



**BENEMÉRITA UNIVERSIDAD
AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

**SISTEMA GENERADOR DE CURSOS EDUCATIVOS
MULTIMEDIA EN LINEA PARA EL APRENDIZAJE A
DISTANCIA, USANDO PHP Y MYSQL**

**TESIS
QUE PARA OBTENER EL TITULO DE:**

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

**PRESENTA:
JUÁREZ SIERRA AARÓN**

**ASESOR:
DR. MARIO ROSSAINZ LÓPEZ**

**COASESOR:
MC. LETICIA MENDOZA ALONSO**

Puebla, Pue. Otoño 2007

Agradecimientos

A Dios por darme el entusiasmo, la voluntad, la inteligencia y la fuerza necesaria para seguir adelante en la realización de mis metas y proyectos.

A mis asesores por brindarme su conocimiento, paciencia, apoyo y orientación.

A mi familia, por que en ella encontré los ánimos para continuar.

A mi mamá, por su apoyo, cariño y comprensión, que siempre me ha brindado.

A Rebeca por su paciencia, interés y apoyo en la realización de mi proyecto de tesis.

Gracias a Dios y a todas estas personas así como a todos mis amigos y compañeros, por que sin ellos, mi trabajo, esfuerzo, tiempo y dedicación no podrían haber transcurrido.

Índice

AGRADECIMIENTOS	2
ÍNDICE	3
INTRODUCCIÓN.....	6
CAPITULO 1.....	9
MARCO TEÓRICO.....	9
1.1. PRINCIPALES CONCEPTOS DE INGENIERÍA DE SOFTWARE	9
1.1.1. <i>Software</i>	9
1.1.2. <i>Ingeniería de software</i>	9
1.1.3. <i>Métodos de ingeniería de software</i>	9
1.1.4. <i>Proceso de Software</i>	10
1.1.5. <i>Modelos o paradigmas para el desarrollo de software</i>	10
1.2. SERVICIOS WEB.....	12
1.2.1. <i>Internet</i>	12
1.2.2. <i>La WWW</i>	13
1.2.2.1. El Protocolo HTTP.....	13
1.2.2.2. El lenguaje HTML	14
1.2.3. <i>Arquitectura Cliente – Servidor</i>	14
1.2.3.1 Características de la Arquitectura Cliente – Servidor.....	14
1.3. APLICACIONES WEB	15
1.4. CONCEPTOS BÁSICOS DE SERVIDORES WEB	16
1.4.1. <i>Servicio de ficheros estáticos</i>	16
1.4.2. <i>Seguridad y autenticación</i>	17
1.4.3. <i>Contenido Dinámico</i>	17
1.4.4. <i>Servidores virtuales</i>	18
1.4.5. <i>Prestación Extra</i>	18
1.4.6. <i>Actuación como representantes</i>	19
1.4.7. <i>Protocolos adicionales</i>	19
1.5. EL SERVIDOR WEB APACHE	20
1.6. OTROS SERVIDORES WEB DE SOFTWARE LIBRE.....	20
1.6.1. <i>AOLServer</i>	20
1.6.2. <i>Roxen y Caudium</i>	21
1.6.3. <i>thttpd</i>	22
1.6.4. <i>Jetty</i>	22
1.7. ARQUITECTURA DE SOFTWARE PARA WEB (CONTENIDO DINÁMICO).....	22
1.7.1 <i>CGI</i>	22
1.7.2 <i>PHP</i>	23
1.7.2.1. <i>Cómo Funciona PHP</i>	23
1.7.3. <i>Java Servlets y JSP</i>	24
1.7.3.1. <i>Introducción a los Java Servlets</i>	24
1.7.4. <i>Otras opciones de contenido dinámico</i>	25
1.8. BASES DE DATOS.....	27
1.8.1. <i>Historia de los sistemas de bases de datos</i>	27
1.8.2. <i>Sistema de Base de datos</i>	28
1.9. SISTEMA MANEJADOR DE BASE DE DATOS (DBMS)	28
1.9.1. <i>Niveles de abstracción de datos</i>	29
1.10. MODELOS DE DATOS.	29
1.10.1. <i>Modelos lógicos basados en objetos</i>	29
1.10.2. <i>Modelos lógicos basados en Registros</i>	31
1.11. ESTRUCTURA DE DATOS	31
1.12. REGLAS DE INTEGRIDAD.....	32
1.13. LA MANIPULACIÓN DE LOS DATOS.	32
1.14. NORMALIZACIÓN	32

1.14.1. Primer Nivel de Normalización	33
1.14.2. Segundo Nivel de Normalización	33
1.14.3. Tercer Nivel de Normalización	33
1.14.4. Cuarto Nivel de Normalización	33
1.14.5. Quinto Nivel de Normalización.....	33
1.15. EL LENGUAJE RELACIONAL SQL.....	34
1.16. SISTEMA DE GESTIÓN DE BASE DE DATOS MYSQL.....	34
1.16.1. Características generales de MySQL.....	35
1.17. RELACIÓN QUE EXISTE ENTRE LOS DISTINTOS ELEMENTOS QUE INTEGRAN EL MARCO TEÓRICO (ING. DE SOFTWARE, APLICACIONES WEB, ARQUITECTURAS DE SOFTWARE Y BASES DE DATOS).....	35
CAPITULO 2.....	37
ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA	37
2.1. ESPECIFICACIÓN DEL PROBLEMA	37
2.2. IDENTIFICACIÓN DE ACTORES Y CASOS DE USO.....	38
2.2.1. Actores del Sistema.....	38
2.2.2. Casos de Uso del Sistema	38
2.3. DIAGRAMAS DE CASO DE USO Y ESPECIFICACIÓN DE FUNCIONAMIENTO (RUP RATIONAL ROSE).....	40
2.3.1. Caso de Uso Administración de Cursos.....	41
2.3.2. Caso de Uso Administración de contenido didáctico	42
2.3.3. Caso de Uso Administración de evaluaciones	44
2.3.4. Caso de Uso Establecer contacto con alumnos vía correo electrónico.....	46
2.3.5. Caso de Uso Inscripción a cursos.....	47
2.3.6. Caso de Uso Baja de alumnos	48
2.3.7. Caso de Uso Seguimiento de evaluaciones.....	49
2.3.8. Caso de Uso Realizar autoevaluaciones.....	50
2.3.9. Caso de Uso Revisar resultado de autoevaluaciones	51
2.3.10. Caso de Uso Revisar o estudiar contenido didáctico.....	52
2.3.11. Caso de Uso Establecer contacto con profesor vía correo electrónico.....	53
2.4. DIAGRAMAS DE ACTIVIDADES.....	54
CAPÍTULO 3.....	57
MODELO LÓGICO CONCEPTUAL.....	57
3.1. MODELO LÓGICO CONCEPTUAL: DIAGRAMA DE CLASES	57
CAPÍTULO 4.....	62
MODELO DE NAVEGACIÓN.....	62
4.1 ESTEREOTIPOS UTILIZADOS EN EL MODELO DE NAVEGACIÓN	62
4.1.1 Navigation Class.....	62
4.1.2 Form	63
4.1.3 Link	63
4.1.4 Submit	63
4.2 SEMÁNTICA DE ESTEREOTIPOS ASOCIADA CON LA WEB	64
4.2.1 Acerca del estereotipo <<navigation class>>	64
4.2.2 Acerca del estereotipo <<form>>	65
4.3 MODELOS DE NAVEGACIÓN DE LOS SUBMODULOS	66
4.3.1 Modulo administración de cursos.....	66
4.3.2 Modulo administración de contenido didáctico.....	67
4.3.3 Modulo administración de evaluaciones	69
4.3.4 Modulo establecer contacto con los alumnos vía correo electrónico.....	71
4.3.5 Modulo inscripción a cursos.....	72
4.3.6 Modulo baja de alumnos.....	73
4.3.7 Modulo seguimiento de evaluaciones	74
4.3.8 Modulo realizar autoevaluaciones.....	75

4.3.9 Modulo revisar resultado de autoevaluaciones	76
4.3.10 Modulo revisar o estudiar contenido didáctico	77
4.3.11 Modulo establecer contacto con los profesores vía correo electrónico.....	78
CAPÍTULO 5.....	79
ESCENARIOS WEB.....	79
5.1 LOS ESCENARIOS WEB Y LA SEMÁNTICA ASOCIADA A LOS DIAGRAMAS DE ESTADOS	79
5.2 ESCENARIOS WEB DEL SGCAD	81
5.2.1 Escenario Web administración de cursos	81
5.2.2 Escenario Web administración de contenido didáctico	81
5.2.3 Escenario Web administración de evaluaciones.....	83
5.2.4 Escenario Web Establecer contacto con los alumnos vía correo electrónico.....	85
5.2.5 Escenario Web inscripción a cursos.....	86
5.2.6 Escenario Web baja de alumnos.....	87
5.2.7 Escenario web Seguimiento de evaluaciones.....	88
5.2.8 Escenario Web Realizar autoevaluaciones	89
5.2.9 Escenario Web Revisar resultado de autoevaluaciones.....	90
5.2.10 Escenario Web Revisar o estudiar contenido didáctico.....	91
5.2.11 Escenario Web Establecer contacto con los profesores vía correo electrónico	92
CAPITULO 6.....	93
ESQUEMA DE IMPLEMENTACIÓN.....	93
6.1 SEMÁNTICA DE ELEMENTOS ASOCIADOS AL DIAGRAMA DE COMPONENTES	93
6.2 ESQUEMA DE IMPLEMENTACIÓN DEL SGCAD.....	93
6.3. EL SISTEMA GENERADOR DE CURSOS EDUCATIVOS MULTIMEDIA EN LA PRÁCTICA	95
6.3.1. Usuario profesor.....	95
6.3.1.1. Operaciones sobre alumnos	96
6.3.1.2 Operaciones sobre cursos	99
6.3.1.3 Operaciones sobre contenido didáctico	100
6.3.1.4 Operaciones sobre autoevaluaciones	103
6.3.2. Usuario alumno	106
6.3.2.1. Realizar autoevaluaciones	106
6.3.2.2. Calificaciones	107
6.3.2.3 Contenido didáctico.....	107
6.3.2.4 Contacto profesores	108
CAPÍTULO 7.....	109
CONCLUSIONES.....	109
APÉNDICE A	111
A.1 SUGERENCIA DE LENGUAJE DE PROGRAMACIÓN, SERVIDOR DE BASE DE DATOS Y SERVIDOR WEB	111
A.1.1 PHP	111
A.1.2 MySQL.....	112
A.1.3 Apache.....	112
A.2 HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB	113
A.2.1 Editor de Páginas Web: Macromedia Dreamweaver.....	113
A.2.2 Editor UML: Rational Rose.....	114
BIBLIOGRAFÍA	115

Introducción

El sistema que se presenta en esta tesis lleva por nombre “Sistema Generador de cursos Educativos Multimedia En Línea Para el Aprendizaje a Distancia, Usando PHP y MYSQL”, en lo sucesivo SGCAD; como parte del proyecto financiado VIEP 21/ING/06-G denominado “Patrones de Diseño para el Desarrollo de Sistemas Multimedia Educativo y Sistemas de Aprendizaje Colaborativo como Apoyo a la Educación Presencial y a Distancia”, este sistema esta diseñado con el fin de que los usuarios aprendan cualquiera de los tópicos en la modalidad “A distancia”; es decir, no presencial, con acceso directo a través de la Web. Debido a que es un sistema de aprendizaje no presencial, el sistema estará disponible en la Internet, con ello se contarán con varias de las ventajas brindadas por sistemas basados en páginas Web dinámicas usando tecnología de software abierto (PHP, MySQL).

El sistema cuenta con diversas opciones tanto para el alumno como para el profesor, estas por el lado del profesor son respecto a la creación de cursos, desarrollo, creación, incorporación y presentación del material y contenido didáctico de los mismos, administración respecto a la alta o baja de los alumnos en los cursos, creación y aplicación de evaluaciones en línea, visualización, manipulación y autorización de las evaluaciones, contacto personalizado con alumnos por medio de correo electrónico. Por el lado del alumno presentar evaluaciones en línea y visualización de resultados de las mismas, estudio en línea del contenido didáctico del curso y establecer contacto vía correo electrónico con sus profesores.

Así mismo el tema de estudio en cuestión a través de dicho sistema esta basado en un modelo educativo adhoc que garantice el aprendizaje de la persona que lo utilice ya que al ser un sistema educativo a distancia es de vital importancia comprobar el resultado del aprendizaje no sólo a través de instrumentos de evaluación, sino sustentados en un modelo educativo correcto.

Esta investigación resuelve las preguntas de: ¿Cómo surgieron los sistemas educativos multimedia? ¿Cómo han evolucionado?, ¿Qué beneficios se tiene con la utilización de los mismos?, su estructura, su diseño, etc.

La tesis se encuentra estructurada por siete capítulos, los cuales describen un marco teórico, análisis de requerimientos del sistema, modelo lógico conceptual, modelo de navegación, escenarios web, esquema de implementación, y conclusiones; todo ello basado en la metodología UWE específica para la ingeniería de software de aplicaciones bajo la Web utilizando UML.

CAPITULO I. MARCO TEÓRICO. En este capítulo se mencionan los conceptos básicos de ingeniería de software, los distintos métodos de ingeniería de software integrados a un solo enfoque unificado, basado en UML, los procesos de software que involucra detallando, actividades y resultados asociados que conducen a la creación de un producto de software.

Se habla de conceptos fundamentales como (Tipos de servicios web, internet, la www, el protocolo HTTP, el lenguaje HTML, y la arquitectura Cliente – Servidor) que son básicos para entender como se lleva a cabo el proceso de comunicación de la información en las diferentes computadoras.

Los sistemas, tecnologías y lenguajes concebidos para desarrollar contenido Web dinámico, que los hacen, más portables, más potentes, mucho más eficientes, más fáciles de usar, más escalables, etc.

Se detalla la historia de los sistemas de bases de datos, los modelos que existen para representarlos y las reglas que existen para realizar una base de datos sencilla hasta sistemas complejos con criterios bien definidos que hacen un sistema entendible y bien relacionado sin cavidad a ambigüedades.

CAPITULO II. ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA. El capítulo comienza describiendo la especificación del problema que consiste en el desarrollo de un sistema Web dinámico, generador de cursos educativos multimedia en línea, para el aprendizaje a distancia, de distintos tópicos impartidos dentro de la Facultad de Ciencias de la Computación de la BUAP, usando tecnología de software abierto (PHP, MySQL).

El sistema dará servicio a sus dos usuarios principales, que serán: profesores y alumnos; por tanto cada uno tiene distintas necesidades, descritas en este capítulo, para con ello identificar actores y casos de uso, representando las necesidades o requerimientos de los usuarios futuros del sistema.

Se visualizan los distintos diagramas de casos de uso y las especificaciones de funcionamiento con (RUP, Rational Rose), donde se detallan aspectos como: quién inicia el caso de uso, la secuencia que debe seguir el actor, la respuesta del sistema a la actividad indicada y los flujos alternativos en situaciones comunes o especiales.

CAPITULO III. MODELO LÓGICO CONCEPTUAL. En este capítulo se presenta el modelo lógico conceptual del SGCAD. Este modelo describe una primera abstracción de los elementos de la arquitectura de la aplicación Web mediante un diagrama de clases simple de UML; haciendo énfasis en el flujo lógico entre elementos, en su semántica y evitando hacer referencia en la medida de lo posible de cuestiones relacionadas con la navegación y de los aspectos de interacción; además es útil para guiar el diseño de la base de datos de la aplicación.

CAPITULO IV. MODELO DE NAVEGACIÓN. Este capítulo explica la parte más elemental del análisis y diseño del SGCAD: el diseño del modelo de navegación. En un primer apartado se describe la semántica de los elementos de modelado que se utilizan en los diagramas de clases estereotipadas.

Se listan otras consideraciones con relación a convenciones y notaciones que se adoptan en el modelo de navegación.

Se presenta el diagrama de clases estereotipado para cada módulo del sistema: Administración de cursos, Administración de contenido didáctico, Administración de evaluaciones, Establecer contacto con los alumnos vía correo electrónico, Inscripción a

cursos, Baja de alumnos, Seguimiento de evaluaciones, Realizar autoevaluaciones, Revisar resultado de autoevaluaciones, Revisar o estudiar contenido didáctico, Establecer contacto con los profesores vía correo electrónico.

CAPITULO V. ESCENARIOS WEB. El capítulo detalla los Escenarios de Navegación del SGCAD. Los diagramas UML que se utilizan para visualizar estos escenarios de navegación son los diagramas de estados. Se describe la semántica de los elementos de modelado que se utilizan en los diagramas de estado. Posteriormente, se presentan los escenarios de navegación más significativos para el SGCAD. Los escenarios Web son una técnica para la descripción de situaciones de uso de un sitio Web con una meta o espacio concreto. Estas situaciones son el resultado de acciones llevadas a cabo por un personaje (usuario) o bien por la propia aplicación.

CAPITULO VI. ESQUEMA DE IMPLEMENTACIÓN. El capítulo presenta un único diagrama de componentes mediante el cual se proporciona una vista general del esquema de implementación que se propone para el SGCAD. En un primer apartado se describe la semántica asociada a los elementos de modelado que se utilizan en el diagrama. Posteriormente se muestra el diagrama de componentes y los comentarios correspondientes. Además, las asociaciones pueden enriquecer su semántica en el diagrama mediante la utilización de estereotipos que denoten con mayor claridad el tipo de relación que se establece entre los componentes y/o nodos.

Esta propuesta de implementación se plantea con la utilización de tecnologías de uso frecuente en el ámbito del desarrollo de aplicaciones Web tales como PHP y MySQL.

También se muestra un ejemplo del uso del sistema en la práctica; es decir el funcionamiento real del mismo, apoyándose para su explicación en imágenes de pantallas o de la interfaz con la que interactúa el usuario durante el uso del sistema.

CAPITULO VII. CONCLUSIONES. El capítulo presenta las conclusiones obtenidas del desarrollo de esta tesis, cuyo objetivo principal en un principio fue realizar un modelo en UML de una aplicación Web para el aprendizaje a distancia, usando PHP y MySQL, dicho objetivo mejoró como inicialmente se había planteado; es decir el objetivo principal consistió en el desarrollo de un sistema, que no solo se limitara a la enseñanza de un solo tópico, sino que permitiera obtener una aplicación Web para la generación de cursos educativos, incorporando elementos multimedia (audio, imágenes y video), para el aprendizaje a distancia de cualquier tópico impartido dentro de la Facultad de Ciencias de la Computación de la BUAP.

Apéndice A. Se hace referencia a un anteproyecto de software para la implementación del Sistema Generador de Cursos Educativos Multimedia en Línea Para el Aprendizaje a Distancia. En este anteproyecto se incluyen propuestas a las cuestiones que se deben tomar en cuenta en el desarrollo de las aplicaciones de software, en este caso de las aplicaciones Web. Se comentan algunas sugerencias de herramientas útiles en el desarrollo de aplicaciones Web, enfocadas en el SGCAD.

Capítulo 1

Marco Teórico

Actualmente casi todos los países dependen de sistemas complejos basados en computadora. El software en estos sistemas representa una proporción grande y creciente costo del sistema. Por lo tanto, producir software costeable es esencial para el funcionamiento de la economía nacional e internacional. Para el desarrollo de estos sistemas de software es necesario hacer uso de distintas herramientas y además seguir una metodología para el desarrollo del mismo a continuación se hará la descripción de los conceptos más relevantes relativos a los sistemas de información. [1]

1.1. Principales Conceptos de Ingeniería de Software

1.1.1. Software

El software es el conjunto de programas, todos los documentos asociados a estos y la configuración de datos que se necesitan para hacer que estos programas funcionen de manera correcta. [1]

1.1.2. Ingeniería de software

La ingeniería de software es una disciplina que comprende todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. Para aplicar esta disciplina es necesario guiarnos por los métodos de Ingeniería de Software. [1]

1.1.3. Métodos de ingeniería de software

Un método de ingeniería de software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable. Métodos como Análisis Estructurado (DeMarco, 1978) y JSD (Jackson, 1983) fueron los primeros desarrollados en los años 70. Estos métodos intentaron identificar los componentes funcionales básicos de un sistema de tal forma que los métodos orientados a funciones aún se utilizan ampliamente. En los años 80 y 90, estos métodos orientados a funciones fueron complementados por métodos orientados a objetos, como los propuestos por Booch (1994) y Rumbaugh (Rumbaugh *et al.*, 1991). Estos diferentes enfoques se han integrado a un solo enfoque unificado, basado en UML (Lenguaje de Modelado Unificado) (Fowler y Scout, 1997); booch *et al.*, 1999; Rumbaugh *et al.*, 1999a, 1999b). Todos los métodos se basan en la idea de modelos gráficos de desarrollo de un sistema y en el uso de estos modelos como un sistema de especificación o diseño. [1]

1.1.4. Proceso de Software

Es el conjunto de actividades y resultados asociados que conducen a la creación de un producto de software. Estas actividades son realizadas por los desarrolladores de software. Existen cuatro actividades fundamentales que son comunes para todos los procesos del software. [1]

1. *Especificación del software*: Se debe definir la funcionalidad del software y las restricciones en sus operaciones.
2. *Diseño e implementación del software*: Se debe producir software que cumpla su especificación.
3. *Validación del software*: Se debe validar el software para asegurar qué hace lo que el cliente desea.
4. *Evolución del software*: El software debe evolucionar para cumplir los cambios en las necesidades del usuario.

1.1.5. Modelos o paradigmas para el desarrollo de software:

1. **El modelo de cascada**. Esta formado por un conjunto de etapas las cuales son representadas como fases de procesos separados fundamentales para el desarrollo del sistema, como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etc. [1]

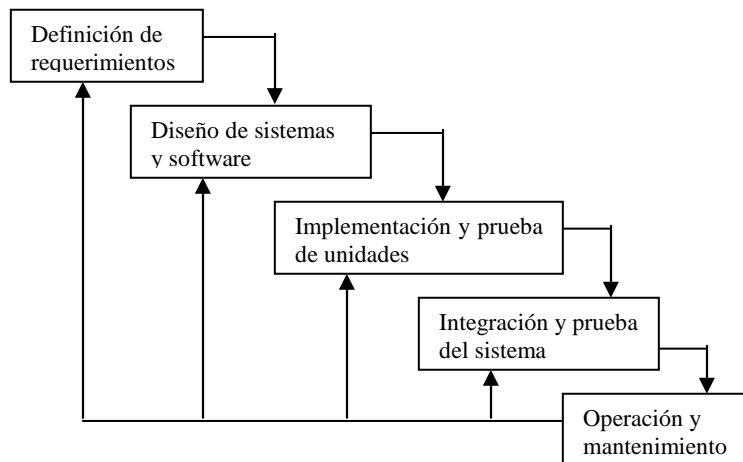


Fig. 1.1. El ciclo de vida del software “Cascada”

2. **Desarrollo Evolutivo.** Éste se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. [1]

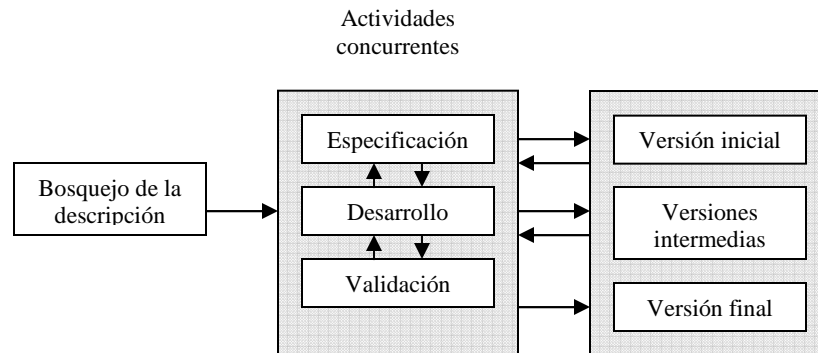


Fig. 1.2. Desarrollo evolutivo

3. **Transformación Formal.** Este enfoque se basa en la transformación de la especificación, estas transformaciones conservan las correcciones anteriores y se puede estar seguro de que el desarrollo del producto cumple con la especificación. [1]

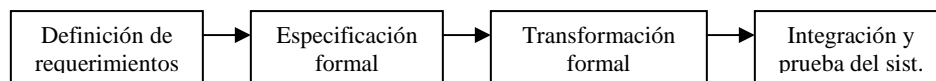


Fig. 1.3. Desarrollo formal de sistemas

4. **Sistema de Ensamblaje de Componentes Reutilizables.** Este enfoque supone que las partes que conformarán al sistema ya existen, y se enfoca a la integración de esas partes más que construirlo desde sus inicios. [1]

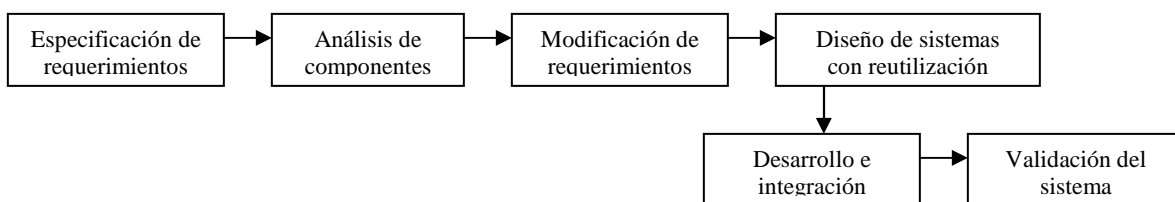


Fig. 1.4. Desarrollo orientado a reutilización

1.2. Servicios Web

Los Servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente, en las cuales el software está distribuido en diferentes servidores.

1.2.1. Internet

Internet, la red de redes, nace a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente.

De estos proyectos nació un protocolo de comunicaciones de datos, IP o Internet Protocol, que permitía a ordenadores diversos comunicarse a través de una red, Internet, formada por la interconexión de diversas redes.

A mediados de los ochenta la Fundación Nacional para la Ciencia norteamericana, la NSF, creó una red, la NSFNET, que se convirtió en el *backbone* (el troncal) de Internet junto con otras redes similares creadas por la NASA (NSINet) y el U.S. DoE (Department of Energy) con la ESNET. En Europa, la mayoría de países disponían de *backbones* nacionales (NORDUNET, RedIRIS, SWITCH, etc.) y de una serie de iniciativas paneuropeas (EARN y RARE). En esta época aparecen los primeros proveedores de acceso a Internet privados que ofrecen acceso pagado a Internet.

A partir de esta época, gracias entre otras cosas a la amplia disponibilidad de implementaciones de la *suite* de protocolos TCP/IP (formada por todos los protocolos de Internet y no sólo por TCP e IP), algunas de las cuales eran ya de código libre, Internet empezó lo que posteriormente se convertiría en una de sus características fundamentales, un ritmo de crecimiento exponencial, hasta que a mediados del 2002 empieza a descender ligeramente el ritmo de crecimiento

A mediados de los noventa se inició el *boom* de Internet. En esa época el número de proveedores de acceso privado se disparó, permitiendo a millones de personas acceder a Internet, que a partir de ese momento ya se empezó a conocer como la Red, desbancado a las demás redes de comunicación existentes (CompuServe, FidoNet/BBS, etc.). El punto de inflexión vino marcado por la aparición de implementaciones de TCP/IP gratuitas (incluso de implementaciones que formaban parte del sistema operativo) así como por la popularización y abaratamiento de medios de acceso cada vez más rápidos (módems de mayor velocidad, RDSI, ADSL, cable, satélite). El efecto de todos estos cambios fue de “bola de nieve”: a medida que se conectaban más usuarios, los costes se reducían, aparecían más proveedores e Internet se hacía más atractivo y económico, con lo que se conectaban más usuarios, etc.

En estos momentos disponer de una dirección de correo electrónico, de acceso a la web, etc., ha dejado de ser una novedad para convertirse en algo normal en muchos países del mundo. Por eso las empresas, instituciones, administraciones y demás están migrando rápidamente todos sus servicios, aplicaciones, tiendas, etc., a un entorno web que permita a sus clientes y usuarios acceder a todo ello por Internet. A pesar del ligero descenso experimentado en el ritmo de crecimiento, Internet está destinado a convertirse en una suerte de servicio universal de comunicaciones, permitiendo una comunicación universal.

1.2.2. La WWW

El éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

1.2.2.1. El Protocolo HTTP

El protocolo HTTP (*hypertext transfer protocol*) es el protocolo base de la WWW. Se trata de un protocolo simple, orientado a conexión y sin estado. La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, *transport control protocol*) de modo conectado, un protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente. El protocolo no mantiene estado, es decir, cada transferencia de datos es una conexión independiente de la anterior, sin relación alguna entre ellas, hasta el punto de que para transferir una página web tenemos que enviar el código HTML del texto, así como las imágenes que la componen, pues en la especificación inicial de HTTP, la 1.0, se abrían y usaban tantas conexiones como componentes tenía la página, transfiriéndose por cada conexión un componente (el texto de la página o cada una de las imágenes).

Existe una variante de HTTP llamada HTTPS (S por *secure*) que utiliza el protocolo de seguridad SSL (*secure socket layer*) para cifrar y autenticar el tráfico entre cliente y servidor, siendo ésta muy usada por los servidores web de comercio electrónico, así como por aquellos que contienen información personal o confidencial.

De manera esquemática, el funcionamiento de HTTP es el siguiente: el cliente establece una conexión TCP hacia el servidor, hacia el puerto HTTP (o el indicado en la dirección de conexión), envía un comando HTTP de petición de un recurso (junto con algunas cabeceras informativas) y por la misma conexión el servidor responde con los datos solicitados y con algunas cabeceras informativas.

El protocolo define además cómo codificar el paso de parámetros entre páginas, el tunelizar las conexiones (para sistemas de *firewall*), define la existencia de servidores intermedios de *cache*, etc.

1.2.2.2. El lenguaje HTML

El otro puntal del éxito del WWW ha sido el lenguaje HTML (*hypertext mark-up language*). Se trata de un lenguaje de marcas (se utiliza insertando marcas en el interior del texto) que nos permite representar de forma rica el contenido y también referenciar otros recursos (imágenes, etc.), enlaces a otros documentos (la característica más destacada del WWW), mostrar formularios para posteriormente procesarlos, etc.

El lenguaje HTML actualmente se encuentra en la versión 4.01 y empieza a proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (*extensible hypertext markup language*) que se suele definir como una versión XML validable de HTML, proporcionándonos un XML Schema contra el que validar el documento para comprobar si está bien formado, etc.

1.2.3. Arquitectura Cliente – Servidor

Gracias a este modelo de desarrollo, las aplicaciones se dividieron en una parte que interacciona con el usuario y otra parte destinada al procesamiento de información. [11]

- Es una forma eficiente de utilizar todos los recursos de máquina.
- Seguridad y fiabilidad en LAN.
- Potencia y simplicidad de las computadoras personales.
- Es un modelo para el desarrollo de sistemas de información en red.
- Las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.
- Cliente: proceso que inicia el diálogo o solicita los recursos.
- Servidor: proceso que responde a las solicitudes. [4]

1.2.3.1 Características de la Arquitectura Cliente – Servidor

- Infraestructura de comunicaciones: mecanismos básicos de direccionamiento y transporte.
- El servidor presenta a todos sus clientes una interface única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interface externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico, ni de su Sistema Operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.
- Cliente: proceso que inicia el diálogo o solicita los recursos, permanece sólo lo particular de cada usuario.

- Servidor: proceso que responde a las solicitudes, contiene la parte que debe ser compartida por varios usuarios. [4]

1.3. Aplicaciones Web

Inicialmente la web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse.

El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (*common gateway interface*) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos.

El esquema de funcionamiento de los CGI tenía un punto débil: cada vez que recibíamos una petición, el servidor web lanzaba un proceso que ejecutaba el programa CGI. Como, por otro lado, la mayoría de CGI estaban escritos en algún lenguaje interpretado (Perl, Python, etc.) o en algún lenguaje que requería *run-time environment* (VisualBasic, Java, etc.), esto implicaba una gran carga para la máquina del servidor.

Además, si la web tenía muchos accesos al CGI, esto suponía problemas graves.

Por ello se empiezan a desarrollar alternativas a los CGI para solucionar este grave problema de rendimiento. Las soluciones vienen principalmente por dos vías. Por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, que evitan que éste tenga que instanciar y ejecutar multitud de programas. La otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que nos permita incluir las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de este momento, se vive una explosión del número de arquitecturas y lenguajes de programación que nos permiten desarrollar aplicaciones web. Todas ellas siguen alguna de las dos vías ya mencionadas. De ellas, las más útiles y las que más se utilizan son aquellas que permiten mezclar los dos sistemas, es decir, un lenguaje de programación integrado que permita al servidor interpretar comandos que “incrustemos” en las páginas HTML y un sistema de ejecución de programas más enlazado con el servidor que no presente los problemas de rendimiento de los CGI.

La que quizás sea la más exitosa y potente de estas aproximaciones, es la seguida por Sun Microsystems con su sistema Java, que está integrada por dos componentes; a saber, un lenguaje que permite incrustar código interpretable en las páginas HTML y que el servidor traduce a programas ejecutables, JSP (*Java server pages*) y un mecanismo de programación estrechamente ligado al servidor, con un rendimiento muy superior a los CGI convencionales, llamado Java Servlet.

Otra de las tecnologías que más éxito ha obtenido y una de las que más se utiliza en Internet es el lenguaje de programación interpretado por el servidor PHP. Se trata de un lenguaje que permite incrustar HTML en los programas, con una sintaxis que proviene de C y Perl.

Además, debido a su facilidad de aprendizaje, su sencillez y potencia, se está convirtiendo en una herramienta muy utilizada para algunos desarrollos.

Otros métodos de programación de aplicaciones web también tienen su mercado. Así sucede con `mod_perl` para Apache, RXML para Roxen, etc., pero muchos de ellos están vinculados a un servidor web concreto.

1.4. Conceptos básicos de servidores Web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). Un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

1. Espera peticiones en el puerto TCP asignado (el estándar para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso en la cadena de petición.
4. Envía el recurso por la misma conexión por donde ha recibido la petición.
5. Vuelve al punto 2.

Un servidor Web que siga el esquema anterior cumplirá los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podrá servir ficheros estáticos.

A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc. A continuación detallaremos algunas de las características principales de los servidores Web, que extienden, obviamente el esquema anterior.

1.4.1. Servicio de ficheros estáticos

Todos los servidores Web deben incluir, como mínimo, la capacidad para servir los ficheros estáticos que se encuentren en alguna parte concreta del disco. Un requisito imprescindible es la capacidad de especificar qué parte del disco se servirá. No resulta en absoluto recomendable que el servidor nos obligue a usar un directorio concreto, si bien puede tener uno por defecto.

La mayoría de servidores Web permiten, además, añadir otros directorios para servir, especificando en qué punto del “sistema de ficheros” virtual del servidor se ubicarán.

Algunos servidores Web permiten, además, especificar directivas de seguridad (para qué direcciones, usuarios, etc., está visible un directorio, etc.), mientras que otros hacen posible especificar qué ficheros se considerarán como índice del directorio.

1.4.2. Seguridad y autenticación

La mayoría de los servidores Web modernos nos permiten controlar desde el programa servidor aquellos aspectos relacionados con la seguridad y la autenticación de los usuarios. El modo más simple de control es el proporcionado por el uso de ficheros .htaccess. Éste es un sistema de seguridad que proviene de uno de los primeros servidores web (del NCSA httpd), que consiste en poner un fichero de nombre .htaccess en cualquier directorio del contenido web que se vaya a servir, indicando en este fichero qué usuarios, máquinas, etc., tienen acceso a los ficheros y subdirectorios del directorio donde está el fichero. Como el servidor de NCSA fue el servidor más usado durante mucho tiempo, la mayoría de servidores modernos permiten utilizar el fichero .htaccess respetando la sintaxis del servidor de NCSA.

Otros servidores permiten especificar reglas de servicio de directorios y ficheros en la configuración del servidor Web, indicando allí qué usuarios, máquinas, etc., pueden acceder al recurso indicado.

Por lo que respecta a la autenticación (validación del nombre de usuario y contraseña proporcionados por el cliente), las prestaciones ofrecidas por los diversos servidores web son de lo más variado.

La mayoría permiten, como mínimo, proporcionar al servidor web un fichero con nombres de usuario y contraseñas contra el que se pueda validar lo enviado por el cliente. De todos modos, es frecuente que los servidores proporcionen pasarelas que permitan delegar las tareas de autenticación y validación a otro software (por ejemplo RADIUS, LDAP, etc.). Si usamos un sistema operativo como Linux, que dispone de una infraestructura de autenticación como PAM (*pluggable authentication modules*), podemos usar esta funcionalidad como modo de autenticación del servidor web, permitiéndonos así usar los múltiples métodos disponibles en PAM para autenticar contra diversos sistemas de seguridad.

1.4.3. Contenido Dinámico

Uno de los aspectos más importantes del servidor Web escogido es el nivel de soporte que nos ofrece para servir contenido dinámico.

Dado que la mayor parte del contenido Web que se sirve no proviene de páginas estáticas, sino que se genera dinámicamente, y esta tendencia es claramente alcista, el soporte para contenido dinámico que nos ofrece el servidor Web es uno de los puntos más críticos en su elección. La mayoría de servidores Web ofrecen soporte para CGI (cabe recordar que los CGI son el método más antiguo y simple de generación de contenido dinámico). Muchos ofrecen soporte para algunos lenguajes de programación (básicamente interpretados) como PHP, JSP, ASP, Pike, etc. Es altamente recomendable que el servidor Web que utilicemos proporcione soporte para alguno de estos lenguajes, siendo uno de los más utilizados PHP, sin tener en cuenta JSP, que usualmente requiere un software externo al servidor Web para funcionar (como por ejemplo, un contenedor de Servlets). La oferta en este campo es muy amplia, pero antes de escoger un lenguaje de programación de servidor tenemos que plantearnos si deseamos un lenguaje muy estandarizado para que nuestra aplicación no dependa de un servidor Web o arquitectura concreta o si, por el contrario, la portabilidad no

es una prioridad y sí lo es alguna prestación concreta que pueda ofrecernos algún lenguaje de programación concreto.

1.4.4. Servidores virtuales

Una prestación que está ganando adeptos y usuarios a marchas forzadas, especialmente entre los proveedores de servicios de Internet y los operadores de alojamiento de dominios, es la capacidad de algunos servidores Web de proporcionar múltiples dominios con sólo una dirección IP, discriminando entre los diversos dominios alojados por el nombre de dominio enviado en la cabecera de la petición HTTP. Esta prestación permite administrar de una forma más racional y ahorrativa un bien escaso, como son las direcciones IP.

Si necesitamos disponer de muchos nombres de servidor (ya sea porque proporcionamos alojamiento o por otros motivos) debemos asegurarnos de que el servidor Web escogido ofrezca estas facilidades, y además, que el soporte que facilita para servidores virtuales nos permita una configuración diferente para cada servidor (directorios, usuarios, seguridad, etc.). Lo ideal sería que cada servidor se comportase como si de un ordenador diferente se tratase.

1.4.5. Prestación Extra

Son muchas las prestaciones que nos ofrecen los servidores Web para diferenciarse de la competencia. Algunas de ellas resultan realmente útiles y pueden decidir la elección de servidor Web. Debemos ser conscientes, no obstante, de que si usamos algunas de estas características, o si éstas se convierten en imprescindibles para nosotros, ello nos puede obligar a usar un determinado servidor Web aunque en algún momento determinado deseemos cambiar.

Algunas características adicionales de algunos servidores Web de código libre son las siguientes:

Spelling (Apache). Esta prestación de Apache nos permite definir una página de error para los recursos no encontrados que sugiera al usuario algunos nombres de recurso parecidos al que solicitaba para el caso de que hubiese cometido un error al escribir.

- Status (Apache). Nos proporciona una página Web generada por el servidor donde éste nos muestra su estado de funcionamiento, nivel de respuesta, etc.
- RXML Tags (Roxen). Añade a HTML algunos tags (comandos de HTML), mejorados para programación y generación de contenido dinámico.
- SQL Tags (Roxen). Añade al lenguaje HTML extendido de Roxen (RXML), comandos para realizar accesos a bases de datos SQL desde las propias páginas HTML.
- Graphics (Roxen). Añade al lenguaje HTML extendido de Roxen (RXML), comandos para la generación de gráficos, títulos, etc., y no requiere un trabajo de diseño gráfico.
- Bfnsgd (AOLServer), mod_gd (Apache). Nos permite realizar gráficos a partir de texto y de fuentes True Type.
- mod_mp3 (Apache), ICECAST, MPEG (Roxen). Nos permiten convertir el servidor Web en un servidor de música (con streaming, etc.).

- Throttle (Roxen), mod_throttle (Apache). Nos proporcionan medios para limitar la velocidad de servicio de HTTP, ya sea por usuario, servidor virtual, etc.
- Nsxml (AOLServer), tDOM (AOLServer), mod_xslt (Apache). Nos permite realizar la transformación de ficheros XML a partir de XSL.
- Kill Frame (Roxen). Envía con cada página Web servida un código que evita que nuestra Web quede como frame (enmarcada) dentro de otra página Web.

1.4.6. Actuación como representantes

Algunos servidores Web nos permiten usarlos como servidores intermedios (*proxy servers*). Podemos usar los servidores intermedios para propósitos muy variados:

- Para servir de aceleradores de navegación de nuestros usuarios (uso como *proxy-cache*).
- Para servir como aceleradores de acceso frontales para un servidor Web, instalando diversos servidores Web que repliquen los accesos a un servidor maestro (*reverse-proxy* o *HTTP server acceleration*).
- Como frontales a algún servidor o protocolo.

Algunos servidores Web nos permiten usarlos como servidores intermediarios para alguno de los usos anteriores. No obstante, para los dos primeros usos (aceleradores de navegación o de acceso) existen programas específicos de código libre mucho más eficientes, entre los que destaca Squid (<http://www.squid-cache.org/>), que se considera uno de los mejores productos de *proxy* existentes.

Existen módulos para diversos servidores Web que nos permiten usarlos como frontales para otros servidores especializados en otro tipo de servicio.

1.4.7. Protocolos adicionales

Algunos servidores, además de atender y servir peticiones http (y HTTPS), pueden atender y servir peticiones de otros protocolos o de protocolos implementados sobre HTTP. Algunos de estos protocolos pueden convertirse en requisitos fundamentales de nuestro sistema y, por ello, su existencia en el del servidor Web puede ser imprescindible.

1.5. El Servidor Web Apache

Apache es un servidor Web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la Web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el *Apache Group*. Además del *Apache Group*, cientos de personas han contribuido al proyecto con código, ideas y documentación.

El nombre Apache es "A PAtCHy server", desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. De acuerdo al estudio hecho por Netcraft, Apache es el servidor WWW más popular del momento y ha demostrado ser substancialmente mas rápido que muchos otros servidores libres y compite de cerca con los mejores servidores comerciales. [14]

Características:

- El nombre Apache es "A PAtCHy server".
- Más del 68% de los sitios Web en Internet están usando Apache.
- Es uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.
- Es libre para uso comercial y no comercial.
- Es multiplataforma (Unix, Linux, Vms, Win32, OS2).
- Las transacciones son seguras.
- Se tiene el código fuente completo. [14]

1.6. Otros servidores Web de software libre

Existen multitud de servidores HTTP de código libre, pero casi todos ellos han quedado eclipsados por la fama de Apache. Algunos de estos servidores presentan características que les dotan de gran interés.

1.6.1. AOLServer

El servidor Web AOLServer es el servidor Web de código libre desarrollado por AOL (América Online, el proveedor de Internet más importante del mundo). AOL utiliza AOLServer como servidor principal de Web para uno de los entornos Web de mayor tráfico y uso de Internet. AOLServer es un servidor Web multi-hebra, basado en TCL, y con muchas facilidades de uso en entornos de gran escala y sitios Web dinámicos. Cabe destacar que todos los dominios y servidores Web de AOL, más de doscientos, que soportan miles de usuarios, millones de conexiones, etc., funcionan gracias a AOLServer. AOLServer tiene una amplia base de usuarios, gracias sobre todo a su integración con OpenACS, un sistema de gestión de contenidos muy potente, de código libre, desarrollado

inicialmente por una empresa llamada ArsDigita y posteriormente liberado bajo licencia GPL.

El binomio AOLServer-OpenACS constituye la infraestructura de proyectos web tan complejos y potentes como dotLRN (un campus virtual universitario de código libre).

1.6.2. Roxen y Caudium

Roxen es un servidor Web publicado bajo licencia GNU por un grupo de desarrolladores suecos que posteriormente fundarían la empresa Roxen Internet Services. El servidor Roxen (que primeramente se había llamado Spider y después, Spinner) ha destacado siempre por la gran cantidad de funcionalidades que ofrece a los usuarios. Este servidor, desarrollado en el lenguaje de programación Pike, ofrece a los usuarios cientos de módulos que permiten desarrollar fácilmente sitios Web muy ricos, dinámicos, etc., sin otra herramienta que el servidor Roxen. Las características principales de Roxen son:

- Multiplataforma, es decir, puede ejecutarse en multitud de plataformas:
- Windows, Linux, Solaris, MAC OS/X, etc.
- Es de código libre.
- Interfaz de administración vía web muy rica y fácil de usar.
- Soporte gráfico integrado que permite, con sólo algunas etiquetas de RXML (la extensión de HTML de Roxen), generar imágenes, títulos, gráficas, etc.
- Acceso a BBDD integrado, posibilita el acceso a PostgreSQL, Oracle, MySQL, etc.
- Base de datos MySQL integrada.
- Programación en el servidor con RXML, Java, Perl, PHP y CGI.
- Soporte de criptografía fuerte.
- Arquitectura modular que permite cargar y descargar extensiones del servidor con éste en marcha.
- Independencia de plataforma en los módulos desarrollados por el usuario.

A mediados del 2000, a raíz de la aparición de la versión 2.0 de Roxen, que rompía la compatibilidad de éste con las versiones anteriores, especialmente la 1.3, que era la más usada, un grupo de desarrolladores que incluía a algunos de los fundadores de Roxen, inició un nuevo proyecto que partía de la versión 1.3 de Roxen para desarrollar un servidor web que mantuviese la compatibilidad con ésta. Dicho servidor web se denominó Caudium. En estos momentos las dos plataformas, Roxen y Caudium, gozan de buena salud, de buenas relaciones (los desarrolladores intentan mantener la compatibilidad entre los API de los dos sistemas) y cuentan, además, con una base de usuarios fiel.

Roxen es uno de los pocos casos en los que un excelente producto (ha sido siempre uno de los servidores web más estables, rápidos y con mayor número de prestaciones y facilidades) no ha triunfado, ya que siempre ha sido eclipsado por Apache.

1.6.3. thttpd

thttpd es un servidor HTTP extremadamente pequeño, muy rápido, portable y seguro. Tiene las mismas prestaciones que otros servidores convencionales, como Apache, aunque en casos de carga extrema su rendimiento es mucho más alto.

Su utilidad como servidor web de propósito general es más bien escasa, aunque su uso principal suele ser como servidor rápido de contenido estático, muchas veces como soporte de servidores Apache para servir contenido binario estático, como puedan ser imágenes u otros, dejando para el servidor Apache las páginas dinámicas o más complejas. Usado como auxiliar de Apache para servir contenido estático ha conseguido reducir la carga del servidor principal a una centésima parte.

1.6.4. Jetty

Jetty es un servidor Web escrito totalmente en Java que incluye, además, un contenedor de Servlets. Tiene un tamaño reducido y un rendimiento alto, lo que lo ha convertido en uno de los preferidos para desarrollar productos embebidos que requieran un servidor HTTP. Si bien no suelen encontrarse muchos servidores Jetty funcionando por sí mismos, sí que suelen encontrarse como servidores Web empotrados en productos como:

- Integrados con servidores de aplicaciones, como JBoss y Jonas.
- Integrados en el proyecto JTXA como base para el transporte HTTP.
- Integrados en productos como Tivoli de IBM, MQ de Sonic y SESM de Cisco, como servidor HTTP.
- En la mayoría de CD de demostración en libros sobre Java, Servlets, XML, etc.
- Ejecutándose en múltiples sistemas embebidos y ordenadores de mano.

1.7. *Arquitectura de Software para Web (contenido dinámico)*

1.7.1 CGI

Uno de los primeros mecanismos para generar contenido dinámico para la Web es el API llamado CGI (acrónimo de *common gateway interface*, interfaz de pasarelas común). Éste es un mecanismo muy simple que permite que un servidor web ejecute un programa escrito en cualquier lenguaje de programación (ya sea como respuesta a un formulario HTML, a partir de un enlace, etc.), que le pueda pasar unos parámetros (bien provenientes del usuario, vía formularios, bien parámetros de configuración del servidor, del entorno de ejecución, etc.) y, finalmente, hace posible que el resultado de la ejecución de este programa sea enviado al usuario como una página Web o cualquier otro tipo de contenido (un gráfico, etc.).

Gracias a este sencillo mecanismo, las páginas Web, que hasta el momento de la aparición de CGI tenían unos contenidos estáticos e inmutables, son generadas dinámicamente en

respuesta a peticiones concretas. Se abre así un nuevo mundo a los programadores de aplicaciones Web.

Al contrario de lo que sucede con los Servlets, etc., no existe ningún tipo de limitación al lenguaje de programación que podemos utilizar para escribir un CGI. Podemos usar desde *scripts* escritos en el lenguaje de la *shell* del sistema operativo hasta programas escritos en ensamblador, pasando por todo el abanico de lenguajes de programación existentes: C, C++, Perl, Python, etc. Hasta ahora, el lenguaje más popular para la escritura de CGI es Perl, ya que proporciona utilidades al programador que simplifican sobremanera la tarea de escribir programas CGI.

1.7.2 PHP

PHP, cuyas siglas responden a un acrónimo recursivo (PHP: *hypertext preprocessor*), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. PHP es un lenguaje ideal tanto para aprender a desarrollar aplicaciones Web como para desarrollar aplicaciones Web complejas. PHP añade a todo eso la ventaja de que el intérprete de PHP, los diversos módulos y gran cantidad de librerías desarrolladas para PHP son de código libre, con lo que el programador de PHP dispone de un impresionante arsenal de herramientas libres para desarrollar aplicaciones.

PHP suele ser utilizado conjuntamente con Perl, Apache, MySQL o PostgreSQL en sistemas Linux, formando una combinación barata (todos los componentes son de código libre), potente y versátil. Tal ha sido la expansión de esta combinación que incluso ha merecido conocerse con un nombre propio LAMP (formado por las iniciales de los diversos productos).

Apache, así como algunos otros servidores Web, Roxen entre ellos, puede incorporar PHP como un módulo propio del servidor, lo cual permite que las aplicaciones escritas en PHP resulten mucho más rápidas que las aplicaciones CGI habituales.

1.7.2.1. Cómo Funciona PHP

Si solicitamos a nuestro servidor una página PHP, éste envía dicha página al intérprete de PHP que la ejecuta (de hecho, no se trata más que de un programa) y devuelve el resultado (generalmente HTML) al servidor Web, el cual, a su vez, se lo enviará al cliente.

Imaginemos que tenemos una página PHP con el siguiente contenido:

```
<?php echo "<h1>¡Hola mundo!</h1>";?>
```

Si tenemos este código en un fichero con extensión `.php` el servidor enviará la página al intérprete de PHP, el cual ejecuta la página y obtiene como resultado:

<h1>¡Hola mundo!</h1>

El servidor se lo enviará al navegador cliente que ha solicitado la página. El mensaje aparecerá en la pantalla de este último. Veremos que PHP permite mezclar en la misma página HTML y PHP, lo que facilita notablemente el trabajo con éste, pero por otro lado supone un peligro, ya que complica el trabajo en caso de que diseñadores de web y programadores trabajen conjuntamente en las páginas.

Disponemos, en los sistemas en los que esté instalado PHP, de un fichero de configuración global de PHP llamado `php.ini` que nos permitirá configurar algunos parámetros globales. Conviene revisar dicho fichero, pues aunque los valores por defecto suelen ser correctos, puede interesarnos realizar algunos cambios.

1.7.3. Java Servlets y JSP

1.7.3.1. Introducción a los Java Servlets

Los Servlets de Java son la propuesta de la tecnología Java para el desarrollo de aplicaciones Web. Un Servlet es un programa que se ejecuta en un servidor Web y que construye una página Web que es devuelta al usuario. Esta página, construida dinámicamente, puede contener información procedente de bases de datos, ser una respuesta a los datos introducidos por el usuario, etc.

Los Servlets Java presentan una serie de ventajas sobre los CGI, el método tradicional de desarrollo de aplicaciones Web. Éstos son más portables, más potentes, mucho más eficientes, más fáciles de usar, más escalables, etc.

Eficiencia

Con el modelo tradicional de CGI, cada petición que llega al servidor dispara la ejecución de un nuevo proceso. Si el tiempo de vida del CGI (el tiempo que tarda en ejecutarse) es corto, el tiempo de instanciación (el tiempo de arrancar un proceso) puede superar al de ejecución. Con el modelo de Servlets, la máquina virtual de Java, el entorno donde se ejecutan, se arranca al iniciar el servidor, permaneciendo arrancada durante toda la ejecución del mismo. Para atender cada petición no se arranca un nuevo proceso, sino un *thread*, un proceso ligero de Java, mucho más rápido (de hecho, casi instantáneo).

Además, si tenemos x peticiones simultáneas de un CGI, tendremos x procesos simultáneos en memoria, consumiendo así x veces el espacio de un CGI (que, en caso de ser interpretado, como suele ocurrir, implica el consumo de x veces el intérprete). En el caso de los Servlets, hay determinada cantidad de *threads*, pero sólo una copia de la máquina virtual y sus clases.

El estándar de Servlets también nos ofrece más alternativas que los CGI para optimizaciones: caches de cálculos previos, pools de conexiones de bases de datos, etc.

Facilidad de uso

El estándar de Servlets nos ofrece una magnífica infraestructura de desarrollo de aplicaciones Web, proporcionándonos métodos para análisis automático y decodificación de los datos de los formularios de HTML, acceso a las cabeceras de las peticiones HTTP, manejo de cookies, seguimiento, control y gestión de sesiones, entre otras muchas facilidades.

Potencia

Los Servlets Java permiten hacer muchas cosas que son difíciles o imposibles de realizar con los CGI tradicionales. Los Servlets pueden compartir los datos entre sí, permitiendo compartir datos, conexiones a bases de datos, etc. Asimismo, pueden mantener información de solicitud en solicitud, facilitando tareas como el seguimiento de las sesiones de usuario, etc.

Portabilidad

Los Servlets están escritos en Java y se rigen por un API estándar bien documentado. Como consecuencia de ello, los Servlets pueden ejecutarse en todas las plataformas que nos ofrezcan soporte de Java Servlets, sin tener que recompilar, modificarse, etc., sean estas plataformas Apache, iPlanet, IIS, etc., y además, con independencia de sistema operativo, arquitectura *hardware*, etc.

1.7.4. Otras opciones de contenido dinámico

Además de las tecnologías vistas hasta el momento, disponemos de otros sistemas, tecnologías y lenguajes concebidos para desarrollar contenido Web dinámico.

Uno de los sistemas más utilizados, como alternativa a los vistos es `mod_perl`, un módulo del servidor Apache que permite escribir páginas Web utilizando el lenguaje de programación Perl de una forma similar a como se usa PHP. Este módulo presenta una serie de ventajas evidentes respecto a la escritura de CGI en Perl:

- Mejor uso de memoria. Su comportamiento es muy parecido al de PHP, ya que el módulo de Perl se inicia una sola vez, al arrancar el servidor Web, permaneciendo en memoria a partir de ese momento. Ello evita el problema que suponía el hecho de tener que iniciar Perl para cada CGI.
- Mayor velocidad de respuesta. El hecho de tener el módulo precargado da una mayor agilidad a la respuesta, como lo hacen los programas de Perl ya precompilados (Perl precompila el código a un código intermedio que posteriormente interpreta).
- Permite a los programas un acceso más directo a información del servidor. El módulo proporciona una pasarela más eficiente y rica que la facilitada por las variables de entorno de CGI.
- Permite escribir extensiones del servidor completamente en Perl.

Dos de las grandes ventajas de `mod_perl` son el fuerte incremento de rendimiento y velocidad de los programas, por un lado, y el hecho de que para convertir un programa CGI escrito en Perl en uno para `mod_perl` sólo hacen falta unos retoques mínimos. Esas dos ventajas lo convierten en una opción muy válida para aquellas situaciones en las que ya tenemos un cierto número de programas CGI escritos en Perl. Una de las desventajas de `mod_perl` reside en el hecho de que solo está disponible para servidores Apache, con lo que si en nuestro entorno de trabajo no existe la posibilidad de usar Apache, no representará una alternativa válida.

Muchos servidores Web, inclusive algunos de los vistos, como Roxen, proporcionan mecanismos de programación con una filosofía similar a la de JSP. Concretamente, Roxen, quizás uno de los de mayor riqueza de opciones para desarrollar aplicaciones, nos ofrece la posibilidad de extender nuestras páginas HTML con:

- Código RXML, una extensión de HTML de Roxen que incorpora todos los elementos de un lenguaje de programación: condicionales, bucles, etc., además de una rica librería de funciones que incluyen cosas como: acceso a bases de datos, a directorios LDAP, comunicaciones, gráficos, manejo de cadenas, etc.
- Código escrito en Pike, el lenguaje orientado a objetos con el que está desarrollado Roxen.
- Código PHP, igualando en este caso las facilidades de Apache.
- Código Perl, sin llegar a ofrecer las mismas prestaciones que ofrece `mod_perl`, pero proporcionando un buen nivel de opciones.

Al igual que Roxen, tanto AOLServer como Apache nos permiten desarrollar módulos de extensión del servidor que permitirían tratar nuevas etiquetas de HTML, nuevas peticiones o protocolos de comunicaciones, etc. Algunos de estos sistemas permiten desarrollar extensiones de programación como lenguajes de plantilla, con una filosofía similar a JSP. Es el caso de Mason, DTL, etc.

Otras opciones pasan por usar un servidor “complejo” que incluya en un sólo producto mecanismos de extensión (con un lenguaje propio o uno de uso general), mecanismos de desarrollo de contenido dinámico y de páginas dinámicas. Uno de los más conocidos es Zope, el cual, basado en el lenguaje de programación Python, es un servidor de aplicaciones de código libre para construir portales, aplicaciones Web, gestores de contenido, etc. Proporciona al programador gran cantidad de facilidades de desarrollo con un API rico y potente para proceso de peticiones HTTP, acceso a bases de datos, etc.

Finalmente, cabe destacar la existencia de opciones de más alto nivel, muchas de ellas basadas en alguno de los productos anteriores y diseñadas para el desarrollo de aplicaciones web complejas. Algunas de ellas, como Hendirá (<http://www.enhydra.org>) están basadas en JSP/Servlets (Enhydra es además un excelente contenedor de Servlets). Otras, como OpenACS, son aplicaciones muy orientadas a un tipo de sitios Web concretos.

Uno de los puntos débiles de OpenACS es su dependencia de AOLServer y TCL. Ésta es una característica común de los paquetes de muy alto nivel, que suelen estar muy

vinculados a algún servidor Web concreto. Por otro lado, OpenACS ofrece una gran variedad de módulos y facilidades para crear sitios Web.

Disponemos, además, de una rica variedad de productos del tipo CMS (*Content Management Systems*) para la mayoría de servidores Web de código libre y contenedores de Servlets que proporcionan funcionalidades de modificación, adaptación y programación suficientes para algunos de los proyectos más complejos. En estos casos no se requiere la complejidad asociada a un desarrollo completo de una aplicación Web.

1.8. Bases de Datos

Las bases de datos son usadas con mucha frecuencia y constituyen una parte esencial de casi todas las empresas actuales. A continuación se mencionan algunas de sus aplicaciones más distintivas: banca, líneas aéreas, universidades, transacciones de tarjetas de crédito, telecomunicaciones, finanzas, ventas, producción y recursos humanos entre las más significativas.

En los primeros días muy pocas personas interactuaban directamente con sistemas de bases de datos, aunque implícitamente lo estaban haciendo al llevar registros de información al realizar búsquedas o realizar operaciones. El usuario empezó a interactuar con estos sistemas con la aparición de los cajeros automáticos. La revolución de Internet a finales de la década de 1990 incrementó el acceso directo del usuario a las bases de datos. [2]

1.8.1. Historia de los sistemas de bases de datos

El procesamiento de datos impulsa el crecimiento de los computadores, así como también la mejora para las técnicas del almacenamiento de datos. [2]

- Década de 1950 y principios de la década de 1960. Se desarrollaron las cintas magnéticas para el almacenamiento de datos, de esta forma la información fue automatizada. Las cintas sólo se podían leer secuencialmente y el tamaño de los datos era mayor que la memoria principal.
- Finales de la década de 1960 y principios de la década de 1970. El uso de los discos fijos a finales de la década de 1960 cambió la forma de acceder a los datos ya que se podía acceder de manera directa a los datos lo cual rompía con la secuencialidad, ésta era una gran ventaja ya que se podía acceder a los datos en cuestión de milisegundos. De esta forma se pudieron desarrollar las bases de datos de red y jerárquicas, que permitieron que las estructuras de datos tales como listas y árboles pudieran almacenarse en disco. En 1970 aparece el artículo histórico de Codd el cuál definió el modelo relacional y formas no procedimentales de consultar los datos en el modelo relacional y la posibilidad de ocultar detalles de implementación.
- Década de 1980. Inicialmente no se puso en práctica el modelo relacional ya que en cuestiones de rendimiento no podía competir con el rendimiento de las bases de datos jerárquicas y de red ya existentes. Esto cambió con la aparición del proyecto System R, su prototipo condujo al primer producto de bases de datos relacionales de IBM:

SQL/DS. Las bases de datos relacionales fueron tan sencillas de utilizar que finalmente reemplazaron a las bases de datos jerárquicas y de red incluso en rendimiento.

- En la década de 1980 también se dio una investigación en las bases de datos distribuidas, paralelas y el trabajo inicial en las bases de datos orientadas a objetos.
- Principios de la década de 1990. El lenguaje SQL se diseñó fundamentalmente para las aplicaciones de ayuda a la toma de decisiones y el objetivo principal de las bases de datos fue las aplicaciones de procesamiento de transacciones, ambas re emergieron como una importante área de aplicación para las bases de datos y se comenzaron a ofrecer las bases de datos relacionales orientadas a objetos.
- Finales de la década de 1990. Se da el crecimiento masivo de World Wide Web, con lo cual las bases de datos se implantaron mucho más extensivamente que antes, se podía administrar un mayor número de transacciones, fiabilidad y disponibilidad. [2]

1.8.2. Sistema de Base de datos.

Un sistema de base de datos tiene como objetivo principal almacenar información y hacer que se encuentre disponible al momento en que se requiera por medio de un sistema de cómputo. Este sistema se encuentra integrado por: el equipo físico, los programas, la información y los usuarios. [3]

Base de datos

Una base de datos es una colección de datos ordenados e interrelacionados, los cuáles se almacenan independientemente de los programas de aplicación, empleando métodos bien definidos para dar de alta, modificar o extraer información, dichos datos son de relevante importancia para cada empresa. [3]

1.9. Sistema Manejador de base de datos (DBMS)

Un sistema manejador de bases de datos es el encargado de manejar las solicitudes de acceso enviadas por los usuarios a la base de datos. Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Sus objetivos principales son proporcionar una forma de almacenar y recuperar la información de una base de datos de forma rápida y eficiente así como proporcionar a los usuarios una visión abstracta de los datos. [2]

1.9.1. Niveles de abstracción de datos.

Estos niveles están definidos en la arquitectura ANSI – SPARC.

- **Nivel físico:** Es el nivel más bajo describe cómo se almacenan realmente los datos. Aquí se describen a detalle las estructuras de datos complejas de bajo de nivel.
- **Nivel Lógico:** Es el siguiente nivel que describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos.
- **Nivel de vistas:** El nivel más alto describe sólo parte de la base de datos completa desde el punto de vista de los usuarios. [2] Para el manejo de un sistema de base de datos se tienen tres tipos de usuarios:
- **Administrador de Base de Datos:** Este usuario tiene todos los permisos para la actualización y mantenimiento a la base de datos.
- **Programadores de Aplicaciones:** Este tipo de usuarios tienen permisos para dar de alta, baja y actualizar la base de datos, por medio del lenguaje de consulta estándar de SQL (Lenguaje Estructurado de Consultas).
- **Usuarios Finales de Aplicaciones:** Este tipo de usuarios son lo que usan las aplicaciones desarrolladas por los programadores. [2]

1.10. Modelos de Datos.

Son una colección de herramientas conceptuales para describir los datos, las relaciones de los datos, la semántica de los datos y las restricciones de integridad. Existen diferentes modelos de datos los cuales se clasifican en tres grupos: *modelos lógicos basados en objetos*, *modelos lógicos basados en registros* y *modelos físicos*.

1.10.1. Modelos lógicos basados en objetos.

Se usan para describir datos en los niveles lógico y de vistas. Proporcionan capacidades estructurales muy flexibles y permiten que las restricciones se especifiquen explícitamente. Los más conocidos son: [2]

- Modelo Entidad – Relación.
- Modelo orientado a objetos
- Modelo de datos semántico.
- Modelo de datos funcional.

En este momento únicamente profundizaremos en el Modelo Entidad – Relación ya que se trabajará sobre este.

A. Modelo Entidad – Relación.

El modelo entidad relación (E-R) se basa en una percepción de un mundo real que consiste en una colección de objetos básicos llamados *entidades*, y *relaciones* entre estos. [5]

Entidad: Es un objeto que es distinguible de otros objetos por medio de un conjunto específico de *atributos*.

Atributo: Son las características únicas asociadas a este objeto, esto hace al mismo único y distinguible.

Relación: Es una asociación entre una o varias entidades.

Conjunto de entidades: Entidades del mismo tipo.

Conjunto de relaciones: Relaciones del mismo tipo. El Modelo E-R presenta ciertas restricciones a las que deben ajustarse los contenidos de una Base de Datos. Una restricción importante es la de cardinalidad de asignación, que expresa el número de entidades a las que puede asociarse otra entidad mediante un conjunto de relación. [5]

La estructura lógica global de una BD puede expresarse gráficamente por el *diagrama E-R* que consta de:

- **Rectángulos:** Representan conjunto de entidades.
- **Elipses:** Representan atributos.
- **Líneas:** que conectan atributos a conjuntos de entidades y conjuntos de entidades a relaciones. [5]

Una entidad se representa por:

Un Rectángulo

Su nombre debe ser escrito con mayúsculas, en singular y debe ser único.

El nombre de los atributos se escribe con minúscula.

Como identificar y modelar una entidad:

- Identificar sustantivo.
- Nombre de la entidad.
- Escribe una descripción de esta.
- Identifica los atributos.
- Dibuja el rectángulo para esta. [5]

Definición de las relaciones

- Definir las ligas de la información.
- Asignar un nombre a ésta.
- Una línea entre dos entidades.
- El nombre de la relación en minúscula.
- Opcionalidad (Cardinalidad mínima). [5]
- Grado (Cardinalidad máxima)

Opcionalidad de los atributos.

Los atributos pueden ser de dos tipos:

- **Obligatorios**, los cuales se denotan por “ * ”
- **opcionales**, los cuáles se denotan por “ o ”

Identificador único: Es un atributo cuyo valor no se repite en el conjunto de datos y se denota por “ # ”.

1.10.2. Modelos lógicos basados en Registros.

Los principales Modelos Lógicos basados en Registros trabajados a través del tiempo son: *el Modelo Relacional, el de Red y el Jerárquico*. El Modelo Relacional se dio a conocer a partir de 1970 por E. F. Codd donde los datos están estructurados a nivel lógico por tablas formadas por filas y columnas, aunque a nivel físico sea diferente, se basa en conceptos matemáticos de relación, en concreto se basa en la teoría de conjuntos y la lógica de predicados. [2]

A. Modelo Relacional.

Se usa una colección de tablas para representar tanto los datos como las relaciones entre ellos. Cada tabla contiene varias columnas y cada columna tiene un nombre único. Este modelo abarca tres aspectos: [2]

- La estructura de los datos.
- La integridad de los datos.
- La manipulación de los datos.

Una vez que se ha diseñado el Diagrama Entidad – Relación de la base de datos, se realiza el mapeo de este haciendo uso del Modelo Relacional para obtener el esquema lógico de la base de datos.

1.11. Estructura de Datos

Los términos a los cuales se hace referencia es esta parte del Modelo Relacional son los siguientes: [5]

- **Relación:** Una relación corresponde a una tabla compuesta por una cabecera (columnas: pares atributo - dominio) y un cuerpo (filas: pares atributo - valor).
- **Atributo:** Es el nombre de una columna de una tabla.
- **Tupla:** Una tupla corresponde a una fila o registro de la tabla.
- **Cardinalidad:** Esta corresponde al número de tóplas.
- **Grado:** El grado se refiere al número de atributos a los cuales se hace referencia en la tabla.
- **Llave primaria:** La llave primaria se elige arbitrariamente del conjunto de llaves candidatas de la relación. Es un identificador único para la tabla, puede ser una columna o la combinación de las columnas que componen a la tabla. Para poder elegir una llave primaria entre las llaves candidatas deberá tener las siguientes propiedades:
 - **Unicidad:** No existen dos tuplas con el mismo valor en el atributo que se haya escogido como llave candidata.
 - **Minimalidad:** Si el atributo que se eligió como llave candidata es compuesto, no será posible eliminar ningún elemento de ese atributo sin destruir la propiedad anterior.
- **Llave foránea:** La llave foránea en una relación R2 es un subconjunto FK de atributos de R2 tal que:

a) Existe una relación R1 (R1 y R2 no necesariamente diferentes) con una llave primaria PK.

b) En todo momento cada valor de FK es igual al valor de PK en alguna tupla de R1 o todos los atributos de FK son nulos.

Donde cada atributo componente de FK se basa en el mismo dominio que el atributo correspondiente en PK.

Dominio: Es el valor escalar que representa la menor unidad semántica de la información en el sentido de que son atómicos.

1.12. Reglas de integridad

Estas reglas se utilizan con el fin de informar al DBMS de ciertas restricciones del mundo real, se refieren principalmente a las llaves primarias y llaves foráneas. [4]

- **Regla de integridad de entidad:** Una llave primaria no puede contener valores nulos o más precisamente, ningún atributo componente de una llave primaria puede tener valor nulo. La llave primaria permite identificar estas entidades.
- **Regla de integridad de referencia:** La base de datos no puede contener llaves foráneas no concordantes. Es decir no puede existir una llave foránea cuyo valor (no nulo) no exista coma llave primaria en alguna tupla de relación referenciada.

1.13. La manipulación de los datos.

El Modelo Relacional ocupa un lenguaje de consultas estandarizado llamado SQL, el cual se basa en el álgebra relacional. La manipulación de datos de acuerdo a este modelo se divide en dos partes: [5]

- Un conjunto de operadores que trabajan sobre una o varias operaciones para obtener como resultado una más, todo esto sin cambiar las relaciones originales. Las operaciones fundamentales son: ***la selección, la proyección, el producto cartesiano, la unión y la diferencia.***
- Una operación de asignación del valor de alguna expresión del álgebra a una relación nombrada.

1.14. Normalización

Las reglas de normalización tienen como finalidad eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas. Para llevar a cabo la misma es necesario pasar por cinco niveles de normalización consecutivos. Antes se hará mención del concepto de Dependencia Funcional para tener un mejor entendimiento de este proceso. [5]

La ***Dependencia Funcional*** se define como: “Dados dos atributos X y Y en una relación R, se dice que Y es funcionalmente dependiente de X, si para cada valor de X existe uno y solo un valor de Y asociado con él”, esto quiere decir que si conocemos el valor de X podemos conocer el valor de Y. La Dependencia Funcional se denota por: **$X \rightarrow Y$** . [5]

1.14.1. Primer Nivel de Normalización

- Eliminar los grupos repetitivos de tablas individuales.
- Eliminar atributos no atómicos.
- Identificar cada grupo de datos relacionados con una clave primaria.
- Crear tablas separadas para los grupos de datos que se aplican a varios registros.
- Relacionar estas tablas mediante una clave externa. [4]

1.14.2. Segundo Nivel de Normalización

- Eliminar aquellos campos que no dependan funcionalmente de la clave. [4]

1.14.3. Tercer Nivel de Normalización

- Eliminar dependencias transitivas a través de la llave primaria. [4]

Forma normal de Boyce/Codd (BCNF)

Una relación esta en BCNF si y sólo si todo determinante es una clave candidata.

Un determinante es un atributo del cuál depende funcionalmente otro atributo.

Los principales Niveles de Normalización son los mencionados hasta el momento, además existen otros niveles que se tendrán en cuenta si es que son necesarios:

1.14.4. Cuarto Nivel de Normalización

- En las relaciones muchos a muchos, las entidades independientes no pueden ser almacenadas en la misma tabla. [5]

1.14.5. Quinto Nivel de Normalización

En la mayoría de los casos este nivel no es necesario, este se lleva a cabo para obtener la mejor funcionalidad de una estructura de datos o aplicación, su principio fundamental sugiere: [5]

La tabla original debe ser reconstruida desde las tablas resultantes.

Los beneficios de aplicar estas reglas aseguran que no se cree alguna columna extraña en el esquema de datos (tablas) y que la estructura de estas tenga el tamaño justo.

1.15. El Lenguaje Relacional SQL

SQL es el lenguaje de gestión de base de datos más conocido en la actualidad, el cual es un estándar internacional, comúnmente aceptado por los fabricantes de generadores de bases de datos. [2]

SQL consta de siete partes:

1. Lenguaje de definición de datos (DDL): Este proporciona órdenes para definir esquemas de relación, eliminar relaciones, crear índices y modificar esos esquemas de relación.

2. Lenguaje de manipulación de datos (DML): Es un lenguaje de consulta basado en el álgebra relacional y el cálculo relacional de tuplas, incluyendo las órdenes para insertar, eliminar y actualizar tuplas en la base de datos.

3. DML inmerso: Este tipo de lenguaje está diseñado para ser utilizado dentro de los lenguajes de programación de propósito general, como lo es C.

4. Definición de vistas: El DDL incluye órdenes para definir vistas.

5. Autorización: El DDL incluye órdenes para especificar derechos de acceso a relaciones y vistas.

6. Integridad: El lenguaje original Sequel incluye órdenes para especificar restricciones de integridad complejas. Las versiones actuales de SQL solo proporcionan una forma limitada de comprobación de integridad.

7. Control de transacciones: SQL incluye órdenes para especificar el inicio y final de las transacciones, varias implementaciones permiten el bloqueo implícito de los datos de control de concurrencia.

SQL trabaja con una estructura cliente/servidor, esto quiere decir el cliente realiza las consultas o peticiones y el servidor es el que atiende esas consultas o peticiones dando el servicio correspondiente, y generalmente permite:

- Definir una base de datos.
- Almacenar información en esa base de datos.
- Consultar la información que se encuentra almacenada.
- Actualizar dicha información.
- Combinar y calcular datos necesarios.

1.16. Sistema de Gestión de Base de Datos MySQL.

Actualmente los sitios Web dependen cada vez más de sistemas complejos, los cuáles almacenan datos de relevante importancia.

MySQL es un lenguaje de base de datos popular y es uno de los más poderosos servidores SQL, su estandarización facilita el almacenamiento, actualización y procesamiento de datos., es de código abierto (software libre), lo que lo hace accesible a cualquier usuario.

El uso de este gestor de bases de datos tiene las siguientes ventajas: es rápido, confiable, robusto y fácil de utilizar tanto para volúmenes de datos grandes como pequeños. Su conectividad, velocidad y seguridad hace a MySQL potente para acceder a bases de datos en Internet. [12]

1.16.1. Características generales de MySQL

- Es un servidor de bases de datos SQL (Structured Query Language) es rápido, multiusuario y multihilos.
- Reduce el costo de licenciar la base de datos hasta 90%.
- Se ahorra tiempo y esfuerzo utilizando Software Certificado que ha sido probado con anterioridad.
- Cuenta con un sistema de contraseñas y privilegios muy flexibles y seguros.
- Todas las palabras de paso viajan encriptadas en la red.
- Se cuenta con ODBC para Windows con lo cual se puede utilizar Access para conectarse con el servidor.
- Los objetivos del diseño de MySQL son la velocidad, robustez y fácil manejo.
- Se encuentra escrito en C y C++.
- Maneja grandes volúmenes de información.
- Es software libre.
- Existen varios lenguajes que se comunican con MySQL como: C, C++, Eiffel, Java, Delphi, Perl, PHP, Python y TCL entre otros. [12]

1.17. Relación que existe entre los distintos elementos que integran el marco teórico (Ing. De software, aplicaciones Web, arquitecturas de software y Bases de Datos)

El desarrollo de Aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información. La Ingeniería de la Web es multidisciplinar y aglutina contribuciones de diferentes áreas:

Arquitectura de la información, que es la disciplina encargada de la fundamentación, planificación, análisis y estudio de la selección, organización, disposición, estructuración de espacios de información y presentación de los datos contenidos en los sistemas de información interactivos.

Ingeniería de hipermedia (conjunto de métodos o procedimientos para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.) / **Hipertexto** (paradigma en la interfaz del usuario cuyo fin es el de presentar documentos que puedan, según la definición de Ted Nelson, "bifurcarse o ejecutarse cuando sea solicitado").

Ingeniería de requisitos, diseño de interfaz de usuario, usabilidad, diseño gráfico y de presentación, diseño y análisis de sistemas.

Ingeniería de software: rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

Ingeniería de datos, indexado y recuperación de información, testeo, modelado y simulación, despliegue de aplicaciones, operación de sistemas y gestión de proyectos.

La Ingeniería de la Web no es un clon o subconjunto de la ingeniería de software aunque ambas incluyen desarrollo de software y programación, pues a pesar de que la Ingeniería de la Web utiliza principios de ingeniería de software, incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de las aplicaciones web.[15]

Para la realización del proyecto de tesis, y debido a las consideraciones y características de los elementos explicados anteriormente, en el presente capítulo, se destaca que se desarrollará una aplicación web basada en la arquitectura cliente servidor, usando el servidor web Apache, junto con el manejador de bases de datos MySQL, y el generador de contenido dinámico PHP, y Java Script.

Capítulo 2

Análisis de requerimientos del sistema

2.1. Especificación del problema

El problema consiste en el desarrollo de un sistema Web dinámico, generador de cursos educativos multimedia en línea, para el aprendizaje a distancia, de distintos tópicos impartidos dentro de la Facultad de Ciencias de la Computación de la BUAP, usando tecnología de software abierto (PHP, MySQL).

El sistema dará servicio a sus dos usuarios principales, que serán: profesores y alumnos; por tanto cada uno tiene distintas necesidades, que se describen a continuación.

En el caso del profesor, necesita crear, modificar, eliminar y visualizar contenido didáctico y evaluaciones de él o los cursos que impartirá en un determinado cuatrimestre. A su vez necesita estar en contacto con sus alumnos vía correo electrónico, inscribirlos o darlos de baja en un curso y llevar un control y seguimiento de sus evaluaciones.

Por parte del alumno se requiere que él mismo realice sus evaluaciones y revise el resultado de las mismas, estudie el contenido didáctico del curso y pueda estar en contacto con sus profesores vía correo electrónico.

Cabe mencionar que el material didáctico de los cursos debe contener o auxiliarse de elementos multimedia como imágenes, audio o video y que todas las operaciones en el sistema, tanto de parte del profesor como del alumno deben ser realizadas en línea.

2.2. Identificación de actores y casos de uso

2.2.1. Actores del Sistema

Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.[10]

A continuación se describen necesidades o requerimientos de los usuarios futuros del sistema:

Usuario profesor:

- crear, modificar, eliminar y visualizar contenido didáctico y evaluaciones de los cursos.
- estar en contacto con sus alumnos vía correo electrónico.
- inscribir o dar de baja alumnos en el curso.
- llevar un control y seguimiento de sus evaluaciones.

Usuario alumno:

- realizar evaluaciones en línea.
- revisar el resultado de sus evaluaciones.
- estudiar el contenido didáctico del curso.
- estar en contacto con el profesor vía correo electrónico.

Basados en los incisos anteriores que representan el análisis de la especificación escrita anteriormente, se pueden identificar entonces los siguientes actores:

Actor Profesor (Es una persona certificada para impartir clases en la Universidad en su modalidad a distancia o presencial)

Actor Alumno(Es una persona que esta inscrita en la universidad para tomar clases en la modalidad a distancia o presencial)

2.2.2. Casos de Uso del Sistema

Un Caso de uso es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. [10]

Los siguientes requerimientos deben ser cubiertos por el sistema:

En el caso del profesor, necesita crear, modificar, eliminar y visualizar el contenido didáctico y las evaluaciones de él o los cursos que impartirá en el cuatrimestre. A su vez

necesita estar en contacto con sus alumnos vía correo electrónico, poder inscribir o dar de baja alumnos en el curso y llevar un control y seguimiento de sus evaluaciones.

Por parte del alumno se requiere que el mismo realice sus evaluaciones y revise el resultado de las mismas, revise y estudie del contenido didáctico del curso y pueda estar en contacto con sus profesor vía correo electrónico.

Basándonos en estas necesidades, se identifican los siguientes casos de uso:

Administración de Cursos

Administración de Contenido Didáctico

Administración de evaluaciones

Establecer contacto con alumnos vía correo electrónico

Inscripción a cursos

Baja de alumnos

Seguimiento de evaluaciones

Realizar autoevaluaciones

Revisar resultado de autoevaluaciones

Revisar o estudiar contenido didáctico

Establecer contacto con profesor vía correo electrónico

2.3. Diagramas de caso de uso y especificación de funcionamiento (RUP Rational Rose)

Un diagrama de casos de uso (*Use Case Diagram*) es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama *Main Use Case*, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones). [16]

El Proceso Unificado de Rational (RUP, el original inglés *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

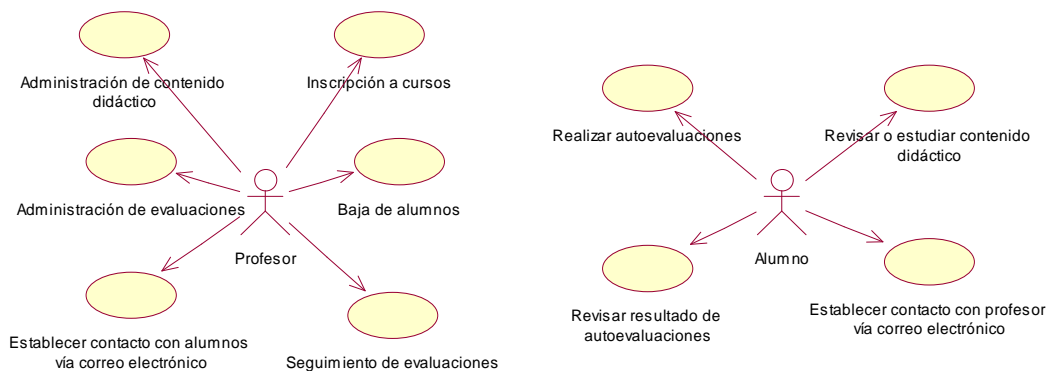


Fig. 2.1. Diagrama de casos de uso (principal) del sistema

2.3.1. Caso de Uso Administración de Cursos

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de crear y eliminar un curso.
- Flujo de Eventos.
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.1.1) para que después el profesor elija de entre las posibles opciones la opción “Cursos” (crear o eliminar curso).

Si la actividad seleccionada es “Crear curso”, el sistema le pedirá que elija el curso (uno de todos los cursos que se ofertan en la licenciatura) y el número de evaluaciones que tendrá dicho curso.

Si la actividad seleccionada es “Eliminar curso”, el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) que quiere eliminar (en el caso de que no haya curso a elegir, se ejecuta el flujo 2.3.1.2), después de elegir el curso, se procede con la eliminación del mismo.
 - Flujos Alternativos
 - 2.3.1.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.1.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Para la ejecución del Subflujo: “Eliminar curso”, es necesario que haya creado o creados por lo menos un curso.
- Post Condiciones. No existen Post-condiciones

2.3.2. Caso de Uso Administración de contenido didáctico

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de crear, modificar, eliminar y visualizar el contenido didáctico de un curso.
- Flujo de Eventos
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.2.1) para que después el profesor elija de entre las posibles opciones la opción “Contenido didáctico” (crear, modificar, eliminar y visualizar el contenido didáctico del curso).

Si la actividad seleccionada es “Crear contenido didáctico”, el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) para el que desee crear contenido didáctico, (en el caso de que no haya curso a elegir, se ejecuta el flujo 2.3.2.2), después de elegir el curso, se procede con la obtención de datos para generar el contenido didáctico como el texto, imágenes, video o audio, que formaran parte de las paginas Web que se le presentaran al alumno, y por ultimo la generación del contenido didáctico.

Si la actividad seleccionada es “modificar contenido didáctico”, en cualquiera de las sub-opciones: “Agregar temas”, “Eliminar temas”, el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) para el que desee modificar el contenido didáctico (en el caso de que no haya curso a elegir, se ejecuta el flujo 2.3.2.2), después de elegir el curso, se procede con la eliminación o agregado de temas al contenido didáctico.

Si la actividad seleccionada es “Eliminar contenido didáctico”, el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) para el que desee eliminar el contenido didáctico (en el caso de que no haya curso a elegir, se ejecuta el flujo 2.3.2.2), después de elegir el curso, se procede con la eliminación del contenido didáctico.

Si la actividad seleccionada es “Visualizar contenido didáctico”, el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) para el que desee visualizar el contenido didáctico (en el caso de que no haya curso, se ejecuta el flujo 2.3.2.2), después de elegir el curso, se elije el tema, y por ultimo se procede con la visualización del contenido didáctico (el tema especifico seleccionado en una nueva pagina web).

- Flujos Alternativos
 - 2.3.2.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.

- 2.3.2.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Para la ejecución de los Subflujos: “modificar contenido didáctico”, “eliminar contenido didáctico” y “visualizar contenido didáctico” es necesario que haya creado o creados por lo menos una contenido didáctico.
Para la ejecución del. Subflujo “crear contenido didáctico”, es necesario que haya por lo menos un curso creado.
Para la ejecución del flujo “Crear contenido didáctico”, es necesario que no haya ningún contenido didáctico creado anteriormente.
- Post Condiciones. No existen Post-condiciones

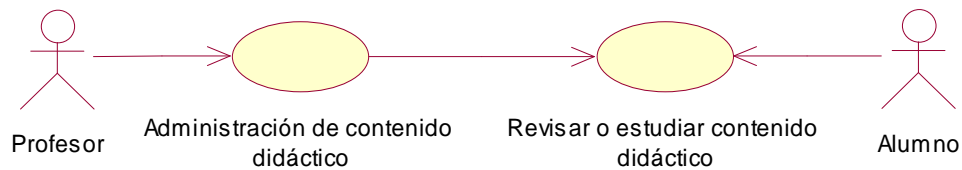


Fig. 2.2. Diagrama de casos de uso Administración de contenido didáctico

2.3.3. Caso de Uso Administración de evaluaciones

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de crear, modificar, eliminar y visualizar evaluaciones de un curso.
- Flujo de Eventos
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.3.1) para que después el profesor elija de entre las posibles opciones la opción “autoevaluaciones” (crear, modificar, eliminar y visualizar evaluación).

Si la actividad seleccionada es “crear evaluación”, el sistema le pedirá que elija el curso y el número de evaluación que desee crear (en el caso de que no haya curso a elegir o evaluación a crear, se ejecuta el flujo 2.3.3.2), después de elegir el curso y el número de evaluación, se procede con la obtención de datos para dicha evaluación; es decir, preguntas con posibles respuestas y la respuesta correcta y por último se genera la evaluación.

Si la actividad seleccionada es “modificar evaluación”, en cualquiera de sus dos sub-opciones: “agregar preguntas” o “eliminar preguntas”, el sistema le pedirá que elija el curso y el número de evaluación a modificar, (el caso de que no haya curso elegir o evaluación a modificar, se ejecuta el flujo 2.3.3.2), después de elegir el curso y el número de evaluación, se procede con la modificación de la evaluación (agregar o eliminar preguntas a la misma).

Si la actividad seleccionada es “eliminar evaluación”, el sistema le pedirá que elija el curso y el número de evaluación que desee eliminar (en el caso de que no haya curso a elegir o evaluación a eliminar, se ejecuta el flujo 2.3.3.2), después de elegir el curso y la evaluación, se procede con la eliminación de la misma.

Si la actividad seleccionada es “visualizar evaluación”, el sistema le pedirá que elija el curso y la evaluación a visualizar (en el caso de que no haya evaluaciones a visualizar, se ejecuta el flujo 2.3.3.2.), y por último se procede con la visualización de la evaluación.

- Flujos Alternativos
 - 2.3.3.1. Contraseña Inválido: Una contraseña inválida ha sido proporcionada. El usuario puede reescribir la contraseña o terminar el caso de uso.
 - 2.3.3.2. No existe curso o evaluación: El usuario recibe un mensaje de aviso y termina el caso de uso

- **Requerimientos Especiales.** No existen requerimientos especiales para este caso de uso.
- **Precondiciones.** Para la ejecución de los Subflujos: “modificar evaluación”, “eliminar evaluación” y “visualizar evaluación” es necesario que haya creada o creadas por lo menos una evaluación.
Para la ejecución del. Subflujo “crear evaluación”, es necesario que haya por lo menos un curso creado.
- **Post Condiciones.** No existen Post-condiciones.

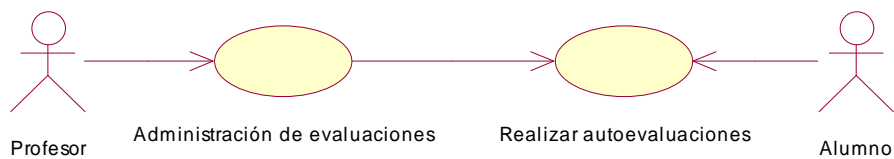


Fig. 2.3. Diagrama de casos de uso Administración de evaluaciones

2.3.4. Caso de Uso Establecer contacto con alumnos vía correo electrónico

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de enviar correos electrónicos a sus alumnos.
- Flujo de Eventos:
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.4.1.) para que después el profesor elija de entre las posibles opciones la opción “Alumnos” (“Enviar correo”), el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) en el cual esta inscrito el alumno al que se va a enviar el correo electrónico (en el caso de que no haya curso, se ejecuta el flujo 2.3.4.3.), A continuación podrá seleccionar de entre una lista de alumnos (si no existen alumnos, se ejecuta el flujo 2.3.4.2.), el o los alumnos a los cuales desea enviar un correo electrónico, después de esto se ejecuta el cliente de correo electrónico definido por el sistema operativo de la computadora del usuario (en este caso algún profesor), con las direcciones de los alumnos puestas en el campo correspondiente, lo restante será escribir texto, adjuntar archivo o cualquier otra opción permitida por la aplicación antes mencionada.
 - Flujos Alternativos
 - 2.3.4.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede reescribir la contraseña o terminar el caso de uso.
 - 2.3.4.2. No existe alumno al cual enviar un correo electrónico: El sistema no avanza en la operación (no hace nada al hacer clic en dicha opción y envía mensaje de notificación al usuario)
 - 2.3.4.3. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Deben existir alumnos inscritos en el curso para poder enviarles correo electrónico.
- Post Condiciones. No existen Post-condiciones

2.3.5. Caso de Uso Inscripción a cursos

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de dar de alta alumnos en un curso.
- Flujo de Eventos:
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválido, se ejecuta el flujo 2.3.5.1.) para que después el profesor elija de entre las posibles opciones, la opción “Alumnos” (“Inscribir”), el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) en el cual se inscribirá al alumno (en el caso de que no haya curso, se ejecuta el flujo 2.3.5.2.), A continuación el profesor introducirá la matrícula del alumno al que desea inscribir, y dará en aceptar para inscribirlo, siempre y cuando sea el alumno correcto.
 - Flujos Alternativos
 - 2.3.5.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.5.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. No existen Pre-condiciones
- Post Condiciones. No existen Post-condiciones

2.3.6. Caso de Uso Baja de alumnos

- Descripción Breve. Caso de uso iniciado por un profesor. Le proporciona la capacidad de dar de baja alumnos en un curso.
- Flujo de Eventos
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.6.1.) para que después el profesor elija de entre las posibles opciones, la opción “Alumnos” (“Dar de baja”), el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) en el cual se dará de baja al alumno (en el caso de que no haya curso, se ejecuta el flujo 2.3.6.3.). A continuación el profesor proporcionará la matrícula del alumno a ser dado de baja, con ella el sistema realizará dicha operación. En caso de que no se encuentre dicha matrícula o sea una matrícula incorrecta, se ejecuta el flujo 2.3.6.2.
 - Flujos Alternativos
 - 2.3.6.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.6.2. No existe dicha matrícula o es un valor invalido. El usuario puede reescribir la matrícula o terminar el caso de uso
 - 2.3.6.3. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. No existen Pre-condiciones
- Post Condiciones. No existen Post-condiciones

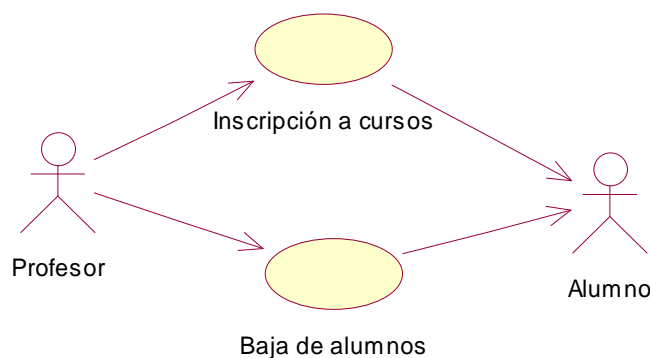


Fig. 2.4. Diagrama de casos de uso Inscripción y Baja de alumnos

2.3.7. Caso de Uso Seguimiento de evaluaciones

- Descripción Breve. Caso de uso iniciado por un profesor. Le permite revisar y modificar el estado de las evaluaciones de sus alumnos, así como negar o dar acceso a la realización de las mismas.
- Flujo de Eventos.
 - Flujo Básico. Este caso de uso inicia cuando el Profesor es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.7.1.) para que después el profesor elija de entre las posibles opciones, la opción “Alumnos”(“Calificaciones”), el sistema le pedirá que elija el curso (uno de todos los cursos que imparte el profesor) en el cual desea revisar el resultado y estado de las evaluaciones de sus alumnos (en el caso de que no haya curso, se ejecuta el flujo 2.3.7.2.), A continuación el profesor podrá revisar, modificar y comprobar el resultado de las evaluaciones en línea, así como otorgar permisos a sus alumnos para poder realizar las evaluaciones correspondientes al curso.
 - Flujos Alternativos
 - 2.3.7.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.7.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Para poder realizar el flujo de “Alumnos”, (“Calificaciones”), debe existir por lo menos un curso creado con sus respectivas evaluaciones y tener alumnos inscritos en el mismo.
- Post Condiciones. No existen Post-condiciones

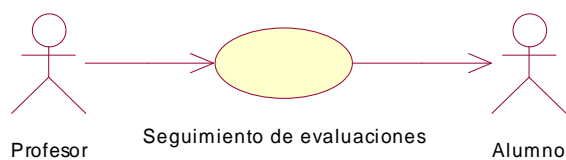


Fig. 2.5. Diagrama de casos de uso Seguimiento de evaluaciones

2.3.8. Caso de Uso Realizar autoevaluaciones

- Descripción Breve. Caso de uso iniciado por un alumno. Le permite realizar su evaluación, contestando los reactivos del cuestionario hecho por su profesor.
- Flujo de Eventos
 - Flujo Básico. Este caso de uso inicia cuando el Alumno es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválido, se ejecuta el flujo 2.3.8.1.) para que después el alumno elija de entre las posibles opciones, la opción “Realizar Autoevaluación”, el sistema le pedirá que elija el curso (uno de todos los cursos que podría estar cursando el alumno) y el número de evaluación que desee realizar (en el caso de que no haya curso o la evaluación no esté disponible para el alumno, se ejecuta el flujo 2.3.8.2.), A continuación el alumno podrá realizar su evaluación, y esta información será almacenada y procesada por el sistema.
 - Flujos Alternativos
 - 2.3.8.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.8.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Para poder realizar este flujo, el alumno necesita estar inscrito en por lo menos un curso que contenga evaluaciones activas.
- Post Condiciones. Una vez que el alumno decidió realizar una evaluación disponible, esta queda bloqueada (si es que desea realizarla nuevamente, el profesor debe darle permiso) y su calificación es guardada en el sistema.

2.3.9. Caso de Uso Revisar resultado de autoevaluaciones

- Descripción Breve. Caso de uso iniciado por un alumno. Le permite revisar el resultado de sus evaluaciones.
- Flujo de Eventos.
 - Flujo Básico. Este caso de uso inicia cuando el Alumno es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.9.1) para que después el alumno elija de entre las posibles opciones, la opción “Calificaciones”, el sistema le pedirá que elija el curso (uno de todos los cursos que podría estar cursando el alumno) en el cual desea revisar el resultado de sus autoevaluaciones (en el caso de que no haya curso, se ejecuta el flujo 2.3.9.2.), A continuación el alumno podrá revisar y comprobar el estado de sus evaluaciones..
 - Flujos Alternativos
 - 2.3.9.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.9.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso.
- Precondiciones. Para poder realizar este flujo, el alumno necesita estar inscrito en por lo menos un curso que contenga evaluaciones activas o inactivas.
- Post Condiciones. No existen Post-condiciones

2.3.10. Caso de Uso Revisar o estudiar contenido didáctico

- Descripción Breve. Caso de uso iniciado por un alumno. Le permite revisar y navegar por el contenido didáctico del curso, esto con el objetivo de estudiar para sus autoevaluaciones.
- Flujo de Eventos.
 - Flujo Básico. Este caso de uso inicia cuando el Alumno es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.10.1.) para que después el alumno elija de entre las posibles opciones, la opción “Contenido Didáctico”, el sistema le pedirá que elija el curso (uno de todos los cursos que podría estar cursando el alumno) en el cual desea revisar el contenido didáctico del mismo (en el caso de que no haya curso o que el curso no tenga material didáctico, se ejecuta el flujo 2.3.10.2.), A continuación el alumno podrá revisar y navegar por el contenido didáctico del curso.
 - Flujos Alternativos
 - 2.3.10.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.10.2. No existe curso o material didáctico: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. Para poder realizar este flujo, el alumno necesita estar inscrito en por lo menos un curso que contenga material didáctico.
- Post Condiciones. No existen Post-condiciones

2.3.11. Caso de Uso Establecer contacto con profesor vía correo electrónico

- Descripción Breve. Caso de uso iniciado por un alumno. Le proporciona la capacidad de enviar correos electrónicos a su profesor.
- Flujo de Eventos.
 - Flujo Básico. Este caso de uso inicia cuando el Alumno es autenticado por el sistema generador de cursos educativos y proporciona su contraseña. El sistema verifica que la contraseña sea válida (si la contraseña es inválida, se ejecuta el flujo 2.3.11.1.) para que después el alumno elija de entre las posibles opciones la opción “Contacto Profesores”, el sistema le pedirá que elija el curso (uno de todos los cursos que toma el alumno) que imparte el profesor al que se va a enviar el correo electrónico (en el caso de que no haya curso, se ejecuta el flujo 2.3.11.2.), después de esto se ejecuta el cliente de correo electrónico definido por el sistema operativo de la computadora del usuario, con la dirección del profesor puesta en el campo correspondiente, lo restante será escribir texto, adjuntar archivo o cualquier otra opción permitida por la aplicación antes mencionada
 - Flujos Alternativos
 - 2.3.11.1. Contraseña Inválida: Una contraseña inválida ha sido proporcionada. El usuario puede volver a escribir la contraseña o terminar el caso de uso.
 - 2.3.11.2. No existe curso: El usuario recibe un mensaje de aviso y termina el caso de uso
- Requerimientos Especiales. No existen requerimientos especiales para este caso de uso
- Precondiciones. El alumno debe estar inscrito en el curso para poder enviar correo electrónico a su profesor.
- Post Condiciones. No existen Post-condiciones.

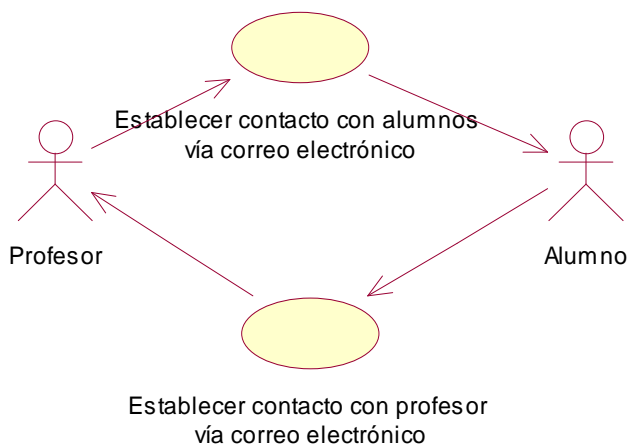


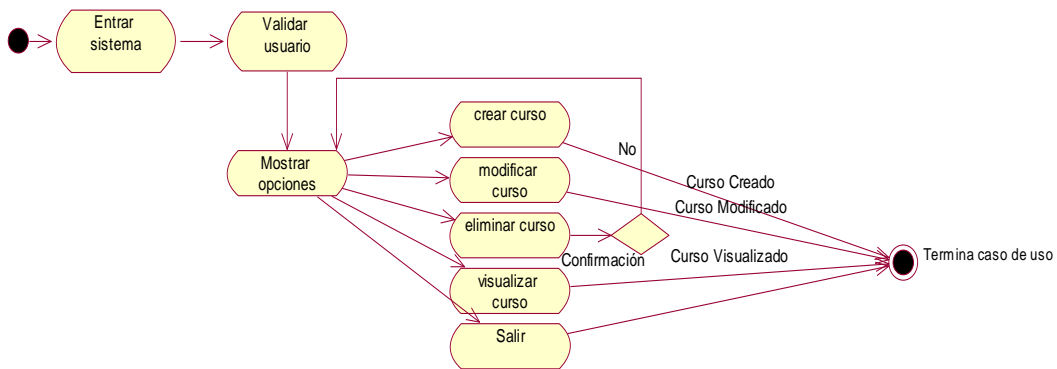
Fig. 2.6. Diagrama de casos Envió de correo electrónico.

2.4. Diagramas de Actividades

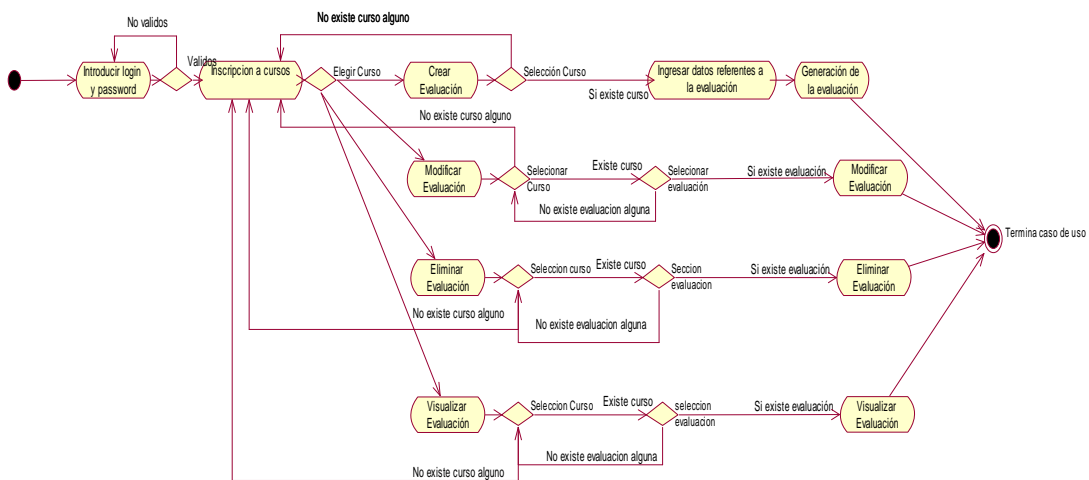
Los Diagramas de Actividades son diagramas de flujo del proceso multi-propósito que se usa para modelar el comportamiento del sistema. Los diagramas de actividades se usan para modelar Casos de Uso, o una clase, o un método complicado.

Un diagrama de actividad es parecido a un diagrama de flujo; la diferencia clave es que los diagramas de actividad pueden mostrar procesado paralelo (parallel processing). Esto es importante cuando se usan diagramas de actividad para modelar procesos 'business' algunos de los cuales pueden actuar en paralelo, y para modelar varios hilos en los programas concurrentes

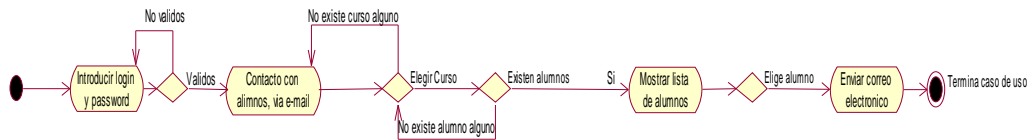
Administración de contenido didáctico



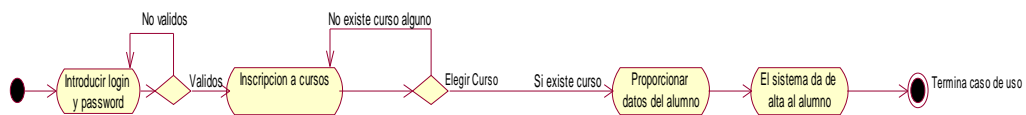
Administración de evaluaciones



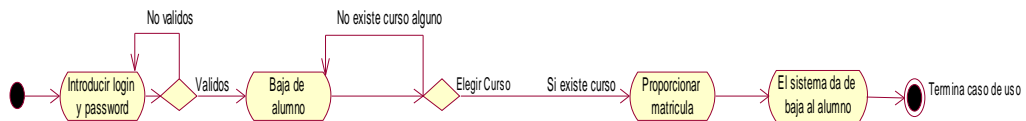
Establecer contacto con alumnos vía correo electrónico



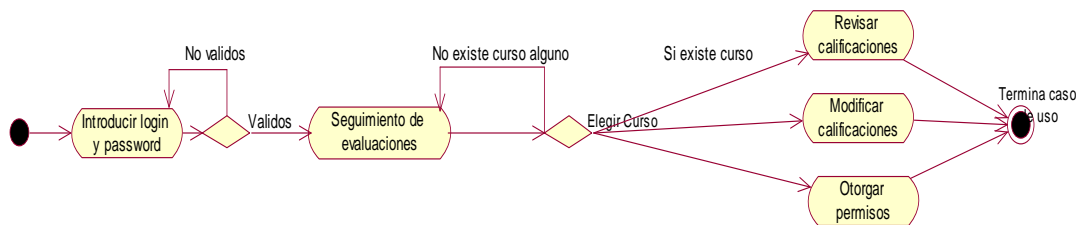
Inscripción a cursos



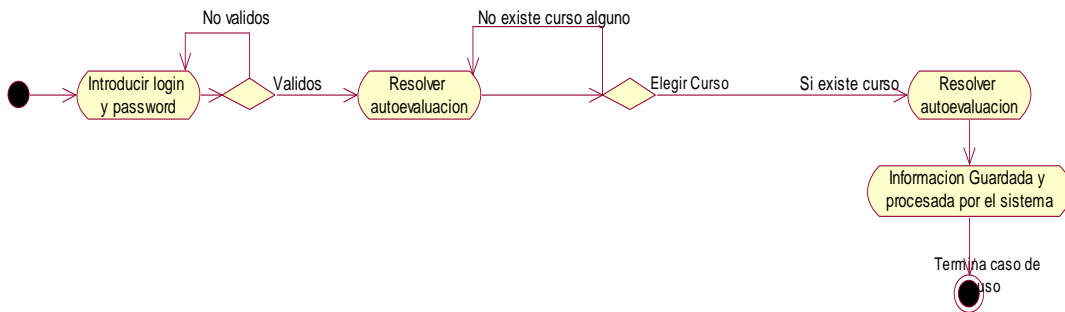
Baja de alumnos



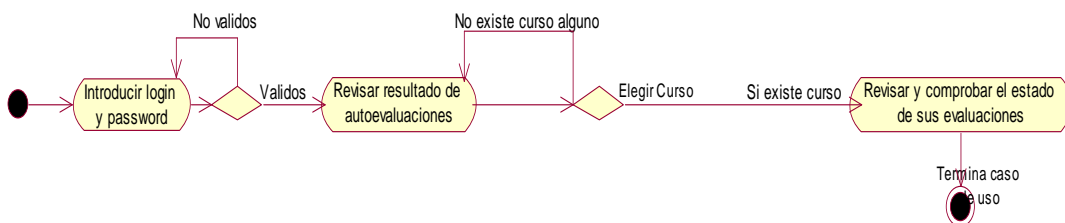
Seguimiento de evaluaciones



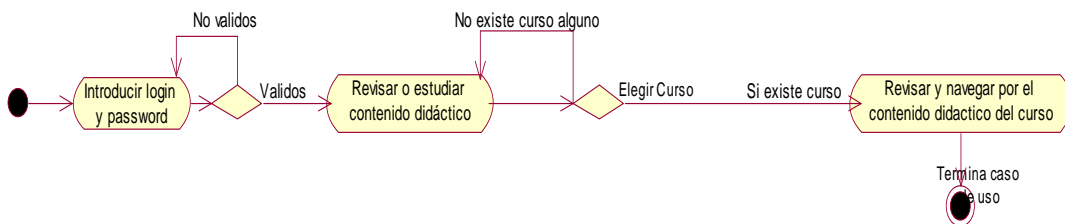
Realizar autoevaluaciones



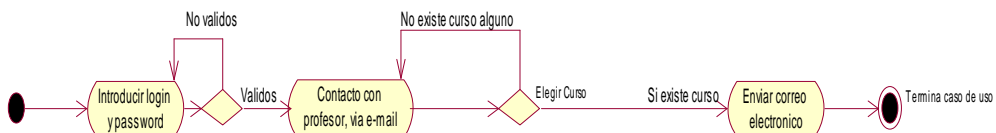
Revisar resultado de autoevaluaciones



Revisar o estudiar contenido didáctico



Establecer contacto con profesor vía correo electrónico



Capítulo 3

Modelo lógico conceptual

En este capítulo se presenta el modelo lógico conceptual del SGCAD. Este modelo describe una primera abstracción de los elementos de la arquitectura de la aplicación Web mediante un diagrama de clases simple de UML. Es en este modelo, donde se exponen asociaciones entre elementos de la arquitectura que se propone, haciendo énfasis en el flujo lógico entre elementos, en su semántica y evitando hacer referencia en la medida de lo posible de cuestiones relacionadas con la navegación y de los aspectos de interacción.

3.1. Modelo lógico conceptual: Diagrama de clases

En las figuras 3.1, 3.2, 3.3 se muestran diagramas de clases simples de UML en donde se presentan los principales elementos del SGCAD y la asociación entre ellos. Esto es, un modelo lógico-conceptual, y como tal, su propósito es el de representar de manera general la estructura del SGCAD. A menudo un modelo lógico-conceptual también, aparte de establecer un marco de referencia que guíe el diseño del modelo navegacional de una aplicación Web, por su semántica, es útil para guiar el diseño de la base de datos de la aplicación. Para el SGCAD se presenta un único diagrama de clases en el cual se muestran los elementos asociados a cada subsistema.

Como se comentó anteriormente, en las figuras 3.1, 3.2, 3.3 se presentan diagramas de clases que muestra una visión general del SGCAD. En ellos se observan clases, multiplicidad y asociaciones de agregación y composición. La semántica de los diagramas presenta estas clases como elementos involucrados en las actividades típicas que el usuario realizará con la aplicación Web, y las asociaciones entre ellos como relaciones de cooperación que, en algún momento, ha de llevarse a cabo para realizar una tarea. Para describir generalmente al sistema se usan tres diagramas de clases: “Artefactos del sistema”, “interfaces”, “infopersonas”, dentro de artefactos del sistema se muestra la forma en que los dos actores del sistema interactúan con el mismo, para administrar o manejar cursos y evaluaciones, por medio de las funciones y capacidades que cada uno de estos actores tiene disponibles dentro del sistema. Dentro del diagrama de interfaces, se presentan también como diagrama de clases las opciones que tienen los alumnos y los profesores dentro del sistema. En el diagrama de infopersonas, se muestra la manera como interactúan los elementos básicos del sistema como son los cursos y las evaluaciones, así como los actores del mismo, los alumnos y los profesores.

Para la realización de los siguientes diagramas de clases, se hace uso de estereotipos, para proporcionar la capacidad de crear un nuevo género para el modelado de una clase. Los estereotipos usados son: entity(entidad), boundary(alcance) y control.

El estereotipo para una clase se coloca dentro del nombre de la clase incluido entre: (<< >>). A continuación se da una explicación del significado de los estereotipos usados.

Entity Classes

Una clase de entidad modela la información y el comportamiento asociado que es generalmente duradero. Este tipo de clase puede reflejar una entidad del mundo real o puede ser necesaria para desempeñar una tarea interna del sistema. Son típicamente independientes de su entorno; es decir, no son sensibles a cómo el entorno se comunica con el sistema. Muchas veces, son independientes del uso, esto significa que pueden ser utilizadas en más de una aplicación.

Boundary Classes

Una clase de alcance maneja la comunicación entre el entorno del sistema y el interior del mismo. Puede proporcionar la interfaz a un usuario o a otros sistemas; es decir es la interface para un actor. En este punto se están documentando los requerimientos de la interfaz del usuario, mas no se esta implementando la misma.

Control Classes

Las clases de control coordinan los acontecimientos necesarios para realizar el comportamiento especificado en el caso de uso. Se piensa en una clase de control como “funcionando” o “ejecutando” el caso de uso; representan la dinámica del caso del uso. Las clases de control son típicamente clases uso-dependientes.

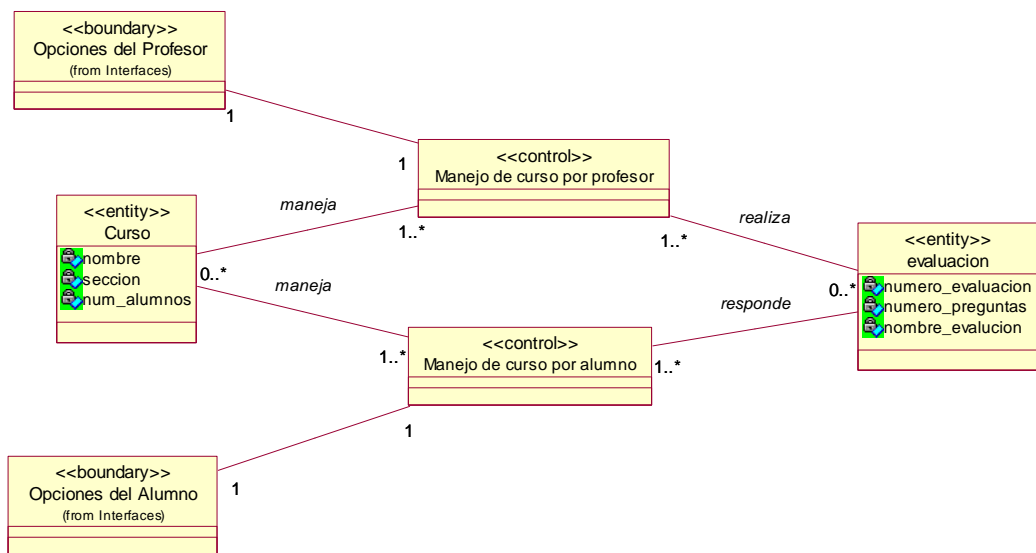


Figura 3.1: Diagrama de clases artefactos del sistema

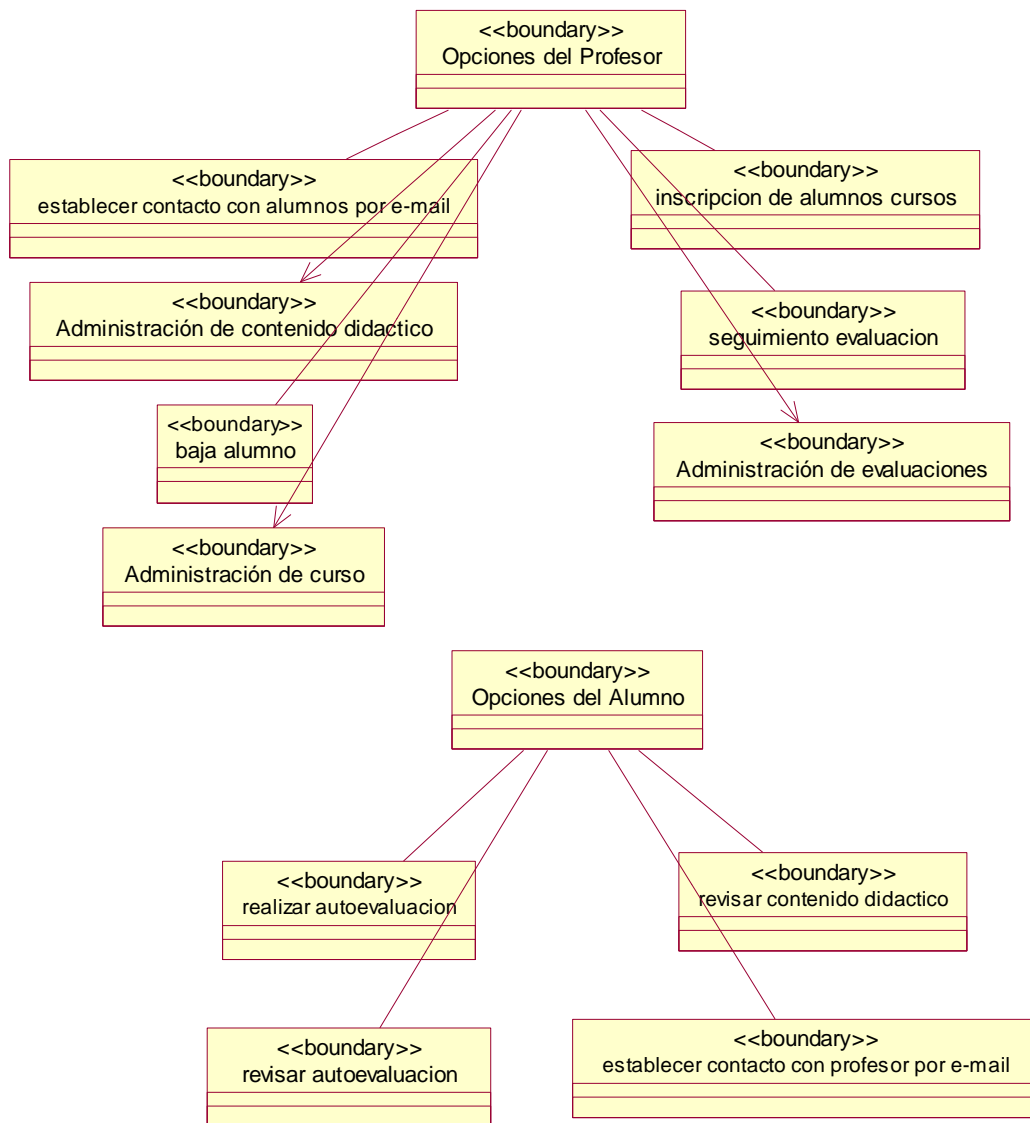


Figura 3.2: Diagrama de clases interfaces

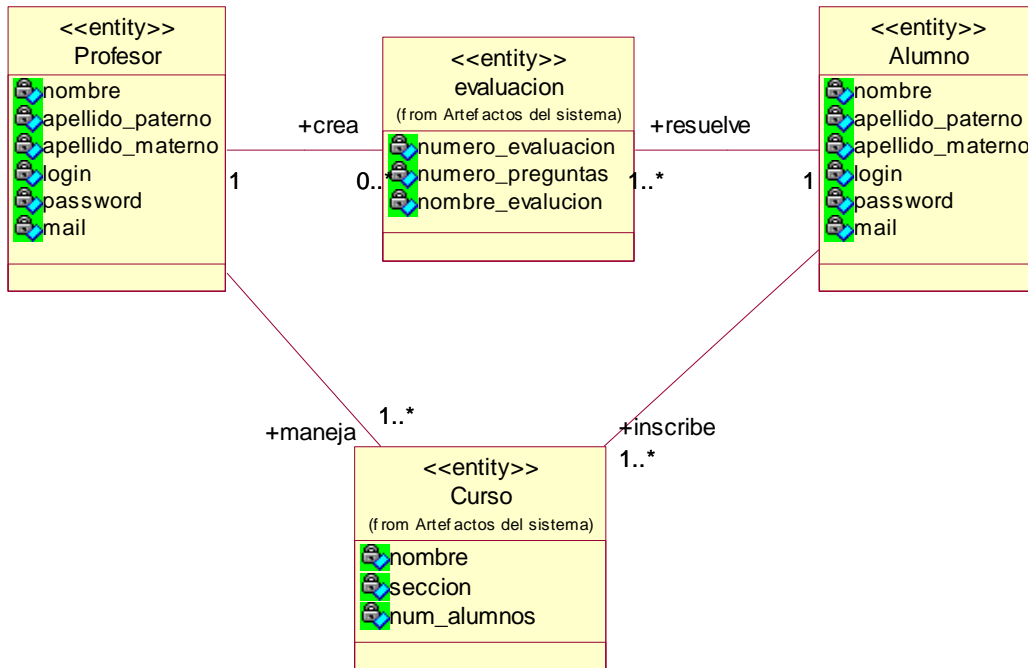


Figura 3.3: Diagrama de clases infopersonas

En el siguiente capítulo se presenta el modelo de navegación para los diferentes subsistemas, tomando como referencia este modelo lógico-conceptual.

Capítulo 4

Modelo de Navegación

En este capítulo se presenta la parte más elemental del análisis y diseño del SGCAD: el diseño del modelo de navegación. En un primer apartado se describe la semántica de los elementos de modelado que se utilizan en los diagramas de clases estereotipadas. Posteriormente se listan otras consideraciones con relación a convenciones y notaciones que se adoptan en el modelo de navegación. Finalmente, se presenta el diagrama de clases estereotipado para cada módulo del sistema: Administración de cursos, Administración de contenido didáctico, Administración de evaluaciones, Establecer contacto con los alumnos vía correo electrónico, Inscripción a cursos, Baja de alumnos Seguimiento de evaluaciones, Realizar autoevaluaciones, Revisar resultado de autoevaluaciones, Revisar o estudiar contenido didáctico, Establecer contacto con los profesores vía correo electrónico.

4.1 Estereotipos utilizados en el modelo de navegación

Con base en la extensibilidad de UML utilizando estereotipos, a continuación se listan y describen estereotipos aplicados a clases y a relaciones utilizados en los diagramas.

De la colección de estereotipos que se presentan a continuación, la mayoría de ellos están definidos por UWE, más sin embargo se han retomado estereotipos de otros trabajos relacionados con técnicas de modelado de aplicaciones Web en UML, esto con la finalidad de presentar una opción más cercana a las necesidades de los analistas y diseñadores de Aplicaciones Web.

4.1.1 Navigation Class

El estereotipo <<Navigation Class>> es aplicable a una clase simple en un diagrama de clases estándar de UML. Este estereotipo representa una clase conceptual cuyas instancias son visitadas o alcanzadas en una actividad o proceso propiciada por el usuario durante la navegación por la aplicación Web. Una <<navigation class>> en la convención del dominio de la aplicación, es decir la Web, puede ser asociada con una página Web la cual combina contenido estático y dinámico que se presenta al usuario o que sirven como pasarelas de procesamiento de código de un proceso requerido en la navegación por la aplicación Web [8]. En un modelo de navegación es el elemento más usual y más importante. Su representación gráfica se muestra a continuación.

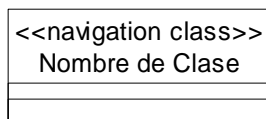


Figura 4.1: Estereotipo Navigation Class

4.1.2 Form

El estereotipo <<Form>> es aplicable a una clase simple en un diagrama de clases estándar de UML. La semántica de este estereotipo tiene que ver con un contenedor de controles (cuadros de texto, botones, cuadros de listas, entre otros) para que un usuario pueda introducir datos a la aplicación, por decirlo de otra manera, este estereotipo representa a los formularios que aparecen en las páginas Web. Este estereotipo aparece siempre como un compuesto o agregado a una <<navigation class>> ya que un formulario aparece siempre embebido en una página Web, y no puede existir de manera independiente [8], [6] y [9]. Su representación gráfica se muestra a continuación.

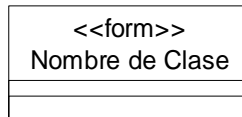


Figura 4.2: Estereotipo Form

4.1.3 Link

El estereotipo <<Link>> es aplicable a una asociación simple unidireccional en un diagrama de clases estándar de UML. Una vez definidas las <<navigation class>> se necesita una relación que haga posible la navegación entre ellas. El estereotipo <<Link>> representa a esa relación [6]. Este estereotipo tiene una clara asociación semántica con los hipervínculos utilizados para ligar páginas Web y por lo tanto su utilización debe acotarse al mismo contexto, condiciones y limitaciones de los hipervínculos en una página web.

4.1.4 Submit

El estereotipo <<Submit>> es aplicable a una asociación simple unidireccional en un diagrama de clases estándar de UML. Este estereotipo representa una relación direccional entre un <<form>> y una <<navigation class>>. Similar a una relación <<Link>>, sin embargo, cuando se define una relación <<submit>>, los atributos de <<form>> son enviados hacia una “navigation class” donde serán recuperados para la operar con ellos [6].

4.2 Semántica de estereotipos asociada con la Web

4.2.1 Acerca del estereotipo <<navigation class>>

Este estereotipo se asocia, en la terminología Web, con una página Web. En la práctica, una página Web se constituye por un archivo hospedado en un servidor, el cual es interpretado por un navegador Web cuando una computadora cliente lo solicita a dicho servidor a través del Internet. Debido a este mecanismo, un archivo de Página Web es un archivo de código, por lo mismo puede tener dos vistas distintas, la vista de contenido real en la que se puede ver el código de la página y la vista que se genera cuando el código que contiene el archivo es interpretado por los navegadores.

Por otra parte, el código de cada página puede generar dos tipos de contenido, el estático y el dinámico. Para cada tipo de contenido existe un gran número de lenguajes para que su generación. El lenguaje universal para el contenido estático es el HTML (Hypertext Meta Language), mientras que para el contenido dinámico los más utilizados son los lenguajes como PHP (Hypertext Pre-Processor), ASP (Active Server Pages) y JSP (Java Server Pages), entre muchos otros. En la figura 4.3 se muestra una <<navigation class>>. En este trabajo se utilizaron ambos tipos de lenguajes.

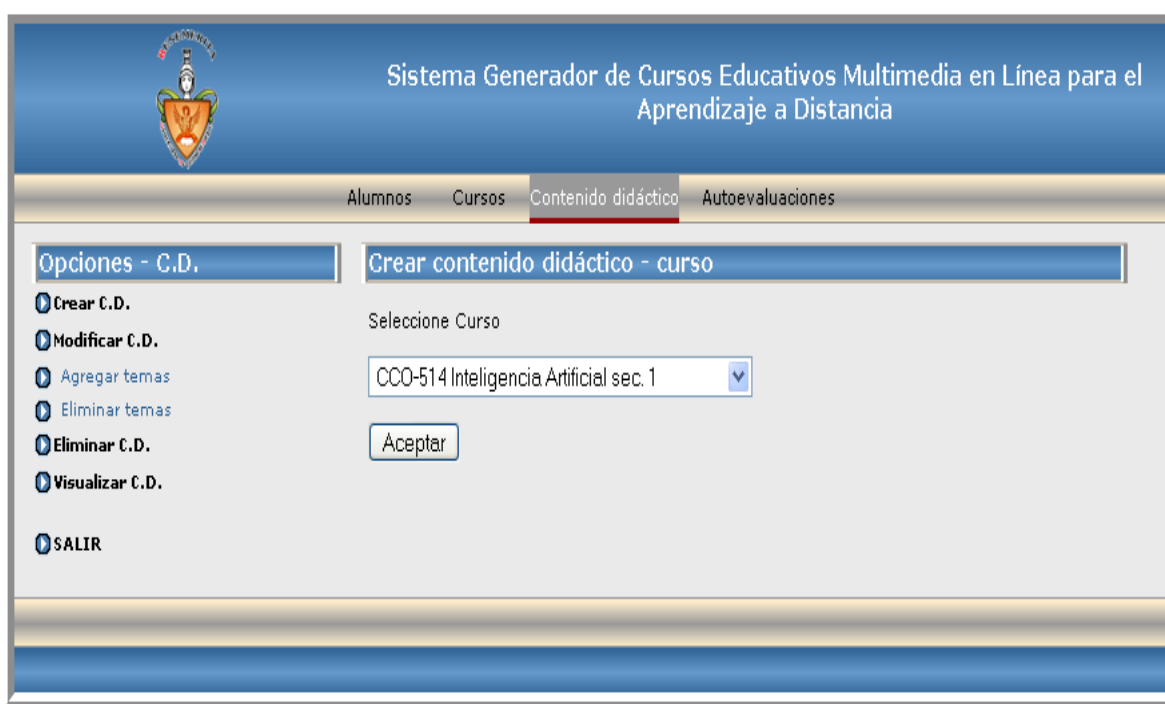


Figura 4.3: Semántica en la implementación de <<navigation class>>

4.2.2 Acerca del estereotipo <<form>>

Este estereotipo es un elemento de modelado que representa a un contenedor de controles para que un usuario pueda introducir datos a la aplicación Web. En la figura 4.4 se muestra un ejemplo de la implementación del estereotipo <<form>> y de los atributos asociados a éste. En el ejemplo se observa la relación entre el estereotipo <<form>> y el estereotipo <<navigation class>>, se observa que un <<form>> se encuentra embebido dentro de una <<navigation class>> y esta es la única forma en que puede ser utilizado. El <<form>> al final de cuentas representa a un formulario en una página Web. Dentro del formulario se encuentran contenidos diversos controles que en un <<form>> del modelado son representados como atributos de tipo especial como lo son input, inputpass, radio, list, hidden y button. En el ejemplo aparecen algunos controles y el tipo de atributo con el cual son representados en una clase estereotipada con <<form>>.

Sistema Generador de Cursos Educativos Multimedia en Línea para el Aprendizaje a Distancia

Alumnos Cursos **Contenido didáctico** Autoevaluaciones

Opciones - C.D. Administrar contenido didáctico - curso

- Crear C.D.
- Modificar C.D.
- Agregar temas
- Eliminar temas
- Eliminar C.D.
- Visualizar C.D.
- SALIR

Introduce en nombre del tema y sus archivos de contenido.

Título del tema

Tema(#)

Archivo de Texto: Examinar...

Archivo de Imagen: Examinar...

Archivo de Audio: Examinar...

Archivo de Video: Examinar...

Enviar

[Cancelar](#)

Figura 4.4: Semántica en la implementación de <<Form>>

4.3 Modelos de Navegación de los submodulos

4.3.1 Modulo administración de cursos

En la figura 4.5 se muestra el diagrama de clases estereotipado correspondiente al modelo de navegación del modulo Administración de cursos. Este modulo comienza con una <<navigation class>> en la cual se presentan dos opciones: “crear curso” y “eliminar curso”, es desde esta interfaz que el usuario profesor tiene la capacidad de crear o eliminar cursos, que serán agregados a la base de datos, dentro de la opción *crear curso* se presenta el estereotipo <<form>> , dentro de una <<navigation class>> en la cual se elige el nombre del curso que se va a crear y el número de evaluaciones que tendrá, la opción *eliminar curso* también dentro de una <<navigation class>> le da la opción de elegir cual de todos los cursos (cursos creados por el profesor que usa el sistema), desea eliminar, junto con ello elimina todo lo referente a dicho curso como contenido didáctico, evaluaciones, alumnos, etc.

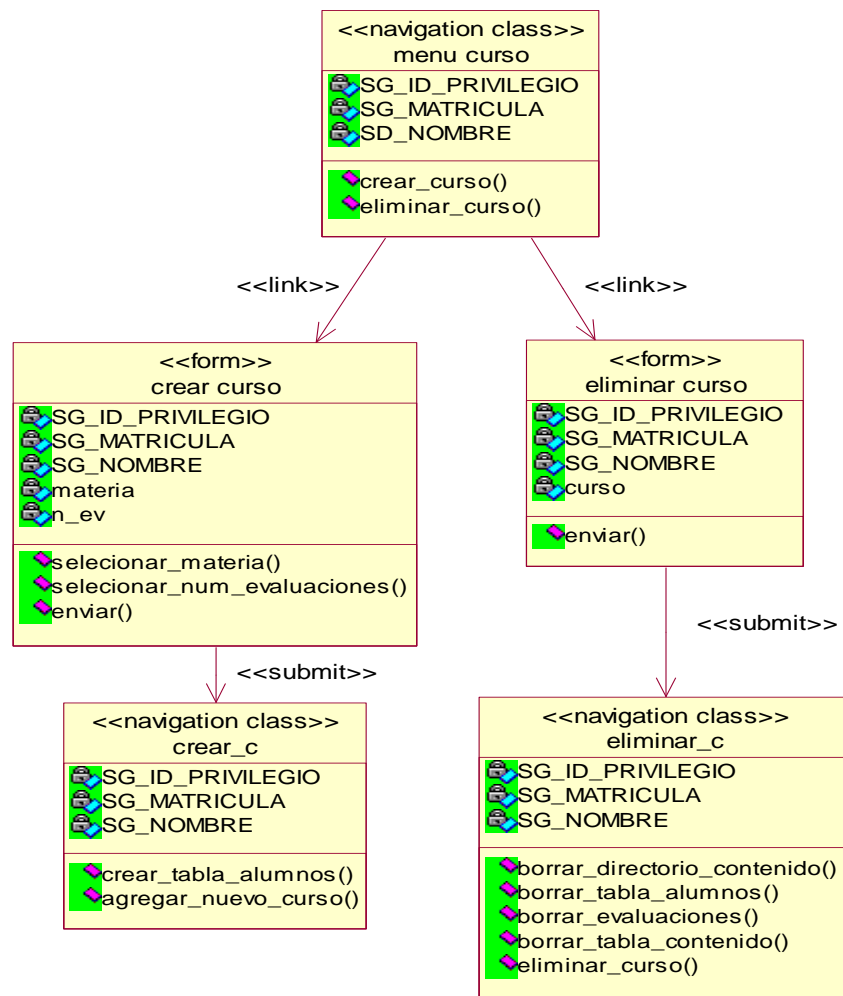


Figura 4.5: Modelo de navegación Administración de cursos.

4.3.2 Modulo administración de contenido didáctico

El modelo de navegación del modulo Administración de contenido didáctico se muestra en la figura 4.6. El modulo consta de la <<navigation class>> *Menú contenido didáctico* en la que se definen los métodos y operaciones necesarias para el procesamiento del contenido didáctico, como son crear, modificar eliminar o visualizar el mismo, todas ellas se presentan dentro de un <<form>>, en el cual siempre se solicita elegir el curso sobre el cual se va a procesar la opción seleccionada (crear, eliminar, modificar o visualizar contenido didáctico), después de este paso, cada uno de los procesos, varia de la siguiente forma, según sea el caso.

- En el caso *crear contenido didáctico*, el <<form >>, requiere de la entrada de los archivos que conformaran la estructura del contenido didáctico (pagina web), como son audio, video, texto, imagen, titulo del tema y tema en numero.
- En el caso *modificar contenido didáctico*, se presenta una <<navigation class>>, llamada menú modificar contenido didáctico, con las opciones: *agregar tema* y *eliminar tema*, para lo cual se procede de la siguiente manera:
 - Para agregar tema, se piden los mismos datos que en el caso de crear contenido (audio, video, texto, imagen, titulo del tema y tema en numero), para agregarlos al contenido del curso que se esta procesando con esta opción.
 - Para eliminar tema, se elige de una lista desplegable el tema que se desea eliminar.
- En el caso *eliminar contenido didáctico*, ya estando seleccionado el curso, se procede a eliminar el contenido didáctico.
- En el caso *visualizar contenido didáctico*, estando elegido el curso, se presenta una <<navigation class>>, con links hacia cada uno de los temas que tenga el curso elegido.

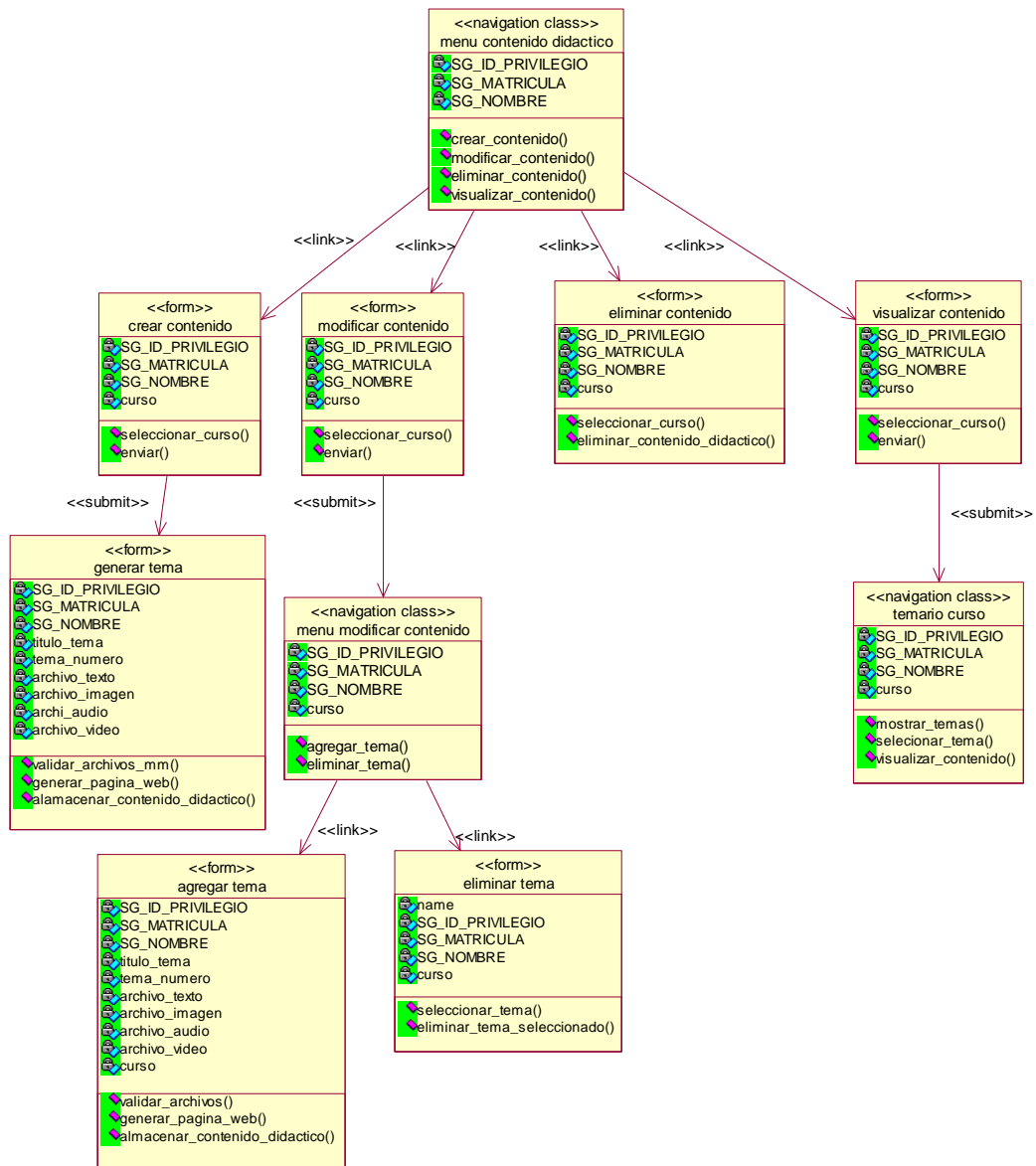


Figura 4.6: Modelo de navegación Administración de contenido didáctico.

4.3.3 Modulo administración de evaluaciones

El modelo de navegación del modulo Administración de evaluaciones se muestra en la figura 4.7. El modulo consta de la <<navigation class>> *Menú administración de evaluaciones* en la que se definen los métodos y operaciones necesarias para el procesamiento de las evaluaciones, como son crear, modificar eliminar o visualizar evaluaciones, todas ellas se presentan dentro de un <<form>>, en el cual siempre se solicita elegir el curso sobre el cual se va a procesar la opción seleccionada (crear, eliminar, modificar o visualizar evaluaciones), después de este paso, cada uno de los procesos, varia de la siguiente forma, según sea el caso.

- En el caso *crear evaluación*, el <<form >>, pide la selección del número de evaluación a crear, después dentro de otro estereotipo <<form>>, se introduce la pregunta, las tres posibles respuestas y se elige cual es la respuesta correcta, este proceso es idéntico para ingresar cada una de las siguientes ‘n’ preguntas a la evaluación.
- En el caso *modificar evaluación*, dentro de un <<form>>, se elige el número de evaluación a modificar, después, se presenta una <<navigation class>>, llamada menú modificar evaluación, con las opciones: *agregar preguntas* y *eliminar preguntas*, para lo cual se procede de la siguiente manera:
 - Para agregar preguntas, se piden los mismos datos que en el caso de crear evaluación (se introduce la pregunta, las tres posibles respuestas y se elige cual es la respuesta correcta), para agregarlas a la evaluación del curso que se esta procesando con esta opción.
 - Para eliminar preguntas, se elige de una lista desplegable la pregunta que se desea eliminar.
- En el caso *eliminar evaluación*, ya estando seleccionado el curso, se procede a elegir el número de evaluación dentro de un estereotipo <<form>>, y se elimina la evaluación.
- En el caso *visualizar evaluación*, ya estando seleccionado el curso, se procede a elegir el número de evaluación dentro de un estereotipo <<form>>, para poder visualizar el cuestionario completo.

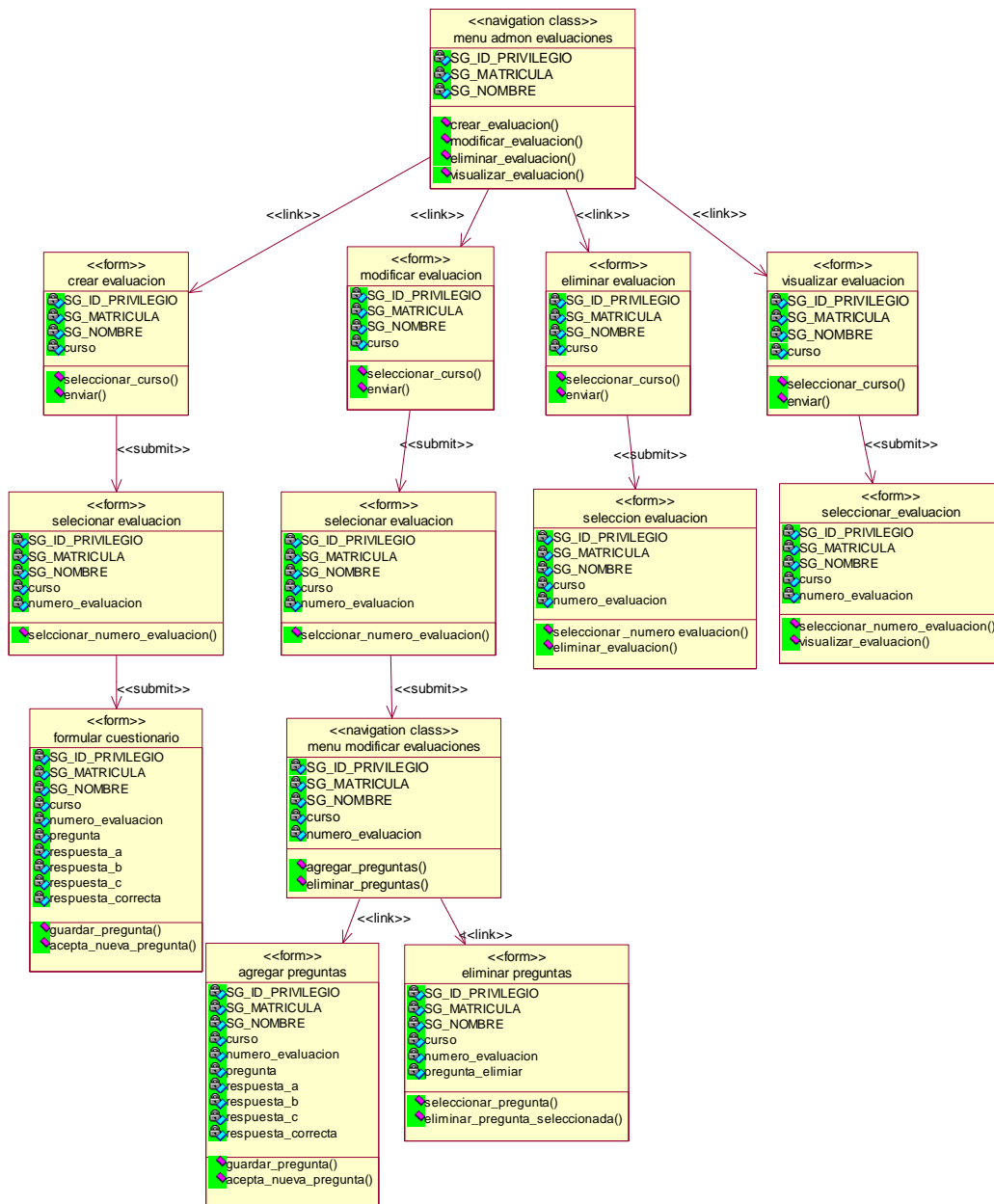


Figura 4.7: Modelo de navegación Administración de evaluaciones.

4.3.4 Modulo establecer contacto con los alumnos vía correo electrónico

El modelo de navegación del modulo *establecer contacto con los alumnos vía correo electrónico*, se muestra en la figura 4.8. El modulo consta de <<form>> *Envía correo alumnos* en la que se elige el curso en donde esta inscrito, el proceso continua en el <<form>> envía correo alumno, donde se elige el o los alumnos a los que se les enviará correo electrónico, enseguida, dentro de una <<navigation class>> se muestra la lista de los correos elegidos en el <<form>> anterior y desde un vinculo ejecuta el cliente de correo electrónico definido por el Sistema Operativo del equipo de computo del usuario para finalizar con la creación y el envío del correo electrónico.

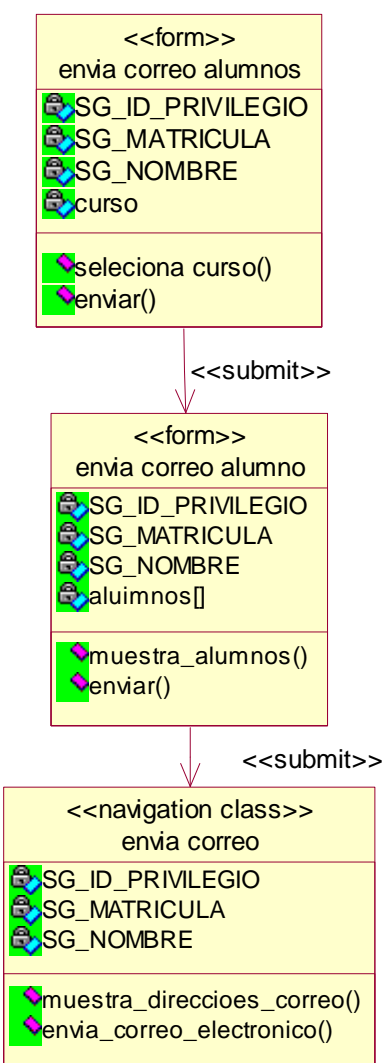


Figura 4.8: Modelo de navegación Establecer contacto con los alumnos vía correo electrónico.

4.3.5 Modulo inscripción a cursos

El modelo de navegación del modulo *inscripción a cursos*, se muestra en la figura 4.9. El modulo consta del <<form>> *Curso inscribir alumno* en la que se elige el curso en donde se inscribirá al o los alumnos, el proceso continua en el <<form>> *registro_a*, donde se busca por matricula al alumno a inscribir, enseguida, dentro de una <<navigation class>> se muestran los datos del alumno (si es que fue encontrado), para finalmente inscribir al alumno encontrado en el curso elegido inicialmente.

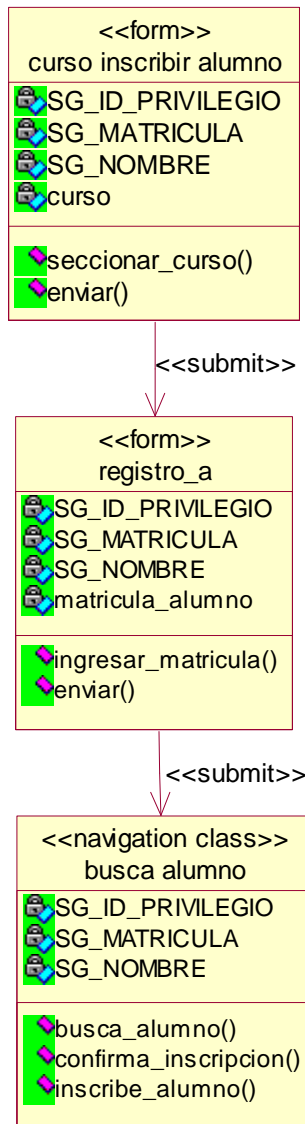


Figura 4.9: Modelo de navegación Inscripción a cursos.

4.3.6 Modulo baja de alumnos

El modelo de navegación del modulo *baja de alumnos*, se muestra en la figura 4.10. El modulo consta del <<form>> *Curso del alumno* en el que se elige el curso en donde se dará de baja al o los alumnos, el proceso continua en el <<form>> *baja_a*, donde se busca por matricula al alumno a dar de baja, si el alumno es encontrado, se da de baja del curso elegido inicialmente.

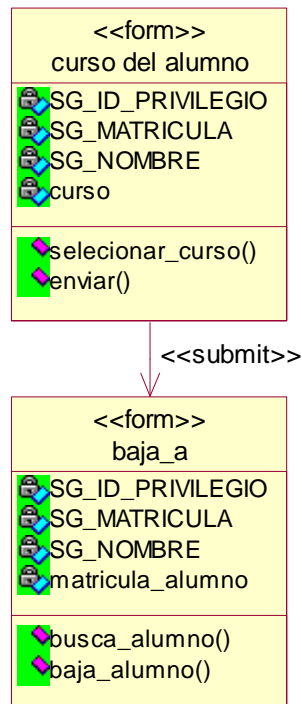


Figura 4.10: Modelo de navegación Baja de alumnos.

4.3.7 Modulo seguimiento de evaluaciones

El modelo de navegación del modulo *seguimiento de evaluaciones*, se muestra en la figura 4.11. El modulo consta del <<form>> *Curso administrar* en el que se elige el curso que se va a administrar, el proceso continua en el <<form>> *administracion curso*, donde se elige a algún alumno, para realizar las operaciones: “dar permiso de realizar evaluación”, “quitar permiso para realizar evaluación” y “modificar calificación”, todo esto como se menciona dentro de un <<form>>, y con ayuda de botones(aceptar o realizar cambios), cuadros de texto(ingresar calificación) y listas desplegables(para mostrar alumnos del curso).

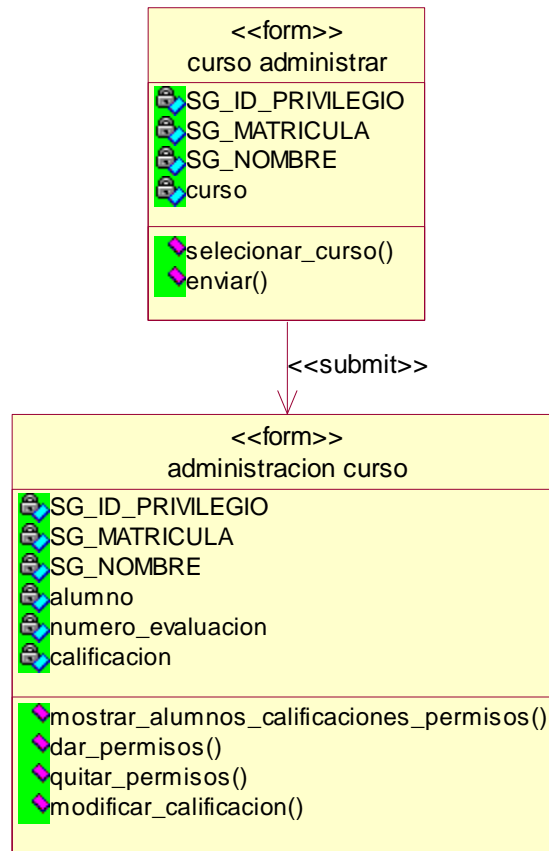


Figura 4.11: Modelo de navegación Seguimiento de evaluaciones.

4.3.8 Modulo realizar autoevaluaciones

El modelo de navegación del modulo *realizar autoevaluaciones*, se muestra en la figura 4.12. El modulo consta del <<form>> *Seleccionar curso*, en el que se elige el curso al que pertenece la evaluación que se va a realizar por un alumno, el proceso continua en el <<form>> *seleccionar evaluación*, donde se elige el número de evaluación a realizar por algún alumno, el paso siguiente consiste en la presentación de un <<form>> en el cual el alumno podrá ver y responder la evaluación, este ultimo paso requiere de validación de permisos para realizar la autoevaluación, y después de él, calificar y guardar resultados de la misma.

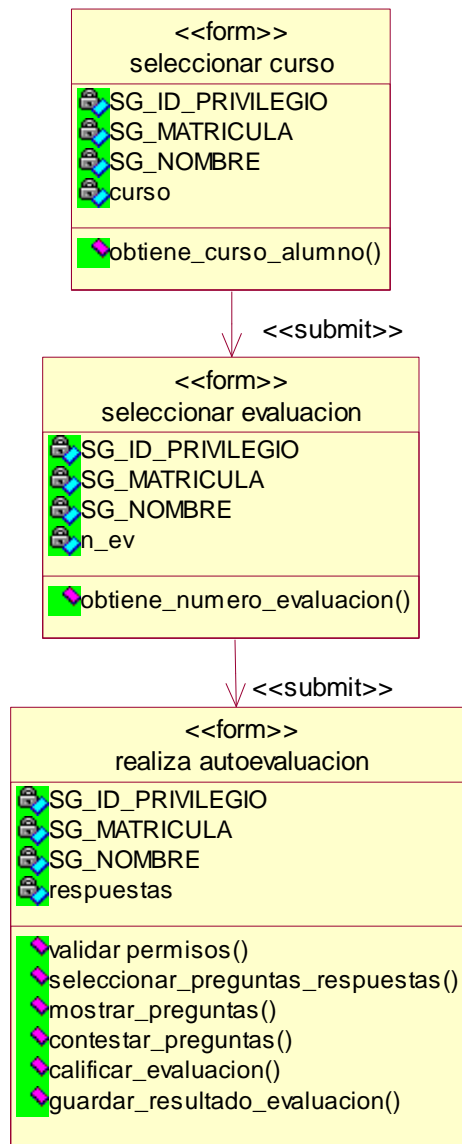


Figura 4.12: Modelo de navegación Realizar autoevaluaciones.

4.3.9 Modulo revisar resultado de autoevaluaciones

El modelo de navegación del modulo *revisar resultado de autoevaluaciones*, se muestra en la figura 4.13. El modulo consta del <<form>> *Seleccionar curso a revisar*, en el que se elige el curso en el que se desea conocer el resultado de las evaluaciones del mismo y de una <<navigation class>>, que muestra el estado de todas las evaluaciones del curso seleccionado.

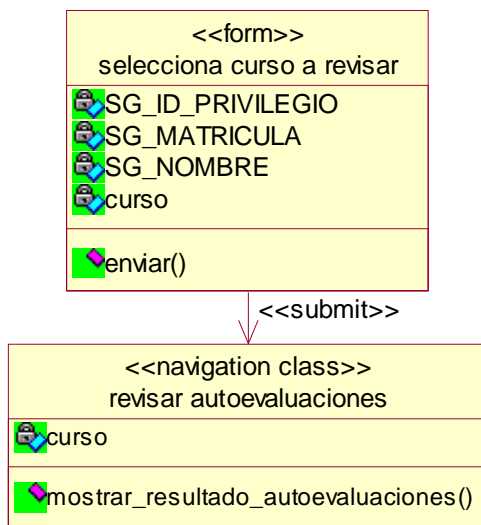


Figura 4.13: Modelo de navegación Revisar resultado de autoevaluaciones.

4.3.10 Modulo revisar o estudiar contenido didáctico

El modelo de navegación del modulo *revisar o estudiar contenido didáctico* se muestra en la figura 4.14. , dentro de un <<form>>, se solicita elegir el curso que se va a revisar o estudiar (visualizar contenido didáctico), después de este paso, se presenta una dentro de una <<navigation class>> , links hacia cada uno de los temas que tenga el curso elegido.

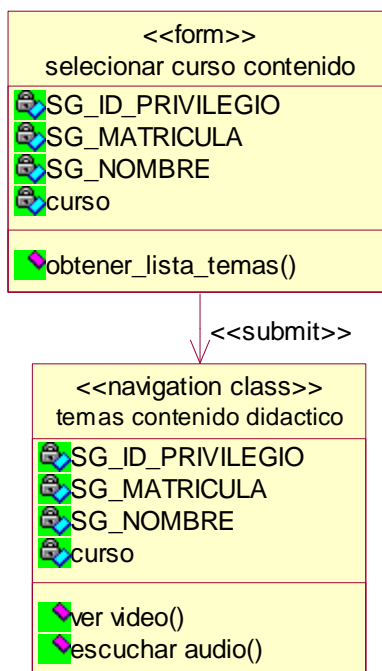


Figura 4.14: Modelo de navegación Revisar o estudiar contenido didáctico.

4.3.11 Modulo establecer contacto con los profesores vía correo electrónico

El modelo de navegación del modulo *establecer contacto con los profesores vía correo electrónico*, se muestra en la figura 4.15. El modulo consta de <<form>> *Seleccionar curso del profesor* en la que se elige el curso que imparte el profesor, enseguida, dentro de una <<navigation class>> se muestran los datos del profesor que imparte el curso elegido en el <<form>> anterior y desde un vinculo ejecuta el cliente de correo electrónico definido por el Sistema Operativo del equipo de computo del usuario para finalizar con la creación y el envío del correo electrónico.

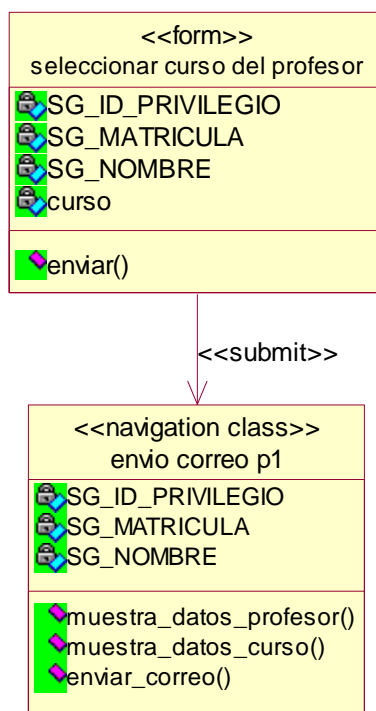


Figura 4.15: Modelo de navegación Establecer contacto con los profesores vía correo electrónico.

Capítulo 5





Escenarios Web

En este capítulo se describen los Escenarios de Navegación del SGCAD. Los diagramas UML que se utilizan para visualizar estos escenarios de navegación son los diagramas de estados. En una primera sección se describe la semántica de los elementos de modelado que se utilizan en los diagramas de estado. Posteriormente, se presentan los escenarios de navegación más significativos para el SGCAD.

5.1 Los escenarios Web y la semántica asociada a los diagramas de estados

Un escenario Web es una técnica para la descripción de situaciones de uso de un sitio Web con una meta o espacio concreto. Estas situaciones son el resultado de acciones llevadas a cabo por un personaje (usuario) o bien por la propia aplicación [7]. De acuerdo a las acciones que realiza, se toman en cuenta para modelar las situaciones que se pueden presentar, los escenarios pueden ser de tres tipos: escenarios que describen acciones de alta frecuencia, escenarios que describen acciones necesarias y escenarios que describen acciones marginales. Los escenarios que describen acciones de alta frecuencia describen situaciones muy frecuentes en la aplicación Web, pero triviales y poco trascendentes. Los escenarios que describen acciones necesarias describen situaciones que en algún momento se llevan a cabo dentro de la aplicación Web como parte de una funcionalidad de la misma. Por último, los escenarios que describen acciones marginales describen situaciones de poca incidencia, y por lo mismo fáciles de ignorar. Los escenarios Web que se modelan en este apartado describen las acciones necesarias de funcionalidades provistas por el SGCAD. Por otra parte, un diagrama de estados de UML denota una secuencia de los estados que un objeto puede adquirir durante su vida, junto con acciones responsivas, disparando eventos y las condiciones asociadas para indicar transiciones. Tomando en cuenta lo anterior, en este apartado se utilizan los diagramas de estados para visualizar escenarios de navegación Web. Estos diagramas permiten visualizar la parte dinámica del modelo de navegación, especificando los eventos que disparan las transiciones, definen condiciones y explícitamente incluyen las acciones a ser realizadas [8].

Debido a lo anterior, la semántica de los elementos que constituyen los diagramas de estado es distinta a la habitual. A continuación se describe la semántica de los elementos de modelado que aparecen en los diagramas de estado.

Elemento	Semántica
	Representa la situación inicial del escenario. Esta situación normalmente no tiene gran representatividad en la aplicación Web, pero se incluye como referencia inicial para el escenario Web que se está describiendo.
	Representa a una situación del escenario Web en la provisión de una funcionalidad. Cada situación involucra actividades y a las entidades que participan en ella. En la nomenclatura de estas situaciones se utilizan descriptores de las entidades que participan en la situación y de las actividades que realizan, concatenadas con el carácter +.
	Mediante las flechas se representan a eventos, acciones o circunstancias que provocan el traslado del flujo de una situación a otra. Las flechas se etiquetan con una descripción breve de dicho evento, acción o circunstancia.
	Representa la situación final del escenario. Esta situación normalmente no tiene gran representatividad en la aplicación Web, pero se incluye como referencia final para el escenario Web que se está describiendo.

En base a esta semántica, a continuación se presentan los escenarios Web más significativos para el SGCAD.

5.2 Escenarios Web del SGCAD

5.2.1 Escenario Web administración de cursos

En este escenario, que se muestra en la figura 5.1 se describe la operación de la administración de cursos de manera global. Cuando un usuario profesor accede al Home SGCAD, en la opción: “cursos”, se presentan dos opciones principales: “crear curso” y “eliminar curso”, la primera opción le permite al profesor crear un nuevo curso a impartir, esto lo hace eligiendo de listas desplegables, el nombre del curso, y el número de evaluaciones que tendrá el mismo. La segunda opción le permite eliminar un curso, el proceso es parecido; se elige de una lista desplegable el curso que se desea eliminar y el curso es eliminado del sistema.

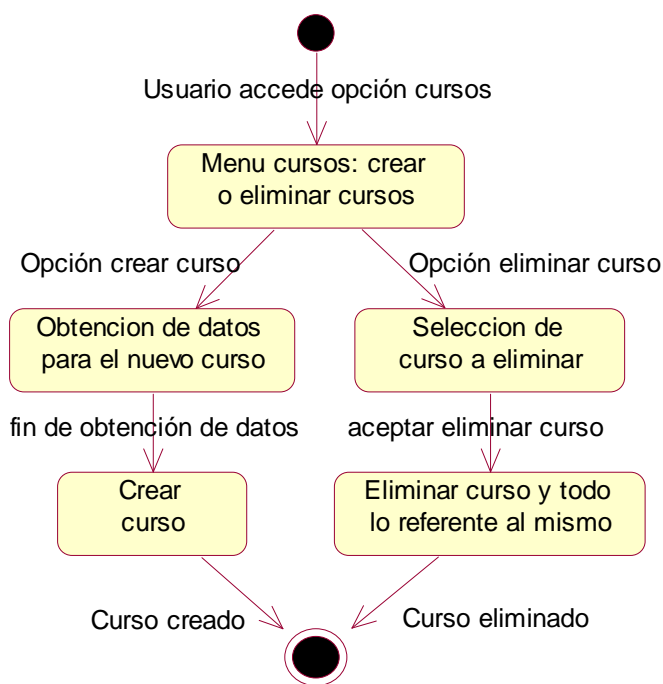


Figura 5.1: Escenario Web Administración de cursos.

5.2.2 Escenario Web administración de contenido didáctico

En este escenario, que se muestra en la figura 5.2 se describe la operación de la administración de contenido didáctico de manera global. Cuando un usuario profesor accede al Home SGCAD, en la opción: “Contenido didáctico”, se presentan en un menú las opciones: “crear contenido”, “eliminar contenido”, “modificar contenido” y “visualizar contenido”. Todas ellas solicitan que el usuario elija el curso sobre el cual se va a procesar

la opción seleccionada (crear, eliminar, modificar o visualizar contenido didáctico), después de este paso, cada uno de los procesos, varía de la siguiente forma, según sea el caso.

- En el caso *crear contenido didáctico*, este se procesa generando tema por tema, por tanto se requiere de la entrada de datos referentes a cada tema a crear, como son Título del tema, Tema en número, (1, 1.1, 1.1.2,...), y archivos multimedia, como son audio, video, texto e imagen, que son buscados en la computadora del profesor, para posteriormente ser subidos al servidor y almacenados en él, al presionar el botón “Crear tema”, se validan los datos, si son correctos, se genera el nuevo archivo (pagina web del tema) y se actualiza la base de datos, el proceso se repite limpiando los campos del formulario para crear de la misma forma cada uno de los temas del curso, la operación termina haciendo clic en el link “cancelar”.
- En el caso *modificar contenido didáctico*, se cuentan con las opciones: *agregar tema* y *eliminar tema*, para lo cual se procede de la siguiente manera:
 - Para agregar tema, se piden los mismos datos que en el caso de crear contenido (audio, video, texto, imagen, titulo del tema y tema en numero), para agregarlos al contenido del curso que se esta procesando con esta opción.
 - Para eliminar tema, se elige de una lista desplegable el tema que se desea eliminar, y con ello se elimina todo lo referente a él.
- En el caso *eliminar contenido didáctico*, ya estando seleccionado el curso, se procede a eliminar el contenido didáctico, presionando el botón “eliminar”.

En el caso *visualizar contenido didáctico*, estando elegido el curso, se presenta una lista de todos los temas creados, con links hacia cada uno de los mismos, cuando el usuario hace clic en uno de ellos, se muestra la pagina web del tema.

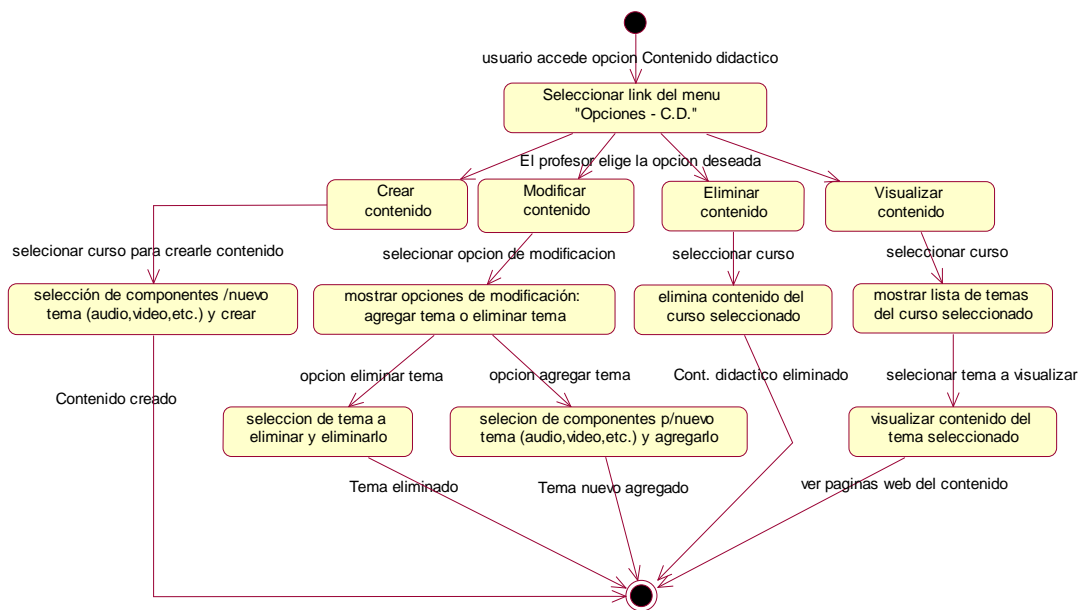


Figura 5.2: Escenario Web Administración de contenido didáctico.

5.2.3 Escenario Web administración de evaluaciones

En este escenario, que se muestra en la figura 5.3 se describe la operación de la administración de evaluaciones de manera global. Cuando un usuario profesor accede al Home SGCAD, en la opción: “Autoevaluaciones”, se presentan en un menú las opciones: “crear evaluación”, “eliminar evaluación”, “modificar evaluación” y “visualizar evaluación”. Todas ellas solicitan que el usuario elija el curso y la evaluación sobre la cual se va a procesar la opción seleccionada (crear, eliminar, modificar o visualizar evaluación), después de este paso, cada uno de los procesos, varía de la siguiente forma, según sea el caso.

- En el caso *crear evaluación*, dentro de un formulario, se introduce la pregunta, las tres posibles respuestas y se elige la respuesta correcta, al presionar el botón “guardar”, la pregunta (para la evaluación elegida) es guardada en la base de datos, a continuación el sistema regresa a la misma página y limpia los campos de texto del formulario para ingresar otra pregunta a la misma evaluación, el proceso es idéntico para ingresar cada una de las siguientes “n” preguntas a la evaluación, para finalizar este proceso y dejar de ingresar preguntas se da clic el link “cancelar”.
- En el caso *modificar evaluación*, se cuenta con las opciones: *agregar preguntas* y *eliminar preguntas*, para lo cual se procede de la siguiente manera:
 - Para agregar preguntas, se piden los mismos datos que en el caso de crear evaluación (la pregunta, las tres posibles respuestas y se elige cual es la respuesta correcta), para agregarlas a la evaluación del curso que se está procesando con esta opción, al presionar el botón “guardar”, la pregunta es guardada en la base de datos, a continuación el sistema regresa a la misma interfaz para ingresar otra pregunta, para finalizar este proceso y dejar de agregar preguntas se hace clic el link “cancelar”.
 - Para eliminar preguntas, se elige de una lista desplegable la o las preguntas que se desean eliminar y se eliminan presionando el botón “aceptar”.
- En el caso *eliminar evaluación*, ya estando seleccionado el curso y número de evaluación, se elimina la misma.
- En el caso *visualizar evaluación*, ya estando seleccionado el curso y número de evaluación, se visualiza el cuestionario completo.

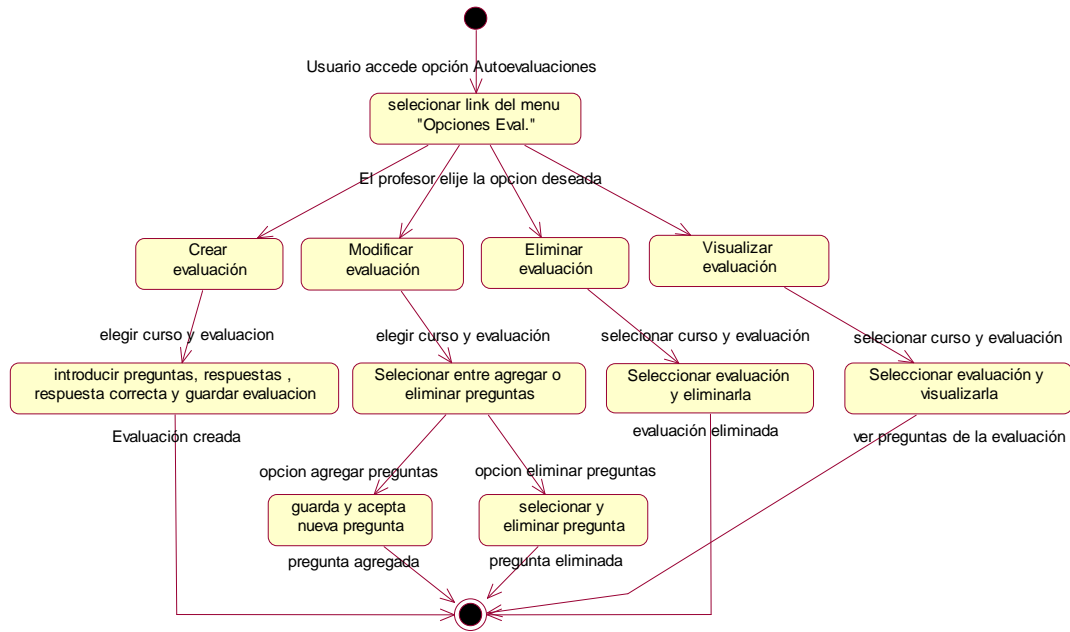


Figura 5.3: Escenario Web Administración de evaluaciones.

5.2.4 Escenario Web Establecer contacto con los alumnos vía correo electrónico

En este escenario, que se muestra en la figura 5.4 se describe la operación de Establecer contacto con los alumnos vía correo electrónico. Cuando un usuario profesor accede al Home SGCAD, en la opción “Alumnos”, “Enviar correo”, lo primero que hace es elegir de un menú desplegable, el curso en donde están inscritos los alumnos a los que se les desea enviar un correo electrónico, en otra pagina web se eligen a los alumnos a los que se les enviará el correo electrónico, se muestra la lista de los correos elegidos en el paso anterior y al hacer clic en el vinculo “enviar”, se ejecutara el cliente de correo electrónico definido por el Sistema Operativo del equipo de computo del usuario para finalizar con la creación y el envío del correo electrónico.

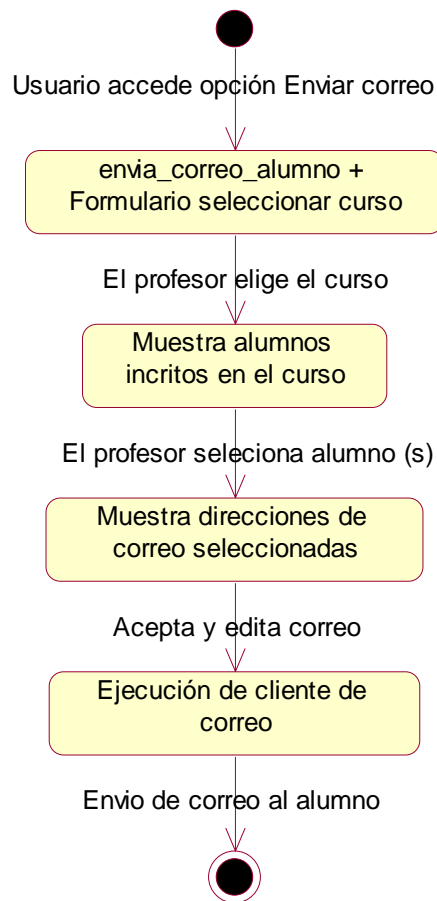


Figura 5.4: Escenario Web Establecer contacto con los alumnos vía correo electrónico.

5.2.5 Escenario Web inscripción a cursos

En este escenario, que se muestra en la figura 5.5 se describe la operación de Inscripción a cursos. Cuando el usuario profesor accede al Home SGCAD, en la opción “Alumnos”, “Inscribir”, lo primero que hace es elegir de un menú desplegable, el curso en donde desea inscribir alumnos, en otra pagina web se proporciona la matricula del alumno a inscribir, la matricula es buscada dentro de la base de datos, si la matricula es encontrada, se muestra el nombre del alumno al que pertenece esta matricula, para inscribir a este al curso basta con presionar el botón “Aceptar” o “Cancelar ” para abortar la operación, después de presionar “Aceptar”, el sistema presenta la pagina web de búsqueda de matricula para inscribir a otro alumno, si ya no fuera el caso se hace clic en el link “Cancelar”.

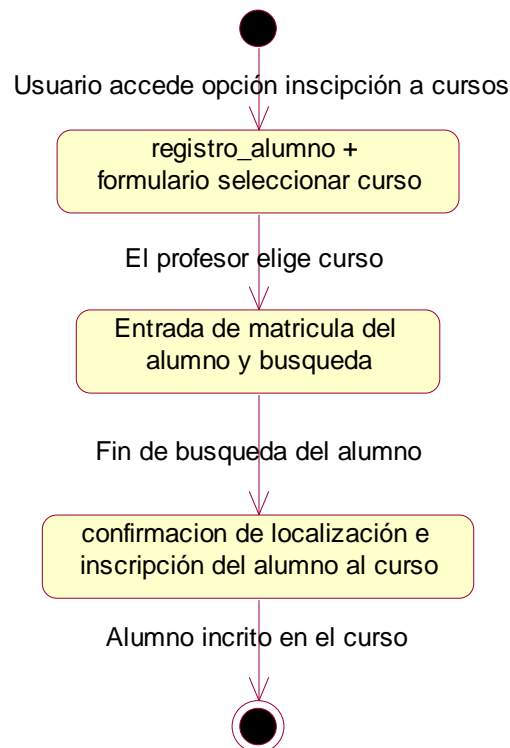


Figura 5.5: Escenario Web Inscripción a cursos.

5.2.6 Escenario Web baja de alumnos

En este escenario, que se muestra en la figura 5.6 se describe la operación de Baja de alumnos. Cuando el usuario profesor accede al Home SGCAD, en la opción “Alumnos”, “Dar de baja”, lo primero que hace es elegir de un menú desplegable, el curso en donde desea dar de baja alumnos, en otra pagina web (dentro de otro formulario) se proporciona la matricula del alumno a dar de baja, la matricula es buscada dentro de la base de datos, si la matricula es encontrada, el alumno con esa matricula es dado de baja del curso, después de presionar “Buscar”, el sistema presenta la pagina web de búsqueda de matricula para dar de baja a otro alumno, si ya no fuera el caso se hace clic en el link “Cancelar”.

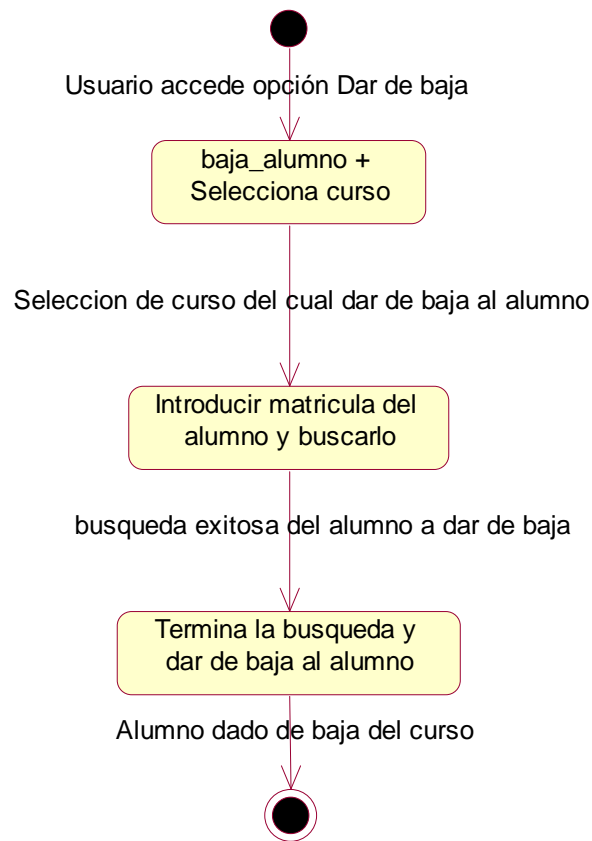


Figura 5.6: Escenario Web Baja de alumnos.

5.2.7 Escenario web Seguimiento de evaluaciones

En este escenario, que se muestra en la figura 5.7 se describe la operación de Seguimiento de evaluaciones. Cuando el usuario profesor accede al Home SGCAD, en la opción “Alumnos”, “Calificaciones”, lo primero que hace es elegir de un menú desplegable, el curso que desea administrar, (con respecto a las evaluaciones) el proceso continua dentro de una nueva pagina web donde se muestra una tabla con matricula, nombre, calificaciones y estado de las evaluaciones de cada uno de los alumnos del curso elegido, junto con ello las herramientas para poder modificar calificaciones y estado de las evaluaciones de los alumnos, en el caso de las evaluaciones (otorgar o quitar permisos para que un alumno realice una determinada evaluación), con menús desplegables se elige al alumno y la evaluación, y al presionar un botón (según sea el caso) se da o quita permiso de realizar la evaluación del alumno elegido.

En el caso de modificar calificación, también se elige el alumno y la evaluación de listas desplegables, y dentro de un campo de texto se escribe la nueva calificación que se le asignará a este, para que esta calificación quede guardada, se presiona el botón “modificar calificación”, para abortar cualquiera de las dos operaciones, se presiona “Cancelar”.

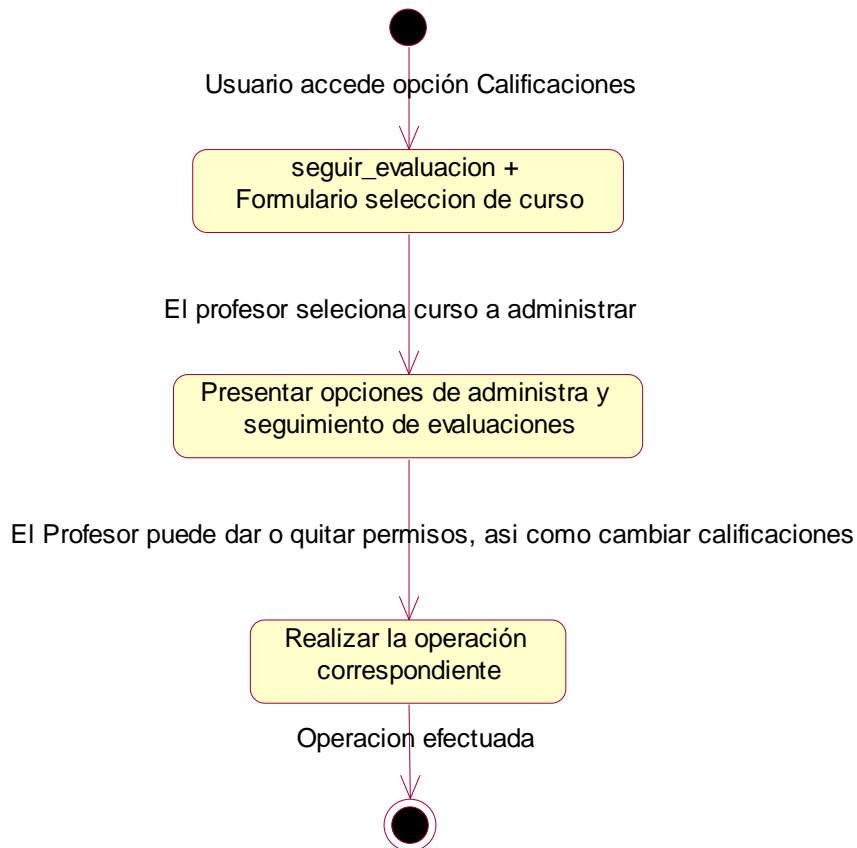


Figura 5.7: Escenario Web Seguimiento de evaluaciones.

5.2.8 Escenario Web Realizar autoevaluaciones

En este escenario, que se muestra en la figura 5.8 se describe la operación de Realizar autoevaluaciones. Cuando el usuario alumno accede al Home SGCAD, en la opción “Realizar autoevaluación”, lo primero que hace es elegir de un menú desplegable, el curso y el número de evaluación, el sistema valida que el alumno tenga permiso de realizar la evaluación seleccionada, si esto fuera así, en otra nueva pagina web se muestra un formulario con todas las preguntas y sus posibles respuestas (evaluación de opción múltiple), para que el alumno conteste la evaluación. Al presionar el botón “Guardar”, la evaluación es calificada y se almacena el resultado en la base de datos, (después de realizar la evaluación, el alumno ya no tiene permiso para realizarla nuevamente), en el caso de que el alumno no tenga autorización para realizar la evaluación se le notifica con un mensaje indicándole lo mencionado anteriormente.

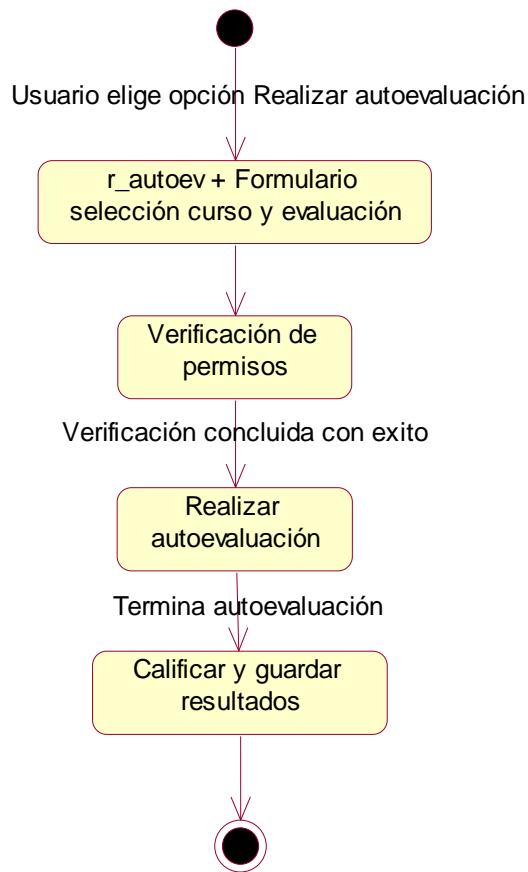


Figura 5.8: Escenario Web Realizar autoevaluaciones.

5.2.9 Escenario Web Revisar resultado de autoevaluaciones

En este escenario, que se muestra en la figura 5.9 se describe la operación de Revisar resultado de autoevaluaciones. Cuando el usuario alumno accede al Home SGCAD, en la opción “Calificaciones”, elige de un menú desplegable, el curso en el que se desea conocer el resultado de las evaluaciones del mismo, después de este paso, se muestra el estado de las evaluaciones (calificación, si ya fue realizada o no) y el numero de evaluación.

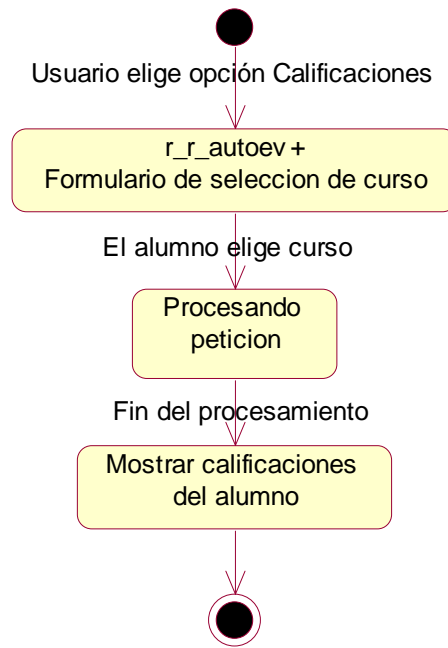


Figura 5.9: Escenario Web Revisar resultado autoevaluaciones.

5.2.10 Escenario Web Revisar o estudiar contenido didáctico

En este escenario, que se muestra en la figura 5.10 se describe la operación de Revisar o estudiar contenido didáctico. Cuando el usuario alumno accede al Home SGCAD, en la opción “Contenido didáctico”, elige de un menú desplegable, el curso que desea estudiar, estando elegido el curso, se presenta una lista de todos los temas creados, con links hacia cada uno de los mismos, cuando el alumno hace clic en uno de ellos, se muestra la pagina web donde se desarrolla el tema.

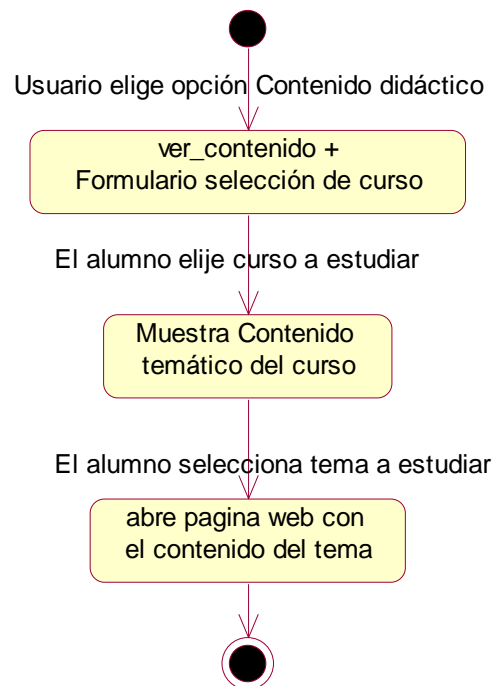


Figura 5.10: Escenario Web Revisar o estudiar contenido didáctico.

5.2.11 Escenario Web Establecer contacto con los profesores vía correo electrónico

En este escenario, que se muestra en la figura 5.11 se describe la operación de Establecer contacto con los profesores vía correo electrónico. Cuando un usuario alumno accede al Home SGCAD, en la opción “Contacto profesores”, elige de un menú desplegable, el curso que imparte el profesor al que se le desea enviar un correo electrónico, en otra pagina web, se muestran los datos del profesor que imparte el curso elegido en el paso anterior y al hacer clic en el vinculo “enviar”, se ejecuta el cliente de correo electrónico definido por el Sistema Operativo en la computadora del usuario, para finalizar con la creación y el envío del correo electrónico.

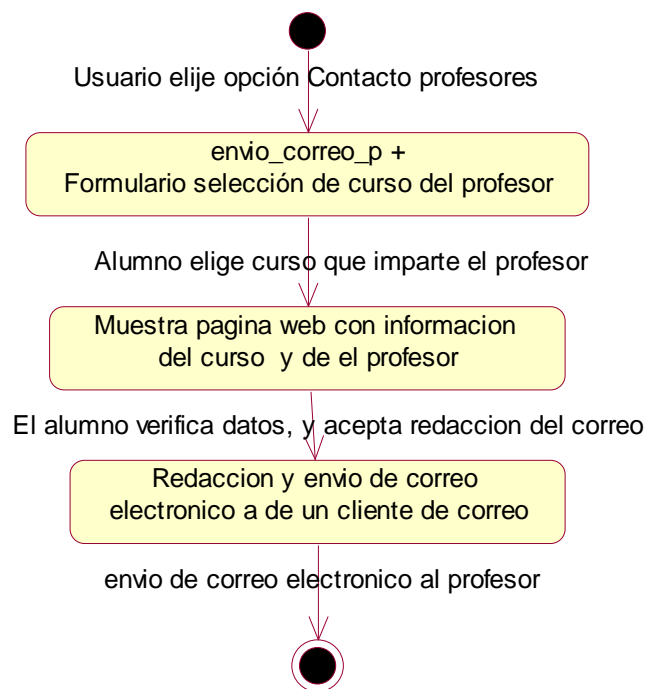


Figura 5.11: Escenario Web Establecer contacto con los profesores vía correo electrónico.

Capítulo 6

Esquema de implementación

En este capítulo se presenta un único diagrama de componentes mediante el cual se proporciona una vista general del esquema de implementación que se propone para el SGCAD. En un primer apartado se describe la semántica asociada a los elementos de modelado que se utilizan en el diagrama. Posteriormente se muestra el diagrama de componentes y los comentarios correspondientes.

6.1 Semántica de elementos asociados al diagrama de componentes

Los elementos de modelado UML que aparecen en el diagrama de componentes que se presenta en este capítulo son: nodos, paquetes, componentes y asociaciones. Un nodo es un elemento físico que existe en el tiempo de ejecución y representa un recurso computacional disponible en la aplicación. Un nodo en el diagrama de componentes se representa con un cubo. Un nodo puede contener objetos y componentes que residen dentro de ese recurso de cómputo. Por otra parte, un componente representa una parte del sistema que proporciona la ejecución de una tarea o servicio específico y se representa como rectángulo con dos pestañas sobre el lado izquierdo. Por último, las asociaciones entre componentes se representan con una flecha. Cada asociación representa en el diagrama algún tipo de relación entre los componentes implicados. Además, estas asociaciones pueden enriquecer su semántica en el diagrama mediante la utilización de estereotipos que denoten con mayor claridad el tipo de relación que se establece entre los componentes y/o nodos [8].

6.2 Esquema de Implementación del SGCAD

En el diagrama de componentes de la figura 6.1 se provee una vista genérica de la implementación del SGCAD. El diagrama se enfoca al SGCAD como elemento principal y las dependencias que sostiene con otros elementos requeridos para su correcto funcionamiento.

Esta propuesta de implementación se plantea con la utilización de tecnologías de uso frecuente en el ámbito del desarrollo de aplicaciones Web tales como PHP y MySQL. En el anteproyecto que se presenta en el próximo capítulo se hace referencia a ambas tecnologías. Como se puede observar la aplicación SGCAD reside en el nodo Web Server. La aplicación SGCAD, tal como se presenta en el modelo de navegación, consiste de una amplia colección de <<navigation class>>, entre otros elementos. Cuando se accede al SGCAD por medio de un navegador Web lo que realmente sucede es que se envía una solicitud al Web Server en donde se encuentra hospedada la aplicación. Cuando el Web Server recibe dicha solicitud, se inicia el proceso de ejecución para satisfacer dicha solicitud. El

contenido dinámico debe ser interpretado por el Web Server y para ello se requiere, en este caso, de un módulo PHP. Con el soporte que provee este módulo se accede a la fuente de datos que reside en servidor dedicado a este fin. En este caso, se propone la utilización de un manejador de base de datos MySQL. Una vez que el código de servidor se interpreta, se accede a la base de datos para la ejecución de instrucciones SQL y se genera el contenido requerido, se envía al Web Browser del Cliente un código (HTML) que incluye tanto el contenido dinámico generado por el servidor, como el estático claramente interpretable por el Web Browser. Una vez que el Web Browser interpreta este código despliega como resultado la <<client page>> para que sea visualizada por el usuario [6].

Este esquema de implementación es el más comúnmente usado en aplicaciones Web. Al respecto de esto, en el próximo capítulo se presenta un anteproyecto de implementación en donde se refinan algunos aspectos asociados al esquema descrito en este diagrama.

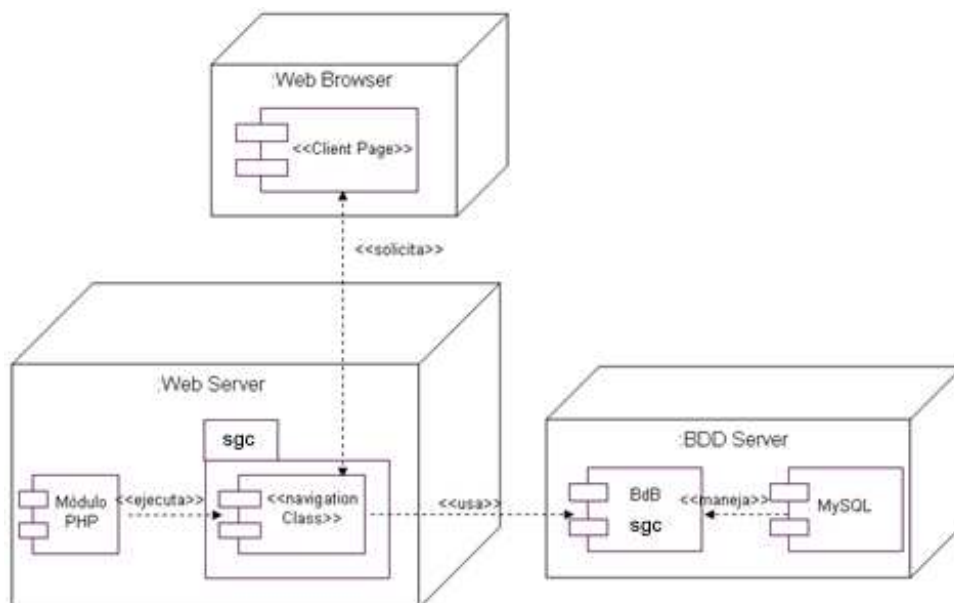


Figura 6.1: Diagrama de componentes, Esquema propuesto de implementación para el SGCAD

6.3. El sistema generador de cursos educativos multimedia en la práctica

En esta sección se describe el funcionamiento del sistema, para los dos tipos de usuarios del mismo. Como se menciona en capítulos anteriores el sistema requiere que el usuario acceda al mismo por medio de la validación del nombre de usuario y clave, en la figura 6.2, se observa la clase de navegación para este proceso.

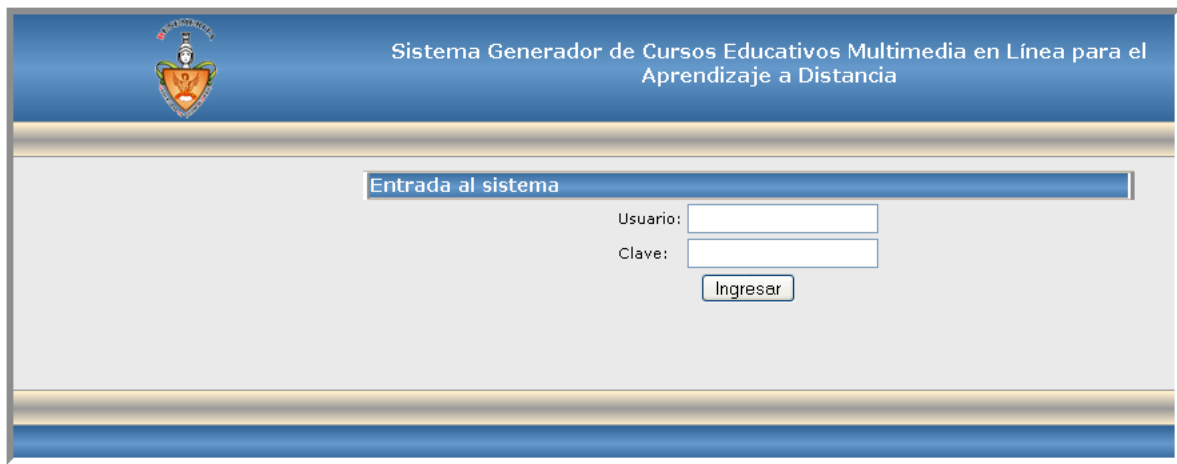


Figura 6.2: Clase de navegación entrada al sistema.

El sistema identifica que tipo de usuario es el que ingresa (profesor o alumno) y por consiguiente se presentan diferentes opciones dependiendo del tipo de usuario que ha ingresado. A continuación se describirán, las clases de navegación para estos usuarios.

6.3.1. Usuario profesor

En la figura 6.3 se observa la representación de la clase de navegación principal para el usuario profesor. En la figura se pueden apreciar las opciones u operaciones sobre Alumnos, Cursos, Contenido didáctico y Autoevaluaciones.

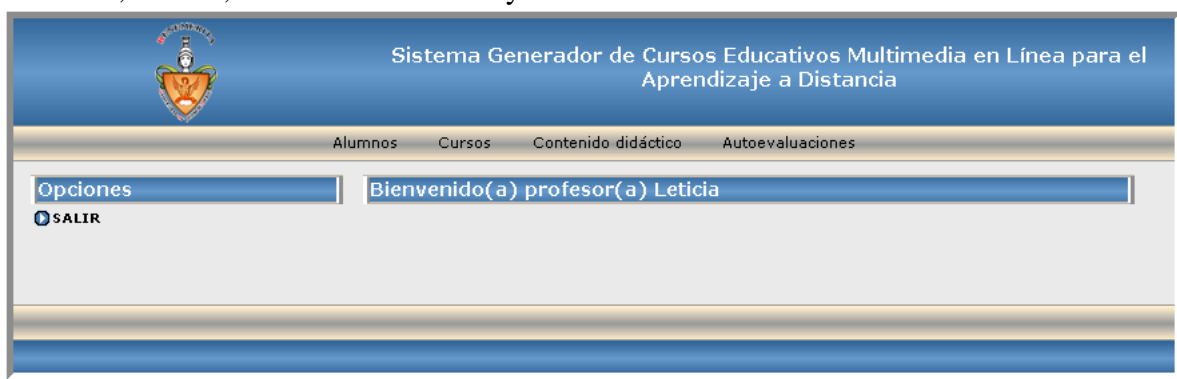


Figura 6.3: Clase de navegación principal del SGCAD (usuario profesor)

6.3.1.1. Operaciones sobre alumnos

Las operaciones sobre los alumnos son: Inscripción o baja de alumnos, envío de correo electrónico y calificaciones. Todas estas opciones empiezan con una clase de navegación en donde se elige un curso, en la figura 6.4, se muestra dicha clase de navegación para la elección de un curso dentro de la opción “Inscribir”.

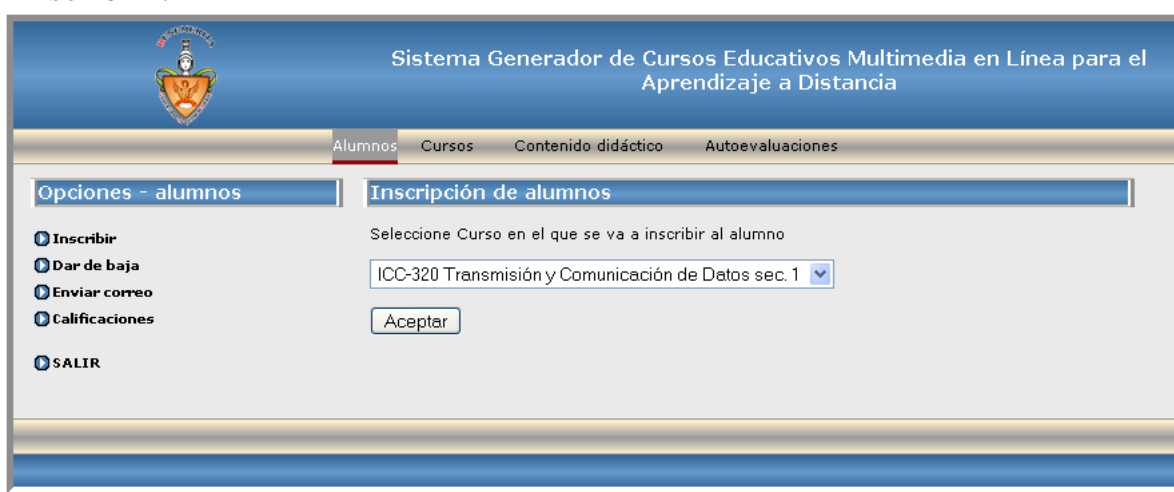


Figura 6.4: Clase de navegación seleccionar curso (Alumnos, inscribir).

Inscripción o baja de alumnos

Para inscribir o dar de baja a un alumno, después de seleccionar el curso (figura 6.4), el paso siguiente es introducir la matrícula del alumno que se desee dar de baja o inscribir el curso (figura 6.5), para finalizar la operación, el sistema envía una ultima confirmación para que el usuario acepte la conclusión de la operación (figura 6.6).

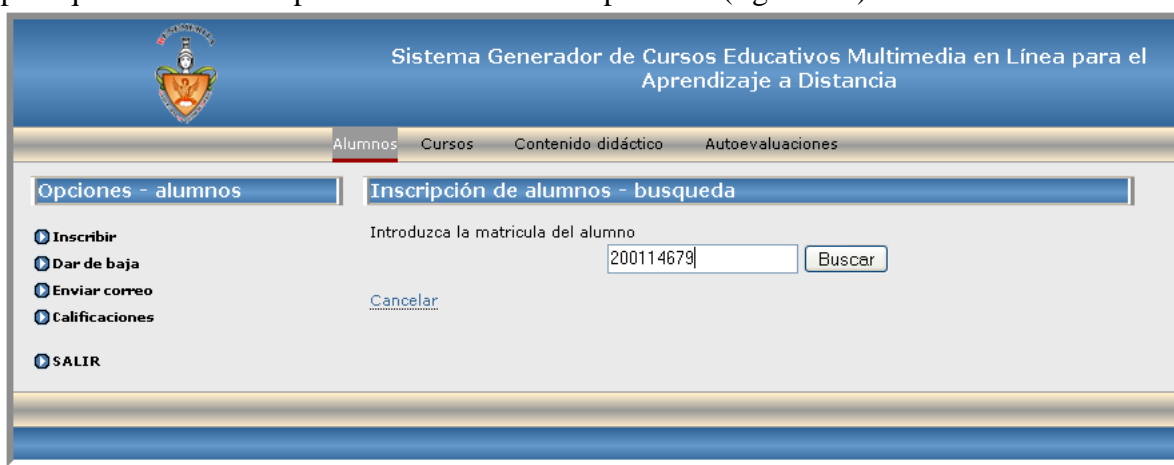


Figura 6.5: Clase de navegación buscar alumno (Alumnos, inscribir, búsqueda).

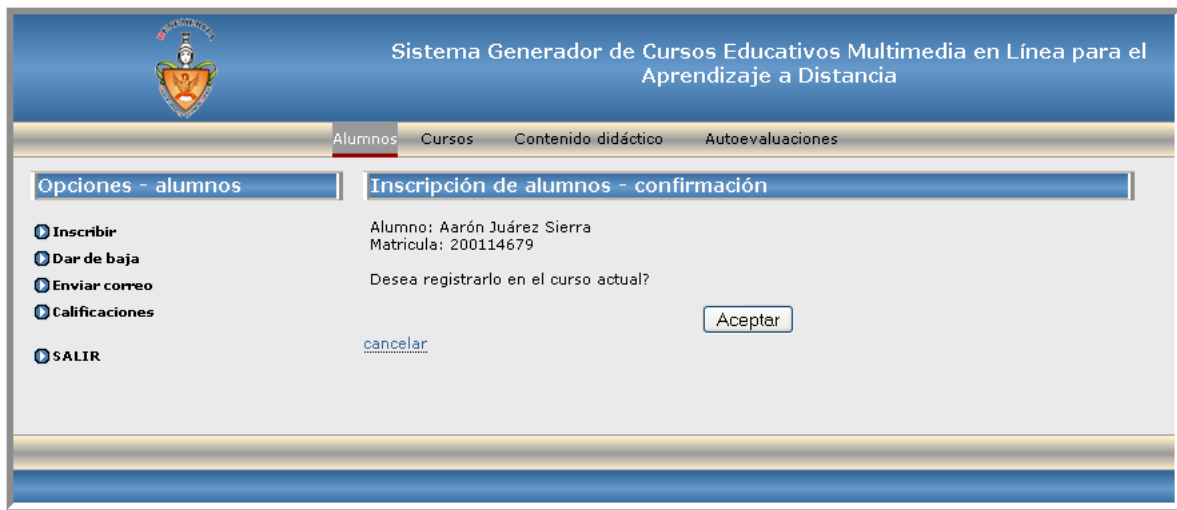


Figura 6.6: Clase de navegación confirma inscripción alumno (Alumnos, inscribir, confirmación).

Enviar correo

Para enviar correo a un alumno o alumnos, después de seleccionar el curso (figura 6.4), el paso siguiente es elegir el o los alumnos, a los que se enviara un correo electrónico (figura 6.7), para finalizar la operación, el sistema envía una última confirmación para que el usuario acepte la conclusión de la operación (figura 6.8).

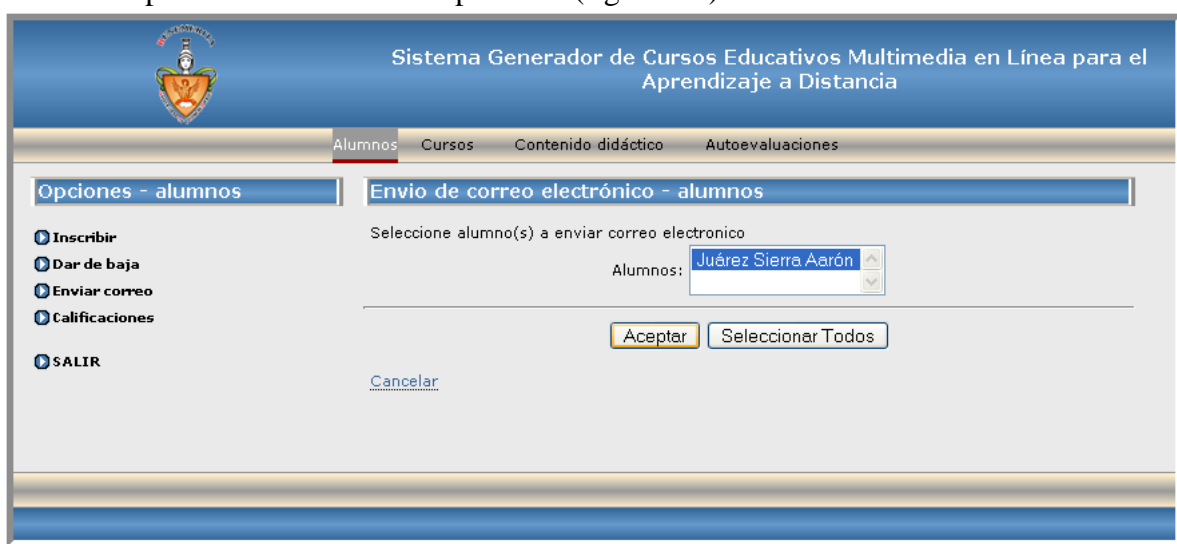


Figura 6.7: Clase de navegación seleccionar alumnos (Alumnos, Enviar correo, Selección alumnos).

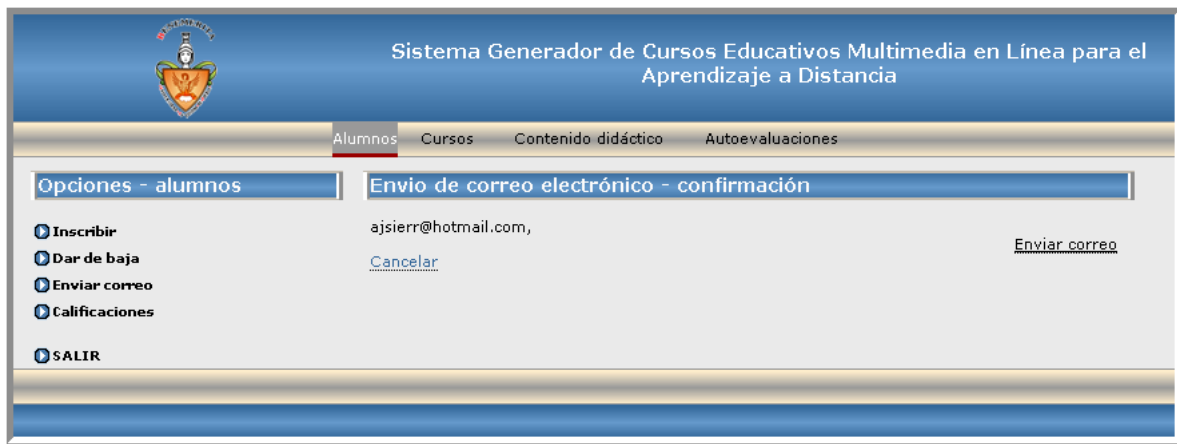


Figura 6.8: Clase de navegación Confirmación correo (Alumnos, Enviar correo, confirmación).

Calificaciones

Dentro de esta opción, el usuario profesor puede dar o no permiso, para que un usuario alumno (elegido previamente), realice o no sus autoevaluaciones, además también puede modificar las calificaciones de las autoevaluaciones realizadas por sus alumnos (figura 6.9).

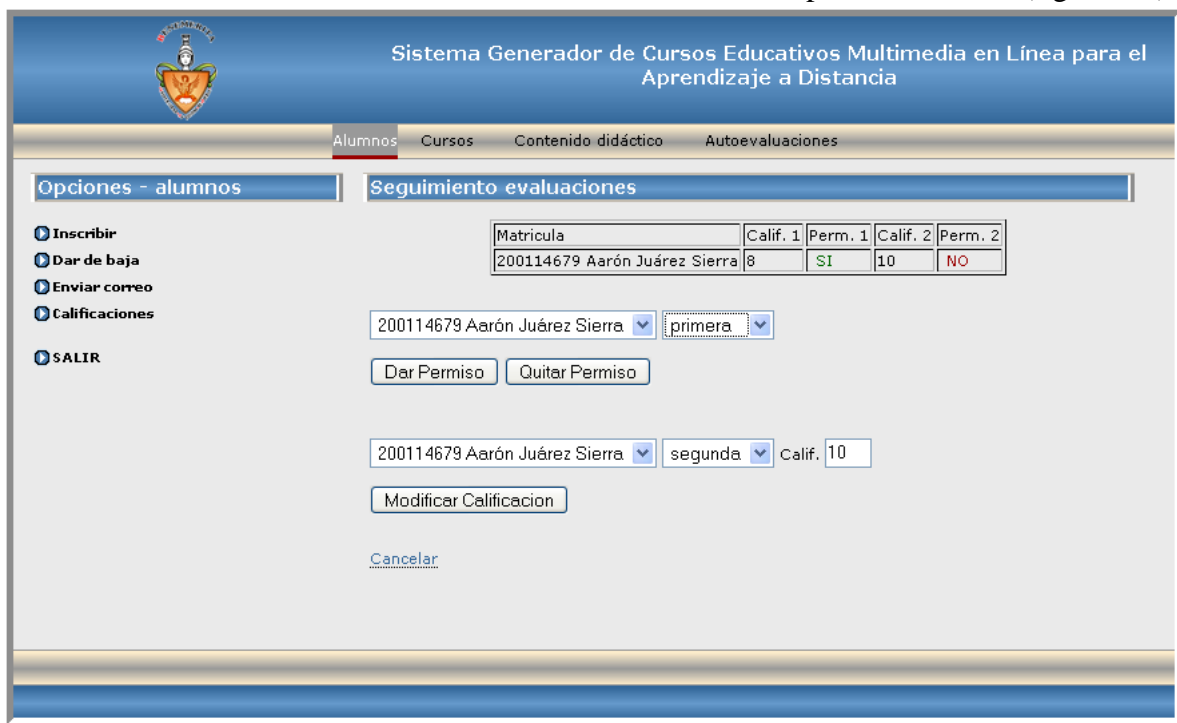


Figura 6.9: Clase de navegación Calificaciones (Alumnos, Calificaciones).

6.3.1.2 Operaciones sobre cursos

Las dos operaciones básicas, son: “Crear curso” y “Eliminar curso”; como sus nombres indican la operación “Crear curso”, permite crear un nuevo curso, eligiendo el nombre del mismo y el número de evaluaciones que tendrá (figura 6.10). La operación “Eliminar curso”, elimina un curso completamente, incluyendo todo lo relacionado con él (alumnos, evaluaciones, material didáctico, etc.), solo se elije el curso que se desea eliminar (figura 6.10).

The screenshot shows the 'Sistema Generador de Cursos Educativos Multimedia en Línea para el Aprendizaje a Distancia' interface. The top navigation bar includes 'Alumnos', 'Cursos', 'Contenido didáctico', and 'Autoevaluaciones'. The 'Cursos' tab is active, and the 'Cursos - Crear' sub-tab is selected. On the left, under 'Opciones - cursos', there are two options: 'Crear Curso' and 'Eliminar Curso'. The main area contains the text 'Seleccione curso y número de evaluaciones'. Below this, there are two dropdown menus: the first is set to 'Inteligencia Artificial' and the second is set to 'tres'. A 'Crear' button is located below the dropdowns.

Figura 6.9: Crear curso (Cursos, Crear curso).

The screenshot shows the same system interface as Figure 6.9, but with the 'Cursos - Eliminar' sub-tab selected. The left sidebar now highlights 'Eliminar Curso'. The main area contains the text 'Seleccione Curso a Eliminar'. Below this, there is a dropdown menu set to 'ICC-320 Transmisión y Comunicación de Datos sec. 1'. An 'Aceptar' button is located below the dropdown menu.

Figura 6.10: Eliminar curso (Cursos, Eliminar curso).

6.3.1.3 Operaciones sobre contenido didáctico

Las operaciones sobre Contenido didáctico: Crear, Modificar, Eliminar y Visualizar contenido didáctico empiezan con una clase de navegación en donde se elige un curso, en la figura 6.11, se muestra dicha clase de navegación para la elección de un curso para la operación “Modificar contenido didáctico, Agregar tema”.

El contenido didáctico es procesado para su modificación, creación y visualización por medio de temas.

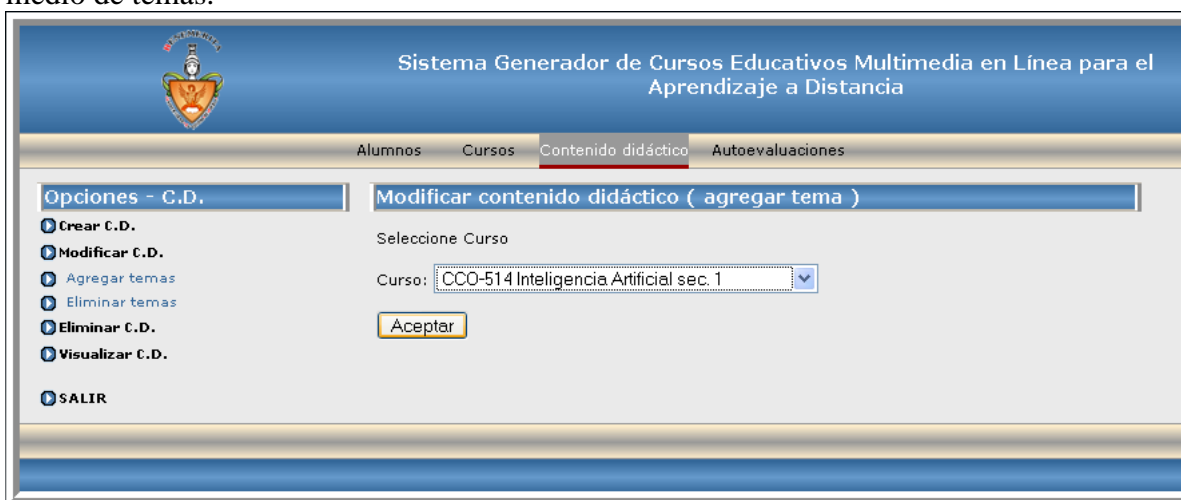


Figura 6.11: Clase de navegación Modificar contenido didáctico (agregar tema – selección de curso).

Crear contenido

Esta opción, permite generar contenido didáctico para el curso elegido previamente, el paso posterior es comenzar con la creación de temas del curso, los cuales son generados a partir de elementos básicos como son el texto que contendrá el tema, elementos multimedia (audio y video) e imágenes, ingresados como archivos y enviados al sistema para que este los procese y genere una pagina Web por cada tema, este ultimo paso (generar tema), se describe a detalle junto con la operación “Modificar contenido, Agregar temas”, por ser en esencia el mismo procedimiento.

Modificar contenido

Esta opción consta de dos operaciones: “Agregar temas y “Eliminar temas”. En la figura 6.12 se observa la representación de la clase de navegación para la opción “Modificar Contenido, Agregar temas”, el proceso consiste en proporcionar al sistema los elementos necesarios para generar un nuevo tema (audio, video, texto e imagen), el sistema verifica y valida estos elementos y finalmente genera el nuevo tema.

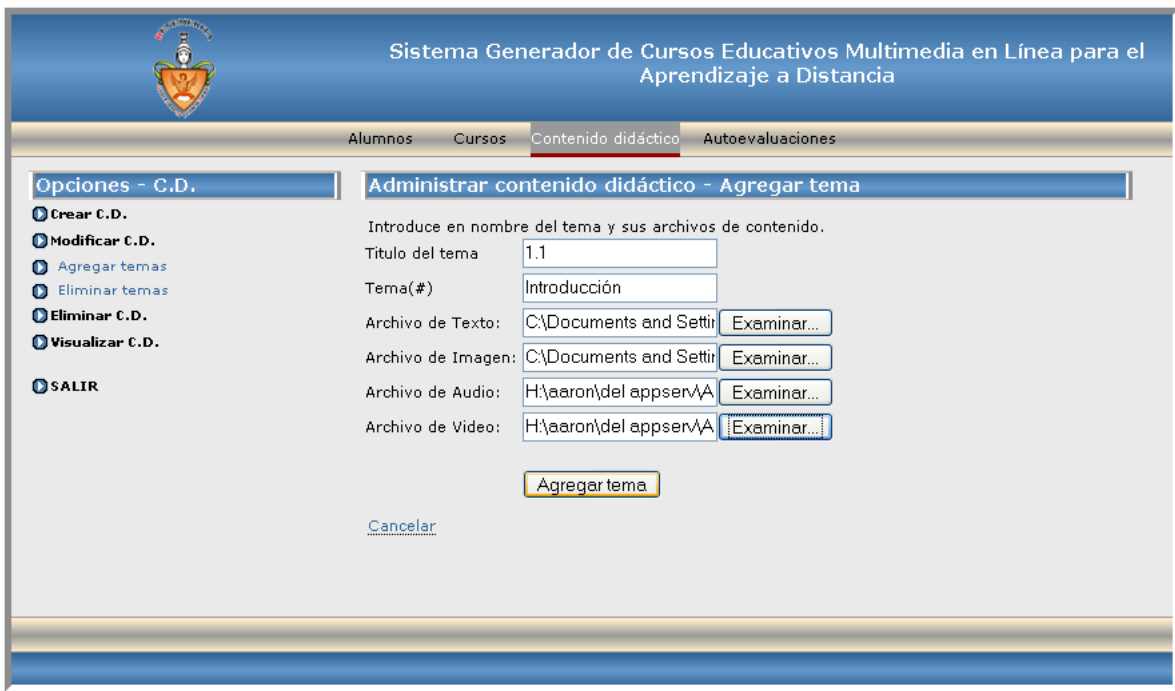


Figura 6.12: Clase de navegación Modificar contenido didáctico (agregar tema – selección de archivos).

La operación “Eliminar tema”, como es de esperarse, elimina temas creados anteriormente desde las operaciones ”Agregar temas ” o “Crear contenido”, esto se hace mediante la selección de una lista de los temas a eliminar (figura 6.13)

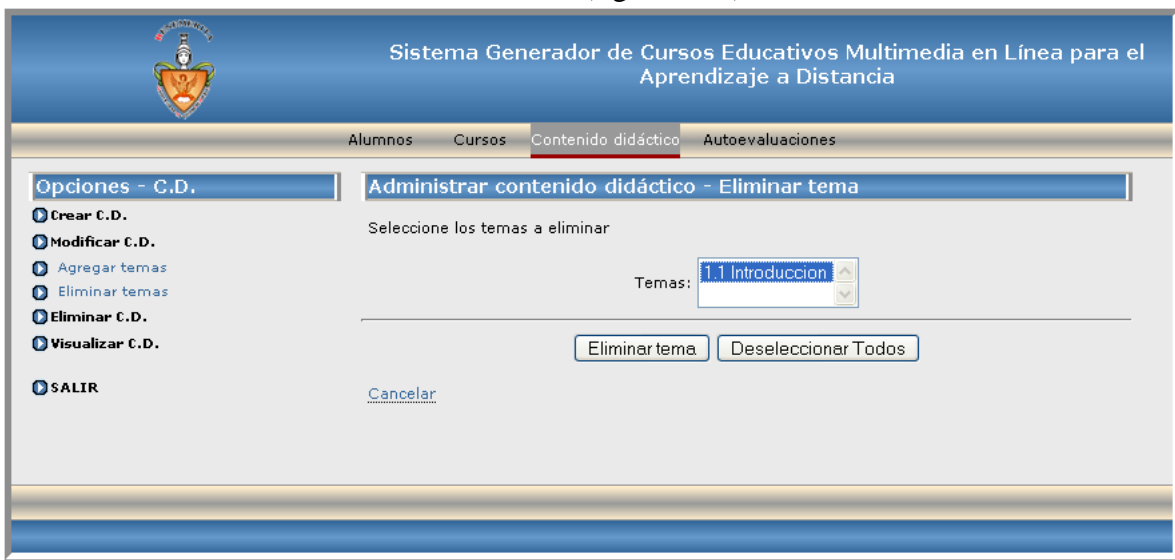


Figura 6.13: Clase de navegación Modificar contenido didáctico (eliminar tema – selección de temas).

Eliminar Contenido

Esta opción se procesa desde el momento en que se elige curso a procesar, y se presiona el botón “Aceptar”, con esto se le indica al sistema que elimine todo el contenido didáctico del curso elegido.

Visualizar Contenido

Para visualizar el contenido didáctico, después de elegir la materia, se elige de clase de navegación (un temario), el tema que se desee visualizar (figura 6.14), para que el sistema presente la pagina Web de contenido didáctico del tema elegido (figura 6.15).

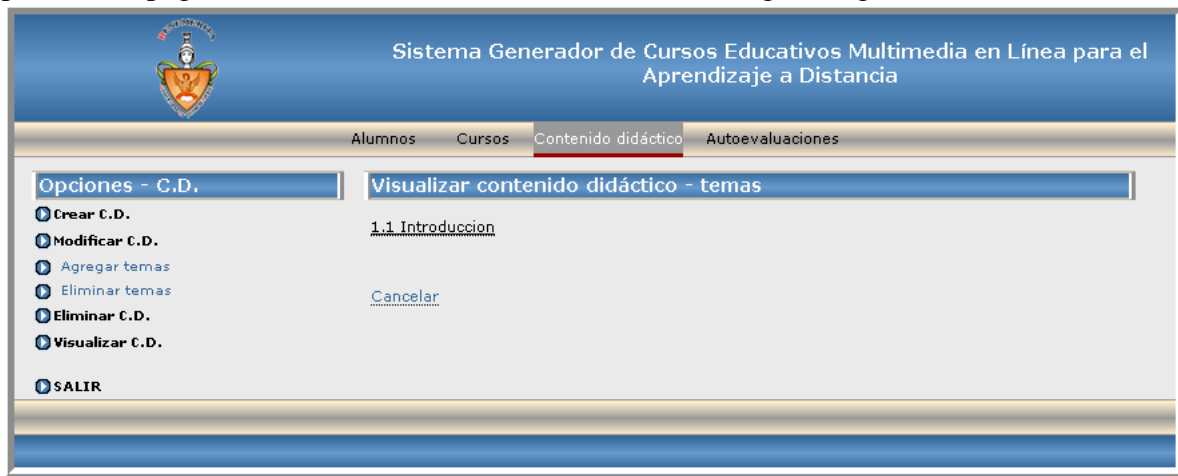


Figura 6.14: Selección de tema a visualizar (Contenido didáctico, Visualizar)

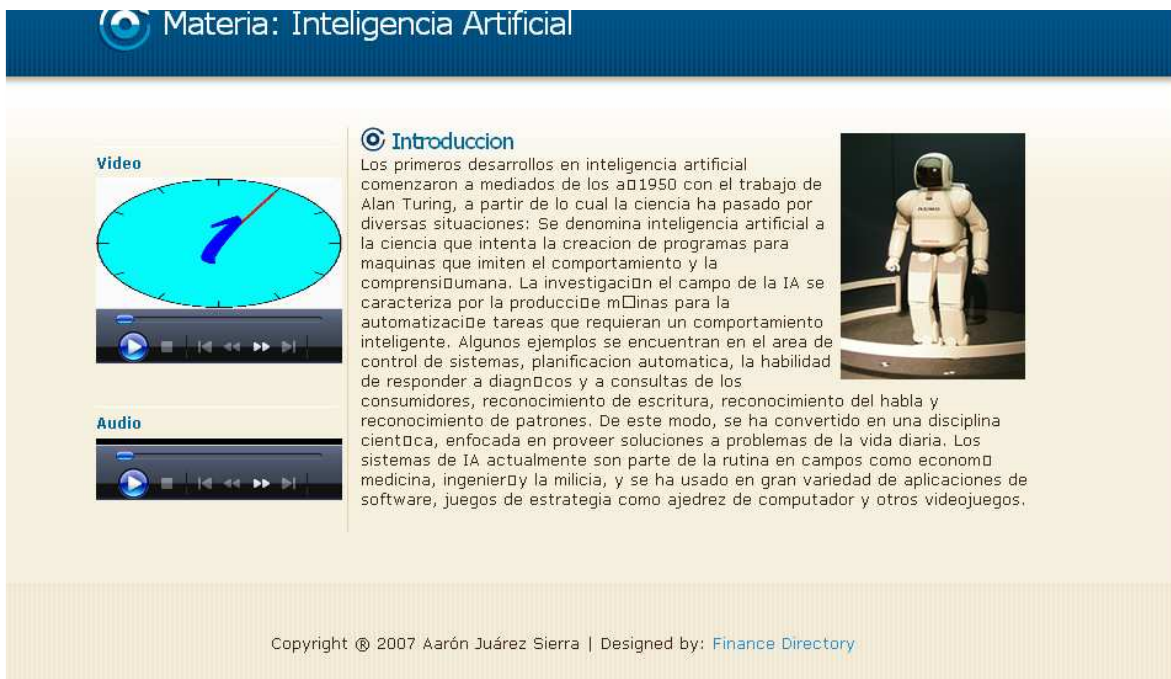


Figura 6.15: Ejemplo de pagina Web de contenido didáctico generada por el sistema

6.3.1.4 Operaciones sobre autoevaluaciones

Las operaciones sobre Autoevaluaciones: Crear, Modificar, Eliminar y Visualizar autoevaluaciones empiezan con una clase de navegación en donde se elije un curso y el número de evaluación, en la figura 6.16, se muestra dicha clase de navegación para la elección de un curso y numero de evaluación para la operación “Modificar autoevaluaciones, Agregar preguntas”.

Las autoevaluaciones son procesadas para su modificación, creación y visualización por medio de preguntas de opción múltiple.

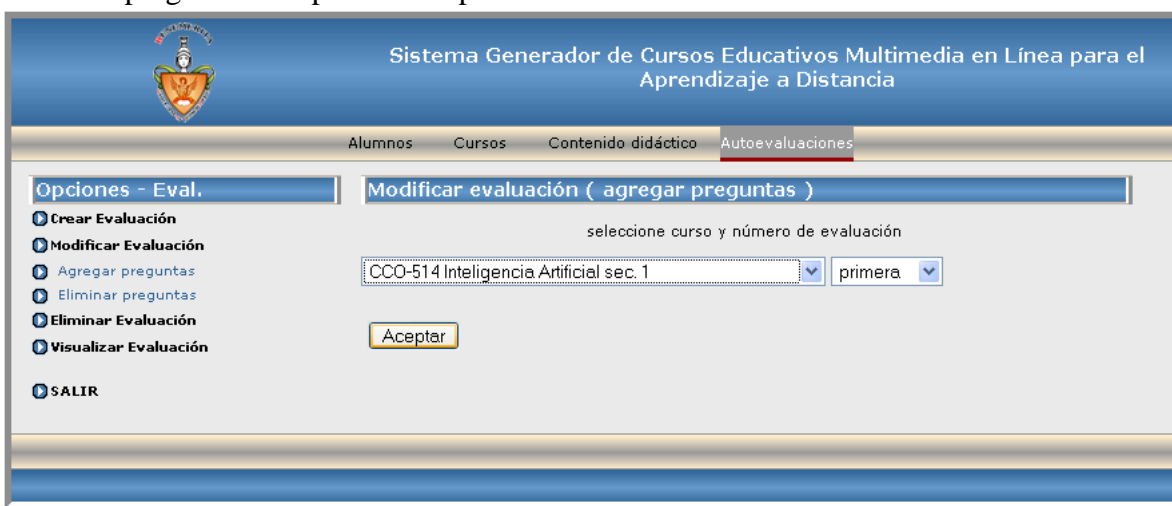


Figura 6.16: Clase de navegación Modificar autoevaluaciones (agregar pregunta – selección de curso y evaluación).

Crear evaluación

Esta opción, permite generar autoevaluaciones para el curso seleccionado previamente, el paso posterior es comenzar con la creación de preguntas, las cuales son generados a partir de elementos básicos como son la pregunta, las posibles respuestas y la elección de la respuesta correcta, ingresados como texto dentro de un formulario, estas son enviadas al sistema para su procesamiento, generando una pregunta nueva para la evaluación, este ultimo paso (generar pregunta), se describe a detalle en la operación “Modificar autoevaluación, Agregar preguntas”, por ser en esencia el mismo proceso.

Modificar evaluación

Esta opción consta de dos operaciones: “Agregar preguntas y “Eliminar preguntas”. El la figura 6.17 se observa la representación de la clase de navegación para la opción “Modificar Evaluación, Agregar preguntas”, el proceso consiste en proporcionar al sistema los elementos necesarios para generar una nueva pregunta (pregunta, respuestas, y la elección de la respuesta correcta), el sistema verifica y valida estos elementos y finalmente genera y agrega una nuevo pregunta a la evaluación.

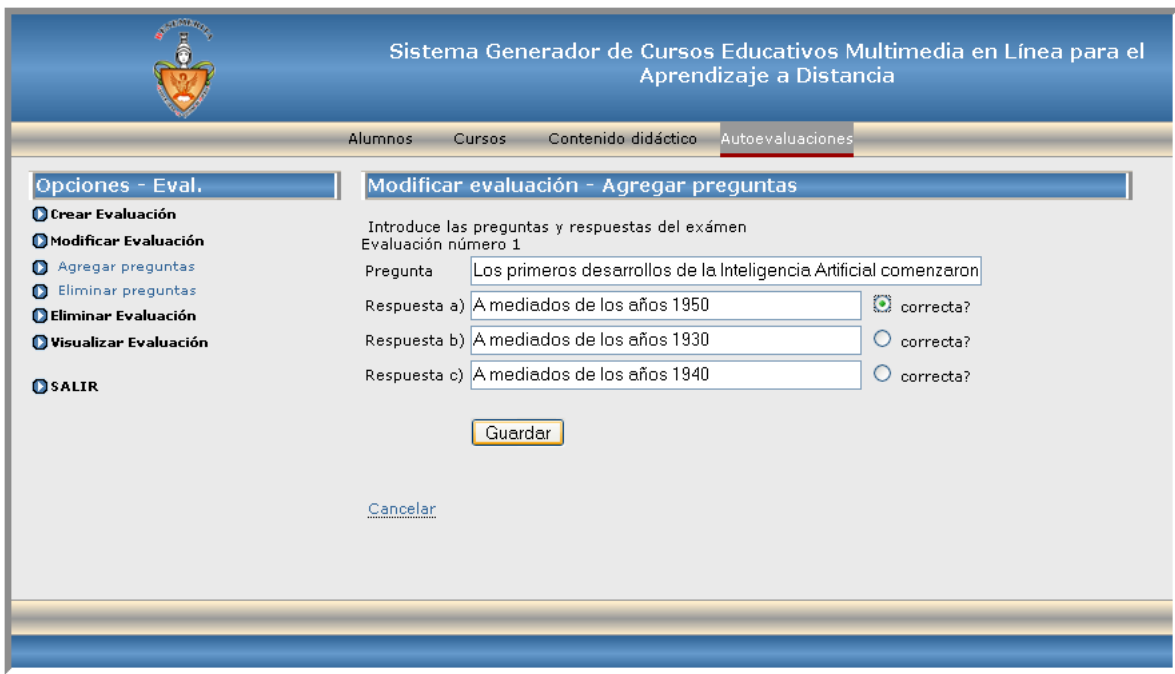


Figura 6.17: Clase de navegación Modificar autoevaluaciones (Agregar pregunta – datos pregunta).

La operación “Eliminar preguntas”, como es de esperarse, elimina preguntas creadas anteriormente desde las operaciones ”Agregar preguntas ” o “Crear evaluación”, esto se hace mediante la selección de una lista de las preguntas a eliminar (figura 6.18)

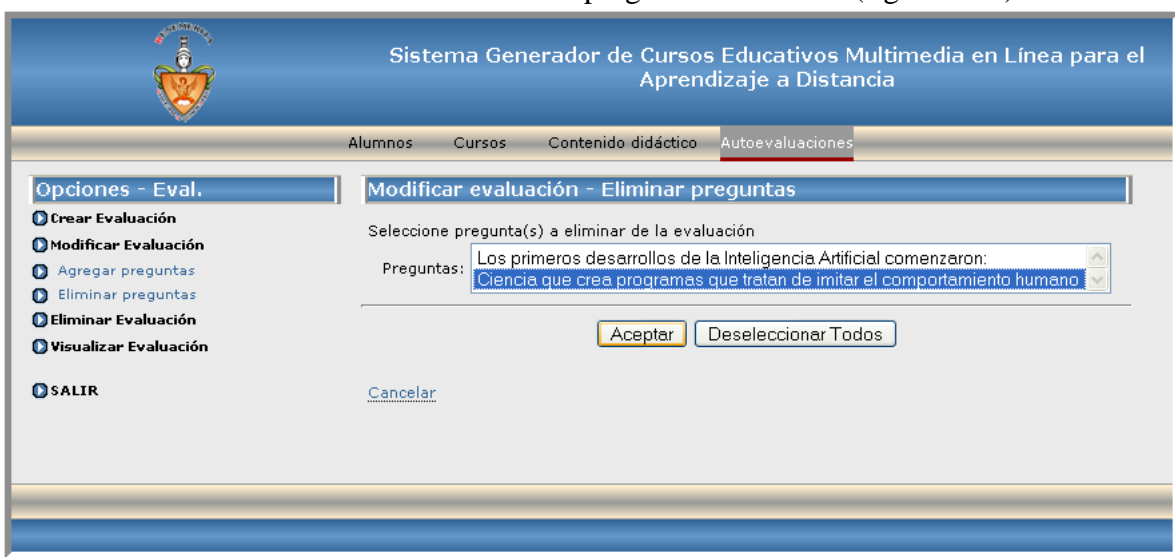


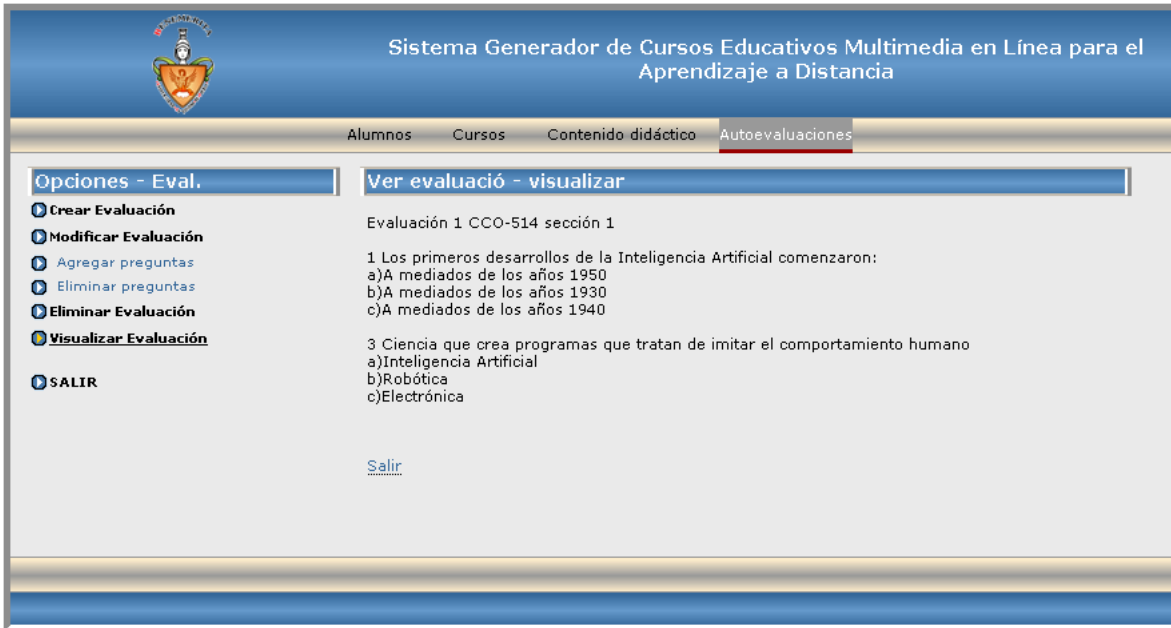
Figura 6.18: Clase de navegación Modificar evaluaciones (Eliminar preguntas – selección de preguntas).

Eliminar Evaluación

Esta opción se procesa desde el momento en que se elige curso y la evaluación; al presionar el botón “Aceptar” se le indica al sistema que elimine la evaluación del curso elegido.

Visualizar Evaluación

Para visualizar la evaluación, después de elegir el curso y el número de evaluación, el sistema presenta la pagina Web de la evaluación elegida (figura 6.19).



The screenshot shows a web interface for a distance learning system. At the top, there is a logo on the left and the title 'Sistema Generador de Cursos Educativos Multimedia en Línea para el Aprendizaje a Distancia' on the right. Below the title is a navigation menu with 'Alumnos', 'Cursos', 'Contenido didáctico', and 'Autoevaluaciones'. The 'Autoevaluaciones' section is active, showing a sidebar with options: 'Crear Evaluación', 'Modificar Evaluación', 'Agregar preguntas', 'Eliminar preguntas', 'Eliminar Evaluación', 'Visualizar Evaluación', and 'SALIR'. The main content area displays 'Ver evaluació - visualizar' and 'Evaluación 1 CCO-514 sección 1'. It lists two questions: '1. Los primeros desarrollos de la Inteligencia Artificial comenzaron:' with options a) A mediados de los años 1950, b) A mediados de los años 1930, c) A mediados de los años 1940; and '3. Ciencia que crea programas que tratan de imitar el comportamiento humano' with options a) Inteligencia Artificial, b) Robótica, c) Electrónica. A 'Salir' link is at the bottom.

Figura 6.19: Visualización de evaluación generada por el sistema

6.3.2. Usuario alumno

En la figura 6.20 se observa la representación de la clase de navegación principal para el usuario alumno. En la figura se pueden apreciar las opciones u operaciones: Realizar autoevaluaciones, Calificaciones, Contenido didáctico y Contacto profesores.

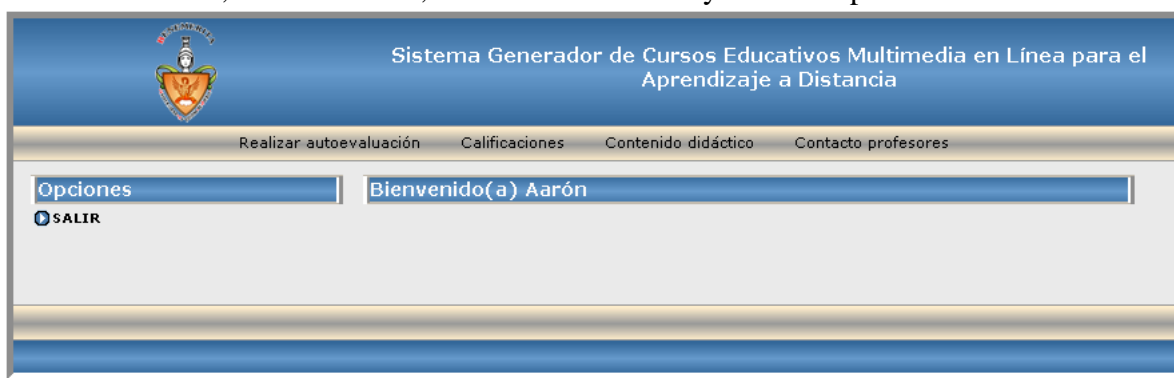


Figura 6.20: Clase de navegación principal del SGCAD (usuario alumno)

6.3.2.1. Realizar autoevaluaciones

Esta operación comienza con la elección del curso y el número de evaluación a realizar (figura 6.21), para que finalmente, el alumno conteste las preguntas de la evaluación (figura 6.22).

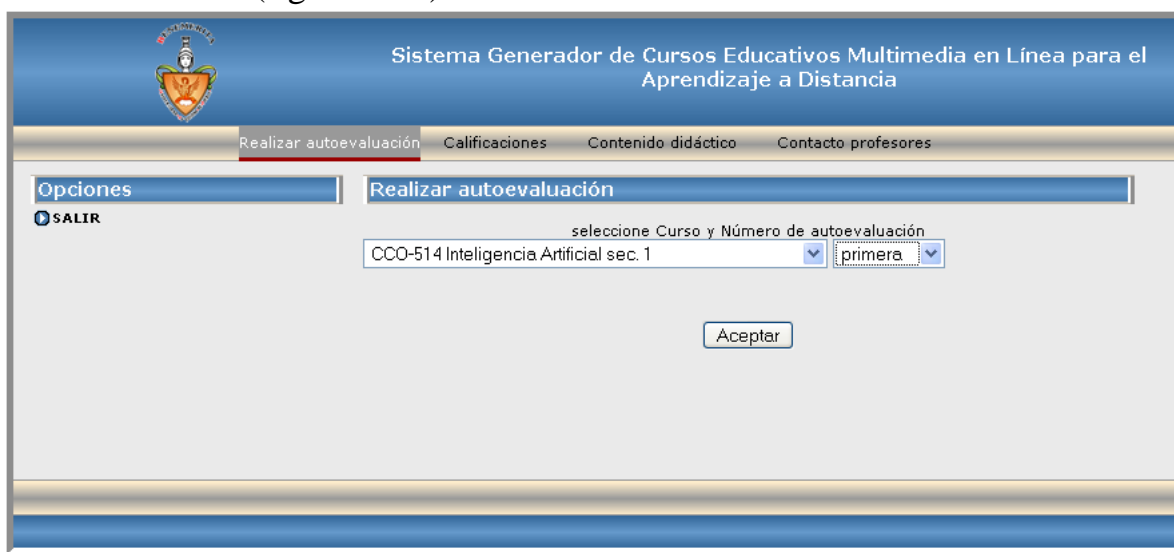


Figura 6.21: Clase de navegación Realizar autoevaluación (seleccionar curso y numero de evaluación).

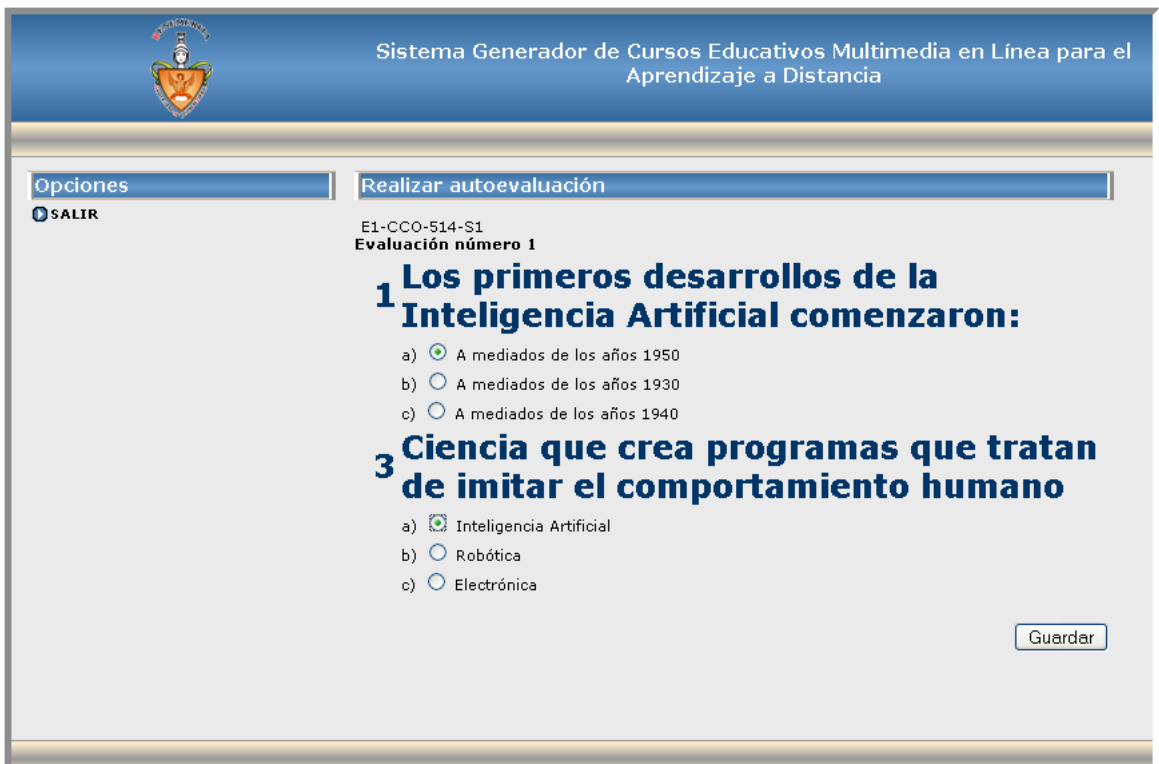


Figura 6.22: Clase de navegación Contestar autoevaluación.

6.3.2.2. Calificaciones

Esta operación muestra los resultados (calificaciones) de las evaluaciones y el estado en el que se encuentran (figura 6.23), de un curso y numero de evaluación elegidos previamente.

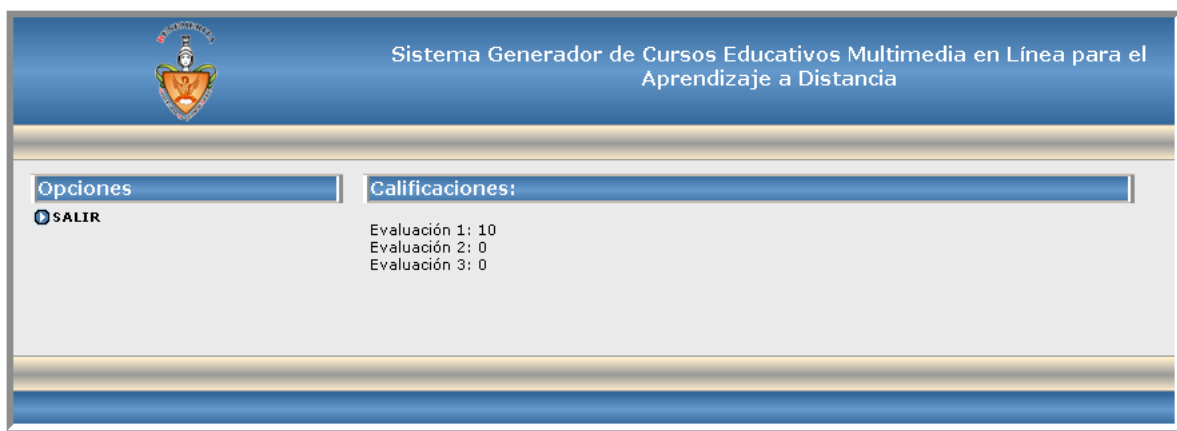


Figura 6.23: Clase de navegación Calificaciones.

6.3.2.3 Contenido didáctico

Esta opción es la misma operación presentada en la sección 6.3.1.3., “Visualizar contenido”, la diferencia es que en este caso la visualización de contenido es para un usuario alumno.

6.3.2.4 Contacto profesores

Esta operación permite enviar un correo electrónico a un profesor, seleccionando el curso que este imparte (figura 6.24), después dentro de otra clase de navegación se presenta la confirmación de envío de correo electrónico, junto con los datos de la materia y del profesor que la imparte (figura 6.25), finalmente al aceptar la confirmación, se procesa la redacción del correo electrónico dentro del cliente de correo definido por el sistema operativo del equipo de computo del usuario alumno.

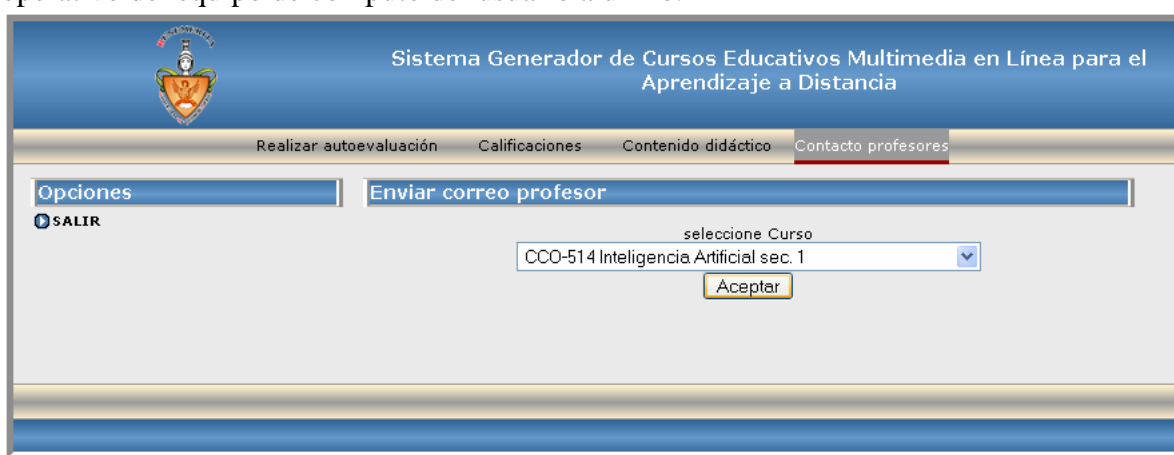


Figura 6.24: Clase de navegación Seleccionar curso del profesor a enviar correo electrónico.

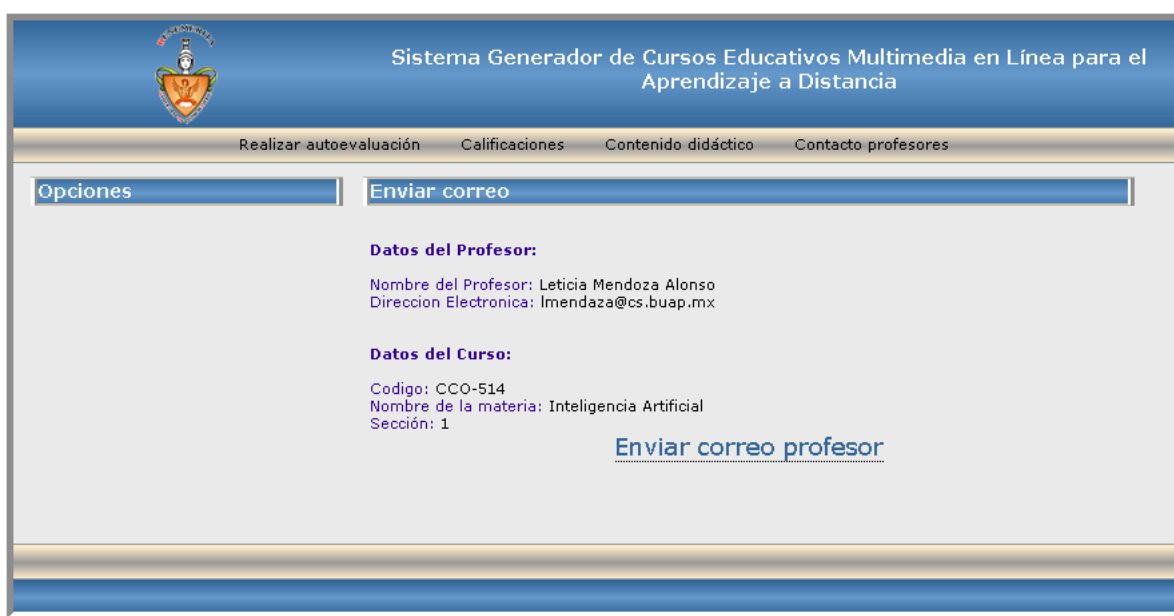


Figura 6.25: Clase de navegación Confirmación envío correo electrónico.

Capítulo 7

Conclusiones

Para finalizar, se presentan las conclusiones obtenidas del desarrollo de esta tesis, cuyo objetivo principal en un principio fue realizar un modelo en UML de una aplicación Web para el aprendizaje a distancia de los “Sistemas Operativos Distribuidos” usando PHP y MySQL, dicho objetivo mejoro como inicialmente se había planteado; es decir el objetivo principal consistió en el desarrollo de un sistema, que no solo se limitara a la enseñanza de un solo tópico, sino que permitiera obtener una aplicación Web para la generación de cursos educativos, incorporando dentro de los mismos elementos multimedia (audio, imágenes y video), para el aprendizaje a distancia de cualquier tópico impartido dentro de la Facultad de Ciencias de la Computación de la BUAP. En torno a este nuevo objetivo se concluye lo siguiente:

- Se realizó el análisis y diseño del Sistema Generador de Cursos. Resultado de este trabajo se obtuvo: la especificación de requisitos, un modelo lógico-conceptual, un modelo de navegación y una colección de escenarios Web bajo la metodología UWE de Ingeniería de Software para el desarrollo de aplicaciones Web usando UML. Se incluye también un anteproyecto de implementación (Apéndice A), en el que se presentan algunos lineamientos generales para la futura implementación del SGCAD propuesto. Dicho anteproyecto consiste en una serie de sugerencias en relación a la conformación del equipo de desarrollo, una planificación preliminar del proceso de implementación, sugerencia de lenguaje de programación, de servidor de base de datos y de servidor Web, propuestas de herramientas de software apropiadas y un mecanismo de prueba para el modelo. Con lo realizado se cubrió el 100% de los objetivos planteados en el desarrollo de la tesis.
- En la realización de cada uno de los diagramas del Sistema Generador de Cursos aquí presentado se utilizó el Lenguaje Unificado de Modelado (UML) en el marco de la Ingeniería Web Basada en UML (UWE). UWE es una propuesta enfocada al modelado de aplicaciones Web basada en la extensión de la semántica del UML mediante la utilización de estereotipos. Tomando como referencia base a UWE se definió un perfil UML (extensión) basado en estereotipos adecuados para el modelado del Sistema Generador de Cursos. Con el perfil UML definido se logró la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso las aplicaciones Web.
- La aportación de este trabajo se puede destacar en la generación de un documento de tesis, como apoyo a desarrolladores de software o estudiantes interesados en la generación de aplicaciones Web enfocadas sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso

de desarrollo de una aplicación Web, en la constante búsqueda de hacer cada día mejores productos, apegándose a los lineamientos establecidos por la ingeniería de software para la calidad de los productos.

- La generación de una aplicación Web, que constituya una herramienta de apoyo a la educación presencial dentro de la Facultad de Ciencias de la Computación.

Apéndice A

En el presente apéndice se presenta un anteproyecto de software de referencia para la implementación del Sistema Generador de Cursos Educativos Multimedia en Línea Para el Aprendizaje a Distancia. En este anteproyecto se incluyen propuestas a las cuestiones que se deben tomar en cuenta en el desarrollo de las aplicaciones de software, en este caso de las aplicaciones Web.

A.1 Sugerencia de lenguaje de programación, servidor de base de datos y servidor Web

El proceso de desarrollo de una aplicación Web, con respecto a la implementación, se debe tomar en consideración la correcta elección de estos tres elementos implantados por naturaleza a cualquier aplicación Web. Este apartado presenta una propuesta para cada uno de ellos la cual se plantea como la alternativa adecuada para el Sistema. La sugerencia se realizó con base a la experiencia personal en el desarrollo de aplicaciones web y no en un análisis comparativo con otras suites de desarrollo.

A.1.1 PHP

PHP significa PHP: Pre-procesador de Hipertexto (PHP: Hypertext Preprocessor). PHP es un lenguaje de scripting embebido en HTML. Mucha de su sintaxis es tomada de C, Java y Perl con un par de características adicionales únicas y específicas de PHP. El propósito del lenguaje es permitir que los desarrolladores Web escriban páginas generadas dinámicamente con rapidez. Existen diferentes versiones de PHP. PHP 4 es la generación actual más estable de PHP, aunque ya se encuentra disponible la versión 5. PHP ha sido diseñado, desde su origen, para ser un lenguaje más seguro para escribir programas, a diferencia de Perl o C y, con la correcta selección de las opciones de configuración, del servidor de base de datos y el servidor Web, es posible conseguir una combinación equilibrada de libertad y seguridad que requieren las aplicaciones que operan en un ambiente naturalmente inseguro como lo es la Web.

La característica primaria de PHP es la generación dinámica de páginas. Pero en si, existen un par de razones muy importantes que justifican la popularidad y éxito de este lenguaje. La primera es el soporte de PHP para operar con una gran variedad de manejadores de bases de datos, como por ejemplo MySQL, Interbase, Oracle, PostgreSQL, Informix, entre muchos otras. La segunda es que PHP es multiplataforma, ya que funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server o Apache) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra. Además de esto, PHP es libre, por lo que es un lenguaje de fácil acceso para todos. Esta información y más se encuentra disponible en la pagina oficial www.php.net. Estas características han convertido a PHP en el lenguaje más utilizado actualmente en el

desarrollo de aplicaciones Web, de ahí que se proponga su utilización en la implementación del SGCAD.

A.1.2 MySQL

MySQL es un sistema de administración de bases de datos (SQL) desarrollado también bajo la filosofía de código abierto y es uno de los más populares en el desarrollo de aplicaciones Web. Entre las características disponibles en las últimas versiones (la versión 4 y la versión 5) se pueden destacar las siguientes: capacidad de manejar un amplio subconjunto del lenguaje SQL estándar, disponibilidad en gran cantidad de plataformas y sistemas, proporciona conectividad confiable y segura, permite la ejecución de transacciones y el manejo de claves foráneas, permite replicación y provee la capacidad de realizar búsquedas e indexaciones de campos de texto, entre otras características más. Además, MySQL es el servidor de base de datos más comúnmente utilizado con el lenguaje PHP. Es por esta robustez que, según las cifras del fabricante (MySQL AB), existen más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos. Esta información y más se encuentra disponible en la página oficial dev.mysql.com. Estas características de MySQL la hacen una excelente alternativa para que se utilice en la implementación del SGCAD.

A.1.3 Apache

El Apache es un servidor Web código abierto multiplataforma. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches aplicados al servidor de NCSA (National Center for Supercomputing Applications, Centro Nacional de Aplicaciones de Supercómputo). Era, en inglés, “a patchy server” (un servidor parchado). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta, entre muchas otras, las siguientes características: utiliza mensajes de error altamente configurables, maneja bases de datos de autenticación, sus módulos se cargan de manera dinámica, soporta una gran cantidad de lenguajes (entre ellos PHP) y tiene un alto desempeño bajo concurrencia de operaciones. Además, en sus versiones 2.x, el núcleo de Apache tiene varias mejoras con respecto al núcleo de apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6. Esta información se encuentra disponible en la página oficial de Apache www.apache.org. En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft). Debido a esto, y su excelente desempeño cuando se utiliza juntamente con PHP y MySQL, se sugiere utilizar el servidor Web Apache en la implementación del SGCAD.

A.2 Herramientas para la implementación de aplicaciones Web

Una vez definidas las tecnologías a utilizar en la implementación de una aplicación Web, se procede a seleccionar las herramientas más adecuadas para el desarrollo de la misma, de la mejor manera posible. Es por esta razón que a continuación se comentan algunas sugerencias de herramientas útiles en el desarrollo de aplicaciones Web, enfocadas en el SGCAD.

A.2.1 Editor de Páginas Web: Macromedia Dreamweaver

En la implementación del Sistema Generador de Cursos, el uso de esta herramienta se propone para el diseñador Web y de los desarrolladores Web. Dreamweaver es la herramienta más utilizada en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Esta herramienta cumple perfectamente el objetivo de diseñar páginas con aspecto profesional de gran funcionalidad, y soporta gran cantidad de tecnologías como lo son las hojas de estilo, capas, DHTML, JavaScript, XML, ASP, PHP, entre varias más. La gran fortaleza de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que las rutinas de este programa (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en JavaScript-C lo que le ofrece una gran flexibilidad. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de JavaScript que hace que sea un programa muy fluido, y que permite que desarrolladores y editores Web puedan hacer extensiones para el programa. Además de estas características, tiene las funciones típicas de un editor de código fuente para la Web: un administrador de sitios para agrupar los archivos según el proyecto al que pertenezcan, un cliente FTP integrado que permite subir los archivos editados inmediatamente al sitio en Internet y la función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP. La versión libre de este software es el D4L y funciona sobre plataforma Linux.

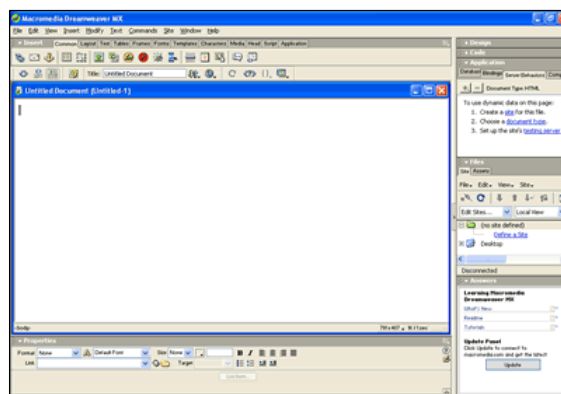


Figura A.1: Interfaz Macromedia Dreamweaver

A.2.2 Editor UML: Rational Rose

En la implementación del CCV, el uso de una herramienta de este tipo se propone exclusivamente para el Arquitecto de Software. Rational Rose es una herramienta CASE que tiene como eje de su funcionamiento el Proceso Unificado de Desarrollo de Software, sin embargo, en virtud que el modelado de este CCV no se realizó estrictamente bajo esta metodología su uso en el proceso de implementación se restringe como un editor de UML para que el Arquitecto de Software pueda tener acceso al modelo del CCV que aquí se presenta y, de esta forma, poder supervisar la correcta implementación de las especificaciones del modelo. Ahora bien, puede utilizar otras herramientas de edición de UML como lo son Visual Paradigm, UML Studio, entre otros.

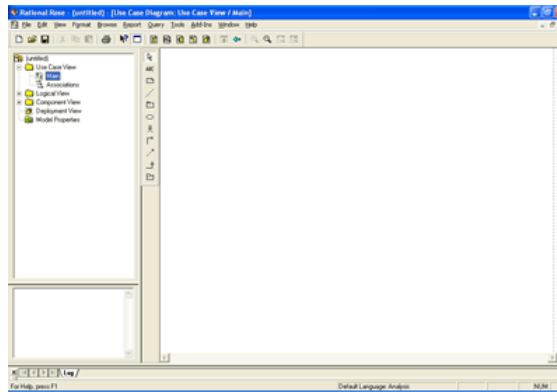


Figura A.: Interfaz de Rational Rose

Bibliografía

- [1] Sommerville, Ian; **“Ingeniería de Software”**. Pearson Educación, 6ta. Edición. México. 2002.
- [2] Silberchatz, Kort, Sudarshan; **“Fundamentos de Bases de Datos”**. Mc Graw Hill 4ta. Edición. 2000.
- [3] Elmasri and Navathe; **“Fundamentos de Sistemas de Bases de Datos”**. Addison Wesley 3ra. Edición. 2000.
- [4] Dra. Somodevilla García María Josefa; **“Tendencias de Bases de Datos”**. Ed: Fac. Cs. de la Computación, BUAP. 2005.
- [5] Dr. Pinto Avendaño David Eduardo; **“Fundamentos de Bases de Datos”**. Ed: Fac. Cs. de la Computación, BUAP. 2003.
- [6] Conallen Jim; **“Building Web applications with UML”**. Ed. Addison-Wesley. 2002.
- [7] Manchón Eduardo; **“Herramientas de diseño de aplicaciones Web: los personajes y escenarios”**. En Master-NET.net, Publicación sobre marketing, publicidad, ecommerce, y promoción en Internet. 2004.
- [8] Koch Nora, KRAUS Andreas. **“The expressive Power of UMLbased Web Engineering”**. Universidad de Munich Alemania, 2002.
- [9] Koch Nora, BAUMEISTER Hubert, HENNICKER Rolf, MANDEL Luis. **“Extending UML to model navigation and presentation in Web Applications”**. Universidad de Munich Alemania, Mayo 2002
- [10] Salinas Patricio, **“Tutorial de UML”**, DCC, Universidad de Chile, 2000, www.dcc.uchile.cl/~psalinas/uml
- [11] **Servicios Web**. <http://www.desarrolloweb.com/articulos/957.php>
- [12] **MySQL**. <http://www.mysql.com>.
- [13] **Php**. <http://www.php.com.mx/index.php?modulo=php>.
- [14] **MySQL, Apache y Php**. <http://es.tldp.org/Presentaciones/200102linuxcol/linux-servinternet/linux-serv-internet/apache-10.html>.
- [15] **Servicios Web**. http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_la_Web
- [16] Fernández Vilas Ana;” **Diagramas de Casos de Uso”**. <http://www-gris.det.uvigo.es/~avilas/UML/node25.html>