidad de descargar bibliografías referentes a diversos temas.

Estos sistemas a pesar de tener gran utilidad no se han podido poner en práctica para cumplir el objetivo fundamental, pues no permiten la autonomía sobre la planificación de los contenidos de los cursos que se encuentran publicados.

Por esta razón se decide implementar un sistema Web que permita la autonomía total sobre los cursos que se encuentren publicados en el mismo, pues de esta manera se dotará al cursista de las capacidades suficientes para que pueda elaborar su propio cronograma de temáticas y así, según sus necesidades y expectativas, sólo recibirá el contenido de las materias que le son de su interés según lo que esté publicado.

1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES

1.3.1 *Tendencias y Desarrollo del Software libre*

Los expertos afirman que el software libre es hoy en día un recurso potente y maduro, capaz de dar soluciones prácticas y democráticas a las necesidades de la sociedad moderna. Bajo el lema “Innovación y Libertad”, que alude a la aplicación práctica de los sistemas de fuente abierta para conseguir mayores cotas de desarrollo y bienestar. Las principales personalidades en la materia coinciden en señalar que el software libre es clave en la educación digital y su universalización. Con su promoción se pretende contribuir a superar la denominada “brecha digital”, poniendo a disposición de los usuarios sistemas informáticos abiertos sin costo alguno de licencias y sin distinción por posibilidades económicas. Este movimiento fue iniciado por universitarios y otros expertos en respuesta al software comercial, por el que se pagan derechos y cuyo núcleo central no es accesible. El software libre es una realidad consolidada, una solución potente, robusta y segura para construir los sistemas de información de cualquier organización, y al mismo tiempo un motor de inclusión ciudadana, capaz de vencer las barreras sociales y económicas que sufre la revolución del conocimiento.

Se argumenta que: “El software libre, es un movimiento tecnológico que ha revolucionado la sociedad por sus implicaciones propias. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, y nuevas normas legales. Muchas empresas informáticas y compañías a nivel mundial, han apostado por el nuevo modelo en el uso del software libre. Este es el caso de IBM, la Administración del Espacio Aeronáutico Nacional (NASA) en los sistemas de misión crítica, el Banco del Estado de Río Grande del Sur en su red de cajeros electrónicos, el motor de búsqueda Google, el portal Yahoo, el Departamento de Energía de Estados Unidos, los gobiernos de la Unión Europea, Australia, China, Brasil y muchos otros, las cuales han confirmado la calidad de los programas que son desarrollados sobre software libre, su gran estabilidad, seguridad y confiabilidad. [2]

El software libre es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión”. Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. [1]

1.3.2 Lenguajes de programación para la Web y Servidores Web

Un lenguaje de programación es aquel que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Un servidor Web es un programa que implementa el protocolo HTTP (*hypertext transfer protocol*). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (*hypertext markup language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Sobre el servicio web clásico podemos disponer de aplicaciones web. Estas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o JavaScript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje JavaScript y Java, aunque pueden añadirse más lenguajes mediante el uso de plugins.
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

1.3.2.1 Servidores Web

Internet Information Services

IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT, luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (Servidor Web).

Este servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de *Active Server Pages* (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes como PHP o Perl.

Apache

Dentro de sus puntos fuertes se encuentran:

- Tiene interfaz de autenticación con todos los sistemas.
- Facilita la integración como "plugins" de lenguajes de programación de páginas Web dinámicas.
- Tiene integración en estándar del protocolo de seguridad SSL.
- Provee interfaz a todas las bases de datos.
- Servidor altamente configurable de diseño modular: Se pueden escribir módulos para realizar determinadas funciones lo que implica que haya gran cantidad de ellos disponibles para su utilización.

El Apache fue hecho para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Está ligado a la plataforma (Linux, Windows, UNIX) sobre la cual los individuos o instituciones pueden construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización. [2]

1.3.2.2 Lenguajes de programación y tecnologías en el servidor.

Perl

Es fácil de utilizar, y soporta diferentes paradigmas como programación estructurada y programación orientada a objetos. Una de las ventajas que presenta es que es un lenguaje extensible, ya que permite hacer llamadas a múltiples programas desarrollados en otros lenguajes de programación. En sus inicios solamente corría en la plataforma Unix, pero en la actualidad es un lenguaje multiplataforma, que puede correr en diferentes sistemas operativos, libres o propietarios.

PHP

PHP (*Hipertext Preprocessor*) es un lenguaje interpretado, usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. PHP contiene un alto nivel embebido en páginas HTML, este lenguaje ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en pdf hasta analizar código XML, y además soporta gran cantidad de bases de datos.

PHP es un producto de código abierto, debido a esto goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades. PHP es interpretado por el servidor (apache) generando un HTML con el resultado de sustituir las secuencias de instrucciones PHP por su salida. [3]

PHP en cada nueva versión soporta nuevas funcionalidades por lo que el mejor método para ver qué nuevas librerías incluye es cuando compilamos. Si dentro del directorio con las fuentes de PHP ejecutamos el comando "*configure help*" nos dará información de todas las posibles opciones que tiene PHP al ser compilado y, entre esta información, la de todos los módulos que podemos añadir a PHP si disponemos de las librerías adecuadas. Dentro de su manual se dispone también de una referencia a todas las funciones disponibles. [7]

PHP ha sido diseñado de forma muy modular y su sencillez está dada por el surgimiento de nuevas librerías. Toda esta funcionalidad está basada en librerías que en su mayor parte no han sido desarrolladas por el equipo de PHP. [5]

XML

XML en sus siglas en inglés por *Extensible Markup Language* (lenguaje de marcas extensible), no es realmente un lenguaje en particular, sino un protocolo de comunicación entre aplicaciones Web. Algunos de los lenguajes que usan XML son Java y PHP. Presenta grandes ventajas como: intercambio de información entre diferentes aplicaciones sin importar la plataforma que las soporten, permite visualizar la información a los clientes finales en celulares, puestos de trabajos, ordenadores de mano (PALM), entre otros.

Es conocido como el lenguaje Esperanto para la Web por sus potencialidades. Es muy fácil la migración de datos de una aplicación a otra en dicho formato.

1.3.2.3 Lenguajes de programación y técnicas de desarrollo Web en el cliente.

JavaScript

Es un lenguaje interpretado lo que significa que no necesita ser compilado para obtener resultados. Es basado en prototipo, donde las nuevas clases se generan clonando la clase base y extendiendo su funcionalidad [4]. Presenta como ventaja que el código es ejecutado en el cliente, sin tener necesidad de ir al servidor, ganando en que el tiempo de respuesta es sumamente rápido. Uno de los principales problemas es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del derecho de autor. El código script tiene capacidades limitadas por razones de seguridad, por lo cual no es posible hacer todo con JavaScript, sino que es necesario usarlo conjuntamente con HTML.

Ajax

Sus iniciales se corresponden con dos lenguajes de programación, JavaScript y XML que interactúan asincrónicamente (*Asynchronous JavaScript And XML*). Hay que destacar que no es un lenguaje de programación, sino la integración de varias tecnologías para acelerar la comunicación del lado del cliente con el servidor.

Se ejecuta en el cliente y mantiene una comunicación asíncrona con el servidor, facilitando la actualización de parte de la información, sin necesidad de recargar nuevamente toda la página. Ésto en primer lugar es eficiente, porque aumenta principalmente la interactividad y la velocidad de respuesta a las solicitudes del cliente. Presenta como ventaja que se puede utilizar con cualquier lenguaje de programación en el servidor.

Plataforma J2EE

A J2EE se le denomina plataforma porque proporciona especificaciones técnicas que describen el lenguaje, pero además, provee las herramientas escalables para implementar productos de software, y cuenta con las siguientes características:

- Portable: Se puede reutilizar el código una y otra vez.
- Escalable: Soporta el aumento de prestaciones en una aplicación Web sin tener que reescribir todo el código de nuevo, solamente añadiendo nuevos componentes J2EE.
- Altamente soportada: Existen empresas sólidas que respaldan el desarrollo de la plataforma.
- Segura: Permite y soporta una gran complejidad de implementaciones de seguridad.

Esta plataforma, contiene varios *frameworks* de trabajo, que facilitan el trabajo durante el desarrollo de las aplicaciones.

HTML

HTML es el acrónimo de *Hypertext Markup Language* (lenguaje de marcas hipertextuales) que fue creado en el año 1990 por Tim Berners-Lee y diseñado principalmente para mostrar información, animaciones en forma de hipertexto. Algunas ventajas que presenta, son la facilidad con que se pueden actualizar los contenidos y que permite utilizar estilos en formato CCS (hojas de estilos en cascada) en las páginas para una mayor facilidad en su modificación. En la actualidad, es el lenguaje que utilizan todos los navegadores para mostrar la información final.

1.3.2.4 Fundamentación del Servidor Web y los lenguajes de programación para la web a utilizar.

Como servidor Web se utiliza Apache (versión 2.4) por ser compatible con el CMS Drupal, pues se aplica este CMS para la administración del sistema. Además es un software libre de código abierto para plataformas como Windows. Dentro de sus ventajas se encuentran que es modular, multiplataforma, extensible, popular (fácil de conseguir ayuda/soporte) y gratuito.

Se utiliza lenguaje PHP como lenguaje para el servidor en su versión 5.2.3 debido a que:

- Es software libre. Se puede obtener en la web y su código está disponible bajo la licencia GPL. [6]
- PHP brinda una serie de ventajas que corroboran su empleo como lenguaje primario en la aplicación, las cuales son:

Está disponible para sistemas operativos diferentes. Puedes escribir código PHP en todos los sistemas operativos gratuitos de tipo UNIX, como Linux y FreeBSD, versiones comerciales de Unix, como Solaris

y IRIX o en las diferentes versiones de Microsoft Windows. Su código funcionará sin necesidad de aplicar ninguna modificación a los diferentes sistemas que ejecuta PHP. Es de fácil aprendizaje por lo que no se tardará mucho en utilizar PHP de manera productiva. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos, es multiplataforma y con capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad tales como Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; destaca su conectividad con MySQL lo cual permite la creación de Aplicaciones web muy robustas. Puede interactuar con los servidores web más populares ya que existe una versión CGI, módulo para Apache, e ISAPI. Drupal está implementado utilizando lenguaje PHP, lo que hace que sea más necesaria su utilización.

Como lenguaje de programación en el cliente se emplea HTML, debido a que es el lenguaje de marcado predominante para la producción de páginas web. Este lenguaje es utilizado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. También puede describir hasta cierto punto la apariencia de un documento.

Por estas razones son utilizados estos lenguajes, por sus características, sus perspectivas para un futuro inmediato, además de la necesidad que existe de la elaboración del sistema de forma rápida y eficiente.

1.3.3 Gestores de Base de Datos

Una base o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos. [4]

PostgreSQL

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzados del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle.

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, optimización de consultas y herencia. Tiene soporte para lenguajes procedurales internos (son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado y están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados), incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl o Python.

Oracle

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de *Relational Data Base Management System*), fabricado por Oracle Corporation. Surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre SGBD (Sistemas Gestores de Base de Datos) de George Koch. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total, la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, pero recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como las bases de datos mencionadas anteriormente. [4]

Para desarrollar en Oracle utilizamos PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos, también por norma general se suele utilizar SQL al crear un formulario. [3]

SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD), basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Revoluciona el concepto de Base de Datos para la empresa. Reúne en un único producto la potencia necesaria para cualquier aplicación empresarial crítica, junto con unas herramientas de gestión que reducen al mínimo el costo de propiedad. Con Microsoft SQL Server, la empresa tiene miles de soluciones disponibles, tendrá libertad de elección, ya que todas las aplicaciones de gestión del mercado corren sobre Microsoft SQL Server.

SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos. Los clientes están buscando soluciones para sus problemas de negocios. La mayoría de las "soluciones" de bases de datos solamente traen múltiples niveles de costos y complejidad. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas. [10]

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET. Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft. [10]

MySQL

MySQL es un sistema de gestión de base de datos multihilo, cualidad que le permite soportar una gran carga de forma muy eficiente y multiusuario con más de seis millones de instalaciones. El objetivo que ha perseguido la empresa open source MySQL AB consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad. Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para acceder a bases de datos en Internet. Desde enero de 2008 MySQL AB pertenece a Sun Microsystems.

MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso.

Este gestor de bases de datos es, probablemente, considerado el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta enorme aceptación se debe, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y contraseñas manteniendo un muy buen nivel de seguridad en los datos. [11]

1.3.3.1 Fundamentación del Gestor de BD a utilizar

Se seleccionó MySQL en su versión 5.0.45 debido a que cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento.
- Bajo costo.
- Facilidad de configuración y aprendizaje.
- Portabilidad.
- Accesibilidad a código fuente.
- Presenta un rendimiento alto debido a que es muy rápido.

MySQL está disponible de manera gratuita, bajo una licencia de código abierto en esta versión, o por un precio reducido en forma de licencia comercial si resultara necesario para su aplicación garantizando su bajo costo. Las bases de datos más modernas utilizan SQL. Si ha utilizado otros RDBMS, no debería tener problemas para adaptarse a este sistema. MySQL resulta además más sencillo de configurar que otros productos similares, como resultado esto traerá que sea fácil de aprender. MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes, así como bajo Microsoft Windows garantizando una gran portabilidad. Su código fuente como en el caso de PHP, se puede obtener y modificar. [11]

1.3.4 Metodologías de Desarrollo de Software

En los primeros momentos de la historia del software, no existían guías, procedimientos, ni metodologías para el desarrollo del mismo, ya que las demandas de los clientes eran muy pobres y el software no jugaba un papel importante en la sociedad. A medida que fue evolucionando el desarrollo de la informática y la exigencia de los clientes fue cada vez mayor, los especialistas se dieron a la tarea de desarrollar diferentes metodologías para organizar el trabajo, basándose en métodos comunes para el desarrollo.

Estas metodologías fueron diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones.

RUP

La metodología RUP es una metodología pesada, está basada en una notación gráfica, la cual permite especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Sus principales características son:

- Guiado por casos de uso: Los casos de uso son el instrumento para describir el comportamiento del software y extraer los casos de prueba con los que se valida el sistema.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño, describe la arquitectura del producto a desarrollar.
- Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones superiores.

RUP utiliza como lenguaje de modelado el *Unified Modelling Language* (UML) para describir todo el proceso. Se divide en ciclos de trabajo, teniendo un producto superior como resultado de cada ciclo. Éstos se componen en su interior por varias fases, en la cuales se llevan a cabo un conjunto de flujos para el desarrollo de todo el proyecto.

Extreme Programming (XP)

La metodología Programación Extrema o *Extreme Programming* (XP), es conocida como metodología ágil o ligera, orientada al cliente, y de iteraciones cortas. La base para el desarrollo del software que usa esta metodología son las llamadas *User Stories*, historias escritas por el cliente en las que describen escenarios sobre el funcionamiento del sistema y que no sólo están limitados los formularios de usuario, sino que también pueden describir modelos.

Estas *User Stories* junto a la arquitectura que se persigue, sirven de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente. Propone que en el equipo de desarrollo se necesita un representante constante del cliente que conozca al dedillo el negocio y que esté a disposición para cualquier duda o necesidades que los desarrolladores necesiten. El cliente se mantiene todo el tiempo informado paso por paso de las cosas que se están desarrollando, y a medida que se de la liberación de cualquier entregable se discutirá con el representante, y se repite la nueva iteración del software.

La programación del software siempre se define en pareja con el objetivo principal de lograr mayores resultados y los menores errores posibles, mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

FDD. Desarrollo Manejado por Rasgos.

Es un proceso que se puede considerar a medio camino entre RUP y XP, aunque es más similar al último. Está pensado para proyectos de corto tiempo de duración (menos de un año), un proceso iterativo con iteraciones cortas (aproximadamente 2 semanas).

Las iteraciones se deciden en base a las funcionalidades, que son pequeñas partes del software con significado para el cliente.

Un proyecto con FDD se divide en 5 fases:

1. Desarrollo de un modelo general.
2. Construcción de la lista de funcionalidades.
3. Plan de *releases* en base a las funcionalidades a implementar.
4. Diseñar en base a las funcionalidades.
5. Implementar en base a las funcionalidades.

El trabajo se realiza en grupo aunque siempre habrá un responsable (arquitecto jefe o jefe de programadores, en dependencia de la fase en la que se encuentre) con mayor experiencia. Las funcionalidades se dividen entre los subgrupos del equipo. FDD también define métricas para seguir el proceso de desarrollo de la aplicación. [12]

Puntos clave: Ligero, a medio camino entre el desarrollo y la organización, existe una jerarquía dentro del equipo, el código fuente tiene propietario, los equipos varían en función de la funcionalidad a implementar, el conocimiento de la aplicación se reparte a través del trabajo en equipo y revisiones. Documentación aceptable.

1.3.4.1 Fundamentación de la Metodología a utilizar

Se decidió utilizar XP debido a que se adapta en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. A continuación se exponen varias de las razones que llevaron al uso de esta metodología.

- El proyecto es pequeño: XP está concebida para ser utilizada dentro de proyectos pequeños.
- Los requisitos del sistema cambian frecuentemente: Con la aceptación de nuevos requerimientos, el sistema debe cambiar y ampliar sus funcionalidades de forma que sea capaz de adaptarse a cada nueva situación. Uno de los principios básicos de XP es que el cambio frecuente de los requerimientos es algo normal en el proceso de desarrollo. Esta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.
- El cliente forma parte del equipo de desarrollo: Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- El riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos: XP está diseñada a mitigar los riesgos en proyectos con estas características.
- Poca disponibilidad de personal: El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.
- Propiedad colectiva del código: XP plantea que todos los programadores pueden realizar cambios en cualquier parte del código en cualquier momento. Enfatiza la comunicación de los programadores a través del código, utilizando líneas directivas para la codificación que están bien establecidas.

1.3.5 Sistemas de Gestión de Contenidos

Un Sistema de gestión de contenidos permite la creación y administración de contenidos principalmente en páginas Web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

Drupal

Es un sistema de administración de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y 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. [8]

Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos y administración de listas de correo electrónico. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones. [8]

Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

Joomla

Es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. En Joomla se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexamiento Web, versiones imprimibles de páginas, flash con noticias, foros, *polls* (encuestas), calendarios, búsqueda en el sitio Web, e internacionalización del lenguaje. [9]

De forma similar a otros proyectos, Joomla mantiene dos versiones de la aplicación: una estable y otra Beta o en desarrollo. La versión estable es la que se considera para usuarios y a medida que aparecen errores se corrigen, esta versión se publica sin nuevas funcionalidades. La versión Beta, sí viene con nuevas funcionalidades y mejoras a las fallas reportadas en versiones anteriores, de igual forma se tiene que en este tipo de versión se reflejan las directivas del proyecto para usuarios avanzados y desarrolladores. [9]

El 22 de enero de 2008 se ha lanzado la versión 1.5 estable de Joomla, que incorpora notables mejoras en el área de seguridad, administración y cumplimiento con estándares W3C. Bien es cierto que para esta sección todavía no existen gran cantidad de componentes, módulos y *plugins* (mambots para Joomla 1.0. X). Por ello, existe la posibilidad de activar un *plugin* incluido por defecto que mejora considerablemente (no por completo) la compatibilidad con la mayoría de ampliaciones. [15]

Apache Lenya

Es un CMS de código libre escrito en Java, manejando el formato de representación XML. Apache Lenya está basado en Apache Cocon, es una herramienta de manejo suficientemente sencilla, lo que permite que la administración se realice sin la necesidad de tener conocimientos técnicos. Presenta las siguientes características:

- Control de flujo de aprobación (*workflow*) permite aceptar, rechazar, publicar o desactivar.
- Control de versiones.
- Auditoría de los cambios (historial).
- Planificador horario para la publicación de contenidos.
- Previsualización de producción y desarrollo.
- Organización de los contenidos por secciones y subsecciones.
- Posibilidad de crear, modificar y/o eliminar secciones y contenidos del Website.
- Separación limpia entre contenido y presentación con XML y plantillas XSLT.
- Gestión de usuarios y grupos.

Desventajas de Apache Lenya:

- Dificultad de aprendizaje.
- Escasos componentes desarrollados.
- Comunidad pobre.
- Es difícil encontrar un *hosting* que soporte Lenya, se debe tener un servidor propio.

1.3.5.1 Fundamentación del Gestor de Contenidos a utilizar

Se selecciona Drupal en su versión 5.6 debido a las facilidades que brinda, pues es utilizado en diversos lugares del mundo. Drupal no ha dejado de perfeccionarse, hace ya 7 años de su existencia, y tiene más de 300 módulos hechos por terceras partes, más de 55 000 sitios Web que funcionan soportados en este CMS. Una comunidad tan grande, con tantos temas que explorar, con tanto que programar no se detiene, los errores en Drupal se corrigen rápidamente gracias a la colaboración de todos; la publicidad generada por el proyecto y por sus usuarios es excelente.

- *A nivel nacional:*

En Cuba se capacitan a futuros ingenieros, profesionales y técnicos para aprender a trabajar con el CMS Drupal y actualmente existen varios sitios Web desarrollados en él, los cuales tienen gran importancia debido a que a través de ellos se pueden intercambiar foros, debates, comentarios, artículos, por ejemplo, el Instituto de Información Científica y Tecnológica (IDICT) utiliza el CMS Drupal para el Servicio de Publicación Web, que tiene como objetivo que los integrantes de este grupo tengan un lugar donde publicar sus trabajos relacionados con el proyecto que desarrollan.

- *En la Universidad de las Ciencias Informáticas:*

En la Universidad de las Ciencias Informáticas se utiliza el Sistema de Gestión de Contenido Drupal para migrar la intranet actual, pues es seguro y estable, algunas personas tienen conocimiento acerca del CMS en la universidad y han trabajado con buenos resultados, además existen en Drupal sitios que la universidad ha realizado para proyectos importantes en la República Bolivariana de Venezuela.

También se han creado con el uso de este CMS las comunidades de C++ y PHP, así como la Intranet versión 1.5.

1.4 CONCLUSIONES

Con el estudio de los sistemas de gestión de contenidos y las búsquedas realizadas con el objetivo de encontrar sistemas similares, se determinaron las características propias del sistema que se va a desarrollar, llegando a la conclusión que no existen sistemas que brinden la posibilidad de ser autónomos en el aprendizaje del contenido y en su planificación. Esto proporciona la medida de las ventajas que se darán a los usuarios que interactúen con el sistema a implementar. También se realizó un estudio de las tecnologías que actualmente son utilizadas para la implementación de sistemas semejantes al propuesto, seleccionando las que serán utilizadas a lo largo del desarrollo del sistema, fundamentándose las elecciones del lenguaje de programación, el sistema gestor de base de datos, la metodología de desarrollo y el sistema de administración de contenidos.

2

CAPÍTULO CARACTERÍSTICAS DEL SISTEMA

2.1 INTRODUCCIÓN

El presente capítulo tiene como objetivo hacer una valoración de las características principales del sistema a desarrollar, prestando gran atención a la situación problemática que dio origen al mismo. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que serán objeto de automatización. Por último se presentará una propuesta del software a implementar, especificando detalladamente los requerimientos funcionales y no funcionales.

2.2 DESCRIPCIÓN DE LOS PROCESOS VINCULADOS AL CAMPO DE ACCIÓN

Actualmente el proceso de desarrollo para la matrícula en un curso en la universidad incluye una serie de flujos y acciones que rigen el comportamiento del mismo, estos flujos permiten la estandarización del entorno de trabajo para la matrícula de cada uno de los cursos que son ofertados por parte de la dirección de postgrado de manera general.

2.2.1 Flujo actual del proceso

El desarrollo de este proceso se realiza de la siguiente manera: Se inscribe el curso por parte del docente encargado en la dirección de postgrado, presentando el programa y la planificación de las actividades, las cuales son valoradas para su aprobación y aceptación por parte de la dirección. De esta manera se asientan los datos correspondientes del profesor que lo impartirá y posteriormente se emite la convocatoria del curso para la matrícula por parte de los interesados.

De esta forma se inicia el proceso de matrícula por los estudiantes donde deben presentar la planilla de solicitud. Dichos datos son verificados por la Teca diez.

Al concluir el proceso se circula a los interesados la aceptación de su registro y se le hace llegar el listado de los mismos al docente que lo impartirá; al finalizar el curso, el profesor entrega los resultados alcanzados por los estudiantes y hace una valoración de los que por un motivo u otro no lo vencieron. A partir de este momento se emite una certificación acreditativa a los estudiantes por haber cursado y aprobado los contenidos y al profesor por haberlo impartido.

2.2.1 Objeto de automatización

Durante el ciclo de desarrollo de esta actividad existen varios procesos que deben ser automatizados, puesto que su ejecución de forma manual resulta tediosa y propensa a errores, además de consumir una valiosa porción de tiempo de los desarrolladores. Muchos de estos procesos son de vital importancia para lograr un nivel de calidad óptimo en los productos finales.

Se automatizarán los procesos de publicación y matrícula de los cursos mediante una aplicación Web, la cual mostrará los bloques de contenidos que serán ofertados para la capacitación de todas las personas interesadas en el aprendizaje de temas informáticos. Este sistema brindará la posibilidad de estudiar de forma autónoma, confeccionando el paquete de contenidos que son del interés individual, proporcionando en gran medida ahorro de tiempo, pues se brindará la posibilidad de escoger dentro del curso sólo aquel o aquellos bloques de contenidos que le son necesarios o que desea.

En el sistema se manejará información referente a los cursistas, docentes y cursos de manera general, se facilitará la planilla de inscripción para cursos de postgrado y el administrador del sistema contará con la responsabilidad para definir entre alumnos y profesores.

2.3 PROPUESTA DEL SISTEMA

El presente trabajo propone la implementación de un sistema que provea las mismas funcionalidades que el flujo de trabajo actual. Dicho sistema debe permitir la gestión (crear, modificar y eliminar) de una cuenta de suscripción al sistema, recogiendo los datos que son de interés para los administradores del sistema. Debe permitir la publicación de convocatorias de cursos para su posterior divulgación,

además de brindar la posibilidad de acceder a las diferentes planificaciones confeccionadas para cada uno de los bloques de contenidos. Los estudiantes tendrán la posibilidad de matricular en los diferentes bloques de contenidos ofertados, y se le otorgarán los permisos pertinentes para acceder a los materiales publicados referentes a estos contenidos. Los profesores por su parte serán los encargados de gestionar la publicación de los diferentes libros y/o tutoriales para propiciar la preparación de los estudiantes, así como de los exámenes que le proporcionarán a los cursistas. El administrador del sistema será el encargado de gestionar toda la información referente al sistema y dará soporte en cada una de las instancias necesitadas.

El sistema tendrá habilitado un fórum que permitirá el intercambio entre docentes y alumnos sobre las diferentes temáticas abordadas. Además de poseer una encuesta que permitirá a los desarrollares medir la utilidad del sistema para los usuarios.

El sistema será una aplicación Web administrada con el CMS Drupal, y cada una de sus funcionalidades será implementada como parte de un módulo Drupal de propósito específico para la posterior integración de todos en lo que se denomina “Herramienta Web para la autonomía en el Aprendizaje y la Consulta de contenidos de informática”.

2.3.1 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 2.1 Personas relacionadas con el sistema

Personas relacionadas con el sistema.	Justificación
Invitado	Es la persona que navega por el sistema sin haberse creado una cuenta aún, fluye dentro de éste sin privilegios. Tiene la posibilidad de crearse una cuenta, ver su perfil, navegar sobre las diferentes opciones para irse relacionando con el mismo, así como indagar y descargar, en caso que desee, las diferentes bibliografías que se muestran de manera general.
Estudiante	Es la persona que se encuentra autenticada en el sitio cumpliendo este rol. Se le brinda la posibilidad de matricular desde 1 hasta 5 de los bloques de contenidos que se ofrecen en los diferentes cursos que se ofertan, además tendrá acceso a diferentes módulos debido al rol que cumple dentro de la aplicación.
Profesor	Es la persona encargada de atender los diferentes bloques de contenidos que se proponen en el sistema. Además tendrá la responsabilidad de crear las matrículas de los diferentes cursos, publicar los materiales referentes a los contenidos.
Administrador	Es la persona facultada para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación.

2.3.2 Requerimientos funcionales del sistema

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran a través de requerimientos funcionales las prestaciones que el sistema será capaz de brindar. Dentro de ellas se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema. De acuerdo a los objetivos planteados, el sistema debe ser capaz de:

R1. Crear cuenta en el sistema

R1.1. Mostrar el formulario de creación de la cuenta.

R1.2. Permitir entrar los datos importantes para la cuenta: nombre de usuario, dirección de correo electrónico, contraseña, conformación de contraseña.

R1.3. Permitir entrar los datos de identificación personal: nombre y apellidos, sexo, dirección particular.

R1.4. Validar los datos introducidos por el usuario.

R1.4. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R2. Ver perfil de usuario

R2.1. Mostrar el formulario con datos del perfil del usuario.

R3. Modificar cuenta del usuario

R3.1. Mostrar el formulario con datos del perfil del usuario.

R3.2. Permitir al usuario modificar los datos que prefiera.

R3.3. Validar los datos introducidos por el usuario.

R3.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R4. Eliminar cuenta de usuario

R4.1. Mostrar el formulario con datos del perfil del usuario.

R4.2. Permitir la posibilidad al usuario de eliminar su cuenta.

R5. Autenticar

R5.1. Mostrar el formulario de autenticación de usuario.

R5.2. Pedir nombre de usuario y contraseña.

R5.3. Validar los datos introducidos por el usuario.

R5.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R5.4. Verificar qué rol cumple dentro del sistema para asignar permisos.

R5.5. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos asignados una vez autenticado correctamente.

R6. Matricular en cursos

R6.1. Mostrar el formulario que permita la matrícula en los diferentes bloques.

R6.2. Permitir al usuario marcar los bloques de contenidos de los cursos (máximo 5) que sean de su interés.

R6.3. Validar los datos introducidos por el usuario.

R6.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R7. Publicar materiales

R7.1. Mostrar el formulario para publicar los materiales.

R7.2. Permitir al usuario entrar los datos referentes al nombre del material, un resumen de lo que trata y adjuntar el archivo pertinente.

R7.3. Validar los datos introducidos por el usuario.

R7.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R8. Mostrar materiales

R8.1. Mostrar formulario con los materiales a los cuales tiene acceso según su matrícula.

R8.2. Mostrar un vínculo que permita descargar el archivo adjunto.

R9. Eliminar materiales

R9.1. Mostrar el formulario de los diferentes materiales publicados.

R9.2. Permitir al usuario marcar el/los materiales que desea eliminar.

R9.3. Validar los datos introducidos por el usuario.

R9.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R10. Publicar bibliografía

R10.1. Mostrar el formulario para publicar las diferentes bibliografías.

R10.2. Permitir al usuario entrar los datos referentes a: título, resumen, categoría, ISBN y adjuntar el archivo pertinente.

R10.3. Validar los datos introducidos por el usuario.

R10.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R11. Mostrar bibliografía

R11.1. Mostrar formulario con las bibliografías publicadas.

R11.2. Mostrar un vínculo que permita descargar el documento adjunto.

R12. Eliminar bibliografía

R12.1. Mostrar el formulario de administración de los contenidos.

R12.2. Permitir al usuario buscar el tipo de contenido que desea modificar.

R12.3. Validar los datos introducidos por el usuario.

R12.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R12.4. Permitir al usuario marcar la bibliografía que desea eliminar.

R13. Publicar programas de cursos

R13.1. Mostrar el formulario para publicar los diferentes niveles de los contenidos de los cursos que serán ofertados.

R13.2. Permitir al usuario entrar los datos referentes a: nombre del curso, un pequeño resumen y adjuntar el archivo correspondiente.

R13.3. Validar los datos introducidos por el usuario.

R13.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R14. Mostrar programas de cursos

R14.1. Mostrar formulario con los datos de los niveles de los contenidos de los cursos que serán ofertados.

R14.2. Mostrar un vínculo que permita descargar el documento adjunto.

R15. Eliminar programas de cursos

R15.1. Mostrar el formulario de los diferentes cursos que han sido publicados.

R15.2. Permitir al usuario marcar el curso que desea eliminar.

R15.3. Validar los datos introducidos por el usuario.

R15.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R16. Buscar información

R16.1. Mostrar el formulario de búsqueda.

R16.2. Permitir al usuario teclear la palabra o frase que desee buscar dentro del sitio.

R16.3. Validar los datos introducidos por el usuario.

R16.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R16.4. Mostrar un vínculo al contenido encontrado correspondiente al término de búsqueda.

R17. Ver listado de estudiantes

R17.1. Mostrar el formulario de búsqueda de estudiantes.

R17.2. Permitir al usuario marcar el curso del cual quiere conocer los estudiantes matriculados.

R17.3. Validar los datos introducidos por el usuario.

R17.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R17.4. Mostrar un listado con el nombre de los estudiantes que pertenecen a dicho curso.

R18. Ver listado de profesores

R18.1. Mostrar el formulario de búsqueda de profesores.

R18.2. Permitir al usuario marcar el curso del cual quiere conocer los profesores encargados de impartirlo.

R18.3. Validar los datos introducidos por el usuario.

R18.3. a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.

R18.4. Mostrar un listado con los nombres de los profesores y la foto en caso de tenerla, que pertenecen a dicho curso.

2.3.3 Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Diseño e implementación

- Aplicación Web escrita sobre el lenguaje de programación PHP 5.2.3.
- Usar el gestor de Base de Datos MySQL 5.0.45.
- Utilizar como servidor web Apache 2.4.
- Desarrollar bajo CMS Drupal 5.6.
- Utilizar como metodología de desarrollo XP.
- Correr sobre el sistema operativo Windows XP.

Apariencia o interfaz externa

- Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

- No debe tener animaciones ni imágenes pesadas que obstaculicen la rapidez de las transiciones del ancho de banda.
- Diseño perfectamente encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.
- El diseño garantizará la selección de un esquema de colores a la vez atractivo pero que no canse, conjugando equilibrio entre los mismos y un contraste suficiente pero no agresivo.
- El diseño debe permitir el uso de los colores corporativos. Para la realización del diseño se tendrá en cuenta la identidad corporativa de SISLIBAUT, para lo cual se utilizarán los colores que se muestran a continuación:

Colores	Codificación hexadecimal
Azul	# 5255FF
Azul	# E7EFFF

Usabilidad

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Funcionalidad

- Capacidad de búsqueda con un tiempo menor de 5 segundos.
- Mínima cantidad de páginas para ejecutar todas las funciones posibles, es decir, agrupar funciones afines en las mismas páginas.

Rendimiento

- La eficiencia de esta aplicación debe ser óptima en cuanto a la velocidad de procesamiento, disponibilidad, tiempo de respuesta y aprovechamiento de los recursos, entre otros.

Seguridad

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Proteger la información manejada por el sistema de accesos no autorizados.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

- Proteger la información manejada por el sistema contra la corrupción de ficheros o estados inconsistentes.
- Verificación sobre acciones irreversibles (eliminaciones).

Ayuda y documentación en línea.

- El usuario recibirá una ayuda que le servirá para orientarse en lo que va a ser, así como toda la documentación que necesite para lograr aprobar su curso de manera satisfactoria, por ejemplo: libros, ficheros complementarios, tutoriales, presentaciones, entre otros.

Portabilidad

- Que corra en cualquier plataforma, es decir, que sea multiplataforma.

2.4 CONCLUSIONES

En este capítulo se inicia el desarrollo de la propuesta de solución que se desea implementar, tras el análisis de los flujos de trabajos actuales descritos por el cliente. Se obtuvo un listado de funcionalidades que debe tener el sistema, expresados en los requerimientos funcionales. Partiendo de este punto, base de todo proceso de desarrollo, se puede comenzar con la construcción de la propuesta, velando por el cumplimiento de todos los requerimientos y funcionalidades consideradas.

3

CAPÍTULO EXPLORACIÓN Y PLANIFICACIÓN

3.1 INTRODUCCIÓN

En este capítulo se hace alusión a las fases de exploración y planificación, propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone. Se exponen además los artefactos generados durante el transcurso de las mismas.

3.2 FASE DE EXPLORACIÓN

La metodología de desarrollo *Extreme Programming* comienza con su fase de exploración. Durante esta etapa se realiza el proceso de identificación de las historias de usuario (UH, del inglés *User Histories*), así como la familiarización de los equipos de trabajo con las tecnologías y herramientas seleccionadas para la construcción del sistema.

3.2.1 Historias de usuario

Las historias de usuarios son la forma en que se especifican en XP los requisitos del sistema. Éstas se redactan desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido que ellas abarcan debe ser concreto y sencillo. Durante este proceso se identifican 14 historias de usuarios las cuales se detallan a continuación.

Tabla 3.1 HU Gestionar cuenta de usuario en el sistema

Historia de Usuario	
Número: 1	Nombre: Gestionar cuenta de usuario
Usuario: Invitado	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar las cuentas del usuario en el sistema. El usuario al entrar al sistema se le brinda la posibilidad de crear su cuenta para la posterior asignación de permisos dentro del sistema. Se brinda la posibilidad de editar su perfil después de llenados los datos e igualmente eliminar la cuenta.	
Observaciones: Se hace referencia a los requisitos R1, R3 y R4.	

Tabla 3.2 HU Mostrar perfil de usuario

Historia de Usuario	
Número: 2	Nombre: Mostrar perfil de usuario
Usuario: Invitado	
Prioridad en negocio: Bajo	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se brinda la posibilidad de ver los datos que el usuario ha llenado en los campos de creación de su cuenta. Ésto se mostrará como una planilla que identificará al usuario dentro del sistema.	
Observaciones: Se hace referencia al requisito R2	

Tabla 3.3 HU Autenticar

Historia de Usuario	
Número: 3	Nombre: Autenticar
Usuario: Invitado	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que la persona que acceda al sistema introduzca sus datos (usuario y contraseña) con la finalidad de verificar y otorgarle los permisos según el rol que cumpla dentro de la aplicación.	
Observaciones: Se hace referencia al requisito R5	

Tabla 3.4 HU Crear materiales

Historia de Usuario	
Número: 4	Nombre: Crear materiales
Usuario: Profesor	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Se brinda la posibilidad al profesor de publicar en el sistema los materiales que le servirán al estudiante de guía para su autoaprendizaje.	
Observaciones: Se hace referencia al requisito R7.	

Tabla 3.5 HU Mostrar materiales

Historia de Usuario	
Número: 5	Nombre: Mostrar materiales
Usuario: Estudiante o Profesor	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que el estudiante o el profesor puedan descargar los materiales pertinentes a los contenidos de los cuales es matrícula. No siendo así para aquellos que no lo son.	
Observaciones: Se hace referencia al requisito R8	

Tabla 3.6 HU Eliminar materiales

Historia de Usuario	
Número: 6	Nombre: Eliminar materiales
Usuario: Profesor	
Prioridad en negocio: Media	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que el profesor después de un tiempo prudente elimine los materiales referentes a los cursos que él atiende.	
Observaciones: Se hace referencia al requisito R9	

Tabla 3.7 HU Gestionar bibliografía

Historia de Usuario	
Número: 7	Nombre: Gestionar bibliografía
Usuario: Profesor	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 3	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear y eliminar bibliografías. El profesor tiene la posibilidad de añadir y eliminar a su vez bibliografías de carácter general para todas las personas que accedan al sistema, pueden ser de cualquier tema relacionado con la informática y que de una forma u otra atributen a la preparación de las personas.	
Observaciones: Se hace referencia a los requisitos R10 y R12	

Tabla 3.8 HU Mostrar bibliografía

Historia de Usuario	
Número: 8	Nombre: Mostrar bibliografía
Usuario: Invitado	
Prioridad en negocio: Bajo	Riesgo en Desarrollo: Media
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver y descargar la bibliografía que se encuentra publicada en términos generales para todos los visitantes.	
Observaciones: Se hace referencia al requisito R11	

Tabla 3.9 HU Gestionar programas de cursos

Historia de Usuario	
Número: 9	Nombre: Gestionar programas de cursos
Usuario: Profesor	
Prioridad en negocio: Bajo	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se realizan las acciones de crear, modificar y eliminar planificaciones. El profesor tiene la posibilidad de añadir, modificar y eliminar a su vez los programas de cursos que son publicados para la guía de los estudiantes, pues ésto ayuda a que el usuario sepa cuál bloque de contenidos de qué nivel de los diferentes cursos le es más factible según sus necesidades.	
Observaciones: Se hace referencia a los requisitos R13 y R15	

Tabla 3.10 HU Mostrar programas de cursos

Historia de Usuario	
Número: 10	Nombre: Mostrar programas de cursos
Usuario: Invitado	
Prioridad en negocio: Alto	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver y descargar la planificación de los bloques de contenidos según los niveles de los diferentes cursos que se encuentran publicados, para que así pueda escoger según sus expectativas.	
Observaciones: Se hace referencia al requisito R14	

Tabla 3.11 HU Matricular en cursos

Historia de Usuario	
Número: 11	Nombre: Matricular en cursos
Usuario: Estudiante	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 3	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que el estudiante opte desde 1 hasta 5 de los bloques de contenidos de los diferentes niveles de los cursos que son ofertados. A partir de este momento será matrícula de dichos bloques, conformándose de esa manera su programa de cursos específico pudiendo acceder a la bibliografía y exámenes referentes a ellos.	
Observaciones: Se hace referencia al requisito R6	

Tabla 3.12 HU Buscar Información

Historia de Usuario	
Número: 12	Nombre: Buscar información
Usuario: Invitado	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de buscar contenidos que se encuentran dentro del sistema.	
Observaciones: Se hace referencia al requisito R16	

Tabla 3.13 HU Mostrar Listado de estudiantes

Historia de Usuario	
Número: 13	Nombre: Mostrar listado de estudiantes
Usuario: Profesores	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se brinda la posibilidad de que los profesores accedan al listado de los estudiantes que se encuentran matriculados en un curso en específico.	
Observaciones: Se hace referencia al requisito R17	

Tabla 3.14 HU Mostrar Listado de profesores

Historia de Usuario	
Número: 14	Nombre: Mostrar listado de profesores
Usuario: Invitado	
Prioridad en negocio: Medio	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que los usuarios que accedan al sistema en dependencia de un determinado curso vean los profesores que están asignados para atenderlo e impartirlo.	
Observaciones: Se hace referencia al requisito R18	

3.3 FASE DE PLANIFICACIÓN

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Ésto se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario, por ejemplo, las pruebas unitarias, la integración y refactorización del código, y la preparación y ejecución de las pruebas de aceptación.

3.3.1 Estimación de esfuerzos por historias de usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las historias de usuario identificadas, llegando a los resultados que se muestran a continuación:

Tabla 3.15 Estimación de esfuerzos por HU

Historia de Usuario	Puntos de Estimación
Gestionar cuenta de usuario	1
Mostrar perfil de usuario	1
Autenticar	1
Crear materiales	2
Mostrar materiales	1
Eliminar materiales	1
Gestionar bibliografía	3
Mostrar bibliografía	1
Gestionar programas de cursos	1
Mostrar programas de cursos	1
Matricular en cursos	3
Buscar información	1
Mostrar listado de estudiantes	1
Mostrar listado de profesores	1

3.3.2 Plan de iteraciones

Después de ser descritas e identificadas las historias de usuario y estimado el esfuerzo propuesto para la realización de cada una de ellas, se procede a la planificación de la etapa de implementación del sistema. Este plan especifica exactamente cuáles historias de usuario serán implementadas para cada iteración del sistema y las posibles fechas para estas liberaciones.

En base a lo antes mencionado se decide realizar el sistema en tres iteraciones, las cuales se detallan a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario de mayor prioridad. Al finalizar se contará con las funcionalidades descritas en las historias de usuario 1, 3, 4, 5 y 7 las cuales

hacen alusión a la gestión de cuentas del usuario en el sistema, la autenticación del mismo como parte de la seguridad de la aplicación, publicar y mostrar las conferencias de los diferentes cursos y gestionar la bibliografía general para todos los usuarios, respectivamente. Además se tendrá la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

Iteración 2

El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad alta, además de otras con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las historias de usuario 6, 9, 10, 11 y 14, en las cuales se hace mención a la posibilidad de eliminar los materiales publicados, gestionar (insertar y eliminar) los programas de cursos, mostrar las diferentes planificaciones de los cursos, matricular en los bloques de contenidos de los niveles de cursos propuestos y mostrar el listado de los profesores encargados de impartir el curso, respectivamente. La versión de prueba referente a esta iteración junto a las implementaciones anteriores, serán mostradas al cliente con el objetivo de realizar cambios en base a la opinión del mismo.

Iteración 3

En esta iteración serán implementadas las funcionalidades de prioridad baja. Estas funciones tienen el propósito de brindar al cliente comodidad en la gestión de otras tareas asociadas a las de alta prioridad, entre ellas tenemos: mostrar el perfil del usuario con las características que inicialmente insertó en la página de creación de la cuenta, permitir la descarga de bibliografías para todos los usuarios de manera general aunque no se encuentren registrados en el sistema, permitir la búsqueda de manera rápida de información en la aplicación, y mostrar el listado de los estudiantes que constituyen matrícula según el curso. Estas funciones están descritas en las historias de usuario 2, 8, 12 y 13 respectivamente. Como resultado de esta iteración se tendrá la versión 1.0 del producto final, adicionando lo concerniente a la interfaz web. A partir de este momento el sistema será puesto a prueba por un período de tiempo para evaluar el desempeño del mismo.

3.3.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando la Metodología XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de las mismas.

Tabla 3.16 Plan de duración de las iteraciones (Equipo de desarrollo)

Iteraciones	Orden de las Historias de usuario a implementar	Duración total de las iteraciones
Iteración 1	<ol style="list-style-type: none"> 1- Gestionar cuenta de usuario. 2- Autenticar. 3- Crear materiales. 4- Mostar materiales. 5- Gestionar bibliografías. 	8 semanas
Iteración 2	<ol style="list-style-type: none"> 1- Eliminar materiales. 2- Gestionar los programas de cursos. 3- Mostrar los programas de cursos. 4- Matricular cursos. 5- Mostrar listados de profesores. 	7 semanas
Iteración 3	<ol style="list-style-type: none"> 1- Mostrar perfil de usuario. 2- Mostrar bibliografía. 3- Buscar información. 4- Mostrar lista de estudiantes. 	4 semanas

3.3.4 Plan de entregas

A continuación se presenta el plan de entregas elaborado para la fase de implementación. Para facilitar la elaboración de dicho plan se acoplaron las funcionalidades referentes a un mismo tema en módulos, quedando éstos de la siguiente manera:

Tabla 3.17 Módulos y HU abarcadas

Módulos	Historias de usuario que abarca
user	1- Gestionar cuenta de usuario 2- Autenticar
perfil	1- Mostrar perfil de usuario
curso	1- Crear materiales 2- Mostrar materiales 3- Eliminar materiales
bibliografía	1- Gestionar bibliografía 2- Mostrar bibliografía 3- Gestionar programas de cursos 4- Mostrar programas de cursos
matrícula	1- Matricular cursos 2- Mostrar listado de estudiantes 3- Mostrar listado de profesores
search	1- Buscar información

Como producto del plan de entregas se harán *releases* al sistema en las fechas que se indican a continuación:

Tabla 3.18 Plan de duración entrega

Módulos	Final 1ra iteración 3ra semana de febrero	Final 2da iteración 2da semana de abril	Final 3ra iteración 2da semana de mayo
user	1.0	Finalizado	Finalizado
perfil			1.0
curso	0.1	1.0	Finalizado
bibliografía	0.1	0.2	1.0
matrícula		0.1	1.0
search			1.0

3.4 CONCLUSIONES

Durante el desarrollo de este capítulo se hace referencia a todo lo concerniente a la fase de Exploración y Planificación del sistema, haciendo una descripción de cada uno de los artefactos generados en el transcurso del mismo. Se asume una implementación por etapas la cual fue concebida y debidamente detallada. Esta fase se considera como un permanente diálogo entre las partes: *lo deseable* y *lo posible*, determinándose:

- **Ámbito:** ¿Qué es lo que el software debe resolver para que genere valor?
- **Prioridad:** ¿Qué debe ser hecho en primer lugar?
- **Composición de versiones:** ¿Cuánto es necesario hacer para saber si el negocio va mejor con software que sin él?
- **Fecha de versiones:** ¿Cuáles son las fechas que en la presencia del software o parte del mismo pudiesen marcar la diferencia?
- **Estimaciones:** ¿Cuánto tiempo lleva implementar una característica?
- **Procesos:** ¿Cómo se organiza el trabajo y el equipo?
- **Programación detallada:** Dentro de una iteración ¿Qué problemas se resolverán primero?

4

CAPÍTULO IMPLEMENTACIÓN Y PRUEBAS

4.1 INTRODUCCIÓN

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

4.2 DISEÑO DEL SISTEMA

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración) [15]. No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante [16].

Con el objetivo de la comprensión de este trabajo se hace una breve explicación del funcionamiento de Drupal como plataforma de publicación.

El CMS Drupal contiene un tipo de contenido genérico llamado *Node* que puede ser extendido por cualquier desarrollador, este tipo de contenido tiene las propiedades básicas para cualquier publicación como son título, autor, fecha de creación y contenido, además Drupal proporciona los mecanismos para la creación, edición y publicación de este tipo de contenido. Cualquier desarrollador que desee una publicación personalizada sólo debe extender este tipo de contenido y de esta manera aprovechar sus propiedades.

Drupal provee al desarrollador de un potente sistema de seguridad basado en roles, el mismo Sistema de Gestión de Contenidos se encarga de la creación de usuarios y roles, así como del control de accesos a los diferentes módulos según los permisos definidos por el administrador. El desarrollador se limita a exportar en su módulo los tipos de acceso que desea definir, el resto lo maneja Drupal, es decir, cuando un usuario trata de acceder a un módulo la plataforma chequea que el usuario autenticado tenga acceso al módulo en cuestión.

4.2.1 Módulos de Drupal

Los módulos son extensiones para Drupal que amplían las funcionalidades del núcleo. Un módulo es la unión de varias funciones que se juntan en Drupal y ayudan a ofrecerle mayor funcionalidad a la Web. Un módulo para Drupal consta de uno o más ficheros, el fichero principal con extensión *.module* debe implementar una interfaz definida por el propio Drupal. Básicamente existen dos tipos de módulos: los módulos de contenido, son los que definen un nuevo tipo de contenido personalizado y la funcionalidad para su creación, edición y publicación y los módulos funcionales, éstos tienen disímiles propósito dependiendo del objetivo con el que se desarrolla. La tarea de estas funciones es actuar como enganche, al ser llamadas por Drupal a la hora de construir una página Web y gestionar el contenido.

El sistema de bloques de Drupal consiste en una serie de bloques (ya sea definidos por un módulo o en la misma interfaz de Drupal) que se activan o desactivan para ser mostrados en las áreas de menú de la plataforma.

A continuación se utilizará el término “paquete” para referirse a módulos del Sistema de Gestión de Contenido en cuestión.

Paquetes de Drupal:

1. Paquete 1: Drupal
 - a. Sub paquete 1.1: Modules
 - b. Sub paquete 1.2: Includes
 - c. Sub paquete 1.3: Themes

Los paquetes desarrollados son:

- a. Sub paquete 1.1.1: bibliografía

- b. Sub paquete 1.1.2: cursos
- c. Sub paquete 1.1.3: matrícula

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

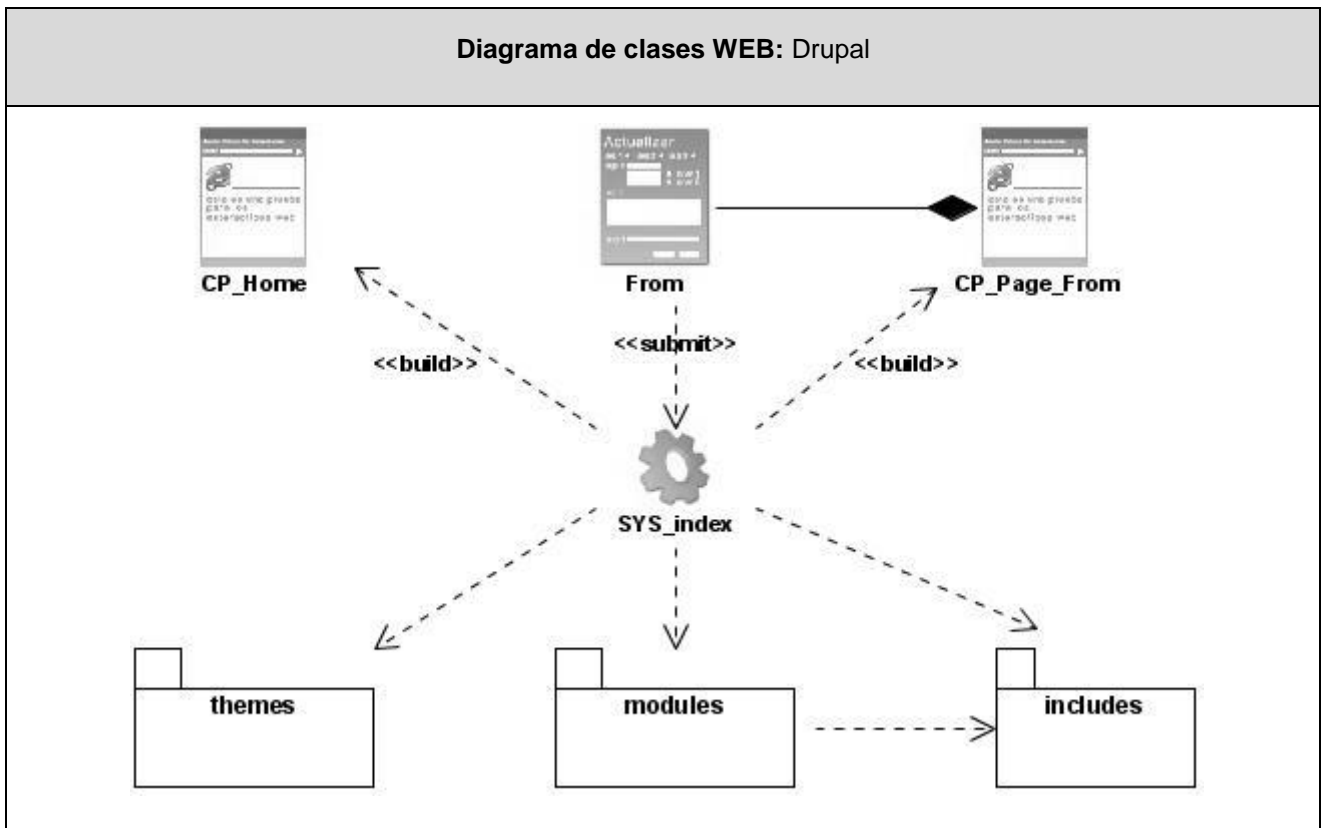
4.2.2 Paquete 1: Drupal

El Paquete 1: Drupal, consiste en la distribución del CMS Drupal. Básicamente contiene un paquete *Themes*, donde se encuentran los mecanismos que soportan el sistema de plantillas, de modo que cuando usted desee cambiar el diseño de la interfaz que presentará el sistema, sólo tiene que definir una nueva plantilla en este archivo; un paquete *Includes*, donde se encuentran ficheros de configuración y ficheros utilitarios, es este paquete donde se incluyen las API de acceso a datos; y por último un paquete *Modules*, que proveen a Drupal de sus funcionalidades, de forma tal que cuando desee agregar un nuevo módulo, sólo debe copiarlo dentro de esta carpeta y activarlo a través de la interfaz de Drupal.

Este CMS contiene una única página de servidor, la cual basándose en el sistema de clases genera el contenido de la página final, teniendo en cuenta los argumentos con que se realiza la petición. Las páginas generadas pueden o no contener formularios, ésto depende del módulo en cuestión y del propósito del mismo.

Debido a la naturaleza de los Sistemas de Gestión de Contenidos y a su estructura modular en esta sección se modelará cada módulo por separado, puesto que son independientes uno del otro en cuanto a funcionalidad. Abordados los principales aspectos que influyen en el diseño del diagrama de clases Web que representa el funcionamiento de Drupal, se está en condiciones de presentar el mismo, recordando siempre que la terminología para referirse a un módulo Drupal será la de “paquete”. El diagrama de clases se muestra en la siguiente figura:

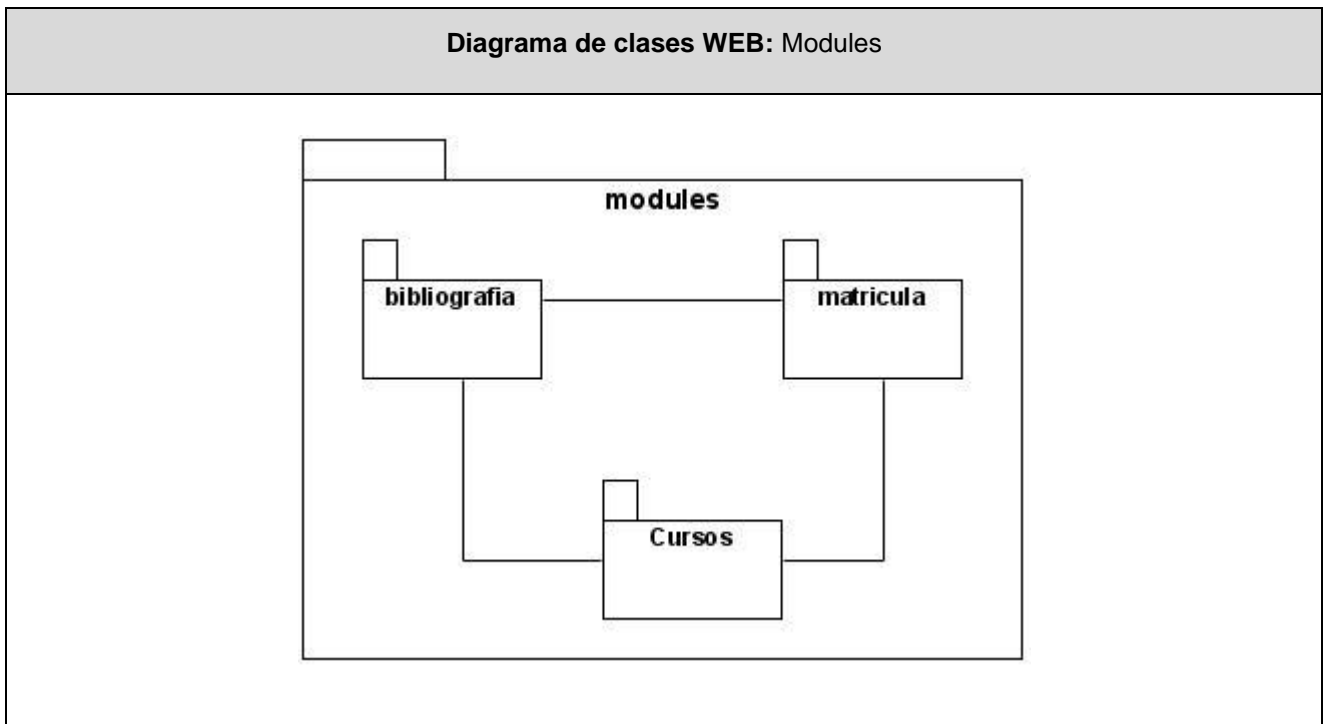
Figura 1 Diagrama de Clases del Diseño: Paquete Drupal



4.2.3 Sub paquete 1.1: Modules

El paquete Modules contiene los módulos que dan soporte a las funcionalidades de Drupal, y los módulos desarrollados en este trabajo. Para simplificar el diagrama no se modelan todos los módulos de Drupal, sino sólo los desarrollados.

Figura 2 Diagrama de Clases del Diseño: Sub Paquete Modules



Con el objetivo de hacer entendible las funcionalidades encapsuladas en estos módulos, se definen una tarjeta CRC por cada uno, con la finalidad de obtener un diseño simple y no incurrir en la implementación de características que no son necesarias.

Tabla 4.1 Tarjeta CRC Módulo Bibliografía

Módulo Bibliografía	
Funcionalidades	Colaboraciones (Módulos)
Adicionar bibliografía	
Modificar bibliografía	Node
Eliminar bibliografía	Upload
Mostrar bibliografía	Menu
Adicionar programas de cursos	Help
Modificar programas de cursos	Block
Eliminar programas de cursos	User
Mostrar programas de cursos	

Tabla 4.2 Tarjeta CRC Módulo Cursos

Módulo Cursos	
Funcionalidades	Colaboraciones (Módulos)
Crear materiales	Bibliografía
Modificar materiales	Upload
Eliminar materiales	Menu
Mostrar materiales	Help
	Block
	User

Tabla 4.3 Tarjeta CRC Módulo Matrícula

Módulo Matrícula	
Funcionalidades	Colaboraciones (Módulos)
	Cursos
Matricular cursos	Bibliografía
Mostrar listado de estudiantes	Menu
Mostrar listado de profesores	Help
	Block
	User

4.3 DISEÑO DE LA BASE DE DATOS

Diseñar la Base de Datos es algo que no se puede pasar por alto, producto de que uno de sus objetivos fundamentales es brindar la persistencia al modelo que se describe en el epígrafe anteriormente desarrollado.

El modelo de datos del problema en cuestión posee un nivel de complejidad bajo, producto a que las entidades son manejadas por el CMS Drupal (por lo que no han sido contempladas en el modelo de datos). A continuación se muestra el diagrama de clases persistentes y el modelo de datos que se utilizó:

Figura 3 Diagrama de clases persistentes

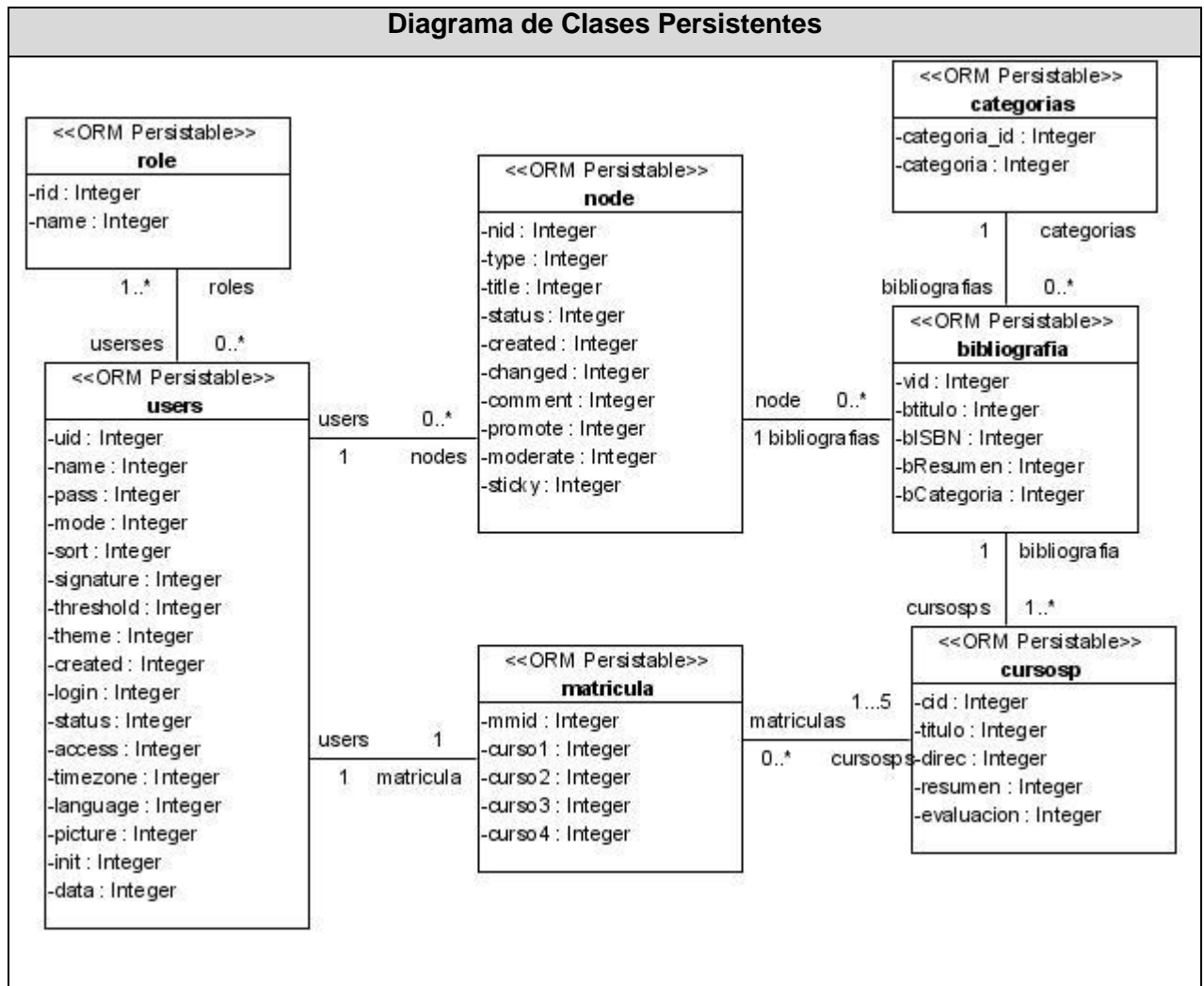
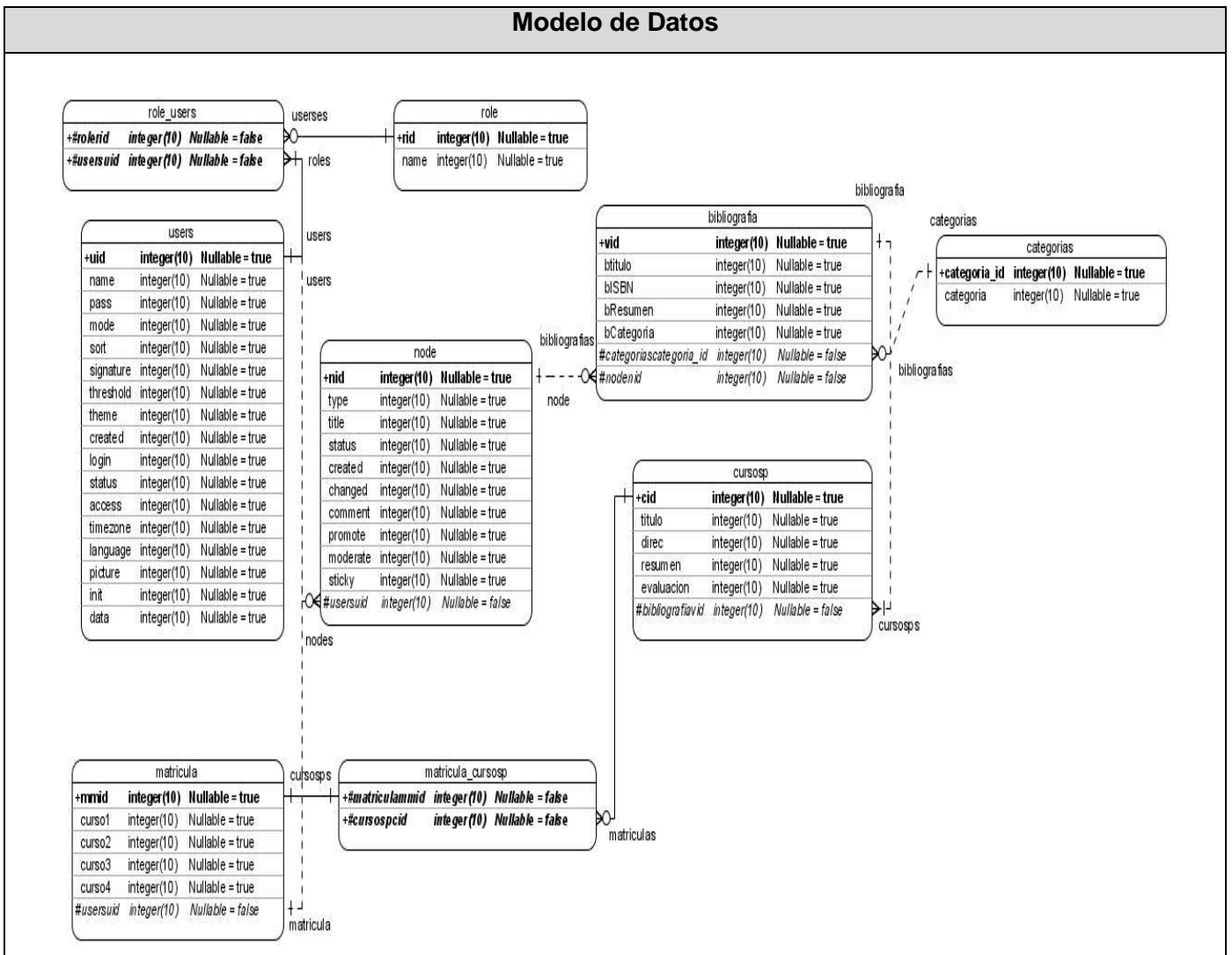


Figura 4 Modelo de Datos



4.4 FASE DE IMPLEMENTACIÓN

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o una persona), responsable de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan cada una de las iteraciones.

4.4.1 Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad, con el fin de obtener una versión del producto con algunas de las funcionalidades críticas para ser mostrado al cliente y tomar nuevas iniciativas de forma rápida.

Tabla 4.4 Módulos abordados en la primera iteración

Módulo	Historias de usuario	Tiempo de Implementación (semanas)	
		Estimación	Real
user	Gestionar cuenta de usuario	1	0.58
	Autenticar	1	0.29
bibliografía	Gestionar bibliografías	3	2.29
curso	Crear materiales	2	2
	Mostar materiales	1	0.14

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

Módulo user

Tabla 4.5 Tarea 1 del módulo user

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Configuración del módulo user.	
Tipo de tarea: Configuración	Puntos estimados: 0.29
Fecha inicio: 10 de diciembre del 2007	Fecha fin: 12 de diciembre del 2007
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se especificarán los diferentes datos que serán de utilidad para la creación de la cuenta. Se especifica el rol al cual estará sujeta la cuenta. Los datos serán almacenados en la Base de Datos de forma persistente	

Tabla 4.6 Tarea 2 del módulo user

Tarea	
Número de tarea: 2	Número de HU: 1
Nombre de la tarea: Manejo de persistencia en la BD.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.29
Fecha inicio: 13 de diciembre del 2007	Fecha fin: 15 de diciembre del 2007
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se gestionará la verificación de los datos introducidos con los almacenados en la Base de Datos. Se darán de alta y baja en el sistema a diferentes roles, reconociendo solamente los que están registrados en él. Un rol puede estar vinculado a varios usuarios.	

Tabla 4.7 Tarea 3 del módulo user

Tarea	
Número de tarea: 3	Número de HU: 3
Nombre de la tarea: Gestionar autenticación.	
Tipo de tarea: Configuración	Puntos estimados: 0.29
Fecha inicio: 07 de enero del 2008	Fecha fin: 09 de enero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se especificará el modo de autenticación por parte del administrador. Se gestionará la verificación de los datos introducidos con los almacenados en la Base de Datos.	

Módulo bibliografía

Tabla 4.8 Tarea 1 del módulo bibliografía

Tarea	
Número de tarea: 1	Número de HU: 7
Nombre de la tarea: Obtención del código fuente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.14
Fecha inicio: 10 de enero del 2008	Fecha fin: 11 de enero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se obtendrá el código fuente del medio de almacenamiento donde se encuentre. Se crearán mecanismos que posibiliten la modificación de la estructura del código.	

Tabla 4.9 Tarea 2 del módulo bibliografía

Tarea	
Número de tarea: 2	Número de HU: 7
Nombre de la tarea: Construcción del módulo bibliografía.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.86
Fecha inicio: 12 de enero del 2008	Fecha fin: 25 de enero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se modificará el código creando el módulo <i>bibliografía</i> que permitirá al profesorado publicar los diferentes materiales de manera general y de diversos contenidos. Se utiliza la integración de los hooks: hook_info, hook_help, hook_access, hook_perm, hook_form, hook_insert, hook_view, hook_block entre otros para posibilitar el resultado esperado.	

Tabla 4.10 Tarea 3 del módulo bibliografía

Tarea	
Número de tarea: 3	Número de HU: 7
Nombre de la tarea: Creación de las tablas en la BD.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.29
Fecha inicio: 26 de enero del 2008	Fecha fin: 28 de enero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se creará un archivo de tipo <i>.install</i> donde se crean y se destruyen las tablas “ <i>bibliografías</i> ” y “ <i>categorías</i> ” para posibilitar la persistencia de los datos.	

Módulo curso

Tabla 4.11 Tarea 1 del módulo curso

Tarea	
Número de tarea: 1	Número de HU: 4
Nombre de la tarea: Creación del código fuente del módulo curso.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29 de enero del 2008	Fecha fin: 05 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se creará el módulo de bloque básico <i>curso</i> que posibilitará al profesorado publicar las diferentes materiales respectivos a los bloques de contenidos de cada uno de los cursos. Se utiliza la integración de diferentes hooks, para lograr el resultado esperado.	

Tabla 4.12 Tarea 2 del módulo curso

Tarea	
Número de tarea: 2	Número de HU: 4
Nombre de la tarea: Creación de la tabla correspondiente al módulo curso.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.29
Fecha inicio: 06 de febrero del 2008	Fecha fin: 08 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se creará un archivo de tipo <i>.install</i> donde se crea y se destruye la tabla “ <i>cursosp</i> ” para posibilitar la persistencia de los datos.	

Tabla 4.13 Tarea 3 del módulo curso

Tarea	
Número de tarea: 3	Número de HU: 5
Nombre de la tarea: Configuración del módulo curso.	
Tipo de tarea: Configuración.	Puntos estimados: 0.29
Fecha inicio: 09 de febrero del 2008	Fecha fin: 11 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se activarán permisos según los diferentes roles para el acceso a la información almacenada.	

4.4.2 Iteración 2

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

Tabla 4.14 Módulos abordados en la segunda iteración

Módulo	Historias de usuario	Tiempo de Implementación (semanas)	
		Estimación	Real
curso	Eliminar materiales	1	0.29
bibliografía	Gestionar los programas de cursos	1	0.43
	Mostrar los programas de cursos	1	0.29
matrícula	Matricular cursos	3	2.87
	Mostrar listados de profesores	1	0.29

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

Tabla 4.15 Tarea 4 del módulo curso

Tarea	
Número de tarea: 4	Número de HU: 6
Nombre de la tarea: Configuración 2 del módulo curso.	
Tipo de tarea: Configuración.	Puntos estimados: 0.14
Fecha inicio: 12 de febrero del 2008	Fecha fin: 14 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se continuará con la activación de permisos según roles sobre la información almacenada, permitiendo la eliminación de la misma.	

Módulo bibliografía

Tabla 4.16 Tarea 4 del módulo bibliografía

Tarea	
Número de tarea: 4	Número de HU: 9
Nombre de la tarea: Modificación del código fuente referente al módulo bibliografía.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.43
Fecha inicio: 18 de febrero del 2008	Fecha fin: 21 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se crearán nuevos <i>hooks</i> integrados al código base para propiciar la organización de los archivos de contenidos en categorías.	

Tabla 4.17 Tarea 5 del módulo bibliografía

Tarea	
Número de tarea: 5	Número de HU: 10
Nombre de la tarea: Configuración del módulo bibliografía.	
Tipo de tarea: Configuración.	Puntos estimados: 0.29
Fecha inicio: 22 de febrero del 2008	Fecha fin: 24 de febrero del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se gestionarán los archivos incluidos sobre ese módulo para su presentación en la interfaz gráfica. El acceso a la información será a través de los permisos según roles.	

Módulo matrícula

Tabla 4.18 Tarea 1 del módulo matrícula

Tarea	
Número de tarea: 1	Número de HU: 11
Nombre de la tarea: Creación del código fuente perteneciente al módulo matrícula.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha inicio: 25 de febrero del 2008	Fecha fin: 10 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se creará el módulo <i>matricula</i> mediante la interacción de diferentes <i>hook</i> .	

Tabla 4.19 Tarea 2 del módulo matrícula

Tarea	
Número de tarea: 2	Número de HU: 11
Nombre de la tarea: Creación de la tabla perteneciente al módulo matrícula.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.29
Fecha inicio: 11 de marzo del 2008	Fecha fin: 13 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se establecerá un archivo de tipo <i>.install</i> donde se crean y se destruyen las tablas “ <i>matricula</i> ” y “ <i>user_matricula</i> ” para permitir la persistencia de datos.	

Tabla 4.20 Tarea 3 del módulo matrícula

Tarea	
Número de tarea: 3	Número de HU: 11
Nombre de la tarea: Configuración del módulo matrícula.	
Tipo de tarea: Configuración.	Puntos estimados: 0.58
Fecha inicio: 14 de marzo del 2008	Fecha fin: 18 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se configurará la interfaz gráfica, permitiendo el acceso a la BD para actualizar los campos correspondientes.	

Tabla 4.21 Tarea 4 del módulo matrícula

Tarea	
Número de tarea: 4	Número de HU: 14
Nombre de la tarea: Configuración 2 del módulo matrícula.	
Tipo de tarea: Configuración.	Puntos estimados: 0.29
Fecha inicio: 19 de marzo del 2008	Fecha fin: 21 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se gestionarán los archivos incluidos sobre este módulo para el acceso a la BD y la presentación del listado en la interfaz gráfica cumpliendo el rol de estudiantes.	

4.4.3 Iteración 3

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

Tabla 4.22 Módulos abordados en la tercera iteración

Módulo	Historias de usuario	Tiempo de Implementación	
		Estimación	Real
matrícula	Mostrar lista de estudiantes	1	0.29
bibliografía	Mostrar bibliografía	1	0.29
perfil	Mostrar perfil de usuario	1	0.29
search	Buscar información	1	0.43

A continuación se muestran las tareas concernientes a los módulos implementados en esta iteración:

Módulo matrícula

Tabla 4.23 Tarea 5 del módulo matrícula

Tarea	
Número de tarea: 5	Número de HU: 13
Nombre de la tarea: Configuración 3 del módulo matrícula.	
Tipo de tarea: Configuración.	Puntos estimados: 0.14
Fecha inicio: 22 de marzo del 2008	Fecha fin: 24 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se gestionarán los archivos incluidos sobre este módulo para el acceso a la BD y la presentación del listado en la interfaz gráfica cumpliendo el rol de profesor.	

Módulo bibliografía

Tabla 4.24 Tarea 6 del módulo bibliografía.

Tarea	
Número de tarea: 6	Número de HU: 8
Nombre de la tarea: Configuración 2 del módulo bibliografía.	
Tipo de tarea: Configuración.	Puntos estimados: 0.29
Fecha inicio: 25 de marzo del 2008	Fecha fin: 27 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se gestionarán los archivos incluidos sobre este módulo para su presentación en la interfaz gráfica.	

Módulo perfil

Tabla 4.25 Tarea 1 del módulo perfil.

Tarea	
Número de tarea: 1	Número de HU: 2
Nombre de la tarea: Configuración del módulo perfil.	
Tipo de tarea: Configuración.	Puntos estimados: 0.29
Fecha inicio: 28 de marzo del 2008	Fecha fin: 30 de marzo del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se activarán archivos incluidos sobre este módulo para su posterior visualización.	

Módulo search

Tabla 4.26 Tarea 1 del módulo search.

Tarea	
Número de tarea: 1	Número de HU: 12
Nombre de la tarea: Configuración del módulo search.	
Tipo de tarea: Configuración.	Puntos estimados: 0.43
Fecha inicio: 31 de marzo del 2008	Fecha fin: 03 de abril del 2008
Programador responsable: Deiler Sevilla Fernández – Yanirys Silva Lemus	
Descripción: Se activará este módulo y se integrará a la interfaz gráfica programada.	

4.4.4 Diagrama de Componentes

Un diagrama de componentes representa la separación del sistema de *software* en componentes físicos (por ejemplo archivos, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes. Se utiliza para modelar la vista estática de un sistema, además de mostrar la organización y las dependencias entre un conjunto de componentes.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

Con el propósito de brindar un mejor entendimiento del sistema se muestra el diagrama de componentes del sistema y de cada uno de los módulos implementados. (Ver Anexo 1, 2, 3 y 4)

4.4.5 Diagrama de Despliegue

El diagrama de despliegue permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor Web y la misma se comunica con un sistema de gestión de base de datos (MySQL). (Ver Anexo 5)

4.5 PRUEBAS

Uno de los pilares fundamentales de XP es el proceso de pruebas [14], el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección [13]. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. [13]

4.5.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario [13]. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable [15]. Una HU no se considera completa hasta que no ha pasado por sus pruebas de aceptación.

Tabla 4.27 Prueba 1 al módulo cursos

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Publicar una conferencia en el sistema.	
Descripción: Prueba para la funcionalidad de publicar una conferencia en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta publicar una conferencia por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: La conferencia es publicada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.28 Prueba 2 al módulo cursos

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Mostrar una conferencia en el sistema.	
Descripción: Prueba para la funcionalidad de mostrar una conferencia en el sistema.	
Condiciones de Ejecución: El usuario debe estar matriculado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta ver las conferencias publicadas, por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de estudiante o profesor.	
Resultado Esperado: Se brinda la posibilidad de ver todas las conferencias publicadas según los bloques de cursos del cual es matrícula.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.29 Prueba 3 al módulo cursos

Caso de Prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Descargar una conferencia del sistema.	
Descripción: Prueba para la funcionalidad de descarga de una conferencia del sistema.	
Condiciones de Ejecución: El usuario debe estar matriculado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta descargar las conferencias publicadas, por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de estudiante o profesor.	
Resultado Esperado: La conferencia es descargada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.30 Prueba 4 al módulo cursos

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Baja de una conferencia del sistema.	
Descripción: Prueba para la funcionalidad de eliminar una conferencia del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta dar baja a alguna conferencia publicada, por un usuario que cumple el rol de profesor.	
Resultado Esperado: La conferencia es eliminada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.31 Prueba 1 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Publicar bibliografías en el sistema.	
Descripción: Prueba para la funcionalidad de publicar bibliografías del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta publicar bibliografías por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: Las bibliografías son publicadas sin generar error.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.32 Prueba 2 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Eliminar bibliografías en el sistema.	
Descripción: Prueba para la funcionalidad de eliminar bibliografías del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta eliminar bibliografías por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: Las bibliografías son eliminadas sin generar error.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.33 Prueba 3 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Mostrar bibliografías en el sistema.	
Descripción: Prueba para la funcionalidad de mostrar bibliografías en el sistema.	
Condiciones de Ejecución: Las bibliografías deben estar publicadas.	
Entrada/ Pasos de ejecución: Se intenta acceder a bibliografías por un usuario que navega en el sistema, sin necesidad de estar autenticado en el mismo.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.34 Prueba 4 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU8_P2	Historia de Usuario: 8
Nombre: Descargar bibliografías en el sistema.	
Descripción: Prueba para la funcionalidad de descarga de bibliografías.	
Condiciones de Ejecución: Las bibliografías deben estar publicadas.	
Entrada/ Pasos de ejecución: Se intenta descargar alguna bibliografía de las publicadas por un usuario que navega en el sistema, sin necesidad de estar autenticado en el mismo.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.35 Prueba 5 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Publicar programas de cursos en el sistema.	
Descripción: Prueba para la funcionalidad de publicar P1 en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta publicar P1 por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.36 Prueba 6 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Dar baja a los programas de cursos en el sistema.	
Descripción: Prueba para la funcionalidad eliminar los P1 del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta eliminar los P1 publicados por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de profesor.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.37 Prueba 7 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Mostrar los P1 en el sistema.	
Descripción: Prueba para la funcionalidad de mostrar los P1.	
Condiciones de Ejecución: Los P1 deben estar publicados.	
Entrada/ Pasos de ejecución: Se intenta acceder a los P1 publicados por un usuario que navegue en el sistema, no necesariamente debe estar autenticado.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.38 Prueba 8 al módulo bibliografía

Caso de Prueba de Aceptación	
Código: HU10_P2	Historia de Usuario: 10
Nombre: Descargar los P1.	
Descripción: Prueba para la funcionalidad de descargar los P1.	
Condiciones de Ejecución: Los P1 deben estar publicados.	
Entrada/ Pasos de ejecución: Se intenta descargar los P1 publicados por un usuario que navegue en el sistema, no necesariamente debe estar autenticado.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.39 Prueba 1 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Matricular en cursos.	
Descripción: Prueba para la funcionalidad de matricular en cursos.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta matricular en los cursos ofertados por un usuario que tiene los datos válidos, es decir, que cumpla con el rol de estudiante o profesor.	
Resultado Esperado: No se generan errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.40 Prueba 2 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU11_P2	Historia de Usuario: 11
Nombre: Matricular en más de 4 niveles de cursos.	
Descripción: Prueba para la funcionalidad de matricular en más de 5 bloques de cursos.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta matricular en más de 5 bloques de los cursos ofertados por un usuario que tiene los datos válidos, es decir, que cumpla con el rol de estudiante o profesor.	
Resultado Esperado: Se genera un mensaje de error, advirtiendo que solo se puede matricular como máximo en 5 bloques de los niveles de cursos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.41 Prueba 3 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU11_P3	Historia de Usuario: 11
Nombre: Matricular en 2 niveles de cursos iguales.	
Descripción: Prueba para la funcionalidad de matricular en 2 niveles de cursos iguales.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta matricular en 2 niveles iguales de un mismo curso, por un usuario que tiene los datos válidos, es decir que cumpla con el rol de estudiante o profesor.	
Resultado Esperado: Se genera un mensaje de error, advirtiendo que no puede registrar dos veces el mismo curso en igual nivel.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.42 Prueba 4 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU11_P4	Historia de Usuario: 11
Nombre: Modificar matrícula.	
Descripción: Prueba para la funcionalidad de modificar matrícula.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con permisos válidos.	
Entrada/ Pasos de ejecución: Se intenta modificar la matrícula de un usuario.	
Resultado Esperado: Se modifican los bloques de contenidos del curso en los que está matriculado el usuario.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.43 Prueba 5 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU11_P5	Historia de Usuario: 11
Nombre: Eliminar matrícula.	
Descripción: Prueba para la funcionalidad de eliminar matrícula.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con permisos válidos.	
Entrada/ Pasos de ejecución: Se intenta eliminar la matrícula de un usuario.	
Resultado Esperado: Se elimina la matrícula del usuario seleccionado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.44 Prueba 6 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU13_P1	Historia de Usuario: 13
Nombre: Mostrar listado de estudiantes.	
Descripción: Prueba para la funcionalidad de mostrar listado de estudiantes.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta mostrar el listado de los estudiantes según el curso seleccionado por un usuario que contenga los datos válidos, es decir, que posea rol de profesor.	
Resultado Esperado: Se muestra la lista de estudiantes pertenecientes a dicho curso.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 4.45 Prueba 7 al módulo matrícula

Caso de Prueba de Aceptación	
Código: HU14_P1	Historia de Usuario: 14
Nombre: Mostrar listado de profesores.	
Descripción: Prueba para la funcionalidad de mostrar listado de profesores.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta mostrar el listado de los profesores según el curso seleccionado por un usuario que contenga los datos válidos, es decir, que posea rol de estudiante o profesor.	
Resultado Esperado: Se muestra la lista de profesores encargados de impartir dicho curso.	
Evaluación de la Prueba: Prueba satisfactoria.	

4.6 CONCLUSIONES

En este capítulo se construyó el modelo necesario para desarrollar el proceso de implementación del sistema. Se diseñaron las clases persistentes que permiten hacer el diagrama de entidad-relación en el sistema de gestión de bases de datos que se utilizará. Se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y las pruebas de aceptación que propician al cliente conformidad y seguridad ante el sistema. Con el fin de este capítulo se da por terminada la propuesta que trae este trabajo.

5

CAPÍTULO ESTUDIO DE FACTIBILIDAD

5.1 INTRODUCCIÓN

Una de las tareas de mayor importancia en la planificación de proyectos de software es el cálculo de la estimación, la cual consiste en determinar con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. En este capítulo se realiza un estudio de la factibilidad para la realización del sistema propuesto, haciendo una estimación del esfuerzo necesario para llevar a cabo el mismo.

5.2 CARACTERÍSTICAS DEL PROYECTO

Tabla 5.1 Entradas Externas

Nombre de la entrada externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o compleja)
Gestionar cuenta de usuario	3	4	baja
Crear materiales	1	1	baja
Eliminar materiales	1	1	baja
Gestionar bibliografía	2	1	baja
Gestionar programas de curso	3	1	baja
Matricular en cursos	1	4	baja
Total	11		

Tabla 5.2 Salidas Externas

Nombre de la salida externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o compleja)
Mostrar materiales	1	1	baja
Mostrar bibliografía	1	1	baja
Mostrar P1	1	1	baja
Total	3		

Tabla 5.3 Consultas Externas

Nombre de la consulta externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o compleja)
Mostrar perfil	1	4	baja
Buscar información	1	1	baja
Mostrar listado de estudiantes	1	1	baja
Mostrar listado de profesores	1	1	baja
Autenticar	1	2	baja
Total	5		

Tabla 5.4 Archivos lógicos

Nombre del archivo lógico	Cantidad de tipos de registros	Cantidad de elementos de datos	Clasificación (baja, media o compleja)
Usuario	1	17	baja
Cursos	1	5	baja
Bibliografías	2	6	baja
Matrícula	1	6	Baja
Categorías	1	2	baja
Total	6		

Tabla 5.5 Archivos de Interfaz Externa

Nombre del archivo lógico	Cantidad de tipos de registros	Cantidad de elementos de datos	Clasificación (baja, media o compleja)
Total	0		

5.2.1 Estimación inicial

Al sumar los aportes de todos los elementos se obtiene los Puntos de Función sin ajustar.

Tabla 5.6 Puntos de Función desajustados

Elementos	Baja		Media		Alta		Aportes
	Cant.	Valor	Cant.	Valor	Cant.	Valor	
Entradas Externas	11	3	0	4	0	6	33
Salidas Externas	3	4	0	5	0	7	12
Consultas Externas	5	3	0	4	0	6	15
Archivos lógicos internos	6	7	0	10	0	15	42
Archivos de interface externos	0	5	0	7	0	10	0
Total							102

5.3 CÁLCULO DE INSTRUCCIONES FUERTES, ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.

El método COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la Cantidad de Líneas de Código (SLOC, *Source Lines of Code*) estimados para un proyecto, como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo que influyen en el esfuerzo requerido para el desarrollo del software. La meta es obtener un número que caracterice completamente al sistema.

5.3.1 Cálculo del esfuerzo nominal.

Se aplica la ecuación de cálculo del esfuerzo nominal:

$$PM_{\text{nominal}} = A * (\text{Size})^E$$

PM_{nominal}: es el esfuerzo nominal requerido en meses-hombre.

A: Es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. El modelo la calibra con un valor de 2.94.

Size: Es el tamaño estimado del software, en Puntos de Función sin ajustar (convertibles a KSLOC). Se calcula el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP (factor de conversión = 53 SLOC/UFP). Entonces:

$$\text{Size} = 53 * 102 = 5406 \text{ SLOC}$$

Size = 5.41 KSLOC

E: Es una constante denominada *Factor Escalar*. Se calcula ponderando las variables escalares, mediante la ecuación:

$$E = 0.91 + 0.01 * \sum (W_i)$$

Donde las W_i se muestran en la siguiente tabla:

Tabla 5.7 Factor Escalar (SF)

Nombre	Valor	Justificación
PREC	1.24	Existen proyectos similares a nivel mundial y nacional.
FLEC	2.03	Cuenta con alta flexibilidad.
RESL	2.83	No se identifican riesgos críticos.
TEAM	3.29	Existe interacción cooperativa.
PMAT	3.12	Existe una experiencia previa en aplicaciones de este tipo.
Total	12.51	

Luego: $E = 0.91 + 0.01 * 12.51 = 1.0351$.

Entonces:

$$PM_{nominal} = A * (Size)^E = 2.94 * (5.4 \text{ KSLOC})^{1.0351} = 16.8441 \text{ meses-hombre}$$

5.3.2 Cálculo del esfuerzo ajustado

Se aplica la ecuación de cálculo del esfuerzo ajustado:

$$PM_{ajustado} = PM_{nominal} * \prod (ME_i)$$

Tabla 5.8 Multiplicadores de esfuerzo (EM)

Nombre	Valor	Justificación
RUSE	1.07	El nivel de reusabilidad es alto.
RCPX	1.00	Confiabilidad, documentación y volumen de datos moderados.
PDIF	1.00	Uso de memoria y almacenamiento normal, plataforma estable.
PERS	0.83	Alta capacidad del personal.
SCED	1.00	Se empleó el tiempo planificado para el desarrollo del sistema.
Total	0.89	

Entonces:

$$PM_{ajustado} = 16.84 * 0.89 = 14.99 \text{ Meses-hombre.}$$

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

Valores calibrados:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.0351 - 0.91) = 0.27$$

$$TDEV \text{ (Tiempo de desarrollo)} = C * (PM_{ajustado})^F = 3.67 * (14.99)^{0.27} = 7.62 \text{ meses}$$

$$CH \text{ (Cantidad de Hombres)} = PM_{ajustado} / TDVE = 14.99 / 7.62 = 1.97 \approx 2$$

Como la cantidad real de hombres a desarrollar la aplicación es 2, al reajustar el tiempo de desarrollo según la cantidad de hombres, resultó un tiempo equivalente a **6.32 meses**.

Salario promedio:

Para determinar el salario promedio se tiene en cuenta que los desarrolladores del sistema pueden ser ingenieros recién graduados pertenecientes a la UCI (Universidad de las Ciencias Informáticas), por lo que se toma como salario correspondiente ha un adiestrado: **\$225.00**.

$$\text{Costo} = 2 * \$225 * 14.99 = \$ 6745.5$$

De lo anterior se obtienen los siguientes resultados:

Tabla 5.9 Resultados alcanzados

Cálculo de:	Valor
Esfuerzo	14.99 Meses-hombre
Tiempo de desarrollo	6.32 Meses
Cantidad de Hombres	2
Salario Medio	\$ 225
Costo	\$ 6745.5

5.3.3 Beneficios tangibles e intangibles

La incorporación de la práctica de la integración continua dentro del proceso de desarrollo de una empresa supone una serie de beneficios, entre los que se destaca la reducción a casi nulo del tiempo de integración de los diferentes módulos que conforman el proyecto. Ésto se debe a que los errores de integración son detectados rápidamente, posibilitando su corrección de forma inmediata.

El desarrollo del producto propuesto en este trabajo brinda como beneficio un sistema incorporado para posibilitar la autonomía total en la planificación de los contenidos que se deseen adquirir, posibilitando la preparación de estudiantes y docentes como solución a las limitantes presentadas en el proceso.

5.3.4 Análisis de Costo

El desarrollo de un producto siempre tiene un costo de producción, el cual debe ser justificado en base a los beneficios reportados por el mismo. El sistema que se propone en este trabajo no conlleva a grandes gastos, puesto que sólo es influyente el salario de los desarrolladores, por lo cual se concluye que su implementación es factible. Ésto se debe en gran medida a la utilización de plataformas, APIs y herramientas libres que no requieren el pago de alguna licencia.

En el presente capítulo se ha realizado un análisis de factibilidad de la solución propuesta, arribando a la conclusión de que es viable su desarrollo comparando los costos de producción con los beneficios reportados en su puesta en funcionamiento.

5.4 CONCLUSIONES

En este capítulo se ha realizado el estudio de la factibilidad, el análisis de los costos y los beneficios, tanto tangibles como intangibles del sistema desarrollado. Se concluye que es factible el desarrollo de esta aplicación dado que los beneficios son de suma importancia y sus costos son pequeños.

CONCLUSIONES

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

Durante el desarrollo de esta investigación se realizó un estudio que permitió tener un conocimiento de la situación actual y las tendencias de los sistemas de gestión de cursos con contenidos informáticos. Se demostró la necesidad de desarrollar un sistema que fuese capaz de automatizar los procesos de gestión de cursos para lograr la autonomía sobre la planificación de los contenidos. Con este trabajo se presenta una aplicación Web que permite la publicación de cursos por bloques y niveles sobre temas informáticos, brindando así al usuario un punto de acceso donde este puede obtener toda la información que necesite acerca de los cursos publicados, podrá realizar su matrícula y contará con la bibliografía adecuada para cumplir con sus expectativas. Además permite la constante actualización de todo su contenido. Un aspecto novedoso, es que se realizarán las diferentes funcionalidades como módulo Drupal, lo que permite que sea altamente configurable y adaptable en dependencia del centro donde se utilice el sistema.

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

RECOMENDACIONES

Como resultado del proceso de investigación y realización de la aplicación han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas:

1. Continuar con la línea de desarrollo del sistema existente por parte del proyecto CICE.
2. Agregar nuevas funcionalidades en la medida del avance y de acuerdo con las expectativas de los usuarios finales por parte de los desarrolladores vinculados al proyecto.

REFERENCIAS BIBLIOGRÁFICAS

1. Colibrí. (25 de Marzo de 2003). "Software libre en empresas internacionales." [En línea] [Citado el: 15 de enero de 2008.] <http://bachue.com/colibri/proyecto.html>.
2. KABIR, M. J. La biblia de Servidor Apache 2, 2003, Vol: I.
3. Desarrollo Web. [En línea] [Citado el: 20 de enero de 2008.] <http://www.maestrosdelweb.com/editorial/phpintro>
4. Enciclopedia Web. [En línea] [Citado el: 12 de Febrero de 2008.] <http://developer.mozilla.org/es/docs/JavaScript>
5. **Castillo, A.** Características de PHP. [En línea] [Citado el: 5 de diciembre de 2007.] http://www.programemos.com/index.php?option=com_content&task=view&id=279&Itemid=223
6. Web dinámica con PHP. [En línea] [Citado el: 19 de enero de 2008.] <http://ascii.eii.us.es/docs/2002-03/php/php4.html>
7. **S, Henst.** Maestros de la Web. [En línea] [Citado el: 10 de diciembre de 2007.] <http://ascii.eii.us.es/docs/2002-03/php/php3.html>.
8. Drupal Web. [En línea] [Citado el: 16 de febrero de 2008.] <http://www.drupalweb.com/>
9. Joomla Spanish. [En línea] [Citado el: 16 de febrero de 2008.] <http://www.joomlaspanish.org/>
10. **Urmachea, M. G.** SQL. [En línea] [Citado el: 12 de diciembre de 2007.] <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>
11. **Schumacher, R. Lentz, A.** MySQL. [En línea] [Citado el: 12 de diciembre de 2007.] <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>
12. **WELLING, THOMSON.** 2003.
13. **Allende, Roberto.** *Desarrollo de Portales y Extranet con Plone.* 2006.
14. **Crispin, L. y House, T.** *Testing Extreme Programming.* s.l. : Addison Wesley, 2002.
15. **Beck, K.** *Extreme Programming Explained.* s.l. : Addison Wesley, 2000.
16. XP. *A gentle introduction.* [En línea] 2006. [Citado el: 21 de febrero de 2008.] <http://www.extremeprogramming.org>.
17. **Beck, K. y Fowler, M.** *Planeando en Programación Extrema.* 2000.

BIBLIOGRAFÍA

1. AJAX un nuevo acercamiento a aplicaciones web. [En línea] [Citado el: 06 de marzo de 2008.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
2. **Bakken, Sæther, Schmid, Stig. y Egon.** Manual de Php. PHP Documentation Group. . [En línea] 2003. [Citado el: 01 de febrero de 2008.] <http://www.php.net/docs.php>.
3. **Beck, K.** *Extreme Programming Explained*. s.l. : Addison Wesley, 2000.
4. **Beck, K. y Fowler, M.** *Planeando en Programación Extrema*. 2000.
5. **Booch, G., Rumbaugh, J. y Jacobson, I.** *El Lenguaje Unificado de Modelado*. 1999.
6. Comunidad de Programadores: Artículo: "Cursos Oracle". [En línea] [Citado el: 29 de marzo de 2008.] <http://www.lawebdelprogramador.com/cursos/mostrar.php?id=66&texto=Oracle>.
7. **Crispin, L. y House, T.** *Testing Extreme Programming*. s.l. : Addison Wesley, 2002.
8. Desarrolloweb.com. "Programación en PHP". [En línea] [Citado el: 29 de enero de 2008.] <http://www.desarrolloweb.com/manuales/>.
9. Desarrollo de Web: Manual: "Qué es cada tecnología". [En línea] [Citado el: 15 de enero de 2008.] <http://www.desarrolloweb.com/manuales/15>.
10. Desarrollo de Web: Manual: "Tutorial de SQL". [En línea] [Citado el: 15 de enero de 2008.] <http://www.desarrolloweb.com/manuales/9>.
11. Desarrollo de Web: Artículo: "Páginas dinámicas". [En línea] [Citado el: 15 de enero de 2008.] <http://www.desarrolloweb.com/manuales/7>.
12. **González, Yanier Castellanos.** *Portal de las Misiones Sociales de la República Bolivariana de Venezuela*. La Habana : s.n., 2005.
13. Guía Breve de Tecnologías XML . [En línea] [Citado el: 19 de febrero de 2008.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>.
14. **Hernández, Marcos Legido.** Programación en JavaScript . [En línea] [Citado el: 29 de enero de 2008.] <http://es.wikibooks.org/wiki/Portada>.
15. Instalar Apache, Internet Information Server (IIS) y MySQL en Windows. [En línea] [Citado el: 26 de enero de 2008.] <http://www.cristalab.com/tutoriales/128/instalar-apache-internet-information-server-iis--y-mysql-en-windows>.
16. **Miranda, Hilda Libia Martínez.** *Herramienta Web para la autonomía en el aprendizaje*. La Habana : s.n., 2007.

-
17. Perl en Español. [En línea] [Citado el: 17 de marzo de 2008.]
 18. ¿Por qué J2EE en empresas? . [En línea] [Citado el: 21 de marzo de 2008.]
<http://listas.inf.utfsm.cl/pipermail/linux/2006-February/027290.html>.
 19. **Pressman, R.** *Software Engineering. A Practitioner's Approach*. USA : Fourth Edition, 1999.
 20. Programación en castellano: Montar una web con apache+php+postgreSQL . [En línea] [Citado el: 28 de marzo de 2008.]
<http://www.programacion.net/comentarios/id=apache&obj=escrito&num=5/>.
 21. **Rodríguez, José Antonio Plá y Arriba, Damián Lizastegui.** *Sistema para la integración continua de proyectos y el control de builds en la empresa Procyon Soluciones*. La Habana : s.n., 2007.
 22. Sistema de gestión de Contenidos: Artículo: "CMS Quick Guide". [En línea] [Citado el: 02 de febrero de 2008.]
<http://www.opensourcecms.com/index.php?option=content&task=view&id=388>.
 23. Sistemas de Gestión de Contenidos: Tutorial: "CMS Tutorial". [En línea] [Citado el: 02 de febrero de 2008.]
<http://www.opensourcecms.com/index.php?option=content&task=view&id=500>.
 24. Sistema de gestión de contenidos: Artículo: So what is a CMS? [En línea] [Citado el: 03 de febrero de 2008.] http://typo3.com/What_is_a_CMS.1351.0.html.
 25. Sistemas de Gestión de Contenidos: Artículo: "What is a content management system?". [En línea] [Citado el: 03 de febrero de 2008.] http://www.steptwo.com.au/papers/kmc_what/.
 26. Sistemas de Gestión de Contenidos: "About". Drupal.org . [En línea] [Citado el: 05 de marzo de 2008.] <http://drupal.org/features>.
 27. Sistemas de Gestión de Contenidos: "Feature Overview". Drupal.org . [En línea] [Citado el: 05 de marzo de 2008.] <http://drupal.org/features>.
 28. Sistemas de Gestión de Contenidos: "Module developer's guide". Drupal.org . [En línea] [Citado el: 10 de marzo de 2008.] <http://drupal.org/node/508>.
 29. Sistemas de Gestión de Contenidos: "User's guide". Drupal.org. [En línea] [Citado el: 10 de marzo de 2008.] <http://drupal.org/node/6261>.
 30. Sistemas de Gestión de Contenidos: "Drupal". CmsMatrix.org. [En línea] [Citado el: 20 de marzo de 2008.] <http://www.cmsmatrix.org/matrix?func=viewDetail&wid=2&listingId=1050>.
 31. Sistemas de Gestión de Contenidos: "Apache Lenya". CmsMatrix. [En línea] [Citado el: 01 de abril de 2008.] <http://www.cmsmatrix.org/matrix?func=viewDetail&wid=2&listingId=1123>.

32. Sistemas de Gestión de Contenidos: “Apache Lenya”. Apache Software Foundation. [En línea] [Citado el: 01 de abril de 2008.] <http://lenya.apache.org/>.
33. Sistemas de Gestión de Contenidos: Artículo: “Introducción a los Sistemas de Gestión de Contenidos (CMS)” . [En línea] [Citado el: 22 de marzo de 2008.] <http://mosaic.uoc.edu/articulos/cms1204.html>.
34. Software Libre: Noticia: “Gobierno decretará uso del software libre para la administración pública”. [En línea] [Citado el: 25 de enero de 2008.] <http://www.rnv.gov.ve/noticias/index.php?act=ST&f=14&t=8930>.
35. <http://perlenespanol.baboonsoftware.com/>.
36. Tutorial Joomla. [En línea] enero de 2007. [Citado el: 22 de febrero de 2008.] <http://www.juntadeandalucia.es/averroes/~14700079/upload/cursojoomlacep.pdf>.
37. XP. *A gentle introduction*. [En línea] 2006. [Citado el: 21 de febrero de 2008.] <http://www.extremeprogramming.org>.

Glosario de Términos

API: Del inglés *Application Programming Interface* - Interfaz de Programación de Aplicaciones es el 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.

ASP: Del inglés *Active Server Pages* es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida.

Build: Resultado del proceso de compilación y empaquetado de un producto de software.

CCS: Las hojas de estilo en cascada (en inglés *Cascading Style Sheets*) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

CGI: Interfaz de entrada común (en inglés *Common Gateway Interface*) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. Especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa.

CICE: Centro de Investigación de la Calidad de la Enseñanza.

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

Código Abierto: Es una tendencia internacional del desarrollo de software que profesa la distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

Frameworks: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las

entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

GPL: Es una licencia pública general de GNU o más conocida por su nombre en inglés *General Public License*. Está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Hook: Del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos de la interfaz que deben implementar los módulos desarrollados para este CMS.

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

ISAPI: Del inglés *Internet Server API*. Es una interfaz de programación de aplicaciones (API) para el servidor web de Microsoft, IIS (*Internet Information Server*). Permite que los programadores puedan desarrollar aplicaciones basadas en web que se procesen mucho más rápido que los programas CGI.

Microsoft: Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

Plugin: “Parche” para un programa que le añade características nuevas.

Release: Nueva versión de una aplicación informática.

SSL: Del inglés *Secure Sockets Layer* es un protocolo criptográfico que proporciona comunicaciones seguras en Internet.

UCI: Universidad de las Ciencias Informáticas.

Unix: Sistema operativo atribuido a Ken Thompson y comercializado por la empresa ATT en la década de los 70s que alcanzó mucho éxito, sobretodo en las universidades y posteriormente en las empresas. Entre sus principales características tenemos que es: portable, robusto, y flexible actualmente goza de gran popularidad dentro de la tecnología de Internet.

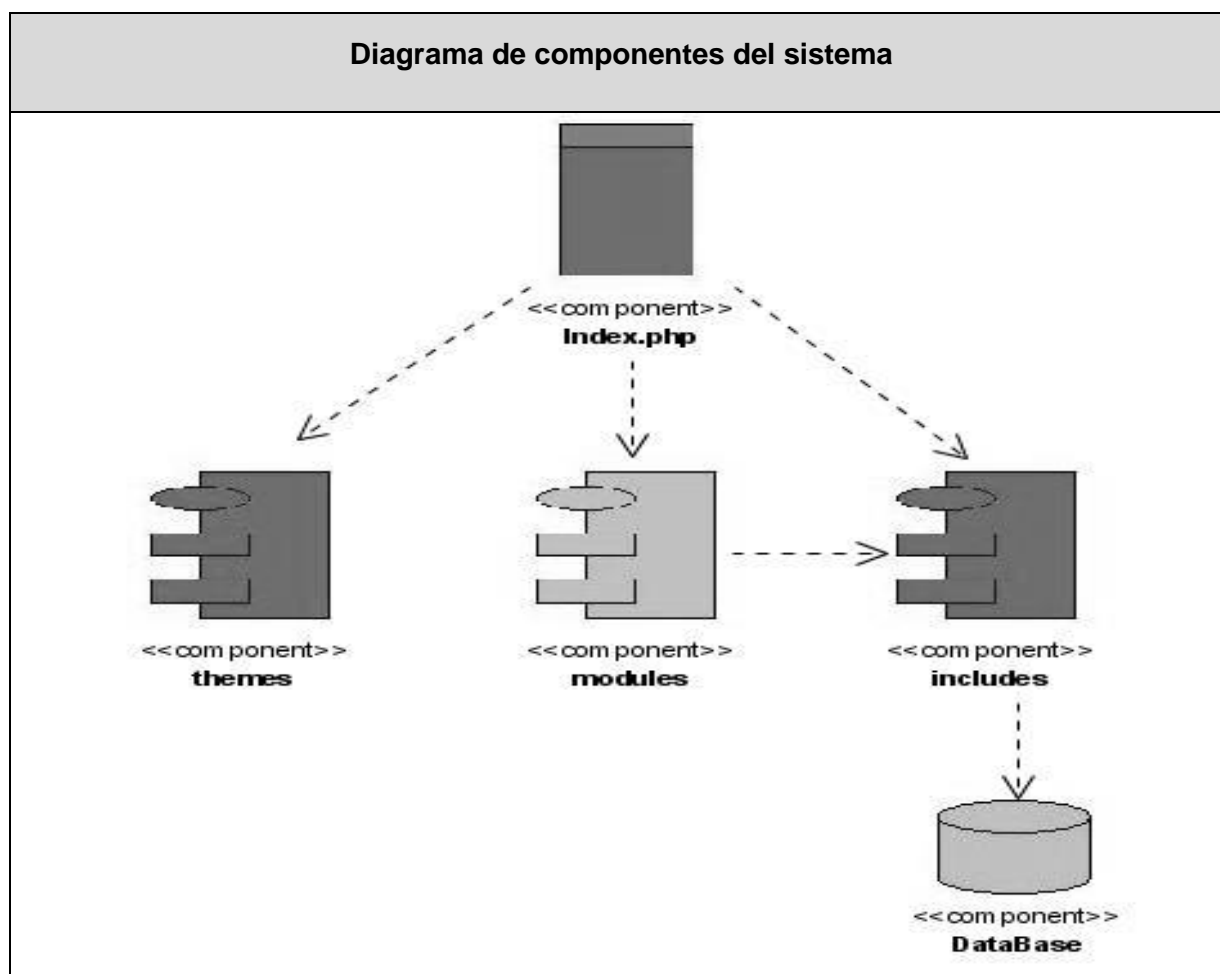
Web Services: Componente de software que puede auto describirse y provee cierta funcionalidad a otras aplicaciones, a través de una conexión de Internet. Esas aplicaciones, acceden los Web Services vía protocolos Web y formatos de datos estándares, tales como http y XML, sin tener en cuenta en absoluto cómo los Web Services están implementados.

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

ANEXOS

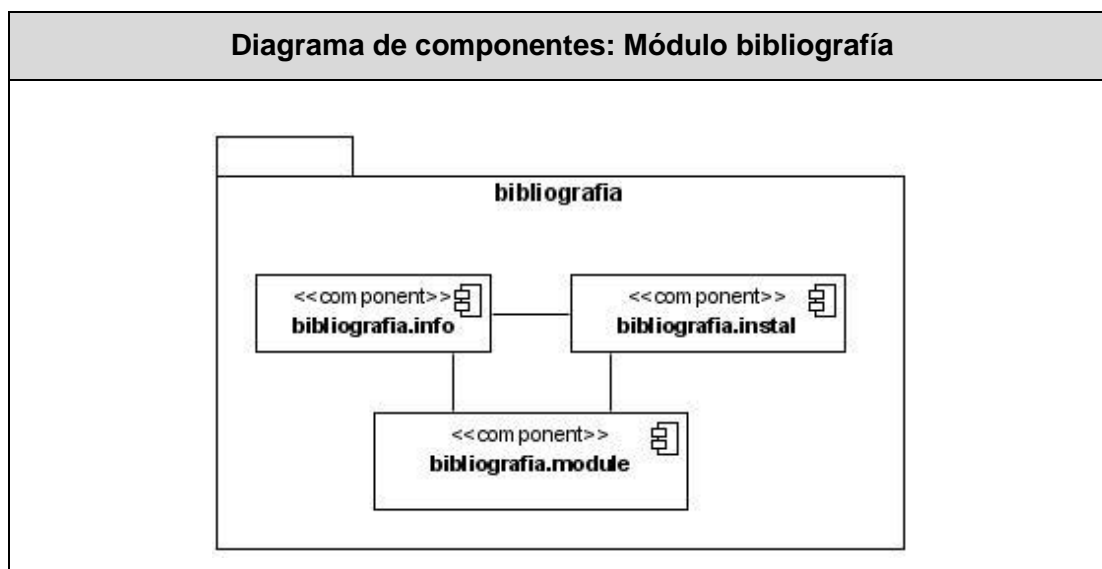
Anexo 1. Diagrama de Componentes del sistema.

Figura 5 A.1 Diagrama de Componentes del sistema.



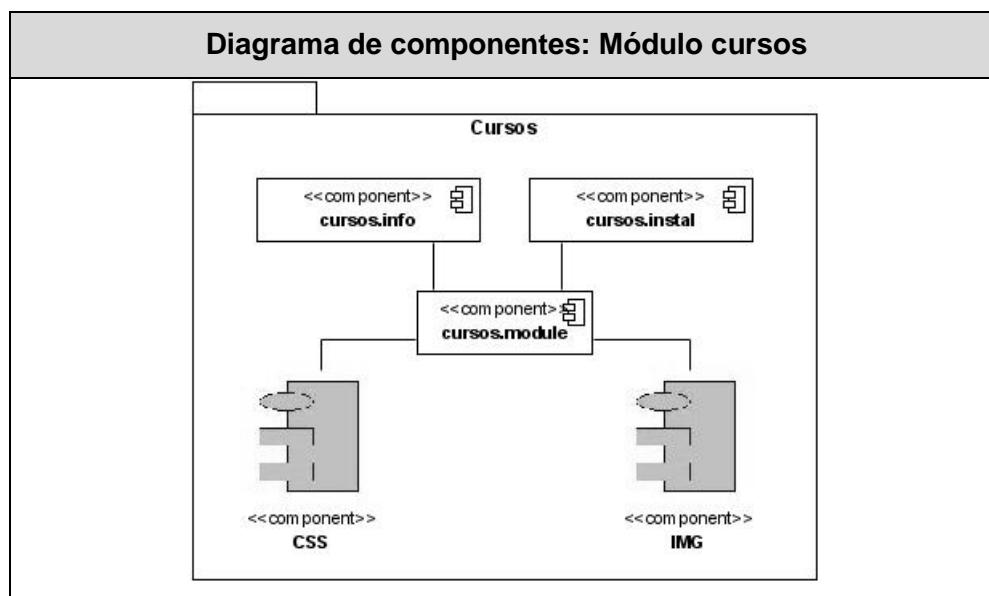
Anexo 2. Diagrama de Componentes: Módulo bibliografía.

Figura 6 A.2 Diagrama de Componentes: Módulo bibliografía.



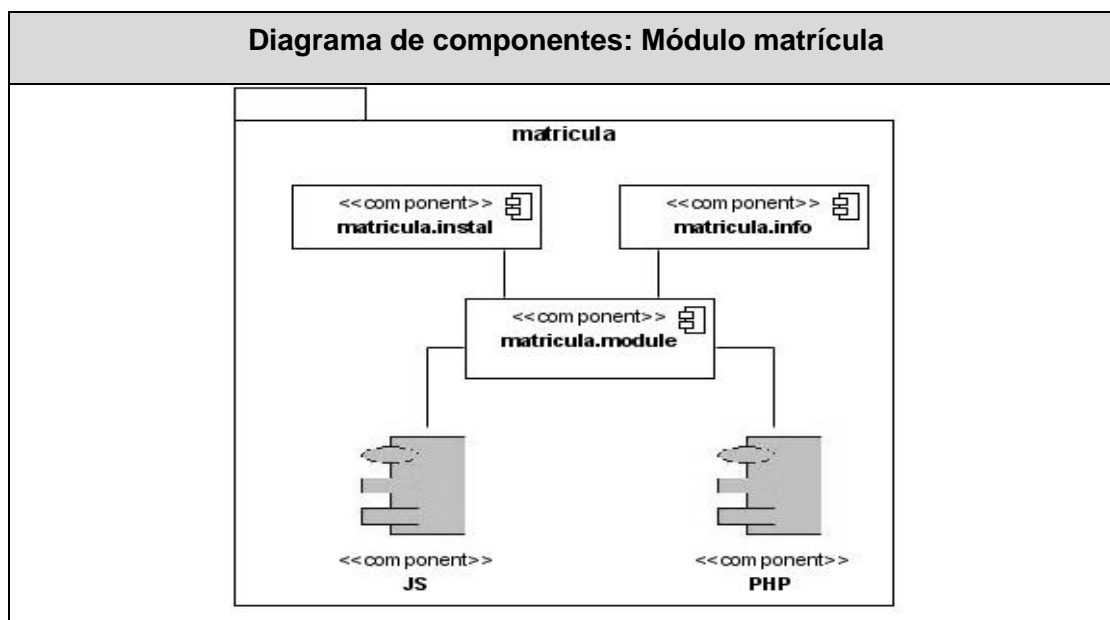
Anexo 3. Diagrama de Componentes: Módulo cursos.

Figura 7 A.3 Diagrama de Componentes: Módulo cursos.



Anexo 4. Diagrama de Componentes: Módulo matrícula.

Figura 8 A.4 Diagrama de Componentes: Módulo matrícula.



Anexo 5. Diagrama de Despliegue.

Figura 9 A.5 Diagrama de Despliegue.

