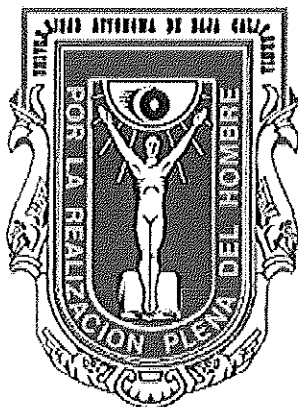


UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS



MODULO DE CURSOS PARA PLATAFORMA PROPIA DE UNIVERSIDAD
VIRTUAL

TESIS

Que para obtener el título de

Licenciado en Ciencias Computacionales

presenta:

Renee Torres Gutiérrez

Ensenada, Baja California, Noviembre de 2002

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS

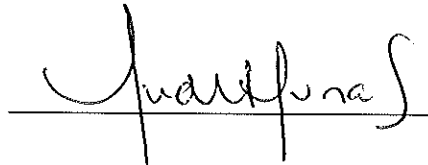
"Módulo de cursos para plataforma propia de universidad virtual"

TESIS PROFESIONAL

QUE PRESENTA

RENEE TORRES GUTIERREZ

APROBADO POR:



MIS. JUDITH ISABEL LUNA SERRANO

Presidente del Jurado



M.A.I. OMAR ÁLVAREZ XOCHIHUA

Secretario



DR. FERNANDO ROJAS IÑIGUEZ

1er vocal

RESUMEN de la tesis de Renee Torres Gutiérrez presentada como requisito parcial para la obtención del título de Licenciado en Ciencias Computacionales. Ensenada, Baja California, México. Noviembre de 2002.

“Módulo de cursos para plataforma propia de universidad virtual”

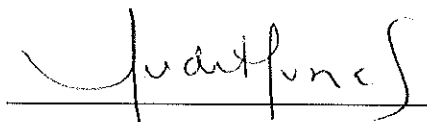
Este trabajo es parte de el proyecto “Desarrollo de una plataforma propia UABC-Universidad Virtual”. El sistema pretende apoyar el modelo de educación en línea de la Universidad Autónoma de Baja California. El desarrollo del sistema esta basado en un modelo educativo centrado en el estudiante, donde el docente se convierte en un facilitador del conocimiento, por medio del modelo pedagógico constructivista descrito por Marzano y colaboradores.

La herramienta se construyo bajo este paradigma; Las características del módulo Cursos visión alumnos reúne varias aplicaciones, se divide en cuatro áreas: 1) Información General: del curso y del docente, 2) Actividades: Acceso a actividades, exámenes asignados y productos. 3) herramientas: material del curso (lecturas, enlaces a sitios, documentos en archivos), Materiales de referencia. 4) Configuración.

La lógica que sigue apoya tanto a docente como a los estudiantes. Los requerimientos fueron proporcionados por docentes y académicos-administrativos (entrevistas y cuestionarios). Este documento muestra el análisis, diseño e instrumentación del módulo.

En el desarrollo del sistema se utiliza el lenguaje de modelado unificado (UML) para el análisis y diseño del sistema. Para su implementación se utilizo la tecnología de Web PHP así como el manejador de bases de datos (SMBD) MySQL y el servidor de Web Apache.

Resumen aprobado por:



M.I.S. Judith Isabel Luna Serrano

Director de Tesis

Dedicatorias

A Dios

Por darme la vida, por permitir que nada me falte, y hacer posible esta etapa de mis estudios.

A mis padres

René Alejandro Torres Lira y Graciela Gutiérrez García por su amor, sus enseñanzas, su apoyo, por estar siempre conmigo, por impulsarme a superarme en todos los aspectos y por los valores que me han inculcado, gracias.

A mis Hermanos

A mis hermanos Ale, Oziel, y Osmin, por su cariño, amistad y consejos, por estar siempre unidos en todos los momentos.

A Hiroshi

Por estar conmigo, por su amor y cariño, por su apoyo brindado en todo momento.

Agradecimientos

A MIS. Judith Luna Serrano, mi director de tesis, por brindarme la oportunidad de trabajar con ella, por que me brindo su ayuda para poder terminar esta etapa. Y por su amistad.

A los miembros del Comité de Tesis Dr. Fernando Rojas Iñiguez y MAI. Omar Álvarez Xochihua, por sus comentarios y aportaciones que permitieron mejorar este trabajo.

A Leopoldo Moran y Solares Y José Ignacio Asensio por sus consejos y amistad .

A mis compañeros y amigos de la carrera, especialmente a los G2K: Carolina Yescas, Vanesa López, Claudia Preciado, Claudia Chavarin, Arlene Aviles, Lupita salgado, Marco Antonio Barbosa, Javier Huerta, Julio C. Márquez, Miguel Pérez, José L. Estrada, por su valiosa amistad.

CONTENIDO

I	INTRODUCCIÓN	1
I.1	MODELO EDUCATIVO	3
II	ANTECEDENTES.....	6
III	OBJETIVOS	8
III.1	OBJETIVO GENERAL.....	8
III.2	OBJETIVOS ESPECÍFICOS	8
III.3	METODOLOGÍA DE INGENIERÍA DE SOFTWARE	9
III.4	LENGUAJE UNIFICADO DE MODELADO (UML)	11
IV	ANÁLISIS	13
IV.1	ANÁLISIS DE REQUERIMIENTOS	13
IV.2	ANÁLISIS DEL SISTEMA	14
IV.2.1	Casos de Uso.....	15
IV.2.2	Análisis de los Casos de Uso.....	20
V	DISEÑO	26
V.1	DISEÑO DEL SISTEMA	26
V.1.1	Diagramas de Secuencia	26
V.1.2	Diagramas de Actividad.....	31
V.1.3	Diagramas de Paquetes	32
V.2	DISEÑO DE LA BASE DE DATOS	38
V.2.1	Especificación de las tablas	39
VI	DESARROLLO.....	40
VI.1	INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR.....	40
VI.2	INSTRUMENTACIÓN DE LA BASE DE DATOS.....	40
VI.3	INSTRUMENTACIÓN DE LA INTERFAZ DE USUARIO (GUI)	41
VI.4	INSTRUMENTACIÓN DEL MÓDULO DE CURSOS VISIÓN-ALUMNOS	41
VII	FUNCIONALIDAD DEL SISTEMA	43
VIII	RESULTADOS.....	47
VIII.1	USABILIDAD DEL SISTEMA UABC-UV	49

VIII.1.1	<i>Plan de pruebas</i>	49
VIII.1.1.1	Prueba de usabilidad del sistema UABC-UV	50
VIII.1.1.2	entrevistas estructuradas	50
VIII.1.1.3	Sesión de laboratorio	50
VIII.1.2	<i>Resultados de las pruebas</i>	51
VIII.1.2.1	Entrevista a estudiantes	51
IX	DISCUSIÓN	52
X	CONCLUSIÓN Y TRABAJO FUTURO	57
X.1.1	<i>Trabajo Futuro</i>	58
	REFERENCIAS	59
	ANEXO A	61
	ANEXO B	65
	ANEXO C	70
	ANEXO D	74
	ANEXO E	83

FIGURAS

Figura 1. Ciclo de vida de cascada.....	9
Figura 2. Estructura global del sistema Universidad Virtual UABC.....	14
Figura 3. Diagrama de casos de uso de acceso al sistema.....	15
Figura 4. Diagrama de caso de uso consulta de información general.....	16
Figura 5. Diagrama de caso de uso Actividades y productos.....	18
Figura 6. Diagrama de caso de uso herramientas.....	19
Figura 7. Diagrama de caso de uso Configuración.....	20
Figura 8. Diagrama de análisis del caso de uso acceso al sistema.....	21
Figura 9. Diagrama de análisis del caso de uso información del curso.....	22
Figura 10. Diagrama de análisis de casos de uso documentos.....	23
Figura 11. Diagrama de análisis de casos de uso consulta y entrega de tareas.....	24
Figura 12. Diagrama de análisis de casos de uso configuración de contraseña.....	25
Figura 13. Diagrama de secuencia acceso al sistema.....	27
Figura 14. Diagrama de secuencia consulta de información del curso.....	28
Figura 15. Diagrama de secuencia consulta de documentos.....	29
Figura 16. Diagrama de secuencia consulta y entrega de actividades.....	30
Figura 17. Diagrama de secuencia cambio de contraseña.....	31
Figura 18. Diagrama de actividades para la consulta de anuncios.....	32
Figura 19. Diagrama general de paquetes.....	33
Figura 20. Diagrama de paquetes interfaz.....	34
Figura 21. Diagrama de paquetes Consulta de información.....	35
Figura 22. Diagrama de paquetes Actividades y Productos.....	36
Figura 23. Diagrama de paquetes Herramientas de apoyo.....	37
Figura 24. Diagrama de paquetes Configuración.....	37
Figura 25. Diagrama de entidades.....	38

Figura 26. Arquitectura que muestra la integración entre el servidor de Web, PHP y MySQL.....	42
Figura 27. Pantalla de bienvenida al sistema.....	43
Figura 28. Pantalla de consulta de información del curso.	44
Figura 29. Pantalla de obtención de documentos.....	45
Figura 30. Pantalla consulta y entrega de actividades.....	46
Figura 31. Pantalla configuración.....	46
Figura 32. Diagrama de análisis de casos de uso Proyectos.....	66
Figura 33. Diagrama de análisis del caso de uso consulta de Información del docente.....	67
Figura 34. Diagrama de análisis del caso de uso consulta de enlaces.....	68
Figura 35. Diagrama de análisis del caso de uso consulta de enlaces.....	69
Figura 36. Diagrama de secuencia consulta de información del docente.....	71
Figura 37. Diagrama de secuencia consulta y entrega de proyectos.....	72
Figura 38. Diagrama de secuencia consulta de enlaces.....	73
Figura 39. Pantalla de consulta de información del curso.....	84
Figura 40. Pantalla consulta y entrega de actividades.....	85
Figura 41. Pantalla de consulta de calificaciones.....	86
Figura 42. Pantalla de consulta de Anuncios.....	86
Figura 43. Pantalla de consulta de Enlaces.....	87

TABLAS

Tabla I. Tabla de la entidad Curso.	75
Tabla II. Tabla de la entidad Alumno.	76
Tabla III. Tabla de la entidad Alumno_curso.	76
Tabla IV. Tabla de la entidad Docente.	77
Tabla V. Tabla de la entidad Docente_curso.	77
Tabla VI. Tabla de la entidad criterios de evaluación.	77
Tabla VII. Tabla de la entidad Unidades.	78
Tabla VIII. Tabla de la entidad Topicos.	78
Tabla IX. Tabla de la entidad Documento.	79
Tabla X. Tabla de la entidad Tarea.	79
Tabla XI. Tabla de la entidad Buzon.	80
Tabla XII. Tabla de la entidad Proyecto.	80
Tabla XIII. Tabla de la entidad Alumno_proyecto.	81
Tabla XIV. Tabla de la entidad Comentario Proyecto.	81
Tabla XV. Tabla de la entidad Anuncio.	81
Tabla XVI. Tabla de la entidad Enlaces.	82

I INTRODUCCIÓN

Entre las aportaciones y posibilidades de Internet en la sociedad actual se pueden mencionar las tres funcionalidades básicas de la red que inciden más en la educación: Información, comunicación y soporte didáctico (<http://educadis.com.ar/intern.html>), describiendo algunos ejemplos concretos de su aplicación en los procesos de enseñanza autogestiva se puede mencionar; clases, documentar trabajos, correspondencia electrónica, edición de páginas WEB, apuntes en línea, tutoría telemática, entre otros.

Los cursos universitarios impartidos a través de las redes de cómputo, denominados Cursos en Línea, se popularizan cada vez más. Las redes de cómputo presentan un medio ideal no solo para la transmisión de información, sino también para la discusión significativa y el aprendizaje dialógico y grupal: sin embargo, debido a que su incursión en la educación es reciente la literatura en este campo es escasa (Pérez, 2000). Se propone que el análisis de la educación a distancia en salones virtuales requiere para fundamentar sus prácticas educativas, de apoyo teórico sustancialmente diferentes a los de la educación a distancia tradicional (Keegan, 1995). Se enfatiza que el estudio de los salones de clases electrónicos es un campo importante y complejo, todavía incipiente, con una contribución única que aportar al conocimiento educativo. (Pérez *op. cit.*) . .

El termino de enseñanza en línea se refiere a la enseñanza impartida cien por ciento a través de una red de cómputo. El calificativo "en línea" significa de forma estricta, la conexión de una computadora con un servidor; en el campo de la educación se utiliza para denotar el trabajo interactivo realizado no solo en una computadora local, sino en conjunto, por lo menos, con una computadora remota (el servidor).

La educación en línea se fundamenta en una concepción constructivista del aprendizaje (O' Malley 1994). Entre los modelos de enseñanza en línea actuales destacan dos: el de

Paulsen (1998) y el de Duchastel (1997). El primero, denominado modelo de enseñanza basada en la comunicación mediada por computadora (*CMC based teaching system*) propone que la enseñanza en este medio se basa en una inter-relación de factores sobre los estudiantes, actividades y técnicas pedagógicas adecuadas; plantea que toda situación de aprendizaje que utiliza métodos y técnicas e instrucciones, a través de dispositivos electrónicos para transmitir los mensajes sobre la instrucción entre maestros y alumnos, separados geográficamente, es un sistema electrónico de educación a distancia. Se involucra al estudiante, al maestro, los contenidos, así como los métodos y técnicas de instrucción y los dispositivos electrónicos utilizados (Pérez, 2000).

Duchaster (1997) parte de la riqueza de información proveída hoy en día por el medio para el desarrollo de las estructuras cognitivas de los estudiantes de los distintos campos del conocimiento; argumenta y fundamenta su propuesta en los cambios del funcionamiento del maestro. Se refiere al paso de la educación centrada en el profesor a una centrada en el estudiante, optimizando la capacidad de la red para la enseñanza.

Por lo tanto, la capacidad de la Red aunada a las aplicaciones tecnológicas diseñadas para sus ambientes de aprendizaje la convierten potencialmente en un medio instruccional eficiente y eficaz sobre todo en el ámbito universitario. El desarrollo de tecnología desde el punto de vista de la interacción entre participantes integradas a los cursos en línea ofrece, en principio, condiciones idóneas para promover aprendizajes significativos (Pérez y Rueda 2000).

Con este tipo de aplicaciones las tecnologías basadas en el Internet representan una oportunidad, para las instituciones interesadas en educación a distancia. La rápida proliferación de estas tecnologías hace necesario el desarrollo de herramientas de educación a distancia basadas en objetivos educativos considerando los recursos disponibles.

Esta tesis forma parte del proyecto de investigación "Desarrollo de una plataforma propia UABC-UV" apoyado con la 6ta convocatoria Interna de Investigación UABC. Se propone el desarrollo del módulo de cursos visión alumnos para el sistema universidad virtual de la Universidad Autónoma de Baja California (UABC).

El desarrollo del sistema plataforma propia UABC-UV, se caracteriza por brindar herramientas necesarias para que la universidad pueda ofrecer educación en línea, está dividido en tres distintas visiones: 1) visión del administrador, enfocado al desarrollo de aplicaciones necesarias para agregar usuarios y cursos al sistema, así como para el mantenimiento de éste; 2) visión del docente, orientado al desarrollo de aplicaciones que permite al docente impartir un curso en línea y 3) visión del alumno, dirigido al desarrollo de aplicaciones que permite a un alumno tomar un curso en línea.

Con este ultimo se ofrecerán herramientas de apoyo para que los usuarios tengan oportunidad de cursar asignaturas de manera remota a través de una computadora. El sistema tiene opciones de consulta variadas para el soporte de la materia, tales como documentos, enlaces y mensajes, además del manejo de tareas y proyectos, los cuales incluyen entrega remota de archivos.

I.1 Modelo Educativo

Las actividades de aprendizaje son aquellas que se deben consumir para cubrir un proceso de enseñanza optimo. Estas actividades se deben llevar a cabo independientemente de la manera en que se apliquen, sea de forma presencial, por computadora, impresiones, video o audio.

En el modelo educativo elegido por la UABC este proceso se caracteriza por la asunción de un papel fundamental por parte del alumno: 1) Propicia que el alumno se convierta en responsable de su propio aprendizaje, que desarrolle las habilidades de buscar, seleccionar, analizar y evaluar la información, asumiendo un papel más activo en la construcción de su propio conocimiento. 2) Conduce a que el alumno asuma un papel participativo y colaborativo en el proceso a través de actividades que le permitan exponer e intercambiar ideas, aportaciones, opiniones y experiencias con sus compañeros, convirtiendo así la vida del aula en un foro abierto a la reflexión y al contraste crítico de criterios y opiniones. 3) Sitúa al alumno en contacto con su entorno para intervenir social y profesionalmente en él a través de actividades como trabajar en proyectos, estudiar casos y proponer solución a problemas. 4) Compromete al alumno con su proceso de reflexión sobre lo que hace, cómo lo hace y qué resultados logra, proponiendo también acciones concretas para su mejoramiento.

En suma, este modelo educativo conduce al estudiante al desarrollo de la autonomía, del pensamiento crítico, de actitudes colaborativas y sociales, de destrezas profesionales y de la capacidad de auto evaluación.

"Por otro lado respondiendo al compromiso establecido en el Plan de Desarrollo Institucional 1999 – 2001 de la UABC, de ampliar las oportunidades educativas a los diferentes sectores de nuestra población y atender la demanda de servicios universitarios, preferentemente a personas que por su situación geográfica y socioeconómica no pueden asistir a las instituciones educativas escolarizadas se implementó el Sistema de Educación en Línea (<http://www.uabc.mx/dqaa/ed1.html>)

Sistema a través del cual la UABC, amplía y diversifica su oferta educativa, propiciando nuevas y variadas formas de aprendizaje, que además de salvar las limitaciones del tiempo y el

espacio, genera las condiciones para hacer de la educación un proceso activo y formativo donde convergen tanto los conocimientos científicos y tecnológicos como el desarrollo de los valores humanos.

La universidad gracias a su peculiaridad de ser simultáneamente la sede natural de la enseñanza y la investigación tiene la posibilidad de definir, los nuevos medios y modos para comunicar el saber, utilizando las nuevas tecnologías informáticas y telemáticas, para activar nuevos procesos de aprendizaje flexibles y diversificados, de ejercer un rol protagónico en la innovación de sus productos, sus procesos y estructuras.

El Sistema de Educación Abierta y a Distancia tiene como sustento un modelo flexible basado en la investigación que permite mejorar de manera continua la calidad de los programas educativos, centrarlos en las necesidades reales de los estudiantes y seleccionar los medios tecnológicos apropiados a cada programa.

II ANTECEDENTES

Actualmente existen universidades virtuales que ofrecen cursos a distancia a estudiantes situados alrededor del mundo, como es el caso de *Open Learning Agency*, el cual es un líder educacional completo y único, que provee un amplio rango de oportunidades de educación formal e informal para sus estudiantes. Ofrece programas internacionales o recursos de aprendizaje disponibles en 12 países distintos. Su visión es crear aprendizaje de calidad disponible para los estudiantes a cualquier hora, en cualquier lugar.

El *Simon Fraser University* ofrece, a través del *Centre for Distance Education*, uno de los programas más grandes de educación a distancia en Canadá. El sistema, *virtual-U*, está diseñado para utilizarse en el Web, y se ejecuta en casi todas las plataformas computacionales. Incluye características como foros, chat, estructuración de cursos, herramientas para asignación de calificaciones, soportando el aprendizaje colaborativo y métodos de enseñanza. En seguridad cuenta con diversos niveles de acceso para proveer a docentes y estudiantes la privacidad necesaria para el manejo de herramientas de calificaciones, asignación de trabajo y archivos personales. (<http://virtual-u.cs.sfu.ca/>)

La UABC ha instrumentado proyectos de investigación aplicados al desarrollo de herramientas para la educación en línea: en la Facultad de Ciencias en 1997, se construyó el banco de notas de cursos en forma de documentos HTML estáticos, los cuales pueden ser accedidos por los estudiantes a través de Internet (http://fciencias.ens.uabc.mx/notas_cursos/). En 1998, se elaboró un software de apoyo para los alumnos del curso de Matemáticas I, dicho sistema cuenta con las notas del curso así como un conjunto de ejercicios para que el alumno se auto evalúe. Posteriormente en 1999, se desarrolló el proyecto Material de Apoyo Docente en Línea (MADL). El cual está formado por 4 módulos: Banco de Notas de Curso, Sistema Global de Seguridad, Banco de Reactivos de Exámenes Tipo y Auto evaluaciones.

A partir de 1999 la UABC ha través de la Dirección General de Asuntos Académicos (DGAA) ha utilizado el software Virtual-U, desarrollado por la *Universidad Simon Frase* de Canadá en asociación con la Red de Tele aprendizaje de Centro de Excelencia (TL NCE) y con *Canarie* organización avanzada de desarrollo en Internet. Virtual-U, sistema descrito anteriormente en esta sección, es un campus virtual para la enseñanza en línea, formado por un conjunto de herramientas basadas en principios pedagógicos anglosajones para apoyar la estructuración de cursos en línea.

En Marzo del 2001 la Dirección General de Asunto Académicos (DGAA), solicito la instrumentación de un sistema de soporte para la educación en línea de la UABC, que se adecuara a las necesidades y características de los cursos impartidos en la universidad, basándose en los principios pedagógicos en los que se asienta.

En agosto del 2001 se inició el proyecto "Desarrollo de una herramienta propia Universidad Virtual" con apoyo de la 5ta convocatoria interna de investigación, sistema propuesto para la impartición de cursos en línea en la UABC. Desarrollar una plataforma particularizada con base en los requerimientos propios de los procesos educativos y administrativos de la UABC, permite, por un lado, satisfacer las necesidades propias de la universidad que hayan sido identificadas; y por otro, que el sistema se integra completamente al Sistema Institucional de Información.

III OBJETIVOS

III.1 Objetivo General

Desarrollar un sistema que cubra la solicitud del módulo de cursos para el sistema Universidad Virtual UABC, en la visión del alumno. El cual de soporte a los alumnos que cursan materias en línea, brindando un conjunto de herramientas variadas que sustenten el seguimiento de actividades del alumno durante un curso.

III.2 Objetivos Específicos

Analizar, diseñar y desarrollar las aplicaciones que darán soporte a los cursos en línea, tales como obtención de información de los cursos y docentes, Consulta de documentos, Entrega remota de actividades asignadas, manejo de proyectos, Consulta de calificaciones y anuncios, consultar y agregar direcciones de Internet y Manejo de configuración personal. Diseñar e implementar la estructura de la base de datos para almacenar la información concerniente al módulo.

El sistema estará disponible para los usuarios a través de Internet, con soporte para un gran número de solicitudes a las que hay que responder con agilidad.

III.2.1.1.1 METODOLOGIA

III.3 Metodología de Ingeniería de software

Para llevar a cabo el desarrollo del sistema se siguió una metodología que permite el progreso de la instrumentación del sistema de una forma ordenada, para esta etapa del desarrollo, se aplicó el modelo de ciclo de vida "método de cascada" (figura 1), que es un conjunto de procedimientos por etapas secuenciales y ordenadas para la dirección de un proyecto, desde el inicio lógico del sistema hasta la etapa de pruebas. Con la utilización de este método, se realiza una revisión final en cada fase del sistema para determinar si se está listo para avanzar a la próxima fase (McConnell. 1999).

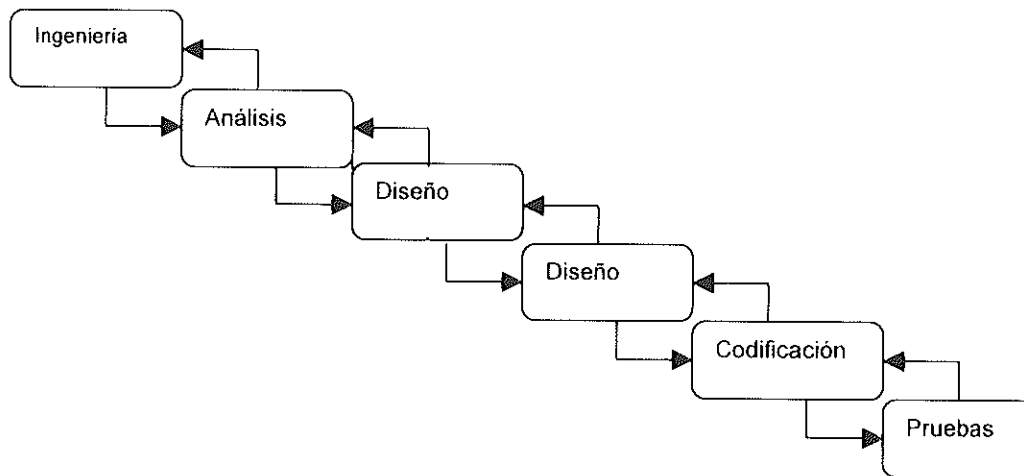


Figura 1. Ciclo de vida de cascada.

Los pasos que se siguieron dentro del ciclo de desarrollo del sistema son los siguientes:

1. Análisis: durante la etapa del análisis se exploran los requerimientos del sistema, y se busca obtener una comprensión mas precisa de estos.

1.1. Diagramas de casos de uso: utilizamos los diagramas de casos de uso para plasmar los objetivos particulares del sistema, tal y como son captados por el usuario.

Un objetivo particular y por lo tanto un caso de uso, es la entrega remota de archivos de tareas o trabajos.

1.2. Análisis de casos de uso: un diagrama de análisis de casos de uso no permite identificar las clases, sus atributos y operaciones. Siguiendo con el ejemplo anterior, para la entrega de una tarea se necesita que exista una tarea asignada, que se tenga el archivo del que se va a hacer entrega, y que haya un buzón para poder entregar. En donde la tarea y el buzón son 2 clases que tienen una relación.

2. Diseño: durante esta etapa logramos estructurar a partir de diagramas de análisis, el diseño del sistema construyendo un esquema para su desarrollo.

2.1. Diagramas de secuencia: este diagrama muestra los mensajes que son intercambiados entre los diferentes elementos del sistema y la secuencia de dichos mensajes.

2.2. Diagramas de clase: con este diagrama obtendremos las relaciones que existen entre los elementos que forman el sistema, así como su cardinalidad, se puede mencionar la clase curso y la clase alumnos las cuales tienen una relación de diez a muchos y de uno a muchos respectivamente, ya que un curso puede tener 10 ó más alumnos y un alumno puede tener de 1 curso a muchos.

2.3. Diagramas de paquete: utilizamos este diagrama para encapsular lógicamente las clases en paquetes y establecer dependencias entre estos.

3. Diseño de la base de Datos

3.1. Mapeo del diagrama de clases a tablas: se crea la estructura de la base de datos a partir del diagrama de clases.

3.2 Normalización de las tablas: se analiza que las tablas sean consistentes, que no dupliquen información y que mantengan la integridad de los datos.

4. Instrumentación

4.1.Desarrollo de la base de datos: se crean las tablas que darán soporte para el almacenamiento de los datos.

4.2.Desarrollo de interfaz del usuario: se crean las paginas Web, que visualizará el alumno al acceder al sistema, y a través de las cuales se involucra con una materia en línea.

4.3.Desarrollo de interfaz WWW-BD: se crean los programas que se encargan de procesar los datos con los que alumno trabaja a través de la interfaz y de almacenarlos en la base de datos.

III.4 Lenguaje Unificado de Modelado (UML)

El presente trabajo se desarrolla sobre la base del Lenguaje de Modelado Unificado (UML), debido a que es un lenguaje estándar de modelado para software, que permite a los desarrolladores visualizar, especificar, construir y documentar las partes de su sistema. Debido a su notación se puede representar software complejo en forma de símbolos gráficos simples, lo que permite una mayor comprensión del software que se modela. (Rumbaugh, et al, 1998).

Un Diagrama es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo conexo de arcos (relaciones) y vértices (otros elementos del modelo). La mayoría de los diagramas de UML y algunos símbolos complejos son grafos que contienen formas conectadas por rutas. La información está sobre todo en la topología, no en el tamaño o la colocación de los símbolos (hay algunas excepciones como el diagrama de secuencia con un eje métrico de tiempo).

Dado que uno de los requerimientos del sistema es la ejecución en línea, se optó por utilizar los estereotipos de UML para modelar aplicaciones que se realizan en el Web. Estos estereotipos permiten representar páginas Web y otros elementos importantes en el modelo.

Las extensiones son expresadas en términos de estereotipos - permite crear nuevos tipos de bloques de construcción que se derivan de otros existente, sin embargo son específicos a un problema particular- y valores etiquetados, es decir, en términos de los mecanismos de extensión proporcionados por UML, así como la notación gráfica empleada para representar algunos de los estereotipos. Combinados, estos mecanismos nos permiten crear nuevos tipos de construcción de bloques que pueden ser usados en el modelo (Conallen, 2001).

IV ANÁLISIS

IV.1 Análisis de Requerimientos

El paso inicial en la etapa de desarrollo, fue especificar los requerimientos y el enfoque de las entrevistas, obtenidos durante el proceso de captura de requisitos. Durante esta fase de inicio, se identifican la mayoría de los casos de uso para delimitar el sistema, el alcance del proyecto y detallar los puntos más importantes. Esta depuración de requisitos nos ayuda a conseguir una visión suficiente del sistema para avanzar en los flujos de trabajo subsiguientes. Los requerimientos obtenidos son los siguientes:

- R1. Sistema en línea, accesado a través de un navegador.
- R2. Manejo de un gran número de solicitudes.
- R3. Agilidad de respuesta.
- R4. Manipulación de archivos remotos
- R5. Empleo de niveles de acceso por cuestiones de seguridad.
- R6. Manejo de tareas y proyectos
- R7. Acceso a fuentes de información proporcionadas por el docente(documentos, enlaces)
- R8. Consulta de información general de cursos y docentes
- R9. Configuración de correo y contraseña

Un requerimiento substancial del desarrollo del proyecto, era adaptar una estructura que permitiera dividir el sistema en módulos, sin crear dependencia de funcionalidad, solo debía existir la transferencia de información de un modulo a otro. Enseguida se describe un diagrama que representa la estructura entre los módulos del sistema (figura 2).

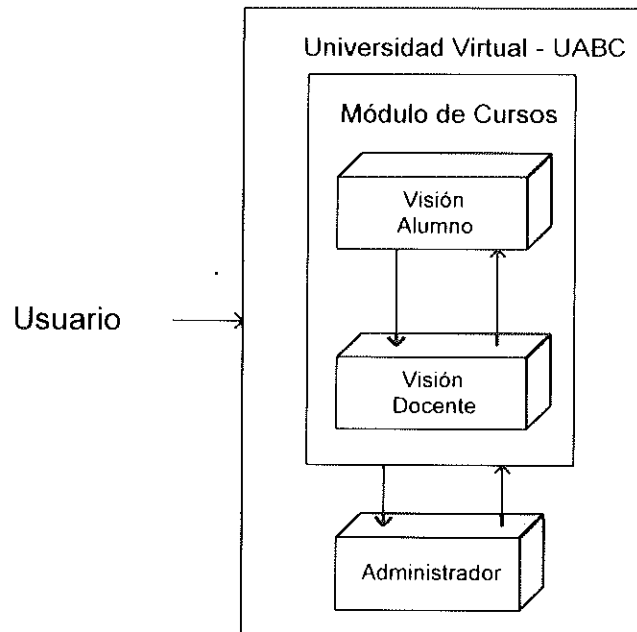


Figura 2. Estructura global del sistema Universidad Virtual UABC.

IV.2 Análisis del sistema

Una vez definidas las condiciones iniciales del sistema, a través del análisis de requerimientos, se comenzó el análisis del sistema. Durante esta etapa, analizamos los requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sean fácil de mantener y que nos ayude a conservar el sistema entero - incluyendo su arquitectura.

Durante esta etapa se desarrollan los diagramas de casos de uso y los diagramas de análisis de casos de uso.

IV.2.1 Casos de Uso

Los casos de uso son los que inician y dirigen el proceso de desarrollo, debido a que la mayoría de las actividades como el análisis, diseño y pruebas se llevan a cabo a partir de estos.

Los casos de uso modelan la funcionalidad del sistema tal como es percibido por los usuarios, llamados actores, de otra forma se dice que un caso de uso es en esencia, una interacción típica entre un usuario y un sistema de cómputo. (Booch, et al 1997). En esta etapa se describen los diferentes casos de uso que muestran la iteración del usuario con el sistema. Los casos de uso se agruparon, sometidos al tipo de actividad a realizar, es decir: acceso al sistema, consulta de información general, actividades del curso, herramientas y configuración personal. Los casos de uso son los siguientes:

Nombre: acceso al sistema (figura 3).

Actor: alumno.

Propósito: tener acceso al sistema.

Descripción: Al alumno se le despliega una forma donde debe introducir su nombre de usuario y contraseña del sistema para poder acceder a este.

Para que el alumno pueda hacer uso de las herramientas, es necesario ingresar al sistema, por lo que se incluye este caso de uso en los diagramas subsiguientes.

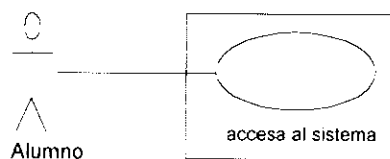


Figura 3. Diagrama de casos de uso de acceso al sistema.

Nombre: consulta de información general (figura 4).

Actor: Alumno

Propósito: brindar la opción de consultar información general de cursos en los que está inscrito y de docentes que lo imparten.

Descripción: Los datos desplegados al usuario en la consulta de información general son, del curso: escuela, carrera, modalidad, presentación general, equipo de trabajo, desarrollo por unidad, objetivo de unidad, problemas guía, actitudes, materiales, producto, tiempo, criterios de evaluación, bibliografía y acreditación. Acerca del docente: nombre, grado, escuela, carrera, dirección académica, horas de localización, tel./fax, correo, página personal, proyectos actuales y áreas de interés.

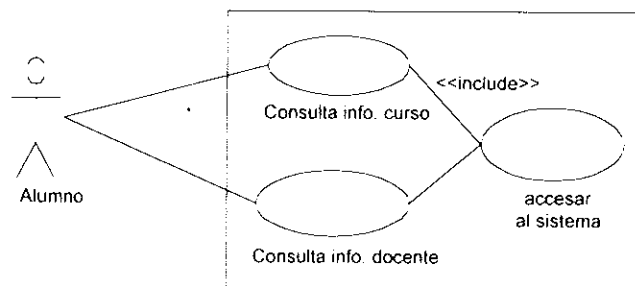


Figura 4. Diagrama de caso de uso consulta de información general.

Nombre: Actividades y Productos (figura 5)

Actor: alumno

Propósito: brindar aplicaciones para consultar trabajos asignados, hacer entrega remota de archivos y adelantos de proyectos, además de llevar seguimientos de los proyectos a desarrollar.

Descripción: Las diferentes actividades asignadas por el maestro pueden ser consultadas en una tabla organizada por criterios de evaluación.

Al consultar una actividad se muestra información descriptiva de la misma: nombre de la actividad, fecha de asignación, fecha y hora límite de entrega, autor, descripción, número de sobre-escrituras, además de un archivo adjunto, el cual es opcional, y pone a disposición de los alumnos documentos específicos: Ejemplo una descripción más amplia de la actividad a realizar o cuestionario.

Junto con cada actividad, se encuentra la aplicación que permite entregar de forma remota un archivo. El estado más general es la opción de subir la tarea, la cual permite entregar el archivo que contiene la respuesta a la tarea asignada.

El alumno puede entregar avances del proyecto de forma remota. Para esto se le presenta una forma con la que debe elegir el archivo o documento que va a entregar. El maestro revisa cada archivo que es entregado y puede o no asignarle un comentario. Este comentario es desplegado al alumno como retroalimentación. La información acerca del proyecto únicamente es presentada a quienes lo desarrollan.

Las calificaciones son presentadas por criterios de evaluación; estos criterios los especifica el docente al iniciar el curso, pueden ser n criterios con diferentes pesos porcentuales, la suma total de los pesos debe ser 100.

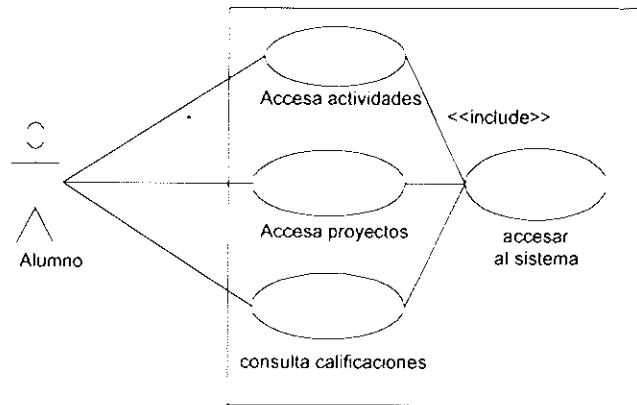


Figura 5. Diagrama de caso de uso Actividades y productos.

Nombre: consulta de información de apoyo (figura 6)

Actor: alumno

Propósito: brindar aplicaciones para recibir información útil en el seguimiento del curso, tales como: documentos, enlaces o anuncios.

Descripción: El alumno puede consultar y obtener documentos que el maestro a proporcionado como notas del curso. Estos documentos están organizados por unidades, y no se tiene un límite mínimo o máximo de documentos por unidad. De la tabla de documentos, el alumno debe seleccionar la liga que le proporciona la información del documento que desea consultar así como el documento mismo, sea para grabarlo o abrirlo inmediatamente. Se puede adquirir un documento las veces que se desee.

También puede consultar los anuncios, que estarán organizados por páginas de 5 anuncios cada una. Las páginas podrán accesorarse a través de ligas. Los anuncios tienen una clasificación, son de carácter normal o urgente, el carácter se distingue por el color del texto de la clasificación sea verde o rojo.

Otra opción son los Enlaces. Donde el alumno puede consultar los enlaces que son de interés para apoyar los temas del curso. En esta opción se despliega una lista de ligas de direcciones de Internet, al accesar cualquiera de las ligas, se abrirá otra ventana del navegador, para ingresar al sitio. Se pueden agregar nuevos enlaces al curso.

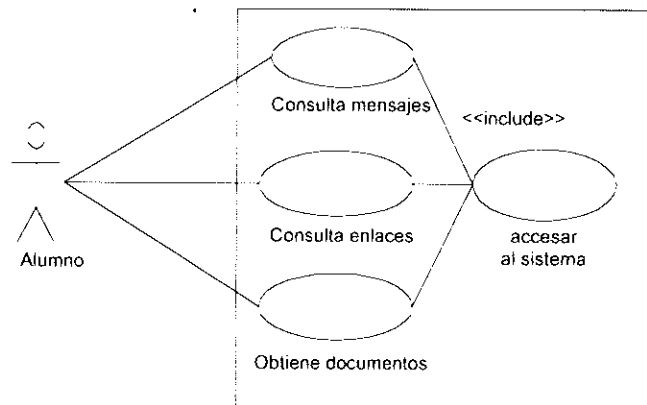


Figura 6. Diagrama de caso de uso herramientas.

Nombre: configuración personal(figura 7)

Actor: alumno

Propósito: Cambiar su contraseña o dirección de correo.

Descripción: al alumno se le permite cambiar contraseña y dirección de correo, con esta opción se muestran dos formas; Para cambiar la contraseña se despliegan 3 campos de texto: a) se introducen contraseña actual b) nueva contraseña y c) confirmación de contraseña.

Para cambiar la dirección de correo únicamente se introducirá la nueva dirección y el sistema procederá al cambio.

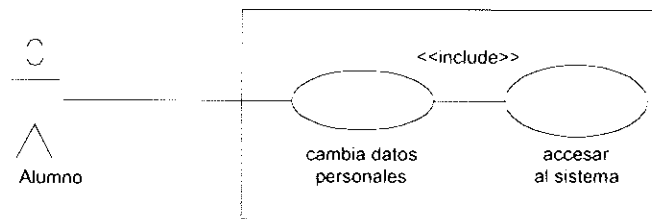


Figura 7. Diagrama de caso de uso Configuración.

IV.2.2 Análisis de los Casos de Uso

El análisis de casos de uso, es una actividad que se realiza cuando los casos de uso se han terminado. Los objetivos de realizar un análisis de casos de uso son: 1) identificar clases y objetos que realizarán el flujo de eventos de los casos de uso. 2) identificar responsabilidades, atributos y asociaciones de clases y 3) percibir el uso de mecanismos de construcción.

En esta etapa del análisis se utilizan tres estereotipos sobre clases: 1) *clases de interfaz*, se modelan la iteración entre el sistema y sus actores; 2) *las clases de control*, representan la coordinación, secuencia, transacciones y control de otros objetos y 3) *las clases de entidad*, se utiliza para modelar información (www.sparxsystems.com.au/).

Los diagramas de análisis de casos de uso, que muestran el comportamiento del sistema utilizando los estereotipos de clases se muestran enseguida. Al igual que los diagramas de casos de uso, se agruparon con los mismos criterios de funcionalidad: a) acceso al sistema, b) consulta de información general, c) actividades del curso, d) herramientas y e) configuración personal. Para consultar los diagramas de análisis de casos de uso que no se muestran en esta sección se pueden encontrar en el Anexo A.

El diagrama de análisis de caso de uso de acceso al sistema (figura 8) muestra el flujo de información que se genera cuando un usuario accesa al sistema. El alumno en el navegador,

introduce la dirección o URL, donde se le presenta una forma para proporcionar su nombre de usuario y contraseña. Los datos suministrados son validados. De ser valido el usuario, se le concede el acceso.

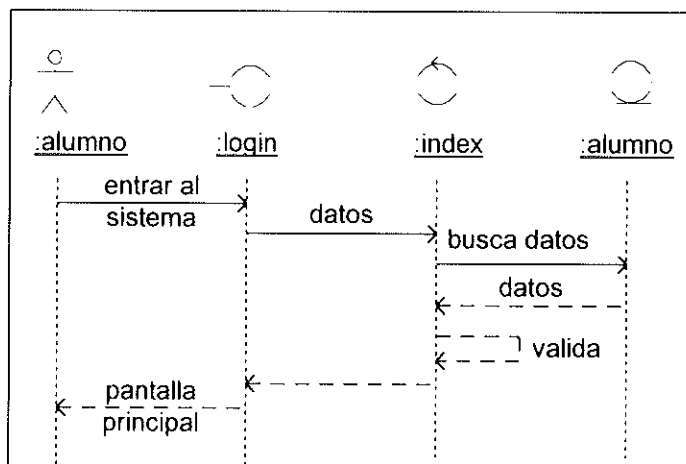


Figura 8. Diagrama de análisis del caso de uso acceso al sistema.

En el diagrama de información del curso (figura 9) se muestra el flujo de información que se crea cuando un usuario consulta la información general del curso. El alumno selecciona, en el menú principal, la opción de información del curso. :menu envía buscar los datos del curso a la BD. Los datos son desplegados al usuario.

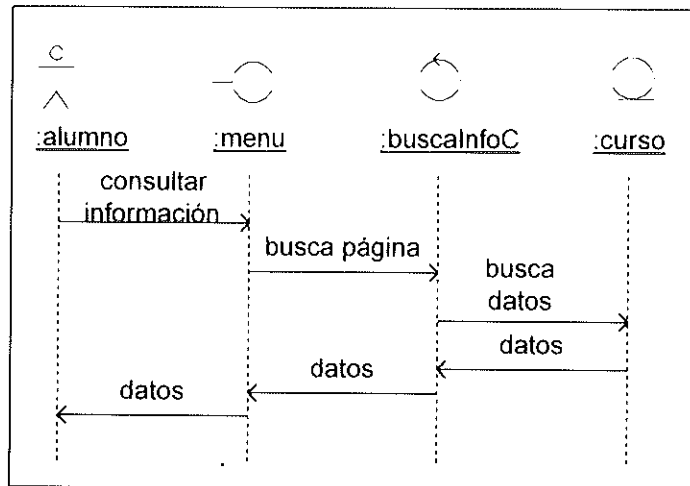


Figura 9. Diagrama de análisis del caso de uso información del curso.

En el diagrama de documentos (figura 10), el usuario elige desde :menu obtener los documentos. :Menú envía consultar los documentos en la BD, al recuperar los datos, son desplegados al usuario. Si el usuario quiere obtener un documento, :documento busca el archivo en el servidor y lo envía hasta la maquina del usuario.

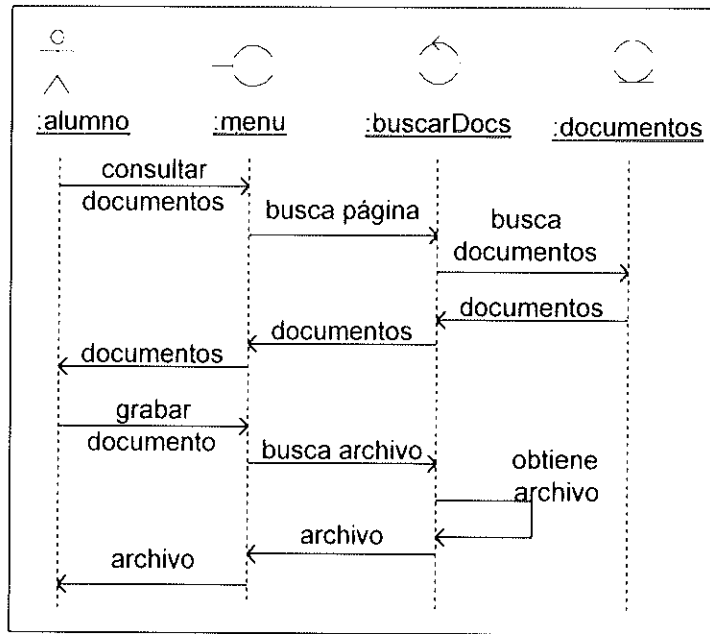


Figura 10. Diagrama de análisis de casos de uso documentos.

En el diagrama de análisis de casos de uso consulta y entrega de tareas (figura 11), muestra el flujo de información que surge cuando el usuario entrega una tarea. El alumno elige desde el menú la liga que le muestra las tareas existentes. :Menú envía buscar las tareas, que son desplegadas en una lista. El alumno elige a partir una tarea específica el archivo o documento que va a entregar. :tareas almacena el archivo en el servidor y envía mensaje al usuario de archivo entregado.

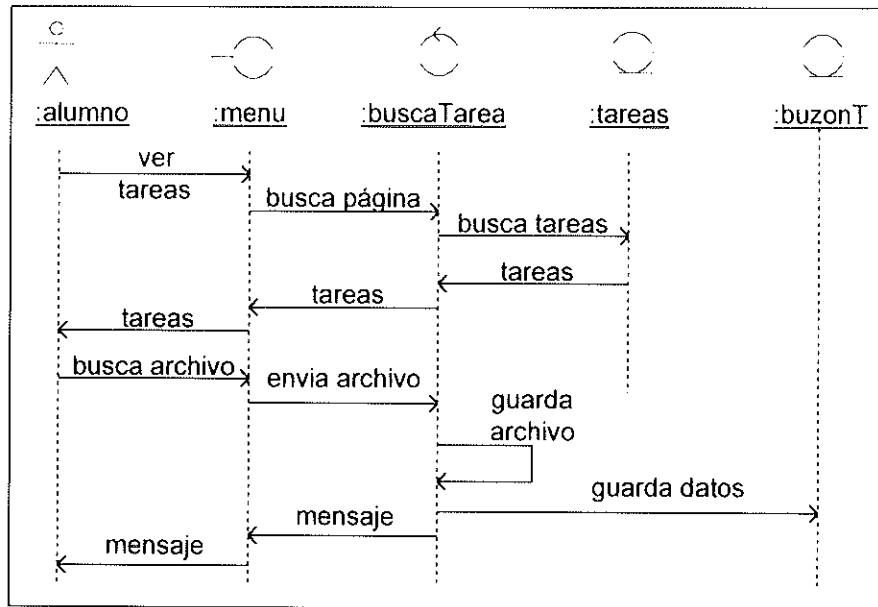


Figura 11. Diagrama de análisis de casos de uso consulta y entrega de tareas.

En el diagrama de caso de uso configuración de contraseña (Figura 12) se muestra el flujo de información que surge cuando el usuario va a cambiar su contraseña. Una vez hecha la petición al menú, éste le despliega una forma donde será introducirá la nueva contraseña. Al introducir la nueva contraseña :configuración guarda los datos. Cuando el usuario va a cambiar su dirección de correo debe introducir en una forma la nueva dirección y el sistema la modificará.

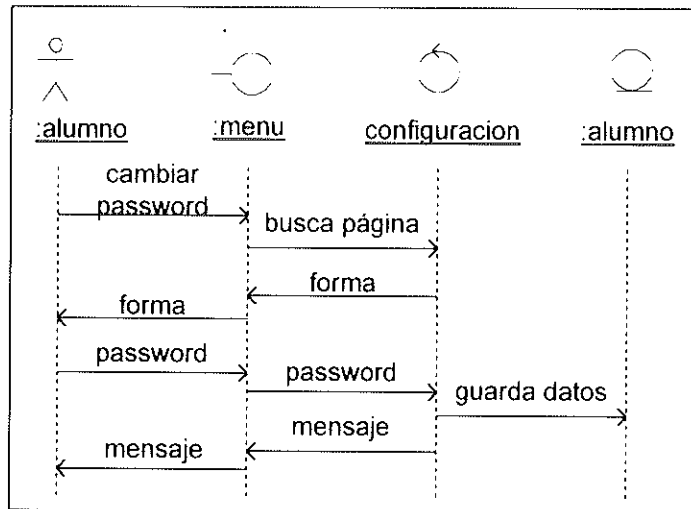


Figura 12. Diagrama de análisis de casos de uso configuración de contraseña.

En la etapa de análisis se examinaron los requerimientos acreditados. Con los casos de uso se definieron las iteraciones que tendrá un alumno con el sistema, a través de al interfaz de usuario. Estas interacciones generalmente son solicitudes al sistema para consultar información y muy pocas veces para procesarla.

Con los diagramas análisis de casos de uso, se obtuvo el flujo de datos que se genera cuando el alumno hace una petición al sistema. Las clases interfaz son el ambiente gráfico que se le presenta al usuario y, a través de él, se hacen las solicitudes al sistema, las clases controladoras se encargan de hacer consultas a las clases entidad y procesar los datos, por lo que las clases entidad representan el almacén de datos.

V DISEÑO

En la etapa de diseño, modelamos el sistema y encontramos su forma para que soporte todos los requerimientos. El resultado del diseño se esfuerza en detallar la estructura del sistema impuesta por el análisis y además sirve como esquema para la implementación. Una entrada esencial en el diseño es el resultado del análisis.

En esta etapa se utilizó la extensión de UML para modelar aplicaciones en el Web. Se muestran clases tipo: *frames*, formas, *targets*, base de datos, paginas servidor y páginas cliente. Del lado del servidor, en el Web, existen aplicaciones que son ejecutadas por el cliente, estas aplicaciones son transparentes para el usuario, por lo que no pueden ver el procesamiento de la información, éstas son llamadas páginas servidor. Del lado del cliente se despliegan documentos en formato HTML, y son desplegadas al usuario a través del navegador, éstas son llamadas páginas cliente.

Debido a que se conocía la herramienta que se utilizaría para el desarrollo del sistema, el diseño se realizó a un nivel bajo enfocándose al manejo de la tecnología de Web PHP, el cual se ejecuta bajo una arquitectura cliente servidor, disponible en Internet.

V.1 Diseño del sistema

V.1.1 Diagramas de Secuencia

Un diagrama de secuencia capta el comportamiento de los casos de uso y describe la secuencia de los mensajes que son intercambiados entre los diferentes roles que son implementados en el comportamiento del sistema.

Estos diagramas se muestran en un nivel de abstracción medio, por lo que no muestran en detalle los datos que fluyen cuando los métodos son aplicados a los objetos. A medida que

crece el nivel de abstracción se proporciona información que permite una mejor comprensión de cualquier sistema que se desarrolle. El diagrama de actividades, que se muestran más, muestra un nivel de abstracción de más bajo nivel. Algunos diagramas de secuencia del sistema se describen a continuación. Los diagramas de secuencia que no se muestran en esta sección se pueden consultar en el anexo B.

El usuario solicita la entrada al sistema (figura 13). Cuando el usuario introduce el URL del sistema en el navegador se despliega la ventana de acceso al sistema, donde se introducen el login y contraseña. Al presionar el botón, se manda llamar `validar()`, que verifica en la base de datos en la tabla `alumno`, que los datos proporcionados sean validos, De ser valido el usuario se brinda el acceso al sistema y se despliega la pantalla principal.

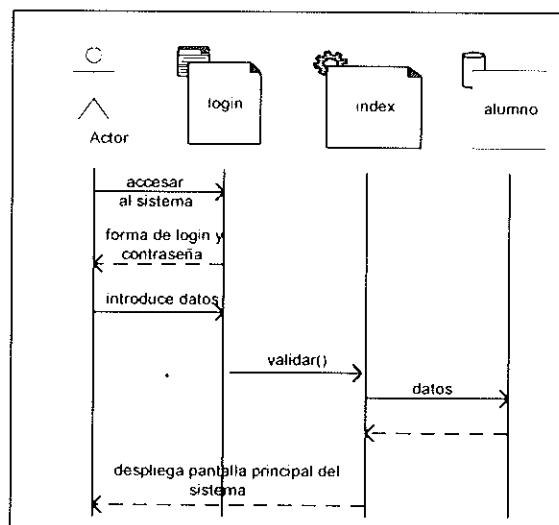


Figura 13. Diagrama de secuencia acceso al sistema.

El usuario solicita la consulta de información del curso (figura 14), a través del menú. EL menú manda llamar una página servidor llamada `infoCurso`, que ejecuta la función `Curso()`

esta función consulta en la base de datos la tabla curso. Cuando infoCurso() obtiene los datos crea una página cliente que despliega al usuario una tabla con la información del curso.

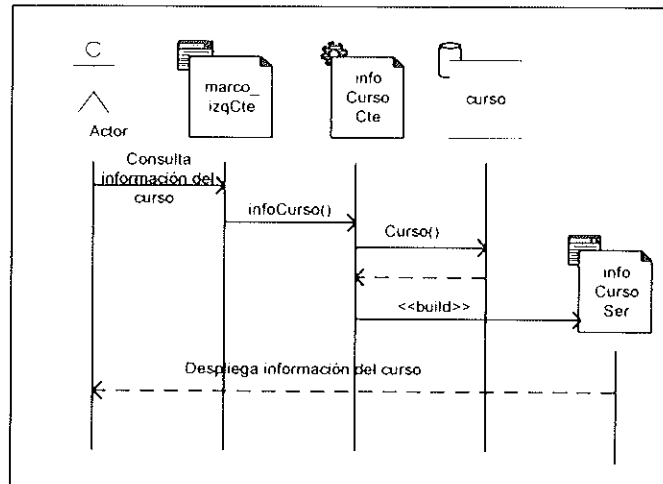


Figura 14. Diagrama de secuencia consulta de información del curso.

El usuario solicita la consulta de documentos (figura 15). EL menú manda llamar una página servidor, DocumentosSer, que ejecuta la función Documento() la cual consulta en la base de datos la tabla documentos. Con los datos obtenidos de la consulta se crea la página cliente que despliega al usuario la lista de documentos del curso. Al elegir el usuario un elemento de la lista se manda llamar DocsSer, el cual ejecuta la función Documento(), que consulta en la base de datos la información específica del elemento elegido de la lista. Cuando DocsSer recibe la información, crea la página cliente que despliega la información al usuario.

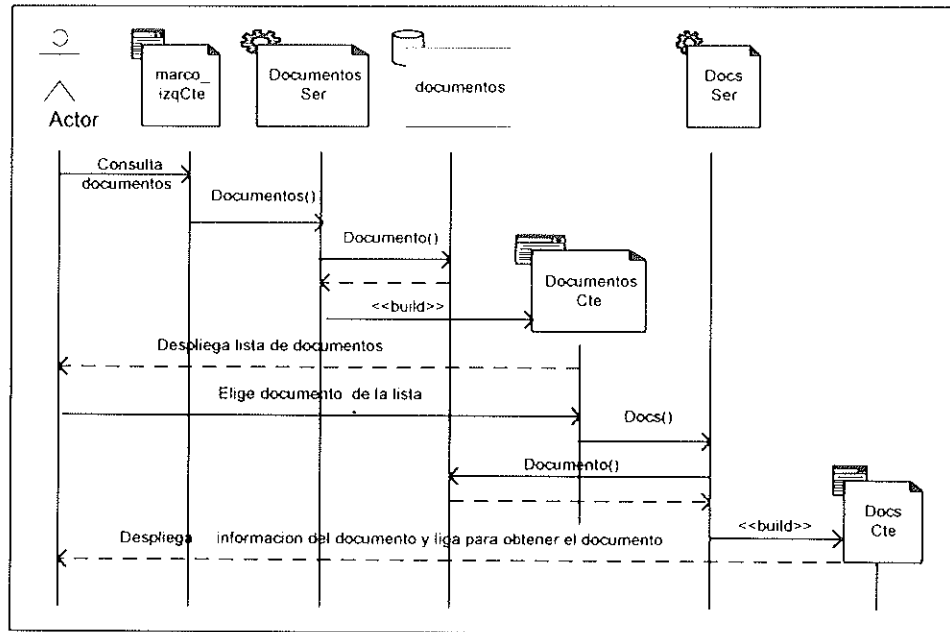


Figura 15. Diagrama de secuencia consulta de documentos.

El alumno consulta las actividades asignadas (figura 16). Se manda llamar `listaTipoActividadesSer`, que ejecuta la función `critérios()`, esta función consulta los criterios de evaluación del curso y las actividades asignadas a cada uno de ellos. El resultado de la consulta es desplegado en una lista a través de una página cliente. El usuario elige un elemento de la lista para acceder a la información de una actividad específica. Al elegir se manda llamar `listaActividadSer`, que ejecuta la función `Actividad()`, esta consulta de la base de datos la tabla `tareas`. `listaActividadSer` obtiene los datos de la consulta y crea una página cliente que despliega al usuario una tabla con datos de la actividad elegida.

Junto con la tabla de datos de una actividad, se despliega también el estado del buzón. El estado más general del buzón es cuando se presenta la forma de entrega de actividades. Para lo cual se elige el archivo desde el sistema de archivos local y se ejecuta el envío a través de un botón.

La acción del botón mandar llamar aSube() que almacena el archivo entregado en el servidor remoto y manda llamar la función actualizarBuzon(), esta actualiza la tabla buzón con los datos de la actividad almacenada. Posteriormente se crea una página cliente que envía un mensaje al usuario.

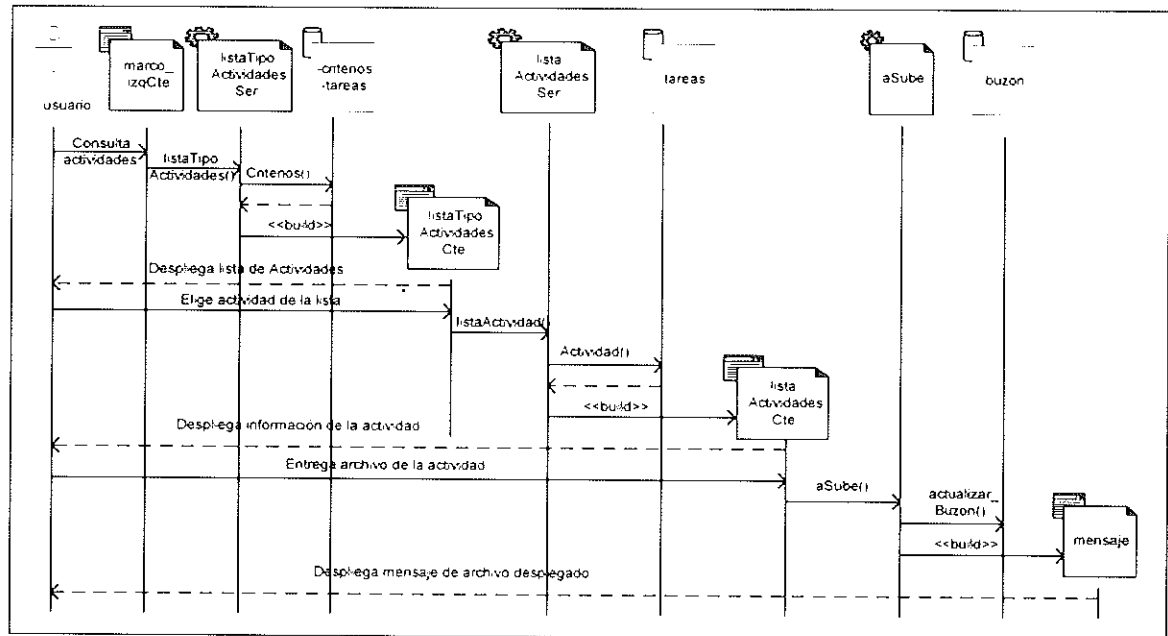


Figura 16. Diagrama de secuencia consulta y entrega de actividades.

El usuario solicita la opción de configuración (figura 17). Esta opción despliega una página cliente llamada Configuración que contiene 2 formas: una de contraseña y otra de dirección de correo. Para cambiar de contraseña se introducen los datos en la forma, y se manda llamar cambiaPassword(), que ejecuta la función password(), esta consulta la contraseña actual almacenada en la base de datos. Valida() recibe el resultado de la consulta y lo compara con la contraseña proporcionada en la forma, se ejecuta la función actualiza() que

actualiza la nueva contraseña en la base de datos y construye una página cliente que despliega un mensaje de operación realizada.

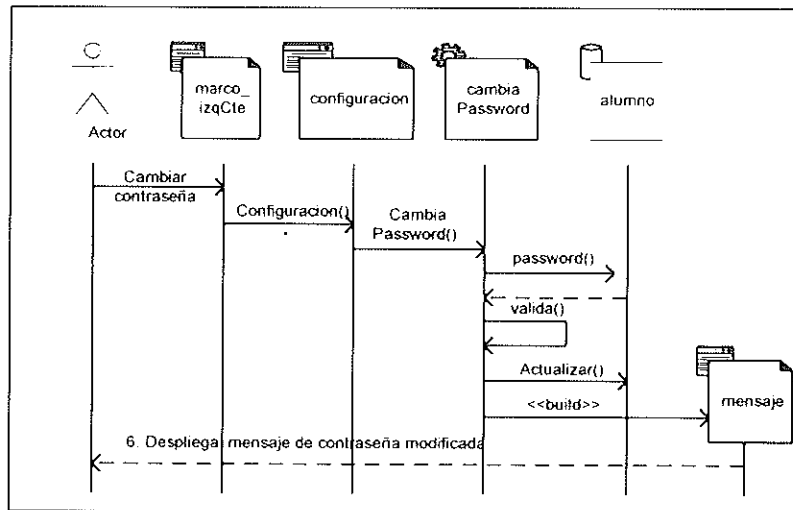


Figura 17. Diagrama de secuencia cambio de contraseña.

V.1.2 Diagramas de Actividad

Para mostrar un nivel de abstracción a un nivel mas bajo del sistema, se desarrolló un diagrama de actividades. Estos diagramas son particularmente útiles en conexión con el flujo de trabajo y para la descripción del comportamiento que tiene una gran cantidad de proceso paralelo. Además se pueden describir procesos más complicados y así ser mejor comprendidos. El diagrama de actividad realizado (figura 18), muestra la secuencia genera al ejecutar la aplicación que consulta los anuncios del curso, esta aplicación fue modelada debido a que es más compleja su ejecución.

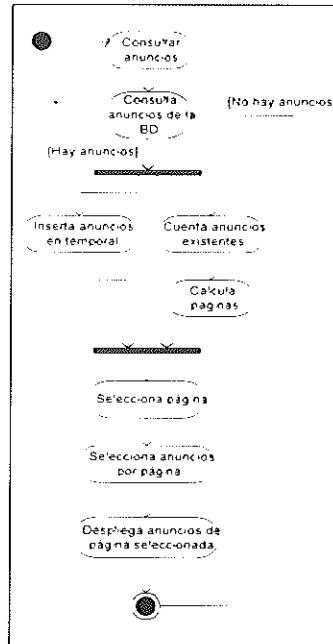


Figura 18. Diagrama de actividades para la consulta de anuncios.

V.1.3 Diagramas de Paquetes

El diagrama de clases describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones estáticas que existen entre ellos. Un diagrama de paquetes es un diagrama que muestra los paquetes de clases y sus dependencias entre ellos. Las clases del sistema se representaron en 5 grupos distintos, dependiendo de su funcionalidad. El diagrama de paquetes (figura19) muestra los diversos paquetes de clases formados y que se describen enseguida.

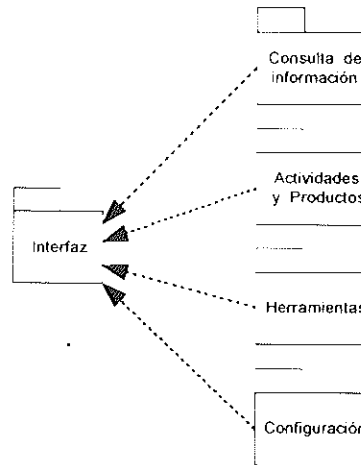


Figura 19. Diagrama general de paquetes.

En primer lugar se describe el diagrama de paquetes interfaz (figura 20), este diagrama contiene las clases que forman la interfaz de usuario. El diagrama esta formado por la clase *index*, que es una página servidor que revisa si el usuario es válido, las clases *frames1* y *frames2* crean la estructura física del sistema, y a la vez están formadas por las clases *arriba* y *mainFrame*, los cuales son páginas cliente que despliegan información. La clase *marco_izqSer* consulta la base de datos para obtener los cursos disponibles al usuario y crea una página cliente que contiene los cursos. La clase *mainFrame* es tipo *target*, y es donde es desplegada la información al usuario.

El diagrama de paquetes interfaz crea: menú de opciones, mapa de imágenes y pantalla principal de despliegue de información.

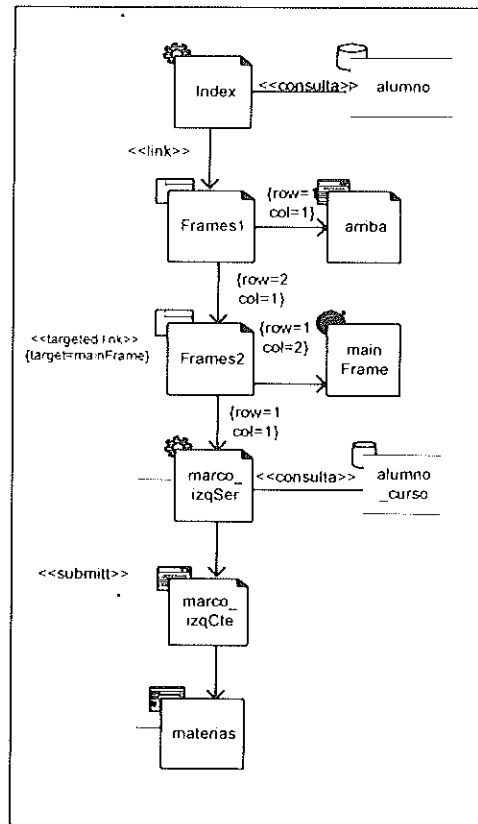


Figura 20. Diagrama de paquetes interfaz.

El diagrama de paquetes, consulta de información (figura 21) muestra las clases `lista_docenteSer`, `docenteSer`, `infoCursoSer` y `desarrolloSer`. Cada una de estas clases es una página servidor que consulta la base de datos y genera una página cliente llamada `lista_docenteCte`, `docenteCte`, `infoCursoCte` y `desarrolloCte` respectivamente. Las páginas cliente creadas despliegan los datos obtenidos de la consulta, en formato HTML para que ésta pueda ser examinada por el usuario a través del navegador. Las clases `docente_curso`, `docente`, `curso`, `tópicos` y `unidades` son tablas de la base de datos que son consultadas por las páginas servidor.

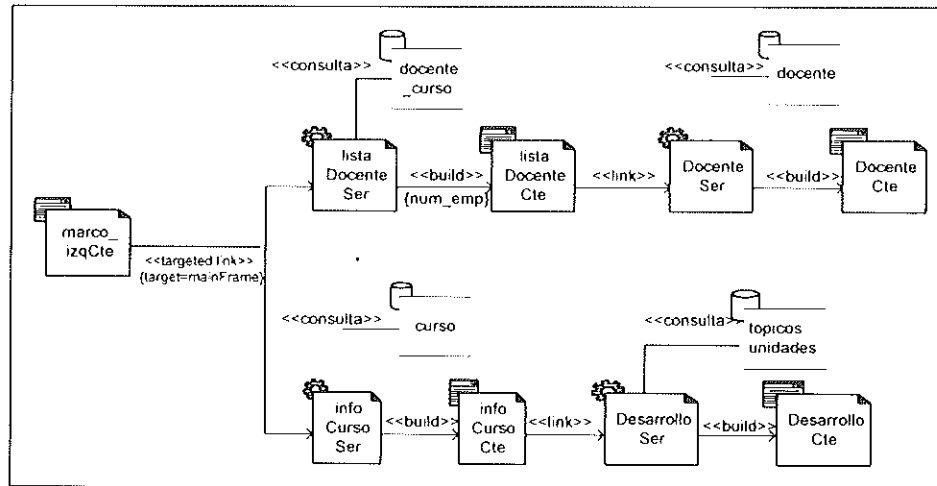


Figura 21. Diagrama de paquetes Consulta de información.

El diagrama de paquetes Actividades y productos (figura 22) muestra el conjunto de clases listaTipoActividadesSer, listaActividadesSer, productoSer, calificaciónSer y calif_tipoSer; estas clases son páginas servidor que consultan la base de datos y crean páginas cliente con la información obtenida para desplegarse al usuario. Las clases buzónProducto y buzónTarea son formas que se le presentan al usuario para elegir el archivo que se va a entregar. Psube y asube son páginas servidor, almacenan el archivo entregado en el servidor.

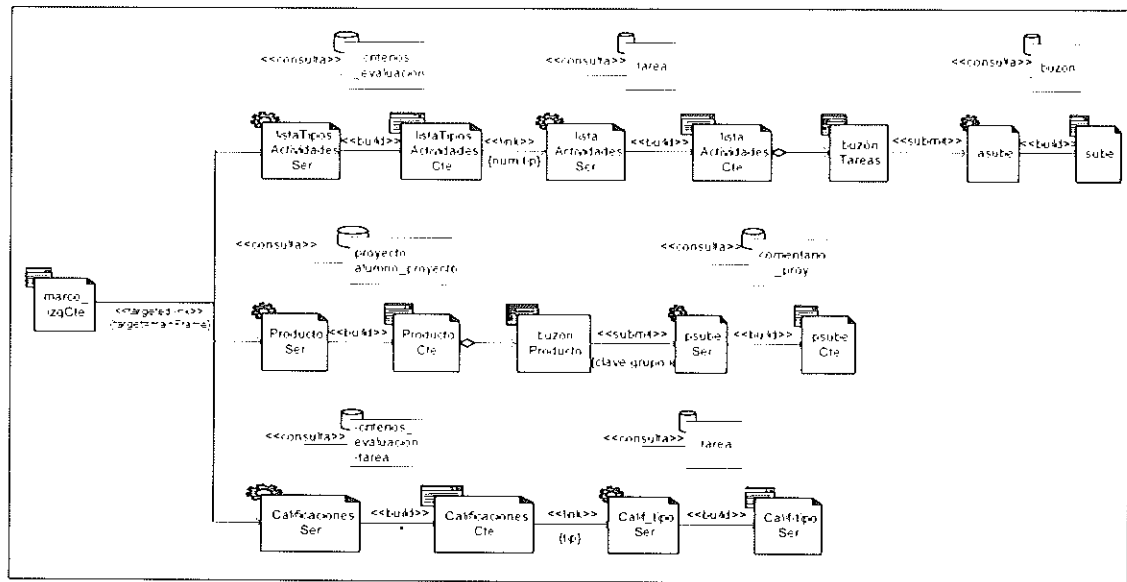


Figura 22. Diagrama de paquetes Actividades y Productos.

El diagrama que se muestra a continuación describe el paquete Herramientas de apoyo(figura 23). Las clases DocumentoSer, DocsSer, listaBookSer, consultaAnunciosSer, pagina y listaBook realizan consultas a la base de datos y generan documentos en formato HTML para desplegar la información obtenida.

La clase `Página` se encarga de organizar los mensajes en conjuntos de 5 mensajes por página. `InsertBook` es una página cliente que despliega al usuario una forma con el mismo nombre para agregar nuevos enlaces. Esta clase manda llamar la página servidor `pbookmarks`, que inserta en la base de datos el nuevo enlace.

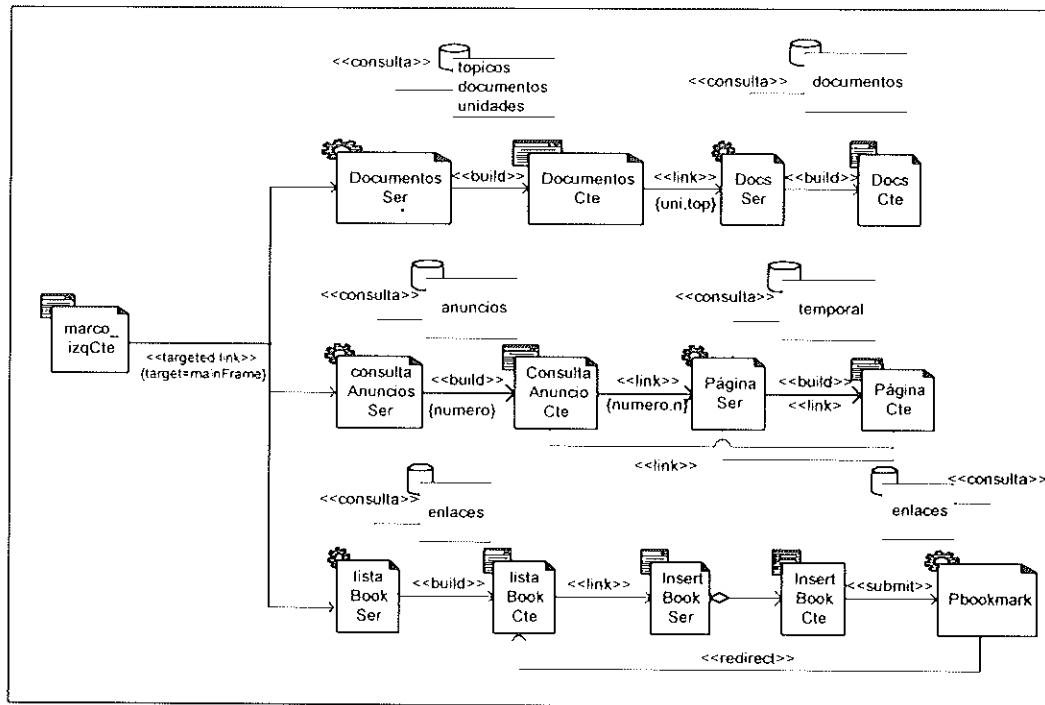


Figura 23. Diagrama de paquetes Herramientas de apoyo.

El diagrama de paquetes Configuración (figura 24) muestra las clases que actualizan la contraseña y la dirección de correo. La página servidor configuración crea la página cliente que despliega las formas: correo y password. Estas mandan llamar cambiaCorreo y cambiaPassword respectivamente para actualizar la base de datos.

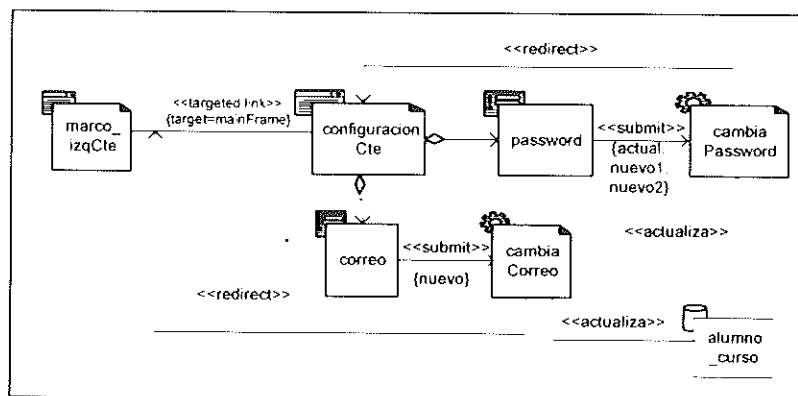


Figura 24. Diagrama de paquetes Configuración.

V.2 Diseño de la base de datos

La arquitectura de un sistema, nos permite crear la estructura estática del diseño de la base de datos, con modelos de datos relacionales lógicos y físicos, a partir de los diagramas de clases. Es decir se pueden convertir los diagramas de clases hechos con UML a un diagrama E-R y de este a un modelo de datos físicos, o viceversa.

El modelo de bases de datos creado para el sistema se desarrolló a partir del diagrama de clases modelado con UML (figura 25). Las clases se trasladaron a tablas en la base de datos. Con la cardinalidad podemos saber cuando es necesaria la creación de tablas intermedias que relacionen 2 tablas cuya relación es de muchos a muchos. Con los atributos de las clases se establecieron los atributos de las tablas. Se aplica la normalización a los datos para estructurarlos de tal forma que nos permitan relacionar fácilmente los datos, facilitar la recuperación y mantenimiento de estos, pero principalmente elimina la redundancia e inconsistencia.

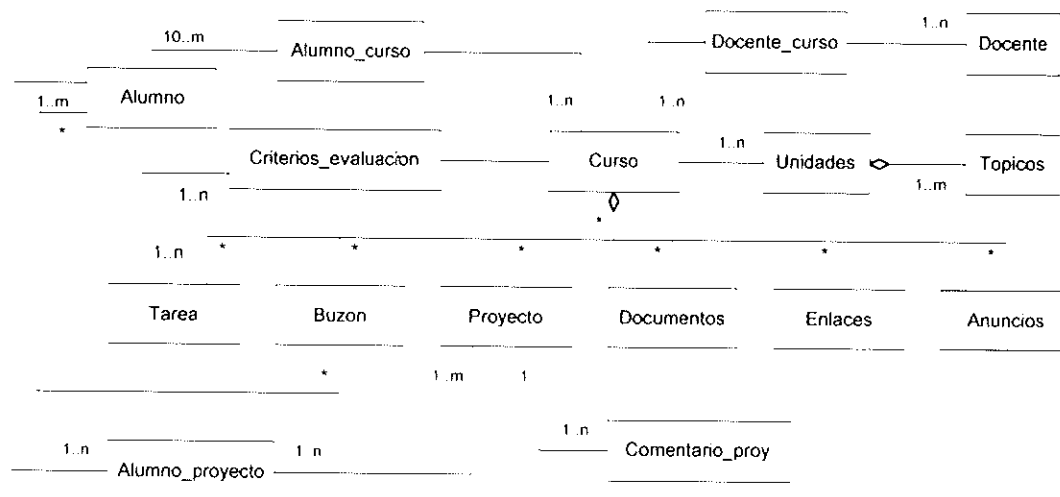


Figura 25. Diagrama de entidades

V.2.1 Especificación de las tablas

Las tablas y los atributos que se construyeron para instrumentar la base de datos son representados en las clases. Los atributos de las clases corresponden a los atributos de las tablas en la base de datos. Las operaciones aplicadas a los atributos se muestran como los métodos de las clases desarrolladas. La descripción de las tablas elaboradas se especifica en el anexo D.

VI DESARROLLO

VI.1 Instalación y configuración del servidor

La instalación de los paquetes se realizó en una máquina con sistema operativo Linux Mandrake V.8.0.

Cumpliendo con el requerimiento de ejecutar el sistema a través de un navegador en Internet, se instaló y configuró un servidor de Web, Apache V.1.3. el cual soporta la integración de módulos de tecnologías de Internet, así como manejadores de bases de datos, entre otros, permitiendo un mejor desempeño entre estos.

Entre los módulos que soporta el servidor de Web se encuentra PHP, lenguaje utilizado para el desarrollo del sistema el cual por ser un lenguaje de script interpretado en el servidor Web, permite la generación de páginas Web dinámicas. El manejo de este lenguaje tiene la ventaja de ser fácil y rápido de comprender la codificación; es menos complejo que otros lenguajes orientados al desarrollo de aplicaciones Web. Este lenguaje ofrece un rango más amplio de funciones predefinidas al integrarlo con el manejador de base de datos MySQL(cita), que con otros manejadores, por lo que reduce el tiempo de desarrollo y las líneas de código.

VI.2 Instrumentación de la Base de datos

Para almacenar la información se utilizó el manejador de base de datos MySQL V.3.23.36. Debido a que PHP cuenta con un conjunto de funciones destinadas a mejorar la integración con MySQL, el uso de este manejador facilitó la implementación del sistema. MySQL no soporta consultas anidadas, por lo que disminuye la agilidad en estas. Una característica importante es que consume pocos recursos tanto de CPU como de memoria. Para la conexión a la base de datos se creó el usuario curso, utilizado para generar las tablas de las entidades encontradas en el sistema.

VI.3 Instrumentación de la interfaz de usuario (GUI)

Del lado del cliente se instrumentaron páginas HTML, las cuales despliegan la información a los usuarios y a través de los cuales estos interactúan con los cursos, esto en un navegador.

VI.4 Instrumentación del módulo de cursos visión-alumnos

El sistema desarrollado cubre el módulo visión del alumno del sistema que da soporte a la educación en línea de la UABC. Como paso inicial se realizaron entrevistas para la recopilación de los requerimientos. Algunas de las características del sistema fueron acreditadas dada la comparación de otros sistemas similares desarrollados por distintas universidades o por el actual sistema Virtual-U, utilizado por la UABC para la impartición de los cursos en línea.

Para el proceso de desarrollo de software del sistema se utilizó el modelo de cascada modificado según McConnell (1996), el cual percibe el desarrollo de un sistema como una secuencia simple de fases, en donde cada fase tiene un conjunto de metas bien definidas y las actividades entre cualquier fase contribuyen a satisfacer las metas de las fases subsiguientes. Este modelo no es suficiente en la aplicación real cuando se están recibiendo requerimientos funcionales en el proceso de desarrollo, sin embargo realizando ciertos ajustes de tiempo de trabajo y revisión en cada fase es posible cumplir con el modelo. Se sugiere que cuando no se tienen claros los requerimientos trabajar con el modelo evolutivo y/o de espiral según Presman (1997) y de esta forma acotar los requerimientos a un nivel que satisfagan al usuario final.

Se realizó el diseño y la instrumentación de la base de datos que daría soporte a las aplicaciones del sistema, basándose en la estructura global lógica resuelta por el diagrama general de clases desarrollado durante la etapa de diseño del sistema y que posteriormente se

tradujo a tablas. Como se mencionó en la etapa de análisis y diseño de la base de datos, es valido trasladar las clases creadas en el diseño, a entidades en el modelo entidad relación, y de éste a la estructuración física del banco de datos.

El desarrollo del sistema, en el lado del servidor se desarrollaron las aplicaciones en PHP, que es transparente para el usuario, al que le parecerá que está visitando páginas HTML que cualquier navegador puede interpretar.

Las aplicaciones en PHP, realizan el procesamiento a los datos proporcionados por los usuarios. El esquema general de estas aplicaciones son páginas servidor que procesan la información, consultándola o insertándola en la base de datos, y posteriormente generan páginas clientes en formato HTML para desplegar la respuesta al procesamiento realizado a los datos.

La arquitectura de instrumentación utilizada es Cliente – Servidor, en donde se integran las capacidades del manejador de bases de datos MySQL con la tecnología de Web PHP, a través del servidor de Web Apache(Figura 26).

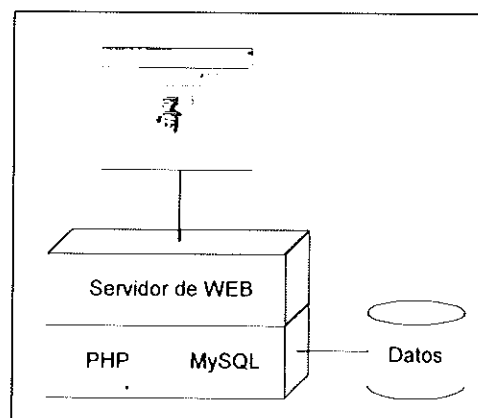


Figura 26. Arquitectura que muestra la integración entre el servidor de Web, PHP y MySQL.

VII FUNCIONALIDAD DEL SISTEMA

Enseguida se muestran algunas de las pantallas que describen físicamente al sistema, estas pantallas corresponden a los diagramas que se han estado mencionando durante el trabajo. Los diagramas que no se muestran en esta sección se muestran en el anexo C.

Una vez que el alumno ha iniciado una sesión de trabajo en el sistema, se le presenta una pantalla de bienvenida. (Figura 27) Como paso posterior al acceso el alumno debe elegir el curso en que va a trabajar, para lo que se le presenta un menú y un botón de acceso al curso.

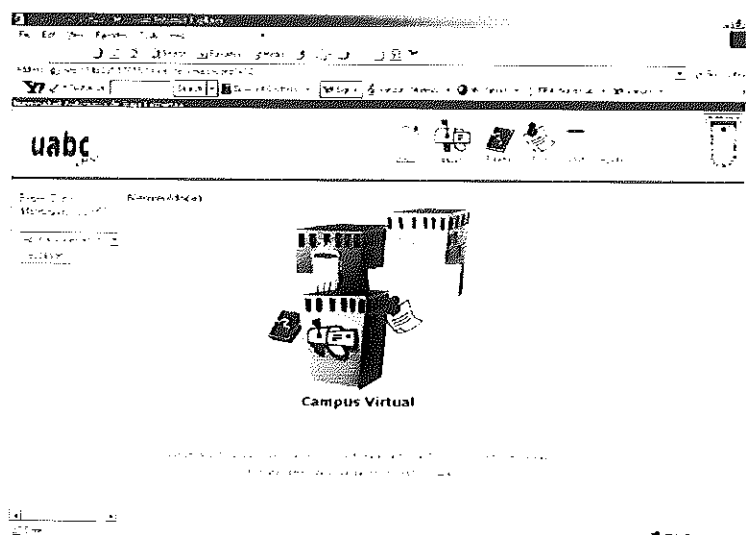


Figura 27. Pantalla de bienvenida al sistema.

Una vez que al alumno ha elegido el curso, el sistema despliega el menú de opciones de trabajo.

Cuando el alumno desea obtener información general del curso, el sistema presenta una tabla que contienen la lista de datos que describen al curso (Figura 28). Estos datos son proporcionados por el docente a cargo.

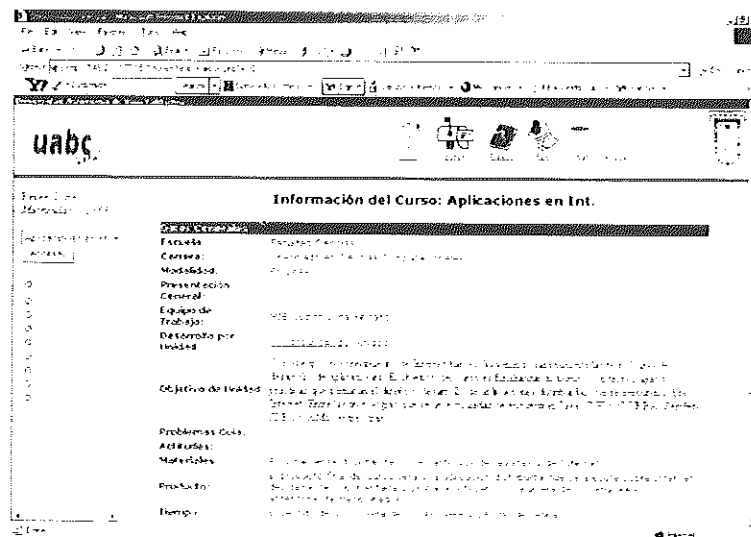


Figura 28. Pantalla de consulta de información del curso.

Al consultar la opción documentos el sistema despliega las unidades del curso, con sus respectivos temas por unidad. Se debe elegir un tema en particular para poder acceder a los documentos existentes, específicos de ese tema. La pantalla (figura 29) muestra los documentos proporcionados por el docente, pertenecientes a un tema del curso, organizados en tablas. Para obtener un documento hay que presionar la liga en tabla que contenga el documento que se desea obtener. El sistema no tiene un numero limite de documentos a almacenar.

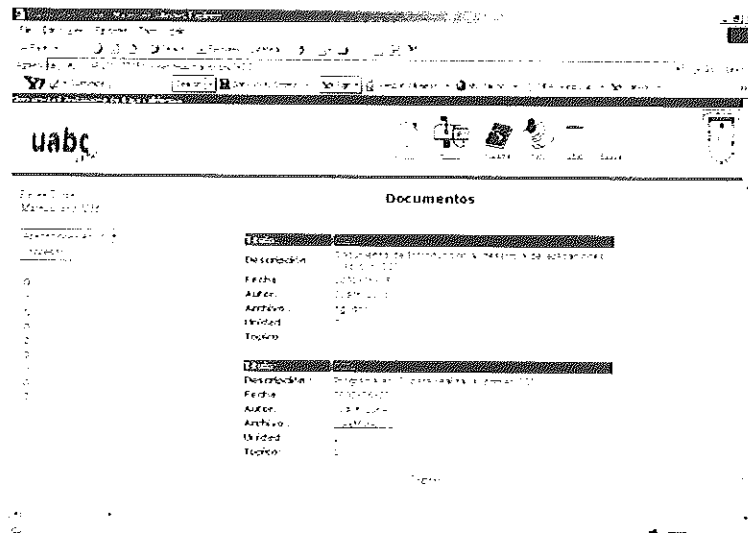


Figura 29. Pantalla de obtención de documentos.

Al elegir la opción de Actividades (Figura 30), el sistema despliega una tabla que contiene las distintas entidades con las que se va a evaluar el curso, mismas que fueron aportadas por el docente a cargo. En este caso propusimos la entidad Tareas, aunque pueden ser exámenes, practicas, etc. Entonces si se elige "Tareas", se obtendrá una segunda tabla que nos proporcionara la lista de tareas asignadas. Cada elemento de la lista es una liga que nos proporciona una descripción de una tarea, y el buzón. El buzón es el medio por el cual el alumno entrega su archivo de tarea. El buzón asume 3 estados distintos: 1) subir archivo; 2) buzón saturado; 3) buzón expirado. En los 2 ultimo casos no se permite subir archivos.

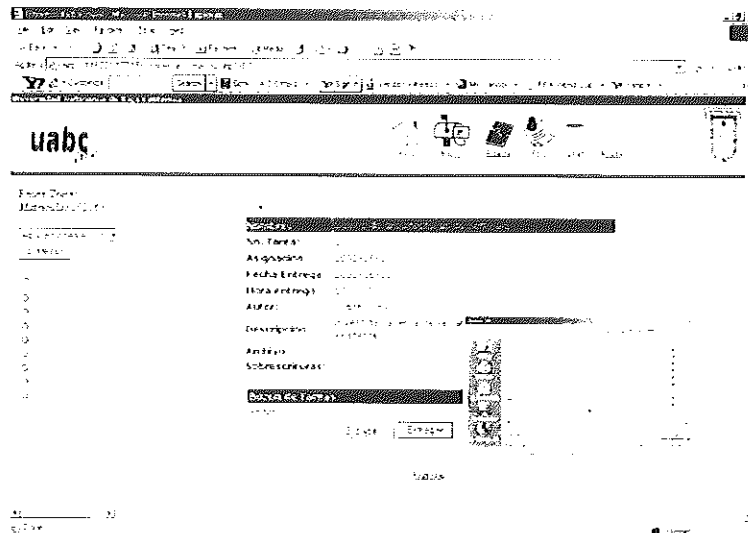


Figura 30. Pantalla consulta y entrega de actividades.

En la opción de configuración (Figura 31), el sistema despliega una pantalla con 2 tablas; para cambio de contraseña y de dirección de correo. Para cambiar de contraseña el usuario debe proporcionar su contraseña actual y una nueva con su confirmación. Para el cambio de dirección de correo solo hay que introducir la nueva dirección.

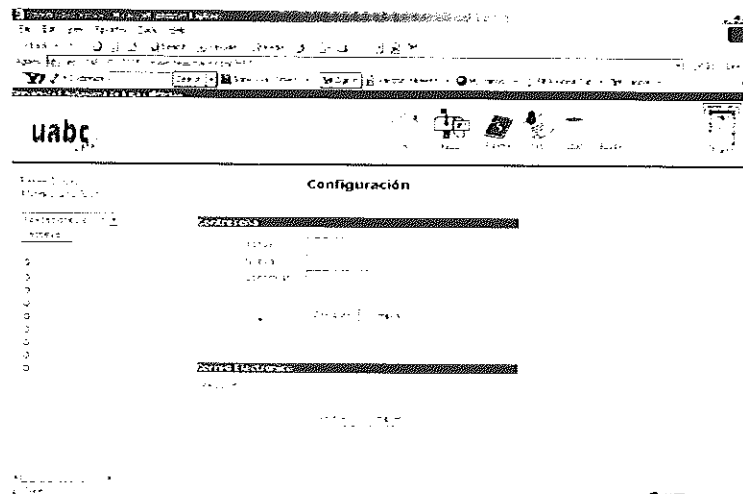


Figura 31. Pantalla configuración.

VIII RESULTADOS

El módulo de la visión del alumno del sistema de educación a distancia desarrollado para la impartición de cursos en línea en la UABC, inició con una planificación que se enfocaba al entendimiento de los requerimientos de los cursos y a las necesidades de los estudiantes según la opinión del maestro. Una vez comprendidas estas exigencias del sistema se eligieron las tecnologías que podían satisfacer el desempeño de éstas.

La metodología de desarrollo de software que se utilizó para el desarrollo del sistema fue el método de Cascada. El uso de esta metodología condujo el progreso del sistema a través de una secuencia de pasos ordenados, estructurándolo y permitiendo generar incrementos en el producto en cada una de sus fases.

El modelo de arquitectura utilizada en el sistema fue Cliente - Servidor de 3 capas, debido que en el sistema debía existir un sitio donde se centraliza toda la información, se distribuiría bajo demanda de un conjunto de personas o máquinas. Además por características como: manejo de un gran número de solicitudes y agilidad de respuesta satisfacía dichos requerimientos.

Como primer paso en el seguimiento de la metodología de desarrollo de software se realizaron entrevistas a las autoridades capacitadas en el área de la enseñanza, esto para obtener los requerimientos que debía tener el sistema. Algunas de las características del sistema fueron acreditadas dada la comparación o exploración de otros sistemas similares desarrollados por distintas universidades o por el actual sistema utilizado por la UABC para la impartición de los cursos en línea, Virtual-U.

Se realizó el análisis del sistema basándose en la metodología de UML. Durante el análisis se crearon los Diagramas de casos de uso y diagramas de análisis de casos de uso. Posteriormente se realizó el diseño del sistema, generando los diagramas de secuencia, diagramas de clases, diagramas de paquetes y un diagrama de actividad. Logrando una representación del sistema basado en diagramas que permitían traducir, entender y lograr de manera más rápida a lo que se quería llegar.

Después se realizó la instrumentación de la bases de datos, trasladando la estructura estática de las clases del sistema a tablas físicamente instrumentadas, permitiendo crear la estructura global de la base de datos del sistema.

Terminadas las fases de análisis y diseño del sistema se dio inicio a la implementación, actividad realizada con PHP. PHP es un lenguaje de programación ejecutado por los servidores Web. Permite crear páginas con contenido dinámico. Las aplicaciones generadas con PHP realizan el procesamiento a los datos del sistema, además de comunicarse con el servidor de base de datos MySQL para almacenar o modificar la información procesada. Todo esto se realiza en el lado del servidor.

Del lado del cliente, páginas en formato HTML son generadas por las aplicaciones en PHP para desplegar los instrumentos de los cursos a los usuarios. Los instrumentos pueden ser accedidos por Internet a través de su navegador. Los distintos instrumentos brindados por el sistema ofrecen: información del curso y del docente, acceso a actividades y exámenes asignados, material del curso (lecturas, enlaces a sitios, documentos en archivos), Materiales de referencia.

VIII.1 Usabilidad del sistema UABC-UV

Las pruebas de usabilidad que se presentan en esta sección fueron realizadas por miembros del equipo de trabajo de desarrollo del modulo de cursos. Las pruebas fueron realizadas por Luna Serrano y Morita Sandoval en noviembre de 2001.

Con las pruebas de usabilidad realizadas al sistema se trató de obtener una primera evaluación del sistema en un medio ambiente real.

El objetivo de estas pruebas fue determinar si los siguientes atributos están presentes en la herramienta UABC-UV: estimular al aprendiz a realizar y proponer tareas que lo motiven a seguir estudiando; permitir al estudiante, explorar una variedad de posibles respuestas; y no controlar el proceso de aprendizaje del estudiante, sino permitirle decidir cuál será su siguiente paso. Según Pearlman (1993) estos atributos deben estar presentes cuando se construye una herramienta para educación en línea.

VIII.1.1 Plan de pruebas

Según Nielsen (2000) cinco usuarios son suficientes para realizar las pruebas de usabilidad y mostrar los problemas en que se incurren al utilizar el sistema y es probable que usuarios subsecuentes encuentran los mismos problemas descubiertos en las pruebas.

La evaluación del sistema fue siempre supervisada por dos observadores, fue diseñado para llevarse a cabo en tres escenarios diferentes y consistió en: sesiones de prueba de usabilidad por usuarios finales, entrevistas a grupos de usuarios y sesiones de laboratorio.

VIII.1.1.1 Prueba de usabilidad del sistema UABC-UV

Se planearon cuatro sesiones de pruebas de usabilidad con 10 estudiantes y dos docentes voluntarios que tuvieran al menos un mes de experiencia en el uso del sistema UABC-UV. Las sesiones de prueba fueron llevadas a cabo en un laboratorio aislado. Los participantes tuvieron acceso al sistema por el navegador de su elección (Internet Explorer o Netscape). Las pruebas de usabilidad se planearon en dos protocolos, correspondiendo dos sesiones para cada uno (una con los estudiantes y otra con uno de los docentes).

Para las pruebas de usabilidad del sistema se evaluaron los siguientes aspectos: diseño de la interfaz, navegación en el sistema, mensajes de error. Las actividades que realizaron los alumnos son: consulta de temario, realizar actividades correspondientes a una unidad específica, cambiar dirección de correo y consulta de calificaciones.

VIII.1.1.2 entrevistas estructuradas

Se aplicó una entrevista a 7 estudiantes y 3 docentes. La entrevista aborda el tema de la forma como se concibe la herramienta, la lógica del manejo del sistema, la densidad del mismo y en general el grado de aceptación. La entrevista admite opiniones, sugerencias y recomendaciones.

El cuestionario y la guía de observación son adaptadas de las desarrolladas por Ruman y Gillete (2001). No hay entrenamiento previo para los observadores en la aplicación de las preguntas.

VIII.1.1.3 Sesión de laboratorio

Se planearon un total de cuatro sesiones de laboratorio con un conjunto de 20 alumnos. Dos sesiones se prepararon con la asignatura de ingeniería de la programación (6 alumnos) y

dos sesiones en la materia de administración de proyectos (14 alumnos). Se dispuso que las sesiones de laboratorio fueran de 40-60 minutos de trabajo, dos días a la semana. A cada participante se le solicitó desempeñar las mismas tareas:

Se selecciona una unidad problema en su respectivo temario.

Se asigna la tarea de buscar 3 direcciones (URLS) relacionadas con la unidad problema seleccionada.

Se envía la tarea por medio del buzón.

VIII.1.2 Resultados de las pruebas

VIII.1.2.1 Entrevista a estudiantes

Los estudiantes mencionan que les gustaría que se agrupara en forma más sencilla el módulo de Cursos sin tener que abrir tantas opciones para poder entender que hacer. Prefieren que se les indique directamente en una sola opción que hacer, con qué, con quién y en cuanto tiempo. Por otro lado, les satisfizo poder revisar calificaciones y ver como se les calificaba su desempeño en cada tarea.

Todos los evaluadores estuvieron de acuerdo en utilizar esta herramienta para sus cursos formales, aunque se requieren ciertos ajustes que permitan una visión más sencilla.

Los estudiantes consideran que es necesario desarrollar el modelo de ayuda, ya que proporciona una sensación de seguridad, cuando se realiza cierta acción.

IX DISCUSIÓN

Para el proceso de desarrollo de software del sistema se utilizó el modelo de cascada modificado según McConnell (1996), el cual percibe el desarrollo de un sistema como una secuencia simple de fases, en donde cada fase tiene un conjunto de metas bien definidas y las actividades entre cualquier fase contribuyen a satisfacer las metas de las fases subsiguientes. Este modelo no es suficiente en la aplicación real cuando se están recibiendo requerimientos funcionales en el proceso de desarrollo, sin embargo realizando ciertos ajustes de tiempo de trabajo y revisión en cada fase es posible cumplir con el modelo. Se sugiere que cuando no se tienen claros los requerimientos trabajar con el modelo evolutivo y/o de espiral según Presman (1997) y de esta forma acotar los requerimientos a un nivel que satisfagan al usuario final.

Para representar el modelo del sistema se empleó el Lenguaje Unificado de Modelado (UML), debido a que es un lenguaje estándar de modelado para el desarrollo de software orientado a objetos. Según Booch, Jacobson y Rumbaugh, 1999.

Su notación nos deja representar software complejo en forma de símbolos gráficos simples, que permiten una mayor comprensión del software que se modela. Por lo que es posible concebir la estructura del sistema modelado. Además se puede modelar el sistema a diferentes niveles de abstracción, se avanza de diagramas generales a diagramas más específicos.

El análisis del sistema se realizó basándose en los requerimientos planteados, se inició con el modelado de los casos de uso que representaban de manera general la funcionalidad del sistema, identificando superficialmente las actividades que debían ser consideradas. Continuando con el análisis de caso de uso, obteniendo detalladamente las actividades que implicaban los casos de uso. No fue necesario dedicar mucho tiempo a esta etapa, pues la

manera de representar los requerimientos iniciales (casos de uso) mostraba claramente las actividades que debía satisfacer el sistema. En el diseño se realizaron los diagramas de secuencia, los cuales nos permitieron identificar objetos y métodos del sistema, en el diagrama de actividades identificamos la secuencia de las funciones u operaciones que se ejecutan en un programa y acciones que son realizadas de manera paralela. Con los diagramas de paquetes reunimos los programas con actividades afines, como se menciona anteriormente. En esta etapa fue necesario implementar un diseño que permitiera comprender la relación de las aplicaciones del Web con el servidor. Se realizaron los diagramas de secuencia y paquetes con una extensión de UML que fortalecía la comprensión del flujo de los datos y procesamiento aplicado a ellos desde una página de Internet.

Dado que uno de los requerimientos del sistema es que se ejecute sobre el Web, se optó por utilizar una extensión de UML (estereotipos) para modelar aplicaciones Web (Conallen, 2001). UML no cuenta con este soporte. Al utilizar esta extensión facilita la comprensión de las relaciones de los objetos que radican en el servidor, con los objetos del cliente.

La arquitectura utilizada fue Cliente – Servidor de 3 capas; permite concentrar la información en un servidor, que este disponible a los usuarios en cualquier momento. Permite atender solicitudes simultaneas, y satisface el requerimiento, que el sistema sea una aplicación para el Web.

El lenguaje utilizado para la instrumentación del sistema es PHP el cual por ser un lenguaje de script interpretado en el servidor Web, permite la generación de páginas Web dinámicas. El manejo de este lenguaje tiene la ventaja de ser fácil y rápido de comprender la codificación; es menos complejo que otros lenguajes orientados al desarrollo de aplicaciones Web. Este lenguaje ofrece un rango más amplio de funciones predefinidas al integrarlo con el

manejador de base de datos MySQL, que con otros manejadores, por lo que reduce el tiempo de desarrollo y las líneas de código. El servidor de Web Apache cuenta con un módulo para soportar la ejecución de PHP. Además una ventaja es que se distribuye de forma gratuita.

La metodología utilizada (UML) provee una secuencia de pasos, para hacer el cambio del diagrama de clases (diseño del sistema) a un diagrama entidad relación (E-R) que estructure la base de datos. De esta forma se verifica la normalización de tablas; evitando inconsistencia; redundancia en la base de datos. (Date, 2000)

El manejador de base de datos utilizado fue MySQL V.3.23.36. Su principal objetivo de diseño fue la velocidad tanto al conectar con el servidor como al servir selects. Se sacrificaron algunas características esenciales en sistemas más robustos con este fin, características tales como ignorar la integridad referencial, dejándola en manos del programador de la aplicación; no soporta transacciones; "roll-backs" quitando la posibilidad de retroceder una operación a los datos; ni subselects, por lo que disminuye agilidad en consultas. Una característica importante es que consume pocos recursos tanto de CPU como de memoria. Además soporta una buena integración con PHP. (<http://www.mmlabx.ua.es/mysql-postgres.html>)

El inicio de una buena codificación es el resultado de una correcta solución y planeación del problema que se ataca. Es decir, realizar un buen análisis y diseño del sistema permite encontrar una estructura para un buen desempeño. Además al lograr lo anterior se puede reducir el tiempo de codificación y líneas de código de manera considerable.

Se debe estar consciente, que no es suficiente con esto, si no que también se debe realizar una programación de calidad. Esto es ventajoso al trabajar en equipo, ahorra tiempo al tratar de comprender el código realizado por otras personas.

El proyecto fue realizado en equipo; instrumentado entre diferentes personas distintos módulos que forman el sistema UABC-Universidad Virtual. Este trabajo se enfoca al modulo orientado al alumno, sin embargo en algunas ocasiones había la necesidad de trabajar en conjunto con el resto del equipo para coordinar y estandarizar segmentos comunes. Durante la instrumentación se observaron algunas características de codificación sobresalientes o que permitirían una mejor comprensión del código: 1) Los nombres de las variables y funciones su elección es muy importantes; ayudan a entender el significado real de las acciones que realizan, y con esto se obtiene una mejor legibilidad, 2) validar a lo largo del código, para detectar rápidamente cualquier procedimiento erróneo de los métodos o funciones, 3) La codificación debe tener una buena visualización, darle un formato ordenado que permita identificar fácilmente las partes que forman el programa, 4) Reutilizar en lo posible funciones, 5) La comunicación debe ser fluida en los miembros del equipo. Si se realiza lo anterior, se hace sencillo el trabajo en equipo y la supervisión por parte de otras personas. Se reitera que con estas características, es posible reducir tiempo de instrumentación y esfuerzo.

Al realizar la instrumentación del modulo visión alumnos se identificaron 4 principales segmentos, que describen de manera global, general la estructura y funcionamiento del modulo. Estos segmentos se pueden denominar: consulta de información general, actividades y productos, herramientas y configuración. Esta clasificación se realizo a partir de programas que realizan operaciones afines. Por ejemplo el segmento que se refiere a consulta se caracteriza por reunir los programas que únicamente realizan consultas a la base de datos, que no efectúan ninguna operación que altere los datos. Con el segmento de actividades y productos clasificamos los programas que recopilan, el resultado de los producto brindados por el alumno (archivos de documentos u de programas), estos productos son la respuesta a una petición hecha por el docente, esta respuesta se hace llegar al docente a través del envío de archivos

por medio de la red de Internet y que finalmente se alojan en el servidor para ser accedidos por el docente. En la parte de herramientas se reúnen las aplicaciones que ayudan a establecer una comunicación unidireccional del docente hacia el alumno, en donde el docente recomienda al alumno documentos de apoyo para el curso, por medio de archivos o ligas de Internet.

X CONCLUSIÓN Y TRABAJO FUTURO

El sistema de educación a distancia desarrollado, propuesto para la impartición de cursos en línea en la UABC, cubre la solicitud del módulo de cursos en la visión del alumno. Debido a que los requerimientos del sistema fueron propuestos por particulares capacitados en el área de la enseñanza, la arquitectura del sistema propone que casi cualquier asignatura pueda ser impartida en esta modalidad.

El sistema está formado por un conjunto de herramientas que se proponen para proveer educación en línea a los estudiantes de la UABC, enfocándose a sus necesidades y en la manera en que estos van a interactuar con los materiales que se les brinden a través del sistema.

Las herramientas desarrolladas que componen al sistema en la visión del alumno son:

Consultas:

Información general del curso y del docente.

Actividades asignadas por el docente

Calificaciones asignadas a cada actividad

Anuncios o mensajes proporcionados por el docente

Lista de documentos y la adquisición de estos

Aplicaciones :

Para hacer entrega de archivos de manera remota

Para hacer entrega de adelantos de los proyectos

Agregar y consultar enlaces que sirven de apoyo al curso.

Cambio de dirección de correo y contraseña de usuario.

X.1.1 Trabajo Futuro

El trabajo futuro predecible en el sistema actual conlleva a la adición de módulos que incrementen la funcionalidad del software, estos módulos se mencionan enseguida.

Es de particular importancia probar la funcionalidad del sistema impartiendo un curso con un grupo de alumnos, para tener la firmeza de que el sistema puede ser utilizado y aceptado como una herramienta para impartir cursos en línea.

El sistema debe asumir una ejecución que le permita establecer conexiones seguras, en donde los datos viajen codificados, y así el usuario tenga la certeza de que la información que transmite es confidencial.

Se propone además, la creación de un editor asíncrono sobre el web, que permita la creación de un documento, programa, etc, entre dos o mas miembros del curso, para que los usuarios puedan desarrollar actividades - tales como proyectos, tareas, practicas - en conjunto.

Referencias

Duchastel, P., "A web-based model for university instruction. *Journal of Education Technology Systems*". 1997.

Fowler M, K. Scott. "UML gota a gota". Editorial Addison Wesley Longman 1999.

I Jacobson, G Booch, J Rumbaugh. "El proceso Unificado de desarrollo de Software". Addison-Wesley-Longman 1999.

J. Rumbaugh . "The unified modeling language reference manual". ADDISON-WESLEY, 1999.

Keegan, D., "Distance education Technology for the new millenium: compressed video teaching". 1995.

Luna Serrano J, Morita Sandoval. 2001. "Evaluación de una herramienta computacional para la educación en línea, visión alumnos". Memorias del X Encuentro Internacional de Educación a Distancia. Noviembre de 2001. Universidad Autónoma de Guadalajara Jalisco (UDG).

McConnell, S.1999. "Rapid Development". Microsoft Press.

Nielsen J. (2000) "Why you Only Need to test whit 5 user Aletbok" <http://www.useit.com/> abril/

Pearlman, R. 1993. "Designing the new american schools". *Technology Education*. Vol. 36, Num. 5.

Pérez Fragoso, C., 2000, "Estado de la docencia en línea: consideraciones sobre su evaluación". En: M. Rueda Beltran y F.Díaz Barriga (eds), *Perspectivas para la evaluación y análisis de la docencia*, México: Paidós.

Pérez Fragoso, C y M. Rueda Beltrán (2000), "Los desafíos de la educación en línea: notas preeliminarias". *Signos Universitarios*, desafíos del 2001 I: Educación, Año XX, num 37, Buenos Aires, Arg. Pp157 -169.

Roger S. Pressman. "Ingeniería de software, un enfoque practico". Mc. Graw Hill 1998.

Ruman C. and Gillatte J. 2001 "Distance Learning Software Usefulnesss and Usability: User-Center Issues in Practical Deploment March". Vol.15: No.30

Direcciones de Internet

Modeling Web Applications with UML

<http://www.conallen.com/whitepapers/webapps/ModelingWebApplications.htm>

Software Engineering with UML, Use Case Model

http://www.sparxsystems.com.au/use_case_model.htm

Guía de estilos de páginas WEB

<http://webestilo.com/php/>

Manual de PHP

<http://www.php.net/manual/es/>

Help for php programmers
<http://phphelp.com>

The resource for php programmers
<http://phpbuilder.com> <http://www.mmlabx.ua.es/mysql-postgres.html>

MySQL Reference Manual
<http://www.mysql.org/documentation/mysql/bychapter/>

MySQL vs. PostgreSQL
<http://www.mmlabx.ua.es/mysql-postgres.html>

UML
<http://www.creangel.com/uml/diagramas.html>

UABC, Sistema de Educación Abierta y a Distancia
<http://www.uabc.mx/dgaa/ed1.html>)

ANEXO A

Diccionario de clases

Las clases del sistema que realizan un procesamiento más complejo se describen enseguida.

Index

Clase que brinda al usuario la pantalla de acceso al sistema y, verifica si el usuario es válido.

Marco_izqSer

Clase que consulta la base de datos para conocer y proporcionar los cursos en los que ésta inscrito el usuario que ha accedido. Genera la página cliente que brinda las opciones que se le proporcionan al alumno para hacer uso de las aplicaciones.

DocenteSer

Esta clase busca la información del docente elegido a través de una liga desplegada por la clase listaDocentesCte. Crea la página cliente que despliega en una tabla los datos generales del docente especificado.

InfoCursoSer

Clase que busca los datos de los cursos en la base de datos. Y crea la página cliente en la que es desplegada la información general del curso.

DocsSer

Una vez hecha la elección de un documento a través de una liga proporcionada por la clase Documentos, esta clase es la encargada de buscar la información del documento elegido y de crear la página cliente que despliegue dicha información en una tabla.

ListaActividadesSer

Esta clase busca la información de una actividad específica. Genera la página en la que se despliega la tabla con la información y además despliega el buzón de tareas. Si es utilizado el buzón, la página cliente generada hace un llamado a la clase que se encarga de almacenar la actividad en el servidor.

Sube

Esta clase se encarga de almacenar en el servidor el archivo entregado a través del buzón. Y almacena o actualiza la base de datos con la información correspondiente a la tarea entregada.

ProductoSer

Consulta la base de datos para recuperar los datos correspondientes al producto, despliega el buzón de productos y manda llamar la clase que almacena los archivos entregados.

Psube

Esta clase se encarga de almacenar en el servidor el archivo entregado a través del buzón. Y almacena en la base de datos los comentarios hechos por el maestro.

Calif_tipoSer

Clase que consulta en la base de datos las calificaciones asignadas a cada actividad. Crea la página cliente que despliega la tabla con las actividades y su calificación correspondiente.

ConsultaAnuncioSer

Consulta en la base de datos los anuncios correspondientes al curso y los despliega en la página cliente que crea.

listaBookSer

Consulta en la base de datos los enlaces del curso y crea la página que despliega los datos de los enlaces. Además direcciona a la forma para agregar un enlace.

CambiaPassword

Clase que verifica la contraseña actual del usuario, proporcionado en la forma. Cuando la contraseña es válida se verifica entonces que la nueva contraseña y su reafirmación sean iguales, procede a cambiar la contraseña.

CambiaCorreo

Clase que toma de la forma la nueva dirección de correo proporcionada y actualiza la base de datos para modificar la información.

ANEXO B

En el diagrama de análisis de casos de uso de Proyectos (figura 32), muestra el flujo de la información que surge cuando el usuario va a ser entrega de archivos del proyecto. El alumno elige desde el menú la liga a proyectos, :Menú busca en :Proyecto envía los datos de archivos que se han entregado previamente, y la información es desplegada en una lista. El alumno elige desde el buzón, el archivo o archivos que va a entregar, :buzonP almacena el archivo en el servidor y envía mensaje al usuario de que el archivo ha sido almacenado.

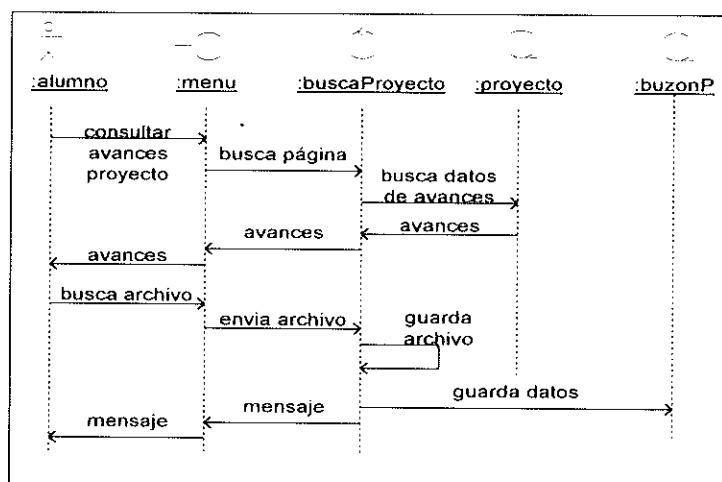


Figura 32. Diagrama de análisis de casos de uso Proyectos.

El en el diagrama de información del docente (figura 33) se muestra el proceso que se crea al consultar la información del docente. El alumno desde el navegador, selecciona en el menú principal la opción de información del docente, :menu envía buscar los datos en :docente. Cuando los datos son encontrados son desplegados al usuario.

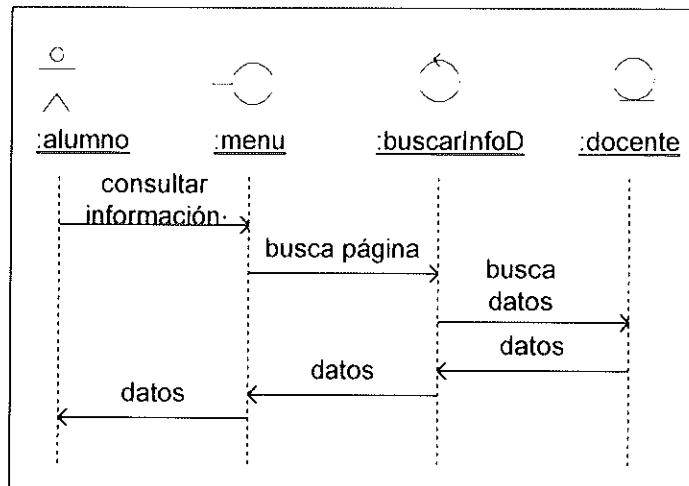


Figura 33. Diagrama de análisis del caso de uso consulta de Información del docente.

El en el diagrama de caso de uso consulta de enlaces (figura 34) se modela el flujo de información que se crea cuando un usuario consulta los enlaces y agrega nuevos. :Menu busca los registros de los enlaces del curso almacenados en :enlaces y los despliega en una lista. Aunado a la lista existe la opción de agregar un nuevo enlace, cuando esta opción se selecciona la información del nuevo enlace es almacenado.

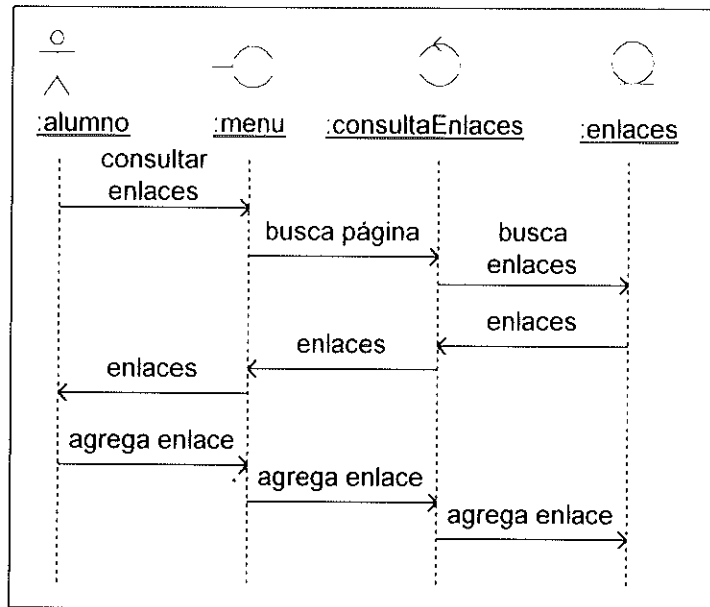


Figura 34. Diagrama de análisis del caso de uso consulta de enlaces.

El en el diagrama de caso de uso de consulta de Anuncios (figura 35) se modela el flujo de información que se forma cuando un usuario consulta los anuncios. :Menu busca los registros de los anuncios almacenados en :anuncios y los despliega en una lista.

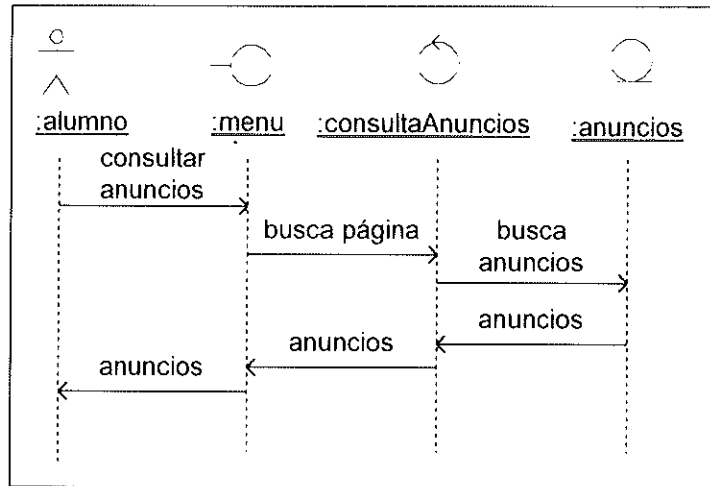


Figura 35. Diagrama de análisis del caso de uso consulta de enlaces.

ANEXO C

El usuario solicita la consulta de información del docente (figura 36) a través del menú. EL menú manda llamar una pagina servidor llamada listaDocente(), que ejecuta la función Docentes() esta consulta en la base de datos la tabla docentes. Con los datos que son obtenidos de la consulta, infoCurso() crea una pagina cliente que despliega un listado con los nombres de los docentes que imparten el curso en, de esta lista generada, el usuario puede elegir al docente del que desea obtener información. esta opción ejecuta la función infoDocente(), que consulta la información del docente elegido. Docente() crea una pagina cliente que despliega la información al usuario en una tabla.

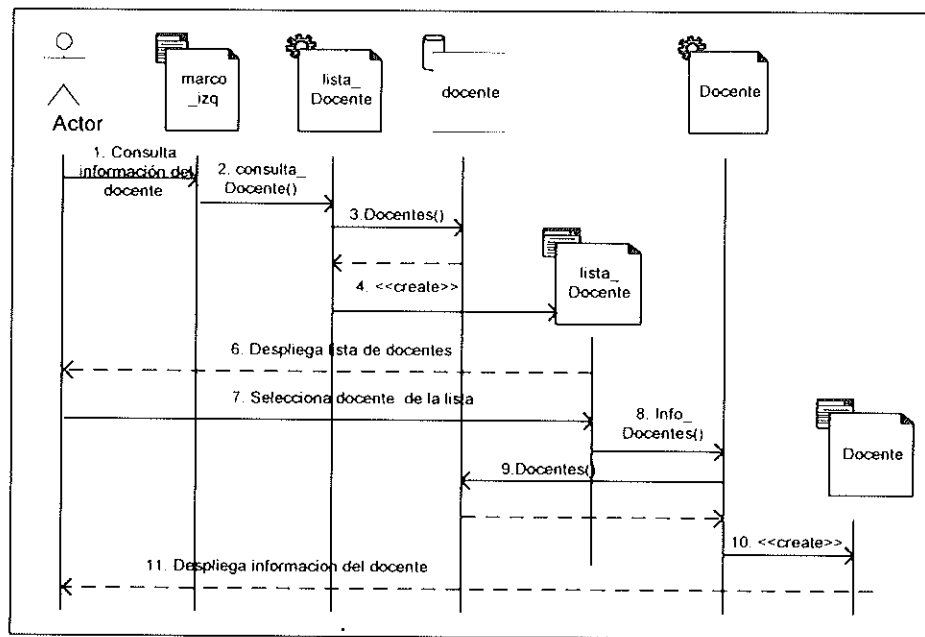


Figura 36. Diagrama de secuencia consulta de información del docente.

Cuando el alumno elige entregar archivos de proyectos (figura 37) se manda llamar Productos, que ejecuta la función infoproyectos(), esta consulta los datos descriptivos de proyecto e información de archivos previamente entregados, la información obtenida es

desplegada al usuario a la par con la opción de buzón de archivos. Si se elige entregar archivos se ejecuta la función `subirAdelanto()`, la cual tiene la función de tomar los archivo y alojarlos en el servidor así como actualizar la base de datos almacenando la información del nuevo archivo. Se crea además al mensaje que avisa al usuario que se ha realizado la operación.

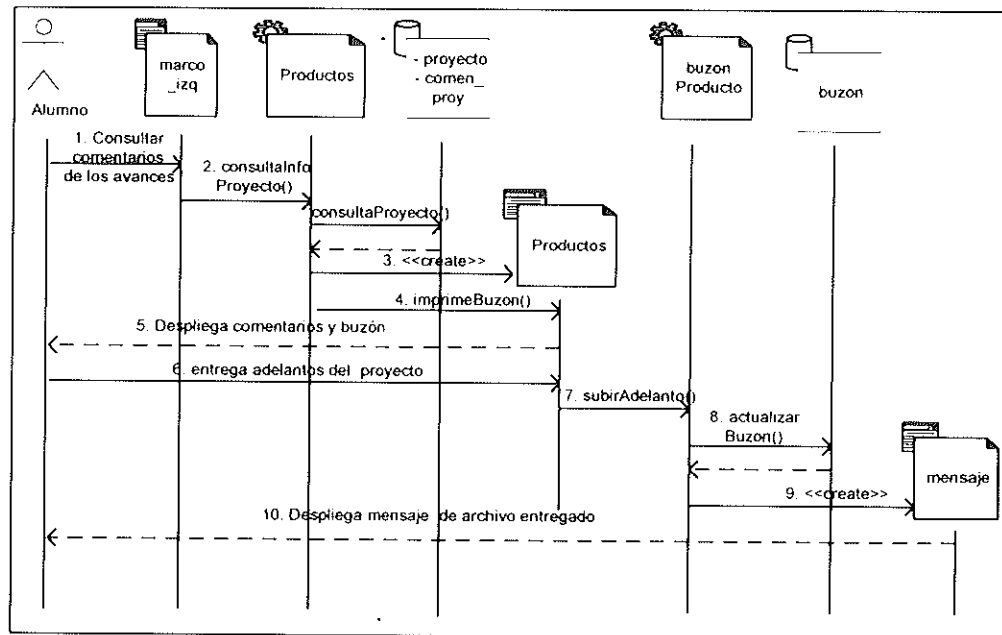


Figura 37. Diagrama de secuencia consulta y entrega de proyectos.

Si el alumno elige consultar los enlaces o agregar un nuevo enlace (figura 38), Se manda llamar `llamarlistaBook`, que ejecuta la función `consultaEnlaces()`, esta consulta los enlaces almacenados para un curso específico, los datos son desplegados al usuario en formato de lista juntamente con la opción de agregar enlaces. Para agregar enlaces se debe llenar la forma, `insertBook` se encarga de almacenar los datos del nuevo enlace en la base de datos, se construye además el mensaje que avisa que la operación fue realizada.

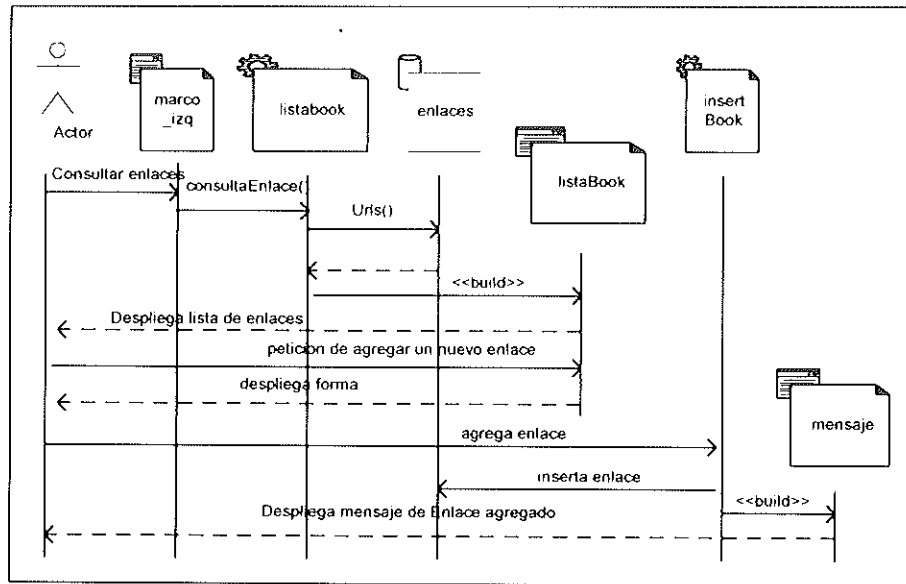


Figura 38. Diagrama de secuencia consulta de enlaces.

ANEXO D

La tabla Curso (tabla I) almacena la información de los cursos impartidos. Los atributos clave_curso, grupo, se utilizan como una llave compuesta para identificar a cada curso y así relacionarlo con los datos que le corresponden. Los atributos restantes contienen información general de los cursos.

Nombre	Tipo	Dominio
clave_curso	Varchar(4)	a-z / 0-9
Grupo	Varchar(5)	a-z / 0-9
Nombre	Varchar(70)	a-z
Carrera	Varchar(5)	integer
Modalidad	Varchar(15)	a-z / 0-9 *
Presentación	text	a-z / 0-9 *
Eq_trabajo	text	a-z / 0-9 *
Objetivo	text	a-z / 0-9 *
Problema_guia	text	a-z / 0-9 *
Actitudes	text	a-z / 0-9 *
Producto	text	a-z / 0-9 *
Tiempo	text	a-z / 0-9 *
Acreditacion	text	a-z / 0-9 *
Materiales	text	a-z / 0-9 *
Bibliografía	text	a-z / 0-9 *
Evaluación	text	a-z / 0-9 *

Tabla I. Tabla de la entidad Curso.

La tabla Alumno (tabla II) almacena los datos de los alumnos inscritos. El atributo matricula es la llave primaria, y se utiliza para recuperar los datos específicos de cada alumno, almacenados en los campos restantes de la tabla. Este atributo además se utiliza para relacionar las actividades efectuadas a lo largo del curso por un alumno, se crea una dependencia para obtener las clases que toma, tareas y proyectos realizados, calificaciones, etc.

Nombre	Tipo	Dominio
Matricula	Varchar(10)	a-z / 0-9
Login	Varchar(20)	a-z / 0-9 / - _
Password	Varchar(20)	a-z / 0-9
Nombre	Varchar(120)	a-z
Correo	Varchar(50)	a-z / 0-9 / - _ @
Dirección	Varchar(50)	a-z / 0-9 / - _
Semestre	Integer	0-9
Carrera	Varchar(5)	0-9

Tabla II. Tabla de la entidad Alumno.

La tabla Alumnos_curso (tabla III) establece la relación de los alumno con los cursos en los que esta inscrito. El atributo matricula es utilizada para relacionar al alumno con el curso a través de los campos clave_curso y grupo.

Nombre	Tipo	Dominio
Matricula	Varchar(10)	0-9
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9

Tabla III. Tabla de la entidad Alumno_curso.

La tabla Docente (tabla IV) almacena información general del docente. Los datos de esta tabla se recuperan con la llave primaria num_empleado. Con este se van generando relaciones a lo largo del curso. Un caso seria la asignación de tareas, dado que más de un docente pueden impartir un curso debemos relacionar que docente asigno que tarea.

Nombre	Tipo	Dominio
Num_empleado	Varchar(10)	0-9
Nombre	Varchar(120)	a-z
Grado	Varchar(20)	a-z
Carrera	Varchar(5)	0- 9
dir_academica	Varchar(100)	a-z / 0-9 *
Hrs_localizacion	Varchar(200)	a-z / 0-9 *
Telfax	varchar(60)	0-9
Correo	Varchar(80)	a-z / 0-9 / @
Dir_url	Varchar(80)	a-z / 0-9 / @ - _ .

Proy_actuales	text	a-z / 0-9 *
Areas_interés	text	a-z / 0-9 *
Login	Varchar(20)	a-z
Password	Varchar(20)	a-z / 0-9 / _ -

Tabla IV. Tabla de la entidad Docente.

La tabla Docente_curso (tabla V) establece la relación de los cursos impartidos con los docentes a cargo. La tabla esta formada por el atributo num_empleado, que es utilizado para recuperar al docente y relacionarlo con los cursos a los que esta signado a través de las claves clave_curso, grupo.

Nombre	Tipo	Dominio
Num_empleado	Varchar(10)	0-9
Clave_curso	Varchar(4)	a-z / 0-9
Grupo	Varchar(5)	a-z / 0-9

Tabla V. Tabla de la entidad Docente_curso.

La tabla criterios_evaluación (tabla VI) almacena las categorías con las que se va a evaluar el curso, estas pueden ser tareas, exámenes, practicas, proyecto, etc; Cada una de estas tiene asignado un valor o peso, es decir grado para definir un estereotipo de calidad. Este grado es asignado por cada maestro a cargo del curso.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(4)	a-z / 0- 9
Tipo	Varchar(20)	a-z
Porcentaje	Integer	0- 9
Descripción	Varchar(255)	a-z / 0- 9 / *

Tabla VI. Tabla de la entidad criterios de evaluación.

En esta tabla (tabla VII) se almacenan las unidades de las que esta formado un curso, son almacenada por cada maestro a cargo del curso. La tabla consiste en almacenar el numero de unidad de manera creciente junto con un nombre descriptivo de la unidad.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Unidad	Integer	0- 9
Nombre	Varchar(100)	a-z / 0- 9

Tabla VII. Tabla de la entidad Unidades.

Cada unidad esta formado de temas, estos temas o tópicos describen detalladamente de que se trata la unidad. La tabla topicos (tabla VIII) almacena el detalle de cada unidad. Sus atributos crean una relación con una unidad especifica de un curso especifico.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Unidad	Integer	0- 9
Topico	Integer	0-9
Nombre	Varchar(100)	a-z/ 0-9 / *

Tabla VIII. Tabla de la entidad Topicos.

En la tabla Documento(tabla IX) trasladada de la clase Documento, se almacena la información de los documentos de los cursos. Como se menciona cuenta con la llave compuesta clave_curso, grupo para la recuperación de datos de un curso especifico. Los atributos restantes relacionan al docente que lo puso a disposición de los alumnos además de describir el documento.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Título	Varchar(80)	a-z / 0- 9
Descripción	Varchar(255)	a-z / 0- 9 / *
Fecha	Date	0- 9
Num_empleado	Varchar(10)	0- 9
Archivo	Varchar(60)	a-z / 0- 9
Unidad	Integer	0- 9
Topico	Integer	0- 9

Tabla IX. Tabla de la entidad Documento.

La tabla Tarea (tabla X) almacena las actividades que el maestro va asignando durante el curso. Además de tener la llave compuesta para relacionarla con un curso, tiene una segunda llave compuesta formada por los atributos tipo y numtarea, esta llave es utilizada cuando se consultan los datos específicos de una actividad, como pueden ser tareas o exámenes.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Nombre	Varchar(60)	a-z / 0- 9
Tipo	Varchar(15)	a-z
Numtarea	Integer	0- 9
Fecha_ins	Date	0-9 / /
Fecha_exp	Date	0-9 / /
Hora	time	0-9 / :
Num_empleado	Varchar(10)	0-9
Nota	Varchar(255)	a-z / 0- 9
Archivo	Varchar(60)	a-z / 0- 9

Tabla X. Tabla de la entidad Tarea.

La tabla Buzon (tabla XI) es requerida para el almacenamiento de actividades que los alumnos entregan durante el curso. Uno de sus atributos es matricula, la cual es una llave foránea para relacionar el archivo entregado con el alumno que realizó la entrega. Los atributos tipo y num_tarea son importantes para generar una correspondencia entre las tareas

entregadas y las asignaciones del maestro. El atributo versión es el equivalente al número de veces que se puede entregar un archivo y, es utilizado para conocer el estado del buzón.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Matricula	Varchar(8)	0- 9
Tipo	Varchar(15)	a-z
Archivo	Varchar(60)	a-z / 0- 9
Fecha_ent	Date	0-9/ /
Hora	Time	0- 9 / :
Versión	integer	0-9
Calificación	Float	0-9
Numtarea	Integer	0-9

Tabla XI. Tabla de la entidad Buzon.

La tabla Proyecto (tabla XII) almacena los datos de los proyectos. El atributo id_proy, se utiliza como llave foránea para recuperar información adicional del proyecto, almacenada en otras tablas como por ejemplo los alumnos que lo desarrollan y los comentarios asignados por los docentes.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Id_proy	Integer	0-9
Nombre	Varchar(80)	a-z / 0- 9
Fecha_inicio	Date	0- 9

Tabla XII. Tabla de la entidad Proyecto.

La tabla Alumno_proyecto (tabla XIII) establece la relación de cada proyecto con el alumno que lo desarrolla. Formado por el atributo matricula, utilizado para referirlo con el alumno que lo desarrolla y el atributo id_proy el cual se utiliza para relacionarlo con el proyecto que desarrolla. En esta tabla se almacena también el dato calificación, dada la relación 1:1.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Matricula	Varchar(8)	0-9
Actividad	Varchar(40)	a-z / 0- 9 / *
Calificación	Flota	0-9
Id_proyecto	Integer	0-9

Tabla XIII. Tabla de la entidad Alumno_proyecto.

En la Tabla comentario_proy (tabla XIV), se almacena los archivos que se han entregado del proyecto, además del comentario proporcionado por el docente dada la revisión del archivo. Se almacena. La tabla relaciona los archivos a un proyecto específico de un curso, esto por los atributos id_proy, clave_curso, grupo.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Id_proy	Integer	0- 9
Comentario	Varchar(255)	a-z / 0- 9 / *
Archivo	Varchar(60)	a-z / 0- 9 / -
Fecha	Date	0-9 / /

Tabla XIV. Tabla de la entidad Comentario Proyecto.

La tabla Anuncio (tabla XV) almacenan los datos de los anuncios de los cursos. Los datos son almacenados con la llave compuesta clave_curso, grupo, ya que pertenecen a un único curso. Los anuncios sólo pueden ser almacenados por el docente es por eso que la tabla tiene el atributo autor, para relacionar el anuncio con el docente que lo agrego.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0- 9
Grupo	Varchar(5)	a-z / 0- 9
Fecha	Date	0-9 / /
Contenido	Varchar(255)	a-z / 0- 9 / *
Autor	Varchar(10)	a-z
Tipo	char(1)	n/u
Vigencia	date	0-9 / /

Tabla XV. Tabla de la entidad Anuncio.

La tabla Enlaces (tabla XVI) almacena los datos de los enlaces proporcionados por los distintos usuarios. Esta tabla únicamente tiene relación con el curso pues cualquier usuario del curso sea maestro o alumno puede agregar enlaces. curso específico.

Nombre	Tipo	Dominio
Clave_curso	Varchar(4)	a-z / 0-9
Grupo	Varchar(5)	a-z / 0- 9
Url	Varchar(255)	a-z / 0-9 / ~& _ - +
Descripción	Varchar(255)	a-z / 0-9
Autor	Varchar(8)	a-z
Fecha	Date	0-9 / /

Tabla XVI. Tabla de la entidad Enlaces.

ANEXO E

Al consultar la información del docente (Figura 39) se despliega una lista de nombres de quienes imparten el curso. Cada nombre en la lista conduce a información específica de un docente. Esta información es proporcionada por cada maestro.

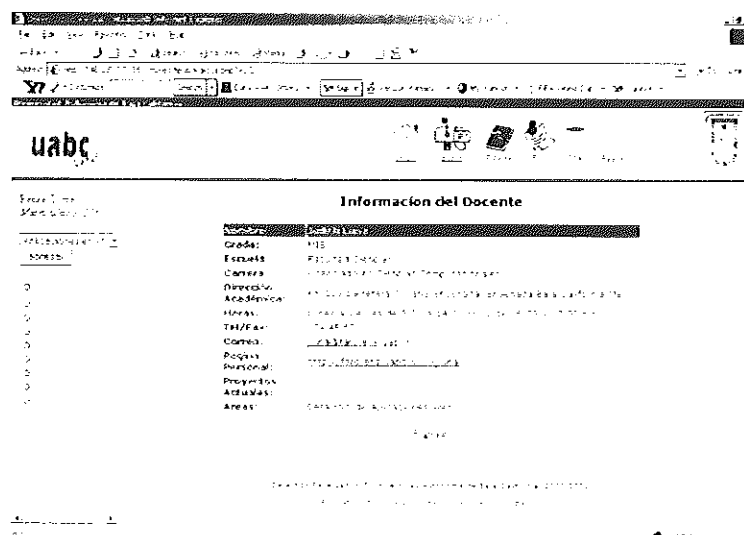


Figura 39. Pantalla de consulta de información del curso.

En la opción de proyectos (Figura 40) se muestra una pantalla que proporciona un buzón de archivos en los que el alumno agrega avances del proyecto del curso. Conforme se agregan archivos, se despliegan en la parte superior del buzón la lista de archivos anteriormente entregados, además del comentario que el docente haya proporcionado al evaluar cada avance.

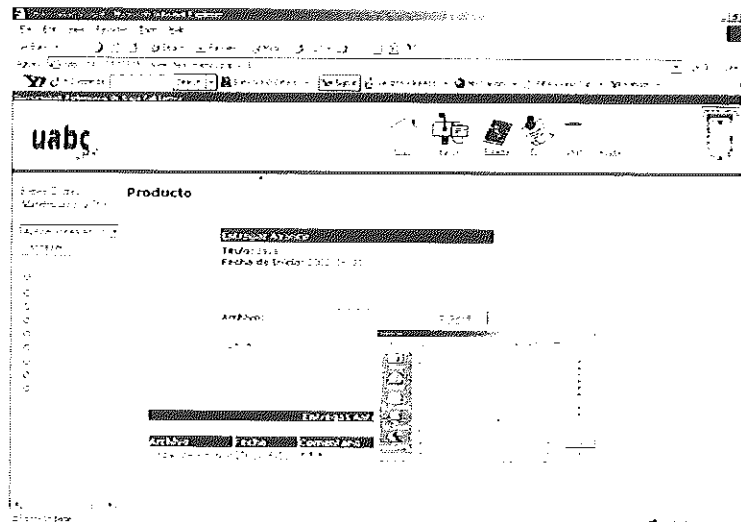


Figura 40. Pantalla consulta y entrega de actividades.

Al consultar la opción de calificaciones (Figura 41), se despliega una tabla que lista el nombre de las entidades con las que se evalúa el curso. Como se menciono anteriormente estas entidades pueden ser practicas, tareas, exámenes, proyectos, etc. Aunado a cada elemento del listado se proporciona el peso que tiene para la calificación del curso, el porcentaje acumulado por el alumno y calificación. También cada elemento de la lista proporciona una liga que conduce al detalle de la calificación.

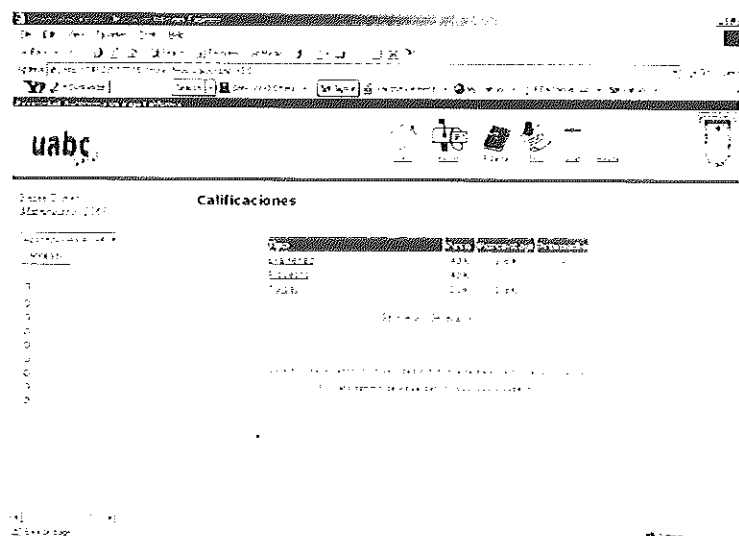


Figura 41. Pantalla de consulta de calificaciones.

En la opción de anuncios (Figura 42) se muestra una lista de mensajes que el docente ha ido colocando durante el curso, cada mensaje tiene una fecha de expiración, al llegar esta fecha el mensaje es borrado del sistema. Los anuncios son clasificados como urgentes o como normales, distinguiéndose por el color, donde el urgente sobresale por ser más llamativo.

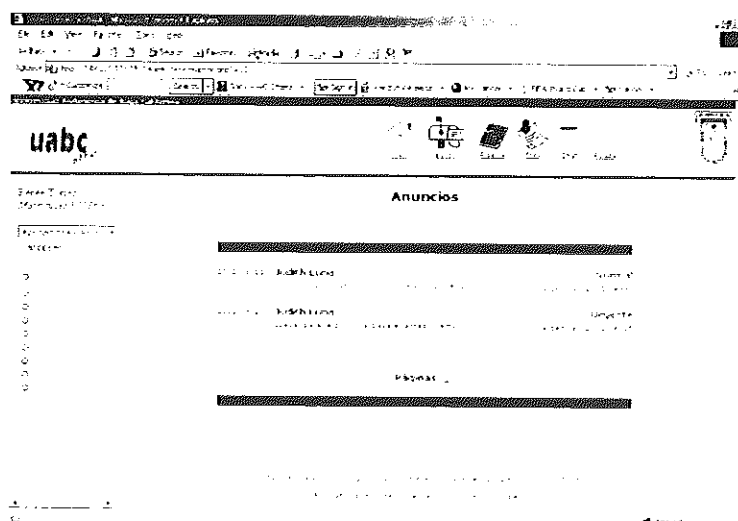


Figura 42. Pantalla de consulta de Anuncios.

Al elegir la opción de enlaces (Figura 43) se muestra una lista de direcciones de Internet, estos han sido agregados tanto por el docente como el alumno a largo del curso. En la lista, las direcciones de Internet existentes son activadas para ser consultadas directamente desde el sistema, abriendo una ventana nueva y continuar la sesión de trabajo.

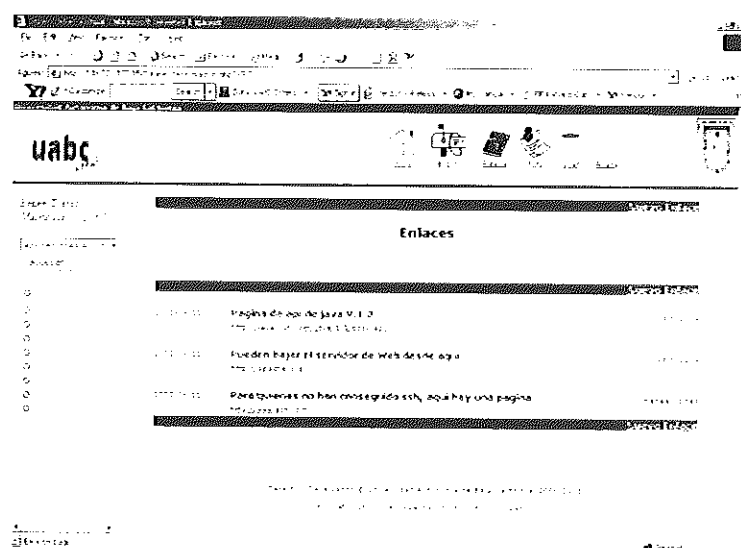


Figura 43. Pantalla de consulta de Enlaces.