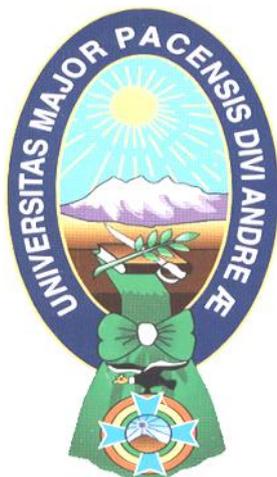


**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



# PROYECTO DE GRADO

**POSTULANTE : QUISPE APAZA TONY**  
**DOCENTE TUTOR : LIC. MARIO LOAYZA MOLINA**  
**DOCENTE REVISOR : MSC. LUCIO TORRICO DIAZ**

**TÍTULO DEL PROYECTO:**

**DESARROLLO DE UN AMBIENTE EDUCATIVO VIRTUAL**

**ÁMBITO ESPACIAL:**

**INSTITUTO SUPERIOR DE ELECTRÓNICA INFORMÁTICA Y  
TELECOMUNICACIONES “SANTO TORIBIO DE MOGROVEJO”**

**LA PAZ – BOLIVIA**

*DEDICATORIA:*

*A mis padres, por haberme dado la oportunidad de estudiar esta carrera. Y a Dios, por darme la inteligencia necesaria para aprovechar esta oportunidad...*

*Gracias.*

## RESUMEN

*El presente Proyecto de Grado titulado “Desarrollo de un Ambiente Educativo Virtual” ha sido desarrollado en el Instituto Superior de Electrónica Informática y Telecomunicaciones “Santo Toribio de Mogrovejo”. Esta es una institución que se especializa en la formación de estudiantes a nivel Técnico Superior.*

*Un Ambiente Educativo Virtual es un sistema de información para la gestión de cursos virtuales, que puede ser utilizado como un complemento para cursos presenciales, como en este caso. Un Ambiente Educativo Virtual es también una solución e-learning y como tal tiene 3 elementos que son la Plataforma, los Contenidos y los Sistemas de Comunicación*

*El Ambiente Educativo Virtual desarrollado es una Aplicación Web que recoge todas las características anteriormente mencionadas.*

*Para el desarrollo del proyecto se utilizó la metodología Ágil SCRUM, que propone un modelo de proceso incremental, basado en iteraciones y revisiones. También se utilizó en cada una de las 3 iteraciones la metodología UWE, que se especializa en el diseño de Aplicaciones Web.*

## ÍNDICE

	Página
CAPITULO I MARCO INTRODUCTORIO .....	8
1.1 INTRODUCCIÓN .....	8
1.2 ANTECEDENTES .....	9
1.3 PLANTEAMIENTO DEL PROBLEMA .....	11
1.4 OBJETIVOS DEL PROYECTO .....	12
1.4.1 Objetivo general .....	12
1.4.2 Objetivos específicos .....	12
1.4.3 Alcances y límites .....	12
1.5 JUSTIFICACIÓN DEL PROYECTO .....	13
1.5.1 Justificación Teórica .....	13
1.5.3 Justificación Económica .....	13
1.5.5 Justificación Social .....	13
1.6 METODOLOGÍA .....	13
CAPÍTULO II MARCO TEÓRICO .....	15
2.1 EDUCACION VIRTUAL .....	15
2.1.1 Tecnologías de la Información y la Comunicación .....	15
2.1.2 E-learning .....	16
2.1.3 Elementos del e-learning .....	16
2.2 METODOLOGÍAS ÁGILES .....	19
2.2.1 El Manifiesto Ágil .....	19
2.3 SCRUM .....	20
2.3.1 Elementos de un proyecto Scrum .....	20
2.3.2 Modelo de proceso .....	22
2.4 METODOLOGÍA UWE .....	24
2.4.1 Análisis de Requerimientos con Casos de Uso .....	25
2.4.2 Representación del Modelo Conceptual .....	26
2.4.3 Modelo de Navegación .....	26
2.4.4 Modelo de Presentación .....	30
2.5 APLICACIONES WEB .....	31
2.5.1 Consideraciones de diseño .....	31
2.5.2 Arquitectura Cliente-Servidor .....	33
2.5.3 Criterios de seguridad .....	34
2.5.4 Pruebas para Aplicaciones Web .....	36
2.5.5 Métricas para Aplicaciones Web .....	37
CAPÍTULO III DESARROLLO DEL PROYECTO .....	39
3.1 PRE-GAME .....	39
3.1.1 Recopilación de requerimientos .....	39
3.1.2 Definición del cronograma de trabajo .....	40
3.1.3 Análisis de riesgos .....	40
3.1.4 Herramientas de desarrollo .....	42
3.1.5 Análisis de costos .....	42
3.2 GAME .....	43
3.2.1 Primera Iteración .....	44
3.2.2 Segunda Iteración .....	45
3.2.3 Tercera Iteración .....	45

3.3 POST-GAME.....	46
3.3.1 Pruebas de la aplicación Web .....	46
3.3.2 Políticas de seguridad .....	57
3.3.3 Métricas de la aplicación Web.....	58
3.3.4 Diseño de la ayuda .....	60
3.3.5 Implementación .....	60
CAPÍTULO IV  MODELADO DEL SISTEMA.....	62
4.1 MODELADO DE LOS REQUERIMIENTOS.....	62
4.2 MODELO CONCEPTUAL .....	74
4.3 MODELO DE NAVEGACIÓN .....	75
4.4 MODELO DE PRESENTACIÓN .....	77
4.5 MODELO ENTIDAD RELACIÓN .....	88
4.6 MODELADO DE LA BASE DE DATOS .....	90
CAPÍTULO V .....	91
5.1 CONCLUSIONES .....	91
5.2 RECOMENDACIONES.....	92
BIBLIOGRAFÍA .....	93
ANEXOS .....	96
ANEXO A  DIAGRAMA GANTT DEL PROYECTO.....	96
ANEXO B  DOCUMENTACIÓN .....	97



## ÍNDICE DE FIGURAS

	Página
Figura 1: Elementos del e-learning .....	18
Figura 2: Backlog del producto .....	21
Figura 3: Backlog del sprint.....	21
Figura 4: Ciclo de vida Scrum .....	24
Figura 5: Elementos de un Modelo de Casos de Uso .....	25
Figura 6: Representación de una clase en UML .....	26
Figura 7: Clase Navegación.....	27
Figura 8: A) Clase Índice y B) Su notación taquigráfica.....	28
Figura 9: A) Clase Visita Guiada y B) su notación taquigráfica .....	28
Figura 10: A) Clase Consulta y B) Sus notaciones taquigráficas .....	29
Figura 11: A) Clase Menú y B) Su taquigrafía .....	29
Figura 12: Clase de Presentación, con elementos del Modelo de Presentación .....	30
Figura 13: Servidor de páginas Web.....	33
Figura 14: Modelo de proceso del proyecto .....	39
Figura 15: Diagrama de casos de uso “registrarse como nuevo usuario”.....	62
Figura 16: Diagrama de casos de uso “ingresar al sistema” .....	63
Figura 17: Diagrama de casos de uso “leer anuncios administrativos” .....	64
Figura 18: Diagrama de casos de uso “configurar perfil de usuario” .....	65
Figura 19: Diagrama de casos de uso “utilizar los foros de un curso” .....	66
Figura 20: Diagrama de casos de uso “acceder a los contenidos de un curso” .....	67
Figura 21: Diagrama de casos de uso “acceder a los comunicados de un curso” .....	68
Figura 22: Diagrama de casos de uso “administrar usuarios” .....	69
Figura 23: Diagrama de casos de uso “administrar estudiantes” .....	70
Figura 24: Diagrama de casos de uso “administrar docentes” .....	71
Figura 25: Diagrama de casos de uso “administrar cursos” .....	72
Figura 26: Diagrama de casos de uso “administrar estilos” .....	73
Figura 27: Diagrama de clases del modelo conceptual.....	74
Figura 28: Modelo de navegación para un Administrativo .....	75
Figura 29: Modelo de navegación para un Docente .....	76
Figura 30: Modelo de navegación para un Estudiante.....	77
Figura 31: Interfaz de ingreso al AEV .....	77
Figura 32: Interfaz de registro para usuarios nuevos .....	78
Figura 33: Interfaz de registro para usuarios nuevo (segundo paso) .....	78
Figura 34: Interfaz de inicio.....	78
Figura 35: Interfaz para los anuncios administrativos .....	79
Figura 36: Interfaz de los cursos.....	79
Figura 37: Interfaz dentro de un curso.....	79
Figura 38: Interfaz de los comunicados .....	80
Figura 39: Interfaz de los contenidos.....	80
Figura 40: Interfaz de los foros.....	81
Figura 41: Interfaz dentro de un foro .....	81
Figura 42: Interfaz de la administración .....	82
Figura 43: Interfaz para administrar cursos .....	82
Figura 44: Interfaz para modificar un curso .....	82
Figura 45: Interfaz para adicionar estudiantes a un curso.....	83

Figura 46: Interfaz para adicionar un curso .....	83
Figura 47: Interfaz para administrar docentes .....	83
Figura 48: Interfaz para modificar docentes .....	84
Figura 49: Interfaz para adicionar un docente .....	84
Figura 50: Interfaz para administrar estudiantes .....	84
Figura 51: Interfaz para modificar estudiantes .....	85
Figura 52: Interfaz adicionar estudiantes .....	85
Figura 53: Interfaz administrar usuarios .....	85
Figura 54: Interfaz modificar usuarios .....	86
Figura 55: Interfaz para adicionar usuarios .....	86
Figura 56: Interfaz para administrar estilos .....	86
Figura 57: Interfaz para adicionar estilos .....	87
Figura 58: Interfaz para configuración de usuarios .....	87
Figura 59: Diagrama entidad relación .....	88
Figura 60: Diagrama Relacional de la Base de Datos AEV .....	90



## ÍNDICE DE TABLAS

	Página
Tabla 1: Áreas de seguridad en una Aplicación Web .....	34
Tabla 2: Backlog del producto .....	40
Tabla 3: Análisis de Riesgos .....	42
Tabla 4: Calculo del esfuerzo por cada iteración .....	43
Tabla 5: Backlog del primer Sprint .....	44
Tabla 6: Backlog del segundo Sprint .....	45
Tabla 7: Backlog del tercer Sprint .....	46
Tabla 8: Pruebas de unidad .....	51
Tabla 9: Caso de prueba “registrarse como nuevo usuario” .....	52
Tabla 10: Caso de prueba “ingresar al sistema” .....	52
Tabla 11: Caso de prueba “leer anuncios administrativos” .....	53
Tabla 12: caso de prueba “configurar perfil de usuario” .....	53
Tabla 13: Caso de prueba “utilizar los foros de un curso” .....	53
Tabla 14: Caso de prueba “acceder a los contenidos de un curso” .....	54
Tabla 15: Caso de prueba “acceder a los comunicados de un curso” .....	54
Tabla 16: Caso de prueba “administrar usuarios” .....	54
Tabla 17: Caso de prueba “administrar estudiantes” .....	55
Tabla 18: Caso de prueba “administrar docentes” .....	55
Tabla 19: Caso de prueba “administrar cursos” .....	56
Tabla 20: Caso de prueba “administrar estilos” .....	56
Tabla 21: Resultados de la implementación del sistema en diferentes configuraciones .....	57
Tabla 22: Descripción de casos de uso “registrarse como nuevo usuario” .....	63
Tabla 23: Descripción de casos de uso “ingresar al sistema” .....	63
Tabla 24: Descripción de caso de uso “leer anuncios administrativos” .....	64
Tabla 25: Descripción de caso de uso “configurar perfil de usuario” .....	65
Tabla 26: Descripción de caso de uso “utilizar los foros de un curso” .....	66
Tabla 27: Descripción de caso de uso “acceder a los contenidos de un curso” .....	67
Tabla 28: Descripción de caso de uso “acceder a los comunicados de un curso” .....	68
Tabla 29: Descripción de caso de uso “administrar usuarios” .....	69
Tabla 30: Descripción de caso de uso “administrar estudiantes” .....	70
Tabla 31: Descripción de caso de uso “administrar docentes” .....	71
Tabla 32: Descripción de caso de uso “administrar cursos” .....	72
Tabla 33: Descripción de caso de uso “administrar estilos” .....	73
Tabla 34: Diccionario de datos del Ambiente Educativo Virtual .....	89

# CAPITULO I MARCO INTRODUCTORIO

## 1.1 INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TICs) se han introducido en nuestra sociedad lenta y progresivamente, esto está afectando a prácticamente todos los campos de la sociedad; entretenimiento, comunicación, gobierno, etc. y la educación no es una excepción.

Una de las bases para la *Sociedad de la Información* es la educación, así las TICs forman un papel crucial en la formación de las nuevas generaciones. Estas tecnologías se presentan cada vez más como una necesidad en nuestra sociedad y una exigencia permanente para los estudiantes.

La educación virtual o e-learning, ambientes educativos virtuales, cursos virtuales, bibliotecas virtuales, etc. forman parte de estas tecnologías y son cada vez más utilizados por los estudiantes de nuestra sociedad.

Un Ambiente Educativo Virtual es un sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso. Originalmente diseñados para el desarrollo de cursos a distancia, vienen siendo utilizados como complementos para cursos presénciales. Estos sistemas funcionan generalmente en un servidor, para facilitar el acceso de los estudiantes a través de Internet.

El presente documento constituye el Proyecto de Grado para el desarrollo de un Ambiente Educativo Virtual en el Tecnológico I.S.E.I.T. Santo Toribio de Mogrovejo. El mismo consta de 6 capítulos. En el primer capítulo se hace una introducción al Proyecto de Grado, sus antecedentes, la problemática, los objetivos y justificaciones. En el segundo capítulo se explican los fundamentos teóricos sobre los cuales se desarrolló el proyecto. En el tercer capítulo se describen las etapas del desarrollo del proyecto, además de las metodologías y herramientas que se utilizaron, y los resultados de las pruebas. En el cuarto capítulo se explica esquemáticamente la arquitectura y el funcionamiento del sistema a través del modelado. Y finalmente en el quinto capítulo se muestran las conclusiones a las que se llegaron con el desarrollo del proyecto.

## **1.2 ANTECEDENTES**

El Instituto Superior de Electrónica, Informática y Telecomunicaciones (I.S.E.I.T.) “Santo Toribio de Mogrovejo”, es un centro educativo de enseñanza técnica a nivel técnico superior, pertenece a Fe y Alegría, bajo convenio iglesia-estado que tiene el apoyo del gobierno Vasco de España en cuanto a su equipamiento e infraestructura.

El instituto se encuentra ubicado en la avenida Montenegro N° 123, plan 145, entre calles 9 y 12 de la avenida Bolivia, zona Villa Adela de la ciudad de El Alto.

**Antecedentes Históricos.-** El I.S.E.I.T. Santo Toribio de Mogrovejo nace como consecuencia de la preocupación de Fe y Alegría respecto a las alternativas de formación de los bachilleres alteños, paceños y del área rural.

El área de formación técnica es uno de los pilares del desarrollo de un país. Fe y Alegría consciente de esta realidad, pone en marcha en 1997, la creación de un instituto que brinde una formación integral, técnica y humana a nivel superior a todos los bachilleres.

En febrero de 1999 y gracias a la colaboración de instituciones tanto bolivianas como extranjeras, se abren las puertas del I.S.E.I.T. Santo Toribio de Mogrovejo haciendo realidad la propuesta de continuidad de formación para bachilleres por parte de Fe y Alegría.

**Misión.-** *“Es una institución comprometida con el desarrollo local, regional y nacional; basada en equidad de género, valores, interculturalidad; formando profesionales técnicos con excelencia y capacidad creativa.”*

**Visión.-** *“Ser una institución modelo en educación técnica y tecnológica haciendo constante y sostenible la mejora de su proceso educativo para lograr profesionales técnicos comprometidos con el desarrollo nacional, con enfoque pro-ambientalista, altos valores y mejores competencias”.*

**La educación en el I.S.E.I.T. Santo Toribio de Mogrovejo.-** Esta institución se especializa en la formación técnica a nivel Técnico Superior, cuenta con las siguientes especialidades:

- *Técnico Superior en Sistemas de Telecomunicaciones.*- En esta especialidad el estudiante adquiere los conocimientos en el uso y mantenimiento de los sistemas de telecomunicaciones, telefonía fija, telefonía móvil, fibra óptica, sistemas de red satelital y antenas.
- *Técnico Superior en Sistemas Informáticos.*- En esta especialidad se adquiere una amplia y sólida base de conocimientos en el análisis, diseño, y programación de sistemas informáticos.
- *Técnico Superior en Sistemas de Regulación y Control.*- Este técnico debe ser capaz de desarrollar e implementar equipos e instalaciones de medida, control y regulación para maquinas, procesos y aplicaciones industriales. Además de coordinar y supervisar la ejecución y mantenimiento de los sistemas automáticos.

El tiempo de estudios es de 3 años, divididos en 6 semestres. Durante los primeros cuatro semestres se llevan materias comunes, luego de los cuales se obtiene un certificado de técnico medio en Electrónica-Infornática. En los últimos dos semestres la enseñanza se especializa según la especialidad elegida por el estudiante. Finalmente se realizan tres meses de pasantías y un examen o proyecto de grado para obtener el título en provisión nacional como Técnico Superior.

**Antecedentes del proyecto.**- En nuestro país, son contadas las instituciones educativas que cuentan hoy en día con un sistema de estas características, entre ellas se encuentra la Universidad Real con el Programa de Bachillerato Virtual, el Aula Virtual de la Universidad Católica Boliviana San Pablo, el Centro de Educación a Distancia de la Universidad Andina Simón Bolívar, la Pizarra-Boliviana y nuestra FCPN con su Aula Virtual, la mayoría de estas desarrolladas bajo la plataforma de libre distribución Moodle.

A nivel internacional se ha avanzado mucho en el área, y existen varias plataformas, entre ellas está la WebCT, que es la más utilizada en el ámbito universitario de los Estados Unidos, seguida a bastante distancia de la plataforma Edustance. Se está empezando a implantar con fuerza la plataforma de licencia libre Moodle. También se utiliza en varias universidades la plataforma de código abierto dotLRN.

Paralelamente se desarrollaron diversos estándares utilizables, como son el AICC (desarrollado por la industria de la aviación de EEUU), LTSC (organismo dependiente de la IEEE), IMS (del Global Learning Consortium), y el mayor utilizado SCORM (Shareable Content Object Reference Model o Modelo de Referencia para Objetos de Contenidos Intercambiables).

### 1.3 PLANTEAMIENTO DEL PROBLEMA

El poco uso de tecnologías adecuadas como un Ambiente Educativo Virtual (AEV), la falta de presupuesto y el poco conocimiento en el área de la educación virtual han ocasionado una obsolescencia en la formación de los estudiantes respecto al uso de nuevas tecnologías, también se ha observado que muchos estudiantes buscan cada vez mas información fuera del instituto y la asimilan mejor que en clases. Con todo esto se prevé que de persistir estos problemas, en unos años la calidad en la educación y el prestigio de la institución disminuirán debido al rápido avance tecnológico.

Debido a esto se identificaron los siguientes problemas específicos:

- ¿Qué se debe hacer para mejorar el uso de tecnologías educativas en el instituto?
- ¿Existe la infraestructura y los recursos necesarios para la implementación de un AEV en el instituto?
- ¿Qué consideraciones se debe tener para buscar la seguridad en el sistema?
- ¿Será posible cumplir con los estándares y las recomendaciones del e-learning, y a su vez satisfacer los requerimientos de la institución?
- ¿Se podrá satisfacer con un único sistema las necesidades de estudiantes, docentes y administrativos de contar con una herramienta para su uso en la educación?
- ¿Será posible utilizar este sistema como base para la elaboración de otros proyectos en el área?

Basado en lo anterior, además sumando la misión de la institución de *“formar profesionales técnicos con excelencia...”* y sobre todo la visión que tiene de *“ser una institución modelo en educación técnica y tecnológica haciendo constante y sostenible la mejora de su proceso educativo...”* me he propuesto responder con el presente proyecto de grado la siguiente cuestión<sup>1</sup>:

**¿Será posible desarrollar un sistema que contribuya a mejorar el proceso educativo en el Instituto Superior de Electrónica Informática y Telecomunicaciones “Santo Toribio de Mogrovejo”?**

---

<sup>1</sup> La problemática inicial propuesta en el Perfil de Proyecto de Grado fue modificada con el fin de adecuarla a los Objetivos propuestos

## **1.4 OBJETIVOS DEL PROYECTO**

### **1.4.1 Objetivo general**

Desarrollar un Ambiente Educativo Virtual en el Instituto Superior de Electrónica, Informática y Telecomunicaciones “Santo Toribio de Mogrovejo”, para facilitar el proceso de enseñanza y como un complemento a los cursos presenciales.

### **1.4.2 Objetivos específicos**

En consecuencia a lo anterior y como solución a los problemas específicos, se proponen los siguientes objetivos específicos:

- Desarrollar un sistema que mejore el uso de tecnologías educativas en el instituto.
- Diseñar la infraestructura necesaria para la implementación del proyecto
- Satisfacer con el sistema las necesidades de los estudiantes, docentes y administrativos
- Tener las consideraciones necesarias para buscar la seguridad en el sistema
- Cumplir con los estándares y las recomendaciones que existen en el e-learning
- Desarrollar un sistema que pueda ser usado como base para la elaboración de proyectos futuros

### **1.4.3 Alcances y límites**

Basado en los elementos del e-learning, que se explican en el Capítulo II, el proyecto tiene los siguientes alcances:

- Desarrollar una Plataforma Educativa para la Web que se encargue de gestionar a los usuarios, los cursos y sistemas de comunicación.
- Desarrollar un Módulo de Contenidos que administre los contenidos de cada curso y los ponga a disposición de los estudiantes.
- Desarrollar sistemas de comunicación asíncrona para su uso dentro del AEV que incluya comunicados, anuncios administrativos y foros para cada curso.

También se presentan las siguientes limitaciones:

- El sistema no está orientado a clases cien por ciento virtuales, pues el instituto no está en las condiciones de impartir este tipo de enseñanza.

- En consecuencia con lo anterior los sistemas de comunicación, los contenidos y la plataforma tendrán únicamente las características necesarias para los fines del sistema.
- Los contenidos de cada curso deberán ser elaborados por el docente de la materia, previo asesoramiento del administrador del sistema.

## **1.5 JUSTIFICACIÓN DEL PROYECTO**

### **1.5.1 Justificación Teórica**

Se justifica el estudio de la educación virtual, pues el proyecto permitirá:

- Adquirir experiencia, para el futuro desarrollo de cursos virtuales.
- Ser una base para la elaboración de proyectos similares en el área de la educación
- Mejorar los métodos tradicionales de enseñanza en la institución, en base a las conclusiones obtenidas.

### **1.5.3 Justificación Económica**

En el sentido económico el proyecto se justifica pues:

- Estudiantes y docentes podrán acceder a la información de sus materias desde cualquier lugar y en cualquier momento, ahorrando tiempo y dinero.
- Se prevé un aumento en la población estudiantil lo cual generara mayores recursos para la institución.

### **1.5.5 Justificación Social**

El Ambiente Educativo Virtual será de gran ayuda para la comunidad estudiantil, pues:

- Los **Estudiantes** podrán obtener una educación de mayor calidad.
- Los **Docentes** contarán con una herramienta muy útil para complementar su enseñanza.
- Los **Administrativos** tendrán un medio fácil de comunicación con los estudiantes.

## **1.6 METODOLOGÍA**

Para el desarrollo del proyecto se utilizó la metodología Ágil SCRUM, pues esta se adapta a las necesidades del proyecto, de poner principal énfasis en el producto, además de su flexibilidad y principalmente por su agilidad en el desarrollo de aplicaciones.

También se utilizó la metodología UWE (*UML-Based Web Engineering*) en las etapas de desarrollo de cada una de las iteraciones del proyecto, ya que esta metodología se especializa en el diseño de aplicaciones Web.

Una de las principales características de la metodología SCRUM es su adaptabilidad, en este sentido se tomaron en cuenta los siguientes aspectos:

- Como modelo de proceso se utilizó el incremental basado en iteraciones y revisiones, como propone SCRUM.
- Se utilizaron los elementos del SCRUM, que son el backlog del producto, el backlog del sprint y los incrementos.
- Como muchas metodologías, SCRUM define los roles para el equipo de trabajo, en este proyecto no se utilizaron estas recomendaciones, pues al tratarse de un proyecto de grado individual el equipo de trabajo se redujo a una sola persona.
- Existen algunos principios del SCRUM que sugieren la ejecución de reuniones diarias, reuniones semanales, reuniones antes y después de cada sprint, etc. Que por razones ya mencionadas en el anterior párrafo no se llevaron a cabo. Sin embargo para cumplir con el principio de colaboración con el cliente se efectuaron reuniones semanales con el cliente.

## **CAPÍTULO II MARCO TEÓRICO**

### **2.1 EDUCACION VIRTUAL**

#### **2.1.1 Tecnologías de la Información y la Comunicación**

Se denominan Tecnologías de la Información y la Comunicación (TICs) al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de la información, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética.

Las Tecnologías de la Información y Comunicación han permitido llevar la globalidad al mundo de la comunicación, facilitando la interconexión entre las personas e instituciones a nivel mundial, y eliminando barreras espaciales y temporales.

En la actualidad las Tecnologías de la Información y la Comunicación están sufriendo un desarrollo vertiginoso afectando a prácticamente todos los campos de nuestra sociedad. Esas tecnologías se presentan cada vez más como una necesidad en una sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de información se convierten en una exigencia permanente. [Rosario, 2007]

##### **2.1.1.1 Características de las TICs**

- **Inmaterialidad.-** Las TICs convierten la información, tradicionalmente sujeta a un medio físico, en inmaterial. Mediante la digitalización es posible almacenar grandes cantidades de información, en dispositivos físicos de pequeño tamaño.
- **Instantaneidad.-** Podemos transmitir la información instantáneamente a lugares muy alejados físicamente, mediante la denominada "autopista de la información" o Internet.
- **Interactividad.-** Las aplicaciones multimedia han sido desarrolladas como una interfaz amigable y sencilla de comunicación, para facilitar el acceso a las TICs de todos los usuarios. Una de las características más importantes de estos entornos es la interactividad. A diferencia de las tecnologías más clásicas (TV, radio) que permiten una

interacción unidireccional. El usuario de las TICs es por tanto, un sujeto activo, que envía sus propios mensajes y toma las decisiones sobre el proceso a seguir.

### **2.1.2 E-learning**

Se entiende por **e-learning** (o *electronic learning*) a la utilización de nuevas Tecnologías de la Información y la Comunicación (TICs) en la educación, como una nueva modalidad de enseñanza o como un complemento a los procesos de enseñanza tradicionales.

La educación a distancia ha creado las bases para el desarrollo del e-learning, el cual viene a resolver algunas dificultades en cuanto a tiempos, sincronización, asistencia, problemas típicos de la educación tradicional. [SA3, 2007]

El e-learning cubre un amplio grupo de aplicaciones y procesos, tales como el aprendizaje basado en la Web, aprendizaje basado en ordenadores, aulas virtuales, entrega de contenidos vía Internet, audio y video grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más. [SA3, 2007]

Entre las soluciones e-learning se encuentra un **Ambiente Educativo Virtual (AEV)** ó *Virtual Learning Environment (VLE)*, es un sistema de software diseñado para facilitar la gestión de cursos virtuales, permitiendo la administración y el desarrollo del curso. El sistema puede ser controlado por los docentes o administrativos y tiene como usuario final al estudiante. Originalmente fue diseñado para el desarrollo de cursos a distancia, pero actualmente viene siendo utilizado como complemento para cursos presenciales. [SA1, 2007]

### **2.1.3 Elementos del e-learning**

Una solución e-learning está conformada por tres elementos fundamentales: Plataforma, Contenidos y Sistemas de Comunicación.

#### **2.1.3.1 PLATAFORMA**

Se conoce como Plataforma de Teleformación o LMS (Learning Management System), es el núcleo alrededor del cual giran los demás elementos. Básicamente se trata de un software para servidores de Internet o Intranet que se ocupa de: [SA3, 2007]

- Gestionar los usuarios: inscripción, control de su aprendizaje e historial, generación de informes, etc.
- Gestionar y lanzar los cursos, realizando un registro de la actividad del usuario, resultados de los test, evaluaciones que realice, y accesos al material formativo.
- Gestionar los servicios de comunicación que son el apoyo al material online, foros de discusión, charlas, videoconferencia. Programarlos y ofrecerlos cuando sean necesarios.

### ***2.1.3.1 CONTENIDOS***

Los contenidos para e-learning pueden estar en diversos formatos, dependiendo de la materia tratada. La calidad de los contenidos es una condición necesaria, aunque no suficiente, para el éxito del programa formativo.

El diseño de los contenidos debe de ser realizado por expertos en metodología didáctica con el objetivo de que respondan a: [SA3, 2007]

- Adecuación a las necesidades y posibilidades del alumno.
- Calidad y cantidad de la información presentada.
- Interactividad.
- Estructura adecuada para su correcta asimilación.

Con la aparición de los estándares, a partir del año 2001, se garantizaba la independencia de los contenidos y las LMS, de forma que se cumplan ciertas especificaciones sobre las que basar el desarrollo de herramientas y contenidos: [SA3, 2007]

- **Accesibilidad.**- Independiente de la plataforma en la que estén los contenidos.
- **Interoperabilidad.**- El contenido puede ser usado en diferentes plataformas.
- **Reusabilidad.**- Los contenidos pueden ser utilizados una y otra vez en diferentes programas educativos.
- **Durabilidad.**- El contenido podrá utilizarse sin importar los cambios en la tecnología con la cual se elaboró.

Actualmente existen diversos estándares utilizables para los contenidos, como son el AICC (desarrollado por la industria de la aviación de EEUU), LOM (del IEEE), IMS (del Global

Learning Consortium), y el más utilizado SCORM (desarrollado por el departamento de defensa de los EEUU), que recoge muchas de las anteriores especificaciones. La principal ventaja de una estandarización es que posibilita la reutilización de los contenidos en diferentes plataformas.

### 2.1.3.1 SISTEMA DE COMUNICACIÓN

Las herramientas de comunicación en este entorno formativo constituyen otra pieza clave, ya que permiten la interacción entre los diferentes agentes del proceso de enseñanza-aprendizaje. Dicha interacción se concreta en la posibilidad de realizar trabajos en grupo, intercambiar experiencias, proporcionar apoyo por parte del tutor, resolución de dudas, etc.

Según que la comunicación sea en tiempo real o no, tenemos: [Foix y Zavando, 2002]

- **Comunicación síncrona.**- Un sistema sincrónico es aquel que ofrece comunicación en tiempo real entre los estudiantes y docentes. Por ejemplo, teléfono, chat, webcam, videoconferencia, pizarra electrónica.
- **Comunicación asíncrona.**- Los sistemas asincrónicos no ofrecen comunicación en tiempo real, son las que le dan al e-learning buena parte de su carácter. Por ejemplo, foros de debate, grupos de noticias, correo electrónico, y blogs.

Esquemáticamente los distintos componentes de una solución e-learning se pueden ver de la siguiente manera:

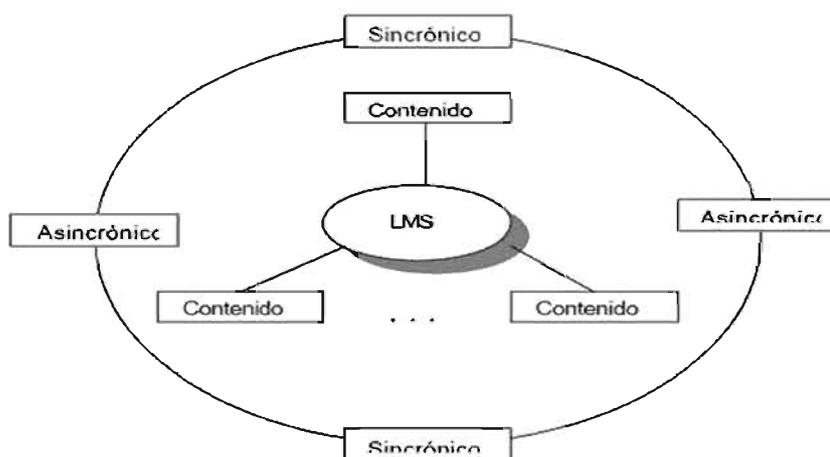


Figura 1: Elementos del e-learning  
Fuente: [Foix y Zavando, 2002]

## 2.2 METODOLOGÍAS ÁGILES

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término “*Ágil*” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. El punto de partida es el *Manifiesto Ágil*, un documento que resume la filosofía “*Ágil*”. [AISI, 2007]

### 2.2.1 El Manifiesto Ágil

El Manifiesto comienza enumerando los principales valores del desarrollo Ágil. Se valora: [Letelier y Penadés]

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- **Desarrollar software que funciona más que conseguir una buena documentación.** La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- **La colaboración con el cliente más que la negociación de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los

requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible.

## **2.3 SCRUM**

Scrum en el área de ingeniería del software es un Método Ágil para el desarrollo de proyectos. Toma su nombre así como los principios, de los estudios realizados por Hirotaka Takeuchi e Ikujiro Nonaka sobre nuevas prácticas de producción a mediados de 1980, quienes se basaron en el juego de Rugby Scrum, las características principales de este juego son el trabajo en equipo y la adaptación. [BSE, 2007]

Scrum surgió como modelo para el desarrollo de productos tecnológicos, pero también se emplea en entornos que trabajan con requisitos inestables y que además requieren rapidez y flexibilidad, estas situaciones son frecuentes en el desarrollo de determinados sistemas software. Scrum es una forma de desarrollo de carácter adaptable más que predictivo, además de ser fácilmente combinable con otros métodos. [AISI, 2007]

En un proyecto Scrum se consideran los siguientes aspectos:

- Desarrollo de software en etapas incrementales.
- Requiere de entregas de software terminado.
- Es fácil de aprender y además adaptable.
- Se enriquece con la experiencia del equipo de trabajo.
- La implementación de Scrum es de bajo riesgo y costo.

### **2.3.1 Elementos de un proyecto Scrum**

Esencialmente se puede identificar 3 elementos principales que se utilizan durante el proyecto: [AISI, 2007]

#### **2.3.1.1 Pila del producto (Backlog del producto)**

Es una lista de requisitos de usuario que se origina con la visión inicial del producto, va creciendo y evolucionando durante el desarrollo del proyecto, las características principales de esta pila son:

- Priorización de acuerdo a la importancia que le da el propietario del producto.
- Debe ser accesible para todos los miembros del equipo del proyecto.
- Todos pueden contribuir y aportar elementos
- El responsable directo es el propietario del producto.

Id	Prioridad	Módulo	Descripción	Est.	Por
1	Muy alta		Plataforma tecnológica	30	JM
2	Muy alta		Prototipos interfaz usuario	40	LR
3	Muy alta		Diseño de datos	40	LR
4	Alta	Trastienda	El operador define el flujo y textos de un expediente	60	JM
5	Alta	Trastienda	Selección del lenguaje de programación.	60	JM
...					
20	Alta	Inventario	Mantenimiento.	999.	XX

Figura 2: Backlog del producto  
Fuente: [AISÍ, 2007]

### 2.3.1.2 Pila del Sprint (Backlog del sprint)

Se define “Sprint” como una iteración que dura alrededor de 30 días. La pila del Sprint es una lista de los trabajos que debe realizar el equipo durante esta iteración para generar el incremento previsto, cuyas características principales son:

- Debe contener las funcionalidades que se van a realizar durante el Sprint.
- El equipo debe estar comprometido a realizar dichas funcionalidades.
- Se debe asignar tareas a los distintos miembros del equipo del proyecto.
- Se debe hacer una estimación de cada funcionalidad.

SPRINT		INICIO	DURACIÓN							
1		01-feb-06	20	X	J	V	L	M	X	J
				01-feb	02-feb	03-feb	06-feb	07-feb	08-feb	09-feb
Tareas pendientes				12	12	11	9	8	6	5
Horas de trabajo pendientes				176	164	144	111	93	76	50
PILA DEL SPRINT										
Prod. ID	Tarea	Tipo	Estado	Responsable						
1	Tecnología web: shtml, ajax, javascript?	Análisis	Terminada	Luis	24	20	14			
1	Tecnología req. Datos	Análisis	Terminada	Luis	8	8	8			
2	Protot. Interfaz cliente web	Prototipado	Terminada	Luis	16	16	16	10	5	
2	Protot. Interfaz trastienda	Prototipado	Terminada	Elena	16	8				
3	Diseño esquema XML	Análisis	Terminada	Elena	16	16	10	5		
3	Diseño Base de datos	Análisis	Terminada	Elena	16	16	16	16	8	
4	Marco de Interfaz y menú	Codificación	Terminada	Luis	8	8	8	8	8	8
4	Editor de flujo grama	Codificación	En curso	Elena	24	24	24	24	24	20
4	Editor de texto	Codificación	En curso	Luis	24	24	24	24	24	16
4	Editor XML	Codificación	En curso	Luis	8	8	8	8	8	8
...										
20	Pruebas de edición de flujo grama.	Pruebas	Pendiente	Elena	8	8	8	8	8	8
21	Pruebas de edición de expedientes	Pruebas	Pendiente	Elena	8	8	8	8	8	8

Figura 3: Backlog del sprint  
Fuente: [AISÍ, 2007]

### 2.3.1.3 Incremento

Es el resultado de cada Sprint, cuyas características principales son:

- Es parte del producto desarrollado en un Sprint.
- Debe estar en condiciones de ser usado.
- Es una funcionalidad.

### 2.3.2 Modelo de proceso

Scrum es un método de desarrollo muy simple, que requiere mayor trabajo, ya que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una Metodología Ágil, y como tal emplea la estructura incremental basada en iteraciones y revisiones. El ciclo de vida Scrum está compuesto de tres fases: pre-game, game y post-game. [AISI, 2007]

#### 2.3.2.1 PRE-GAME

Antes de llevar a cabo el desarrollo del proyecto, se especifica lo que se va a realizar en las iteraciones, además de la prioridad con la que se lo hará, esta fase consta de dos puntos destacables, que se describen a continuación:

a. **Planeación.**- Durante la planeación todos los miembros del equipo incluyendo el cliente contribuyen a la creación de una lista de características del sistema, para el análisis y la conceptualización del problema.

Las tareas que se realizan en esta primera etapa son:

- i. Recopilación de requerimientos para conformar el backlog del producto, priorizándolos de acuerdo a una evaluación del cliente.
- ii. Definición de las fechas de entrega de los sprints y sus funcionalidades.
- iii. Análisis de riesgos y controles apropiados para los riesgos.
- iv. Selección de las herramientas y de la infraestructura de desarrollo.
- v. Cálculo o estimación del costo de cada iteración.

**b. Arquitectura.-** El objetivo de esta etapa es diseñar cómo los elementos del backlog del producto serán puestos en ejecución. Esta fase incluye una revisión de la arquitectura del sistema y diseño de alto nivel.

En esta etapa se realizan las tareas de:

- i. Revisión de los ítems del backlog del producto.
- ii. Análisis de dominio para reflejar el nuevo contexto y requisitos del sistema.
- iii. Revisión de la arquitectura del sistema de acuerdo a los requisitos definidos.
- iv. Revisión del diseño de alto nivel.

#### 2.3.2.2 GAME

Una vez realizada la especificación correspondiente, se lleva a cabo la elaboración del proyecto con un continuo seguimiento a cargo del mismo grupo de desarrollo. En cada iteración del game se realizan las siguientes tareas:

**a. Planeación del Sprint.-** Antes de comenzar cada sprint o iteración, se lleva a cabo dos reuniones consecutivas, en la primera se refina y se prioriza nuevamente el backlog del producto, además de elegir las metas de la iteración. En la segunda reunión se deben considerar cómo alcanzar los requerimientos y crear el backlog del sprint.

**b. Desarrollo de Sprints.-** El trabajo generalmente se organiza en iteraciones de 30 días (o Sprints). El Sprint es el desarrollo de la nueva funcionalidad para el producto. Esta fase provee la siguiente documentación: Backlog del Sprint con las actividades realizadas, los responsables y la duración de cada actividad.

**c. Revisión del Sprint.-** Al final de cada iteración se lleva a cabo una reunión de revisión, en donde se presenta la nueva funcionalidad del producto, las metas incluyendo la información de las funciones, diseño, ventajas, inconvenientes y esfuerzos del equipo.

#### 2.3.2.3 POST-GAME

Luego de haber culminado todas las iteraciones, resta la revisión final, denominada según la metodología Scrum, el cierre:

a. **Cierre.**- En esta última etapa se realiza la preparación operacional, incluyendo la documentación final necesaria para la presentación. También en esta fase se debe realizar dependiendo del tipo de producto ya sea el entrenamiento del personal (usuarios) o el marketing para la venta del nuevo producto.



Figura 4: Ciclo de vida Scrum  
Fuente: [AISI, 2007]

## 2.4 METODOLOGÍA UWE

La metodología UWE (*UML-Based Web Engineering*) presentado por Koch y sus colegas, para el desarrollo de aplicaciones Web, está fundada en un entorno Orientado a Objetos utilizando para esto la notación “*ligera*” de UML. UWE proporciona guías para la construcción de modelos de forma sistemática, enfocándose en personalización y en estudio de casos de uso. Las actividades de modelado principales son el análisis de requerimientos, el diseño conceptual, el diseño de navegación y el diseño de presentación, y producen los siguientes modelos: [Koch, 2000]

- Modelo de Casos de Uso
- Modelo Conceptual
- Modelo de Navegación
- Modelo de Presentación

El Lenguaje de Modelado Unificado UML(*Unified Modeling Language*) es una herramienta lo suficientemente poderosa para cubrir todos los requerimientos que surgen cuando se modela una aplicación Web. Además, tiene la ventaja de ser un lenguaje de modelado bien documentado, que es de hecho un estándar industrial. UML puede ser visto como una familia de lenguajes de modelado, más que un lenguaje simple, si se consideran las extensiones “*ligeras*”. Por “*ligero*”, se entiende que puede ser fácilmente adaptado a otras herramientas de modelado y que no significa un gran impacto en el intercambio de formatos. Los *estereotipos* son un nuevo tipo de elementos de modelado definidos dentro del modelo basado en un tipo de elementos de un modelo existente. Un *valor etiquetado* es un par (etiqueta, valor) que permite adjuntar información arbitraria a cualquier elemento de modelado. Una *restricción* es una condición que permite especificar nuevas semánticas para un elemento de modelado.

#### 2.4.1 Análisis de Requerimientos con Casos de Uso

Para describir los requerimientos funcionales de una aplicación se puede usar un modelo de casos de uso. Este modelo describe un trozo de comportamiento de la aplicación sin revelar su estructura interna. [Koch, 2000]

El modelo de casos de uso está conformado por dos elementos de modelado principales, llamados casos de uso y actores. Un caso de uso es una unidad coherente de funcionalidad provista de aplicaciones que interactúan con uno o más actores externos de la aplicación. Un actor, es el rol que un usuario puede desempeñar con respecto a un sistema o una entidad, tales como otro sistema o una base de datos. Además, existen relaciones entre estos dos elementos, tales como las asociaciones entre actores y casos de uso, y las dependencias «*includes*» y «*extends*» entre casos de uso.



Figura 5: Elementos de un Modelo de Casos de Uso  
Fuente: [Koch, 2000]

## 2.4.2 Representación del Modelo Conceptual

El diseño conceptual está basado en el análisis de requerimientos del paso previo. Incluye a los objetos involucrados en la interacción entre el usuario y la aplicación, especificado en los casos de uso. Apunta a la construcción de modelos de clase con estos objetos, que intentan ignorar tanto como sea posible los caminos de navegación y los pasos de presentación. [Koch, 2000]

Los principales elementos usados para el modelo conceptual son las clases y asociaciones. Sin embargo, el poder del diagrama de clases es dado por una variedad de características adicionales que pueden ser usadas. Entre estas características están los nombres de asociación y los nombres de roles de asociación, la cardinalidad, diferentes formas de asociaciones soportadas por UML como agregación, herencia, composición y la clase asociación, todas estas representadas gráficamente utilizando notación de UML. Una clase es descrita por un nombre, atributos y operaciones que actúan sobre los atributos.

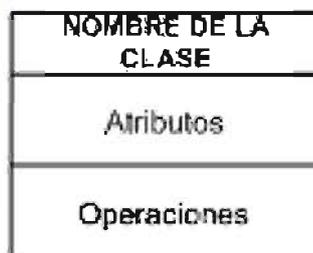


Figura 6: Representación de una clase en UML  
Fuente: [Koch, 2000]

## 2.4.3 Modelo de Navegación

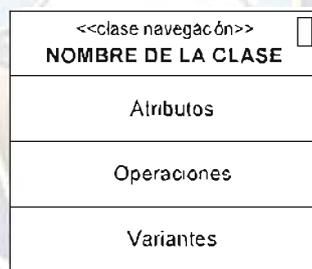
El diseño de navegación es un paso crítico en el diseño de la aplicación Web. El modelo de navegación se comprime en el modelo de espacio de navegación y el modelo de estructura de navegación. El primero especifica qué objetos pueden ser visitados mediante una navegación a través de la aplicación y el segundo define cómo estos objetos son alcanzados. [Koch, 2000]

### 2.4.3.1 Modelo de Espacio de Navegación

El modelo de espacio de navegación es construido con las clases de navegación y las asociaciones de navegación, y están representadas gráficamente por un diagrama de clases de UML. [Koch, 2000]

La clase de navegación modela una clase cuyas instancias son visitadas por usuarios durante la navegación. Se les asigna el nombre que se diera a las correspondientes clases conceptuales. Sin embargo, se diferencia de esta por el estereotipo <<navigation class>>. Además, una clase de navegación puede contener atributos de otras clases del modelo conceptual, siempre que la clase de navegación tenga alguna asociación con la clase de la que se presta el o los atributos. Para diferenciar dichos atributos se coloca una barra inclinada a la derecha (/) antes del nombre.

La navegación directa es representada por asociaciones en el modelo de espacio de navegación que provienen de la clase de navegación de origen. Por lo tanto, sus semánticas son diferentes de las asociaciones usadas en el modelo conceptual. Estas asociaciones son interpretadas como el enlace o vínculo entre la clase de navegación inicial (página Web de inicio) y la clase de navegación final (página Web destino). El estereotipo utilizado para identificar a esta asociación es <<direct navigability>>.



**Figura 7: Clase Navegación**  
Fuente: [Koch, 2000]

### 2.4.3.2 Modelo de Estructura de Navegación

El modelo de estructura de navegación describe cómo la navegación es soportada por elementos de acceso tales como índices, visitas guiadas, preguntas y menús. El resultado es un diagrama de clases UML construido con estereotipos, los cuales están definidos según mecanismos de extensión UML.

Las primitivas de acceso son nodos de navegación adicionales requeridas para acceder a objetos de navegación. Las siguientes primitivas de acceso son definidas como estereotipos UML: índices, visitas guiadas, consultas y menús. [Koch, 2000]

Los índices permiten el acceso directo a las instancias de la clase de navegación. Cualquier índice es miembro de la clase índice, y utiliza el estereotipo <<index>> con su icono correspondiente.

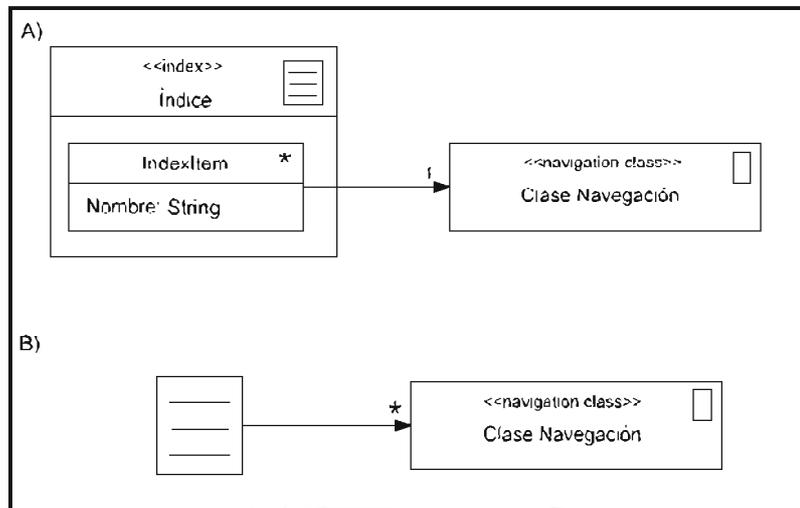


Figura 8: A) Clase Índice y B) Su notación taquigráfica  
Fuente: [Koch, 2000]

Las visitas guiadas proveen acceso secuencial a las instancias de una clase navegación. Para clases que contienen objetos de visita guiada se usa el estereotipo `<<guidedTour>>` y su icono correspondiente. Las visitas guiadas deben ser controladas por el usuario o por el sistema.

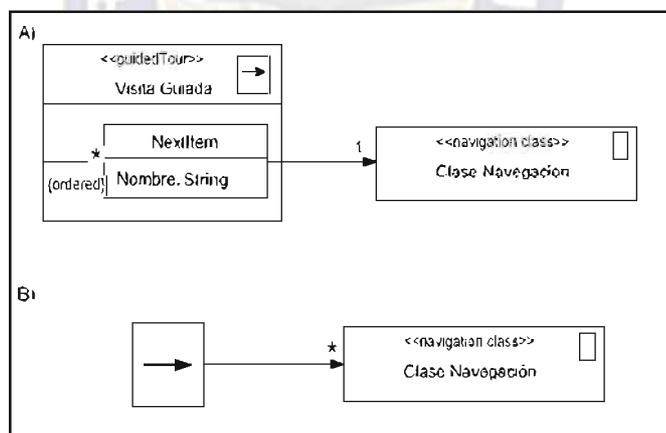
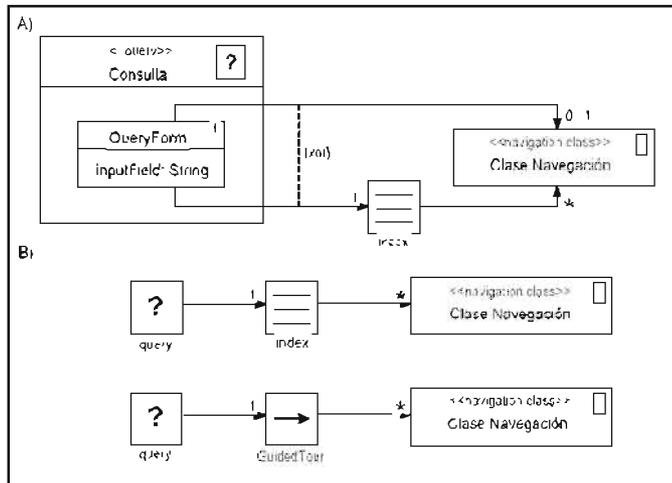


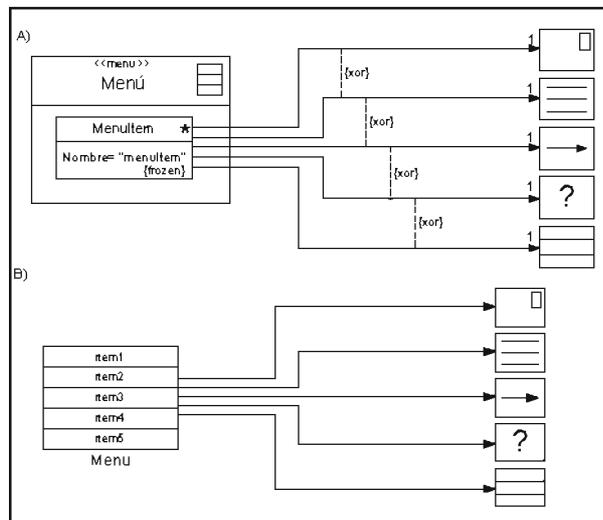
Figura 9: A) Clase Visita Guiada y B) su notación taquigráfica  
Fuente: [Koch, 2000]

Una consulta es modelada por una clase que tiene una serie de preguntas como atributo. Para la clase consulta se utiliza el estereotipo `<<query>>` y su icono correspondiente.



**Figura 10: A) Clase Consulta y B) Sus notaciones taquigráficas**  
 Fuente: [Koch, 2000]

Un menú es una primitiva de acceso adicional, es un índice de un conjunto de elementos heterogéneos, tales como índices, visitas guiadas, consultas, una instancia de una clase navegación u otro menú. Este es modelado por un objeto compuesto que contiene un número fijado de ítems de menú. Cada ítem de menú tiene un nombre constante y posee un enlace, ya sea a una instancia de una clase de navegación o a un elemento de acceso. Cualquier menú es una instancia de alguna clase menú que es estereotipada por <<menú>> con su icono correspondiente.



**Figura 11: A) Clase Menú y B) Su taquigrafía**  
 Fuente: [Koch, 2000]

## 2.4.4 Modelo de Presentación

El diseño de presentación soporta la construcción de un modelo de presentación basado en el modelo de estructura de navegación e información adicional que se recolecta durante el análisis de requerimientos. El modelo de presentación consiste en un conjunto de vistas que muestran el contenido y la estructura de los nodos simples, es decir cómo cada nodo es presentado al usuario y cómo el usuario puede interactuar con ellos. [Koch, 2000]

Las vistas de interfaz de usuario especifican que cada instancia de esta clase es un contenedor de todos los elementos abstractos de interfaces de usuario los cuales están presentados simultáneamente al usuario. Para las vistas de interfaz de usuario se utiliza el estereotipo `<<UI view>>` y su respectivo icono.

La clase presentación es una unidad estructural que permite particionar una vista de interfaz de usuario dentro de grupos de elementos de interfaz de usuario. Para la clase presentación se utiliza el estereotipo `<<presentation class>>`.

El elemento de interfaz de usuario es una clase de abstracción que tiene varias especializaciones describiendo elementos de interfaz particulares.

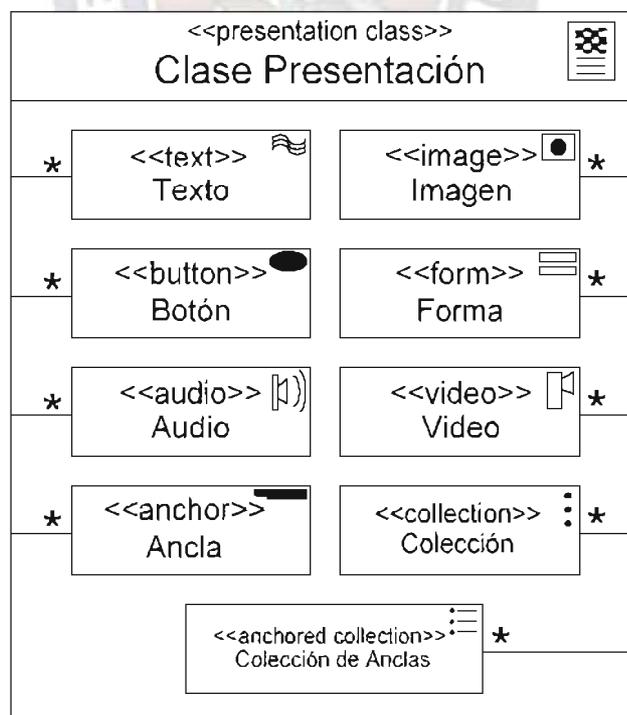


Figura 12: Clase de Presentación, con elementos del Modelo de Presentación  
Fuente: [Koch, 2000]

## 2.5 APLICACIONES WEB

Las aplicaciones basadas en la Web son muy diferentes de las otras categorías de software, pues estas se caracterizan por residir en una red (ya sea intranet o extranet) y deben dar servicio a una comunidad diversa de clientes, además de que deben tener una evolución continua. [Pressman, 2005]

### 2.5.1 Consideraciones de diseño

Se deben considerar los siguientes aspectos en el diseño de componentes de una aplicación Web: [SA5, 2008]

#### *i) Textos:*

- El uso de muchas fuentes en una página dificulta la lectura, es recomendable no utilizar más de dos fuentes distintas.
- Se debe utilizar normalmente una altura de 10 a 12 puntos para el texto.
- Se debe considerar que las líneas de texto cortas son más fáciles de leer que las largas.
- Es conveniente alinear los textos por la izquierda, la alineación derecha dificulta la lectura del mismo.

#### *ii) Hipervínculos:*

- Es necesario que cada hipervínculo contenga una descripción coherente para que el usuario identifique su función.
- Se debe elegir una longitud adecuada para cada enlace.
- No se debe cambiar el color de los enlaces si no es necesario.
- No se debe colocar enlaces a páginas en construcción o que se encuentran fuera de servicio.

#### *iii) Longitud de una página*

- Se debe diseñar páginas cortas y concisas, estas no deben exceder de una pantalla y media.
- Si una página tiende a ser bastante extensa, la mejor opción consiste en dividir en varias páginas el contenido de la Web.
- Las páginas no deben sobrepasar los 800 píxeles de longitud, pues gran parte de los monitores tienen esta medida.

#### *iv) Gráficos*

- Se debe utilizar solo las imágenes necesarias, pues estas aumentan el tiempo de espera del navegador.
  - Es conveniente utilizar iconos o símbolos que puedan sustituir a textos escritos, pues esto contribuirá a internacionalizar la aplicación Web.
  - Se debe utilizar un mismo estilo para la creación de todos los iconos de la aplicación Web, con el mismo tamaño y estilo gráfico.
  - No se debe utilizar fondos gráficos muy recargados, estos pueden desviar la atención del visitante.
- v) *Uso del color*
- Utilizar colores contrastantes y seguros, para facilitar la lectura de un texto.
  - Especificar el color de fondo, ya que este se mostrará en la pantalla inmediatamente.
  - Los colores pastel o neutros para el fondo de la página aumentan considerablemente la legibilidad del texto.
  - Utilizar colores y combinaciones iguales para todas las páginas de una misma aplicación Web.
- vi) *Navegabilidad*
- Es conveniente incluir un encabezado al inicio de cada página, que informe el nombre del sitio y la sección del mismo.
  - Se debe elegir un buen título para el documento, que refleje el contenido general del documento.
  - No dejar un camino cerrado, donde los usuarios tengan que presionar el botón back de su navegador para pasar a la página anterior.
  - Proporcionar siempre un enlace a la página principal, esto será de gran ayuda en aplicaciones Web de muchas páginas.
- vii) *Calidad del documento*
- Probar regularmente cada enlace de la aplicación Web.
  - Revisar la ortografía y la gramática de todas las páginas de la aplicación Web.
  - No utilizar comandos específicos de ciertos navegadores.
  - Se debe ofrecer a los visitantes la posibilidad de comunicarse con el administrador del sitio, pues su opinión es de gran ayuda para mejorar la calidad de la aplicación Web.
  - Debe informarse a los usuarios la fecha de la última modificación de la aplicación Web.

## 2.5.2 Arquitectura Cliente-Servidor

Un Servidor es una computadora que lleva a cabo un servicio que normalmente requiere mucha potencia de procesamiento. Un Cliente es una computadora que solicita los servicios que proporciona uno o más servidores y que también lleva a cabo un tipo de procesamiento por sí mismo. [Pressman, 2005]

Un *Servidor* además se caracteriza por: [SA6, 2008]

- Al iniciarse esperan las solicitudes de los clientes, desempeñando entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud la procesan y luego envían la respuesta al cliente.
- Aceptan conexiones desde un gran número de clientes.
- No interactúan directamente con los usuarios finales.

Por otra parte el *Cliente* está caracterizado por: [SA6, 2008]

- Es quien inicia las solicitudes, tienen entonces un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Puede conectarse a varios servidores a la vez.
- Interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Existen diversos tipos de Servidores, entre ellos los servidores de archivos, los servidores de bases de datos, servidores de objetos, servidores de correo, y los servidores Web que son los que nos interesan en este caso.

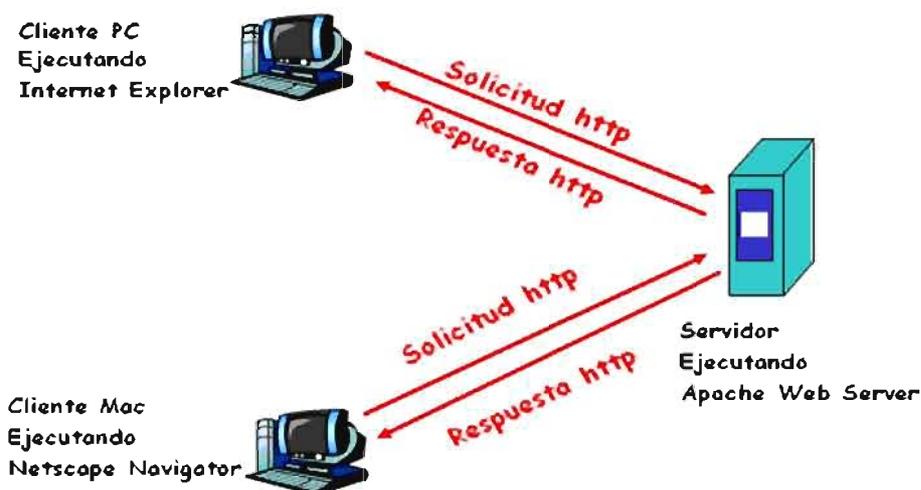


Figura 13: Servidor de páginas Web  
Fuente: [Elaboración propia]

Los Servidores Web almacenan documentos Web, en espera de solicitudes de los navegadores Web que son en este caso los Clientes. Para esto se utiliza el protocolo HTTP (*Hipertext Transfer Protocol o Protocolo de Transferencia de Hipertexto*), que define la sintaxis y la semántica que se utilizan en una comunicación entre cliente y servidor. Para que un servidor este activo debe ejecutarse en el un programa servidor de paginas Web, como Apache Web Server o Internet Information Server (IIS).

### 2.5.3 Criterios de seguridad

El tema de crear aplicaciones Web seguras es un tanto complejo, ya que requiere realizar todo un estudio para comprender los puntos vulnerables de la seguridad en cada aplicación.

En la siguiente tabla se puede identificar las áreas de seguridad en una aplicación Web, y las consideraciones que se deben tener.

Seguridad en el Cliente	<ul style="list-style-type: none"> <li>- Código móvil</li> <li>- Lenguajes de cliente</li> </ul>
Seguridad en el Servidor	<ul style="list-style-type: none"> <li>- Servidor Web</li> <li>- Servidor de base de datos</li> <li>- Lenguajes de servidor</li> </ul>
Seguridad en la Aplicación	<ul style="list-style-type: none"> <li>- Control de acceso</li> <li>- Validación de datos de entrada</li> <li>- Programación segura</li> </ul>
Seguridad en la Comunicación	<ul style="list-style-type: none"> <li>- Protocolos de seguridad</li> <li>- Certificados digitales</li> </ul>

**Tabla 1: Áreas de seguridad en una Aplicación Web**  
Fuente: [Elaboración propia]

La seguridad supone un coste económico y de eficiencia bastante alto. Es por eso que hay que disponer de la adecuada, ni más ni menos. No obstante, aunque no se tenga mucha experiencia en seguridad, existen unas medidas básicas que se deberían adoptar para proteger cualquier aplicación Web. [Gonzalez]

#### 2.5.3.1 Recomendaciones generales de seguridad para aplicaciones Web

Muchas de las recomendaciones de seguridad más básicas son también las más obvias. Sin embargo, incluso los métodos de seguridad avanzados pueden verse comprometidos si un

usuario malintencionado logra obtener acceso a los equipos usando medios simples, a continuación se darán algunas recomendaciones para que esto no suceda: [VWD, 2008]

- Realizar copias de seguridad con frecuencia y guardarlos en lugar seguro.
- Mantener el equipo del servidor en un lugar físico seguro, de forma que los usuarios no autorizados no puedan tener acceso a él, apagarlo, llevárselo, etc.
- Si se utiliza Windows, utilizar el sistema de archivos NTFS, no el FAT32. NTFS ofrece mucha más seguridad que FAT32.
- Proteger el equipo del servidor Web y todos los demás equipos de la misma red con contraseñas rigurosas.
- Cerrar los puertos que no se utilicen y desactivar los servicios no usados.
- Usar un servidor de seguridad (*firewall*), para controlar el tráfico de la información hacia el servidor.
- Ejecutar aplicaciones con privilegios mínimos.
- Mantener los archivos de la aplicación Web en una carpeta ubicada debajo de la raíz de la aplicación, con el fin de negar a los usuarios la posibilidad de acceder a archivos de la aplicación.
- Conocer a los usuarios, no está demás incluir un formulario de autenticación de usuarios al inicio de la aplicación.
- Mantener *secretos* de forma segura. *Secretos* son cualquier información que no se desea que conozcan otras personas como una contraseña o una clave cifrada.
- Crear mensajes de error seguros, un usuario malintencionado puede deducir información importante sobre la aplicación a partir de los mensajes de error que ésta muestra.
- Usar cookies de forma segura. Las *cookies* constituyen un modo fácil y útil de almacenar la información específica disponible sobre los usuarios. Sin embargo, como se envían al explorador del equipo, son vulnerables a la suplantación u otros usos malintencionados. Para evitarlo no almacene información vital como contraseñas en cookies y establezca su fecha de caducidad al periodo de tiempo más corto que sea viable.
- La subida de ficheros permite enviar cualquier fichero al servidor, debe evitarse utilizar el nombre enviado por el navegador, conviene generar un nombre único para el fichero subido. También se debe fijar el tamaño máximo de los ficheros.

- Es recomendable validar todos los datos provenientes de los formularios antes de utilizarlos, pues pueden contener información contaminada como código HTML.
- Un aspecto importante de una aplicación Web protegida es diseñar un modo de que ésta pueda tener acceso a la base de datos de forma segura, no almacenar nunca información proporcionada por el usuario sin filtrar en una base de datos.
- Protegerse de la inyección SQL. La *inyección SQL* es una forma de ataque en la que el usuario asigna de alguna forma código SQL a variables que luego serán utilizadas en consultas SQL. Para evitar esto se debe validar los datos y las variables que se han de integrar en consultas SQL.
- Si la aplicación debe utilizar el acceso anónimo a la base de datos, cree un único usuario con permisos muy limitados, y haga que las consultas se ejecuten iniciando las sesiones como dicho usuario.
- Si se quiere almacenar un nombre de usuario y una contraseña en alguna parte para usarlos en el inicio de sesión con la base de datos, se lo puede hacer de forma segura cifrándolos.

#### **2.5.4 Pruebas para Aplicaciones Web**

Se realizan pruebas a un producto software con el propósito de encontrar errores y finalmente corregirlos. Dado que las aplicaciones Web residen en una red e interoperan con muchos sistemas operativos diferentes, navegadores, plataformas de hardware, protocolos de comunicación, la búsqueda de errores es todo un reto para los desarrolladores. Las etapas que se deben seguir para la prueba de aplicaciones Web son las siguientes: [Pressman, 2005]

1. El modelo de contenido de la Aplicación Web es revisado para descubrir errores, para esto se deben revisar las páginas en busca de errores tipográficos y gramaticales.
2. El modelo de diseño de la Aplicación Web es revisado para descubrir errores de navegación, los enlaces de navegación son revisados para verificar su correspondencia.
3. Se aplican pruebas de unidad a los componentes de proceso. Las pruebas de unidad se encargan de verificar la menor unidad de diseño de software, en este caso una página Web. Para esto se comprueban los procesos encapsulados en la página utilizando una técnica de prueba de software. Una de estas técnicas es la del camino básico, que consiste en hallar la complejidad

cicломática<sup>2</sup> del programa, este valor representa el número de caminos de ejecución y el límite superior para el número de pruebas que se debe realizar.

4. Se construye la arquitectura y se realizan las pruebas de integración, para esto se pueden usar casos de prueba basados en los casos de uso.
5. Luego de ensamblar la aplicación Web se realizan las pruebas de seguridad tomando en cuenta los criterios de seguridad observados.
6. La aplicación Web se implementa en una variedad de configuraciones de diferentes entornos, para así comprobar su compatibilidad con cada configuración. Se pueden usar diferentes sistemas operativos, navegadores y plataformas de hardware.
7. La aplicación Web se prueba con una población de usuarios finales controlada y monitorizada, luego se evalúan los resultados de su interacción con el sistema.

Dado que las aplicaciones Web están en constante evolución, el proceso de comprobación es una actividad continua.

### **2.5.5 Métricas para Aplicaciones Web**

Las métricas que se emplean en los sistemas de información tradicionales son difíciles de aplicar directamente en aplicaciones Web, debido a esto se propone el uso de las siguientes medidas que servirán para la definición de métricas orientadas a aplicaciones Web: [Pressman, 2006]

*Número de páginas Web estáticas (NPE).*- Las páginas Web estáticas son aquellas en las que el usuario no controla el contenido de la página. Estas páginas representan una complejidad relativamente baja y por lo general requieren de menos esfuerzo al construirlas.

*Número de páginas Web dinámicas (NPD).*- Las páginas Web dinámicas son aquellas en las que las acciones del usuario generan contenido personalizado. Estas páginas representan una mayor complejidad y requieren más esfuerzo que las páginas estáticas. Estas dos medidas

---

<sup>2</sup> Métrica de software que proporciona una medición de la complejidad lógica de un programa, se calcula a partir de la fórmula:  $V(G) = P + 1$ , donde  $V(G)$  complejidad cicломática,  $P$  número de nodos predicado.

proporcionan un indicio del tamaño global de la aplicación y el esfuerzo necesario para desarrollarla.

*Número de vínculos internos de la página (NVI).*- Los vínculos internos de la página son hipervínculos que ofrecen un enlace hacia otra página dentro de la aplicación Web. Esta medida proporciona un indicio del grado de acoplamiento arquitectónico dentro de la aplicación Web.

*Número de objetos de datos persistentes (NOP).*- Una aplicación Web puede tener acceso a uno o más objetos de datos persistentes como bases de datos o archivos. Esta medida proporciona un indicio de la complejidad de la aplicación Web.

En base a estas medidas se pueden definir las siguientes métricas:

$$\text{Índice de personalización del usuario} = \text{NPD} / (\text{NPE} + \text{NPD})$$

Este índice varía entre cero y uno dependiendo del grado de personalización que la aplicación Web ofrece al usuario.

$$\text{Grado de acoplamiento} = \text{NVI} / (\text{NPE} + \text{NPD})$$

El grado de acoplamiento indica cuan relacionadas están las páginas de la aplicación Web. Si el valor obtenido es cero significa que no existe acoplamiento entre las páginas de la aplicación Web, si este valor es 1 significa que en promedio cada página contiene un hipervínculo hacia alguna otra página de la aplicación Web, y así sucesivamente dependiendo del número de vínculos que se tengan en la aplicación Web.

$$\text{Grado de complejidad} = (\text{NPE} + \text{NPD}) / \text{NOP}$$

El grado de complejidad indica cuan compleja es la aplicación Web. Si el grado es menor a 1 la complejidad no es mucha pues el número de páginas Web es inferior al número de objetos persistentes. El grado de complejidad crece si es mayor a 1, pues habrán más páginas Web que actuarán sobre los objetos persistentes.

## CAPÍTULO III DESARROLLO DEL PROYECTO

Para el desarrollo del proyecto se utilizó la metodología Ágil SCRUM, que utiliza un modelo de proceso incremental, y se la complementó con la metodología UWE para las etapas de desarrollo de cada iteración. En la siguiente figura se puede apreciar gráficamente el modelo de proceso que se utilizó en el presente Proyecto de Grado:

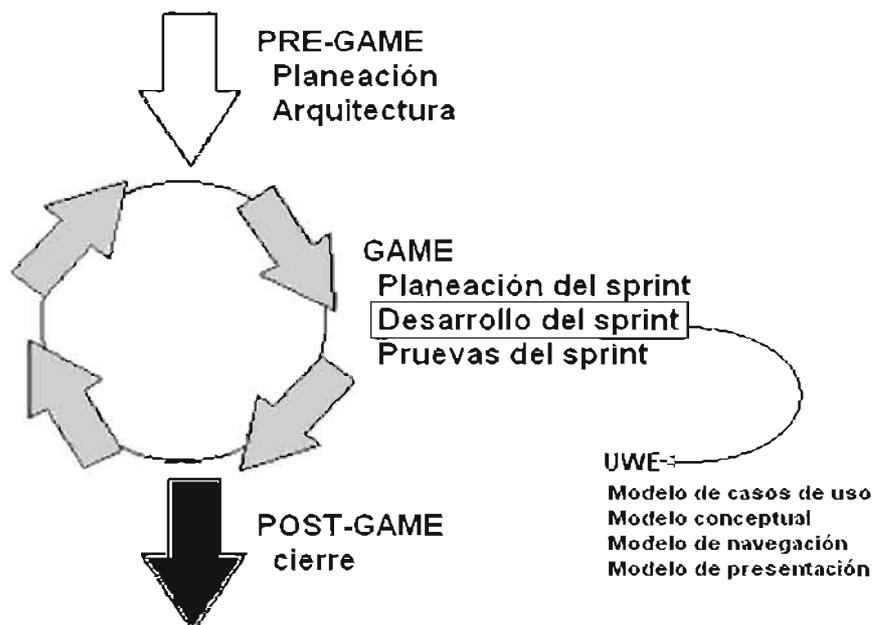


Figura 14: Modelo de proceso del proyecto  
Fuente: [Elaboración propia]

### 3.1 PRE-GAME

#### 3.1.1 Recopilación de requerimientos

Ya que el presente proyecto constituye una solución e-learning, muchos de los requerimientos fueron tomados en base a los elementos del e-learning que son la *plataforma*, los *contenidos* y los *sistemas de comunicación*.

A continuación se presenta el Backlog del producto, que contiene los requerimientos y características finales del sistema.

ID	DESCRIPCIÓN	MÓDULO	PRIORIDAD	ESTADO
R1	Base de datos independiente del sistema académico	Plataforma	Alta	Terminado
R2	Soporte para diferentes tipos de usuarios: estudiantes, docentes y administrativos	Plataforma	Alta	Terminado
R3	Soporte para administración y control de los usuarios	Plataforma	Alta	Terminado
R4	Control de acceso seguro y diferenciado a usuarios	Plataforma	Media	Terminado
R5	Registro para usuarios nuevos	Plataforma	Media	Terminado
R6	Interfaces amigables y personalizables	Plataforma, Contenidos, Comunicación	Media	Terminado
R7	Soporte para la administración de contenidos de cada curso	Contenidos	Media	Terminado
R8	Soporte para la presentación de contenidos en cada curso	Contenidos	Media	Terminado
R9	Soporte para contenidos en forma de documentos	Contenidos	Media	Terminado
R10	Soporte para la administración de foros en cada curso	Comunicación	Baja	Terminado
R11	Soporte para la publicación de anuncios administrativos	Comunicación	Baja	Terminado
R12	Soporte para la publicación de comunicados para cada curso	Comunicación	Baja	Terminado
R13	Soporte para contenidos SCORM	Contenidos	Baja	Postergado

Tabla 2: Backlog del producto  
Fuente: [Elaboración propia]

### 3.1.2 Definición del cronograma de trabajo

El cronograma de trabajo se definió en base al ciclo de vida de la metodología SCRUM, en el cual se identifican 3 etapas principales, que son: el *pre-game*, *game* y *post-game*

La descripción detallada del cronograma de trabajo para el proyecto se la puede observar en el diagrama Gantt del Anexo A.

### 3.1.3 Análisis de riesgos

Un riesgo es la probabilidad de que ocurra algo adverso, existen 3 tipos de riesgo:

1. *Riesgo del proyecto*.- que afectan a la calendarización o recursos del proyecto.

2. *Riesgo del producto*.- afectan a la calidad o rendimiento del software que se está desarrollando.

3. *Riesgo del negocio*.- afectan a la organización que desarrolla o suministra el software.

Durante la gestión de riesgos del presente proyecto se identificaron varios riesgos los cuales se describen a continuación así como las estrategias que se propusieron para combatirlos:

Riesgo	Tipo	Descripción	Probabilidad	Efecto	Estrategia
No se cumplen las fechas establecidas en el cronograma	Proyecto	Es probable que las fechas previstas en el cronograma para cada una de las etapas no se cumplan a cabalidad	Alta	Tolerable	Diseñar un cronograma flexible que considere los posibles retrasos en las etapas más críticas
Cambios en los requerimientos del cliente	Proyecto, producto	Riesgo de que haya cambios en los requerimientos de la institución que no hayan sido considerados	Moderada	Tolerable	- Realizar una revisión constante a los requerimientos.  - Programar reuniones constantes con el cliente
No se cumplen con los plazos de entrega del producto	Producto	Los plazos de entrega del producto están determinados por la Dirección de Carrera y no así por el desarrollador del proyecto, debido a esto es probable que exista cierto retraso en la entrega del producto.	Moderada	Serio	- Agilizar los procesos de desarrollo del producto.  - Realizar una correcta planificación, considerando el tiempo y los alcances del proyecto.
No existe la infraestructura necesaria para la implementación del sistema.	Proyecto	La institución en la que se está realizando el proyecto no cuenta con recursos propios, y existe el riesgo de que no tenga el equipamiento necesario para la implementación del sistema en un corto plazo.	Alto	Tolerable	Solicitar con anticipación el equipamiento y las condiciones necesarias para la implementación del sistema
No se concluya con algunos de los requerimientos	Producto	Es probable que dado el gran número de requerimientos del cliente y el tiempo necesario para	Alto	Tolerable	Aislar los módulos en el sistema de tal forma que la ausencia de uno no

		cumplirlos, no se cumpla con alguno de ellos			afecte el desempeño de los demás.
--	--	--	--	--	-----------------------------------

**Tabla 3: Análisis de Riesgos**  
Fuente: [Elaboración propia]

### 3.1.4 Herramientas de desarrollo

Para el desarrollo del proyecto se decidió utilizar las siguientes herramientas:

Herramientas para el modelado del sistema:

- *Microsoft Office Visio 2003*, para los modelos de casos de uso, modelo E-R y diseño de la Base de Datos.
- *Argo UML 4.0*, para el modelo conceptual, de navegación y de presentación.

Herramientas para el desarrollo del sistema:

- *Macromedia Dreamweaver 8*, para el diseño de las hojas de estilo.
- *Rapid PHP 8.0*, para el desarrollo de las páginas dinámicas.
- *Macromedia FireWorks 8*, para el diseño de los gráficos.
- *PHP 5.1.6*, como lenguaje de programación del lado del servidor.
- *MySQL 5.0.24a*, como sistema gestor de la base de datos.
- *Apache 2.0.59*, para servidor de páginas Web.
- *Internet Explorer 6*, como navegador de páginas Web.
- *Window XP Service Pack 2*, como sistema operativo.

### 3.1.5 Análisis de costos

La estrategia que se uso para calcular los costos de cada iteración fue tomar cada funcionalidad del backlog del producto y estimar su volumen mediante la métrica de *punto de función* (PF)<sup>3</sup>. Luego se procedió a sumar estos volúmenes y hallar el esfuerzo necesario para

<sup>3</sup> Métrica de software orientada a la función, representa una medida de la funcionalidad de una aplicación. Se calcula a partir de la fórmula  $PF = \text{cuenta-total} \cdot (0.65 + 0.01 \sum F)$ . Donde cuenta-total se calcula a partir de 5 valores de dominio de la información y  $F_i (i = 1 \dots 14)$  son valores de ajuste de la complejidad obtenidos a partir de las respuestas a 14 preguntas sobre el sistema.

cada iteración en base a un modelo de estimación. En este caso se utilizó el *modelo de regresión para proyectos pequeños*<sup>4</sup>. [Pressman, 2006]

En la siguiente tabla se puede apreciar los puntos de función obtenidos y el esfuerzo estimado para cada iteración:

Iteración	Funcionalidad	Puntos de función	esfuerzo de la iteración (hombres-mes)
Primera iteración	Página de ingreso al sistema	20.24	$E = (20.24 + 26.4 + 17.6 + 68.64) * 0.504 - 12.88 * 4$ $E = 2.29$
	Registro de usuarios	26.4	
	Configuración de perfil de usuario	17.6	
	Administración de usuarios	68.64	
Segunda iteración	Administración de foros	47.52	$E = (47.52 + 28.16 + 23.76) * 0.504 - 12.88 * 3$ $E = 1.633$
	Soporte para anuncios administrativos	28.16	
	Soporte para comunicados de docente	23.76	
Tercera iteración	Administración de contenidos	35.2	$E = (35.2) * 0.504 - 12.88$ $E = 1.376$
<b>Esfuerzo total estimado</b>			<b><math>E_T = 5.299</math></b>

Tabla 4: Cálculo del esfuerzo por cada iteración  
Fuente: [Elaboración propia]

### 3.2 GAME

Durante esta etapa del proyecto se desarrollaron 3 iteraciones, cada una de ellas corresponde a un elemento del e-learning: *Plataforma, Contenidos y Sistemas de comunicación*. A continuación se desglosan las actividades realizadas en cada una de estas etapas.

La estrategia que se utilizó para el desarrollo de cada iteración fue construir en un principio los modelos propios de la metodología UWE y posteriormente implementarlos utilizando como elemento central la Base de Datos “AEV”. La solución por la que se optó para la persistencia de objetos fue el hacer corresponder cada clase del modelo conceptual con una tabla de la base de datos, en donde las filas representan las instancias de los objetos y las columnas a los atributos de la clase. Las clases y sus métodos fueron implementados en lenguaje PHP. Finalmente se desarrollaron las páginas Web en base a los modelos de presentación y de navegación.

<sup>4</sup> Modelo que utiliza un análisis de regresión en base a datos recopilados de proyectos previos, para obtener empíricamente el esfuerzo necesario en hombres-mes para el desarrollo de un proyecto. Se calcula a partir de la fórmula  $E = -12.88 + 0.405PF$ . Donde PF es el número de puntos de función

### 3.2.1 Primera Iteración

Durante la primera iteración se desarrollaron los elementos pertenecientes a la Plataforma del Ambiente Educativo Virtual. Las actividades realizadas durante esta iteración se las puede observar en la siguiente tabla, que constituye el backlog del Sprint:

		Sprint	Inicio	Duración
		1	28-01-2008	30 días
ID	Tarea	Tipo	Días de trabajo	Estado
1.1	Realizar la planificación de la iteración	Planificación	4	Terminado
1.2	Analizar los requerimientos del backlog del producto	Planificación	2	Terminado
1.3	Analizar los requerimientos de la iteración con casos de uso	Desarrollo	2	Terminado
1.4	Construir el modelo conceptual	Desarrollo	2	Terminado
1.5	Diseñar y construir la base de datos del sistema	Desarrollo	3	Terminado
1.6	Construir el modelo de navegación	Desarrollo	2	Terminado
1.7	Construir el modelo de presentación	Desarrollo	3	Terminado
1.8	Desarrollar la página de ingreso al sistema	Desarrollo	2	Terminado
1.9	Desarrollar el módulo de registro de nuevos usuarios	Desarrollo	2	Terminado
1.10	Desarrollar el módulo de configuración de perfiles de usuario	Desarrollo	2	Terminado
1.11	Desarrollar el módulo de administración del sistema	Desarrollo	3	Terminado
1.12	Probar los módulos elaborados en la iteración	Revisión	3	Terminado

**Tabla 5: Backlog del primer Sprint**

Fuente: [Elaboración propia]

Las funcionalidades correspondientes al incremento de la iteración son:

- Base de Datos independiente del Sistema Académico
- Página de ingreso con control de acceso a los usuarios
- Módulo de registro de usuarios nuevos
- Módulo de configuración de perfiles de usuario
- Módulo de administración del sistema

### 3.2.2 Segunda Iteración

En la segunda iteración se desarrollaron los sistemas de Comunicación para el Ambiente Educativo Virtual, las actividades realizadas en esta iteración se las puede observar en la tabla del backlog del sprint:

		Sprint	Inicio	Duración
		2	10-03-2008	25 días
ID	Tarea	Tipo	Días de trabajo	Estado
2.1	Realizar la planificación de la iteración	Planificación	4	Terminado
2.2	Analizar los requerimientos para los sistemas de comunicación (backlog del producto)	Planificación	2	Terminado
2.3	Analizar los requerimientos de la iteración con Casos de uso	Desarrollo	2	Terminado
2.4	Complementar el modelo conceptual	Desarrollo	2	Terminado
2.5	Complementar el modelo de navegación	Desarrollo	2	Terminado
2.6	Construir los modelos de presentación	Desarrollo	3	Terminado
2.7	Desarrollar el módulo de administración de foros	Desarrollo	3	Terminado
2.8	Desarrollar las páginas para la publicación de anuncios	Desarrollo	2	Terminado
2.9	Desarrollar las páginas para la publicación de comunicados	Desarrollo	2	Terminado
2.10	Probar las funcionalidades elaboradas durante la iteración	Revisión	3	Terminado

**Tabla 6: Backlog del segundo Sprint**  
Fuente: [Elaboración propia]

En la segunda iteración de se desarrollaron las siguientes funcionalidades para el sistema:

- Módulo para administración foros en cada curso
- Páginas para la publicación de anuncios administrativos
- Páginas para la publicación de comunicados de docentes en cada curso

### 3.2.3 Tercera Iteración

Finalmente durante la tercera iteración se desarrolló el soporte para la administración de Contenidos del Ambiente Educativo Virtual. Las actividades realizadas durante esta iteración se las puede observar en la siguiente tabla, que constituye el backlog del tercer sprint:

		<b>Sprint</b>	<b>Inicio</b>	<b>Duración</b>
		3	14-04-2008	20 días
<b>ID</b>	<b>Tarea</b>	<b>Tipo</b>	<b>Días de trabajo</b>	<b>Estado</b>
3.1	Realizar la planificación de la iteración	Planificación	3	Terminado
3.2	Analizar los requerimientos para el módulo de Contenidos (backlog del producto)	Planificación	2	Terminado
3.3	Analizar los requerimientos de la iteración con Casos de uso	Desarrollo	1	Terminado
3.4	Complementar el modelo conceptual	Desarrollo	2	Terminado
3.5	Complementar el modelo de navegación	Desarrollo	1	Terminado
3.6	Construir los modelos de presentación	Desarrollo	2	Terminado
3.7	Desarrollar el módulo para la administración de contenidos	Desarrollo	3	Terminado
3.8	Desarrollar las páginas para la presentación de contenidos	Desarrollo	2	Terminado
3.9	Construir las hojas de estilo para la aplicación Web	Desarrollo	2	Terminado
3.10	Probar las funcionalidades elaboradas en la iteración	Revisión	2	Terminado

**Tabla 7: Backlog del tercer Sprint**  
Fuente: [Elaboración propia]

Durante la tercera y última iteración se desarrollaron las siguientes funcionalidades para el sistema:

- Módulo de administración de contenidos
- Páginas para la presentación de los contenidos de cada curso
- Hojas de estilo para la presentación de las páginas Web

### **3.3 POST-GAME**

Durante esta última etapa se realizaron las actividades de prueba de la aplicación Web, se propusieron las políticas de seguridad para el sistema, se obtuvieron las métricas y se realizó el diseño de la ayuda para los usuarios.

#### **3.3.1 Pruebas de la aplicación Web**

##### **3.1.1.1 REVISIÓN DEL CONTENIDO**

La primera actividad que se realizó como parte de las pruebas de la aplicación Web fue la revisión del contenido de las páginas. Para esto se utilizó es corrector ortográfico de

*Macromedia Dreamweaver 8*. Se encontraron errores ortográficos y gramaticales que fueron corregidos inmediatamente.

### 3.1.1.2 PRUEBAS DE NAVEGACIÓN

Como segunda actividad se verificaron los enlaces y sus correspondencias en busca de errores de navegación en la Aplicación Web. Obteniendo resultados satisfactorios pues todos los enlaces correspondían correctamente.

### 3.1.1.3 PRUEBAS DE UNIDAD

La tercera actividad consistió en realizar las pruebas de unidad a cada uno de los métodos de las páginas Web que contenían código dinámico. Para esto se utilizó la técnica del camino básico. Los resultados de las pruebas fueron los siguientes:

Archivo	Complejidad ciclomática	Camino básico	Resultados
Eliminar_anuncio.php	<b>eliminar_anuncio()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error en la ejecución de la consulta.	1. Se elimina el anuncio de la base de datos y se envía un mensaje 2. Se envía un mensaje de error
Eliminar_comunicado.php	<b>eliminar_comunicado()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error en la ejecución de la consulta.	1. Se elimina el comunicado de la base de datos y se envía un mensaje 2. Se envía un mensaje de error
Eliminar_contenido.php	<b>eliminar_contenido()</b> V(G) = 4	1. La consulta se realiza correctamente y el contenido es un documento existente 2. La consulta se realiza correctamente, el contenido es un documento pero no se encuentra el archivo 3. La consulta se realiza correctamente y el contenido no es un documento 4. ocurrió un error al realizar la consulta	1. Se elimina el contenido de la base de datos y también el archivo 2. Se elimina el contenido de la base de datos pero no el archivo 3. Se elimina el contenido de la base de datos 4. Se envía un mensaje de error
Eliminar_curso.php	<b>eliminar_curso()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se elimina el curso de la base de datos, los comunicados y los foros que le

			<p>pertenećian</p> <p>2. Se envía un mensaje de error</p>
Eliminar_docente.php	<p><b>eliminar_docente()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta.</p>	<p>1. Se elimina el registro de la base de datos</p> <p>2. Se envía un mensaje de error</p>
Eliminar_estilo.php	<p><b>eliminar_estilo()</b></p> <p>V(G) = 5</p>	<p>1. La consulta se realiza correctamente, existen los archivos de vista previa y de la hoja de estilos</p> <p>2. La consulta se realiza correctamente, existe el archivo de vista previa pero no el archivo de la hoja de estilos</p> <p>3. La consulta se realiza correctamente existe el archivo de la hoja de estilos pero no el archivo de la vista previa</p> <p>4. la consulta se realiza correctamente pero no se encuentran los archivos</p> <p>5. ocurrió un error al realizar la consulta</p>	<p>1. Se elimina el estilo de la base de datos y también los archivos</p> <p>2. Se elimina el estilo de la base de datos y el archivo de vista previa pero no el archivo de la hoja de estilos</p> <p>3. Se elimina el estilo de la base de datos y el archivo de la hoja de estilos pero no el archivo de la vista previa</p> <p>4. Se elimina el estilo de la base de datos pero no los archivos</p> <p>5. Se envía un mensaje de error</p>
Eliminar_estudiante.php	<p><b>eliminar_estudiante()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta.</p>	<p>1. Se elimina el estudiante de la base de datos</p> <p>2. Se envía un mensaje de error</p>
Eliminar_estudiante_curso.php	<p><b>eliminar_estudiante_curso()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta.</p>	<p>1. Se elimina el estudiante del curso</p> <p>2. Se envía un mensaje de error</p>
Eliminar_foro.php	<p><b>eliminar_foro()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta.</p>	<p>1. Se elimina el foro de la base de datos y todos los mensajes que contenía</p> <p>2. Se envía un mensaje de error</p>
Eliminar_mensaje.php	<p><b>eliminar_mensaje()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta</p>	<p>1. Se elimina el mensaje del foro</p> <p>2. Se envía un mensaje de error</p>
Eliminar_usuario.php	<p><b>eliminar_usuario()</b></p> <p>V(G) = 2</p>	<p>1. La consulta se realiza correctamente.</p> <p>2. Ocorre un error al ejecutar la consulta.</p>	<p>1. Se elimina el usuario de la base de datos</p> <p>2. Se envía un mensaje de error</p>

Grabar_adicionar _curso.php	adicionar_curso() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el curso en la base de datos 2. se envía un mensaje de error
Grabar_adicionar _docente.php	adicionar_docente() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el docente en la base de datos 2. Se envía un mensaje de error
Grabar_adicionar _estilo.php	adicionar_estilo() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el estilo en la base de datos y se copian los archivos a la carpeta de estilos 2. Se envía un mensaje de error
Grabar_adicionar _estudiante.php	adicionar_estudiante() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el estudiante en la base de datos 2. Se envía un mensaje de error
Grabar_adicionar _estudiante_curso.php	adicionar_estudiante _curso() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el estudiante al curso 2. se envía un mensaje de error
Grabar_adicionar _usuario.php	adicionar_usuario() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se adiciona el usuario en la base de datos 2. Se envía un mensaje de error
Grabar_configuracion.php	grabar_configuracion() V(G) = 4	1. El usuario está registrado y no ingreso una nueva contraseña 2. El usuario está registrado e ingreso una nueva contraseña 3. El usuario se está registrado pero hubo un error en la actualización de los datos 4. El usuario no está registrado	1. Se actualizan los datos que fueron modificados incluyendo el estilo si es que se cambio 2. Se actualizan los datos modificados y la nueva contraseña 3. Se muestra un mensaje de error 4. Se envía un mensaje de error
Grabar_modificar _curso.php	modificar_curso() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se actualizan los datos del curso 2. Se envía un mensaje de error
Grabar_modificar _docente.php	modificar_docente() V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar	1. Se actualizan los datos del docente 2. Se envía un

		la consulta.	mensaje de error
Grabar_modificar_estudiante.php	<b>modificar_estudiante()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se actualizan los datos del estudiante 2. Se envía un mensaje de error
Grabar_modificar_curso.php	<b>modificar_curso()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se actualizan los datos del curso 2. Se envía un mensaje de error
Grabar_registro.php	V(G) = 4	1. El usuario no está registrado, no existe otro usuario con el mismo nombre y no hubo un error al actualizar la tabla 2. El usuario no está registrado, no existe otro usuario con el mismo nombre pero ocurrió un error al actualizar la base de datos 3. El usuario no está registrado pero existe otro usuario con el mismo nombre de usuario 4. el usuario ya está registrado	1. Se registra al usuario y se le muestra su nombre de usuario y contraseña 2. Se informa del error al usuario 3. Se informa del error al usuario 4. Se muestra un mensaje de error
Ingresar_anuncio.php	<b>ingresar_anuncio()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se inserta el anuncio administrativo en la base de datos 2. Se envía un mensaje de error
Ingresar_comunicado.php	<b>ingresar_comunicado()</b> V(G) = 2	1. La consulta se realiza correctamente. 2. Ocurre un error al ejecutar la consulta.	1. Se inserta el anuncio comunicado a los comunicados del curso 2. Se envía un mensaje de error
Ingresar_contenido.php	<b>ingresar_contenido()</b> V(G) = 3	1. El contenido ya existe 2. El contenido es una dirección URL 3. El contenido es un documento o un contenido scorm	1. Se agrega el contenido al curso y se actualiza la base de datos 2. Se inserta el contenido en la base de datos y se la agrega al curso 3. Se inserta el contenido en la base de datos, se la agrega al curso y se copia el archivo del contenido

Ingresar_mensaje.php	<b>ingresar_mensaje()</b> V(G) = 2	<ol style="list-style-type: none"> <li>1. La consulta se realiza correctamente.</li> <li>2. Ocurre un error al ejecutar la consulta.</li> </ol>	<ol style="list-style-type: none"> <li>1. Se adiciona el mensaje al foro</li> <li>2. Se envía un mensaje de error</li> </ol>
Scripts.php	<b>conectar()</b> V(G) = 3  <b>verificar()</b> V(G) = 4	<b>conectar()</b> <ol style="list-style-type: none"> <li>1. No se puede establecer la conexión a la base de datos</li> <li>2. Ocurrió un error en la selección de la base de datos</li> <li>3. No ocurrió ningún problema en la conexión</li> </ol> <b>verificar()</b> <ol style="list-style-type: none"> <li>1. La sesión se ha iniciado, el usuario existe y es quien dice ser</li> <li>2. La sesión se ha iniciado, el usuario existe pero no es quien dice ser</li> <li>3. La sesión se ha iniciado, pero el usuario no existe</li> <li>4. La sesión no se ha iniciado</li> </ol>	Función conectar() <ol style="list-style-type: none"> <li>1. Se envía el mensaje de error y se sale del sistema</li> <li>2. Se envía un mensaje de error y se sale del sistema</li> <li>3. Se retorna la conexión a la base de datos</li> </ol> Función conectar() <ol style="list-style-type: none"> <li>1. Se permite el acceso a la página solicitada</li> <li>2. Se envía un mensaje de error sale del sistema</li> <li>3. Se envía un mensaje de error sale del sistema</li> <li>4. Se envía un mensaje de error sale del sistema</li> </ol>
Verificar_registro.php	V(G) = 3	<ol style="list-style-type: none"> <li>1. El nombre y la clave son correctas, y el usuario no está registrado</li> <li>2. El nombre y la clave son correctas pero el usuario ya está registrado</li> <li>3. El nombre y la clave son incorrectos</li> </ol>	<ol style="list-style-type: none"> <li>1. Pasamos al segundo paso del registro</li> <li>2. Se envía un mensaje de error</li> <li>3. se envía un mensaje de error</li> </ol>
Validar_usuario.php	V(G) = 4	<ol style="list-style-type: none"> <li>1. El usuario existe, es quien dice ser, y tiene un estilo asignado</li> <li>2. El usuario existe, es quien dice ser, y no tiene un estilo asignado</li> <li>3. El usuario existe, pero no es quien dice ser</li> <li>4. El usuario no existe o no está registrado</li> </ol>	<ol style="list-style-type: none"> <li>1. Se almacenan las variables de sesión, se carga la hoja de estilos y se ingresa al sistema</li> <li>2. Se almacenan las variables de sesión, se carga la hoja de estilos por defecto y se ingresa al sistema</li> <li>3. Se muestra un mensaje de error</li> <li>4. Se muestra un mensaje de error</li> </ol>

**Tabla 8: Pruebas de unidad**  
Fuente: [Elaboración propia]

Estas tres primeras actividades de la prueba de la aplicación Web fueron realizadas en las etapas de revisión de cada iteración. Las tres siguientes actividades se desarrollaron luego de concluir con las iteraciones, durante la etapa de cierre.

### 3.1.1.4 PRUEBAS DE INTEGRACIÓN

Luego de concluir con las iteraciones se realizaron las pruebas de integración, para esto se usaron los siguientes casos de prueba, basados en los casos de uso especificados en el modelado del sistema:

Caso de Prueba N° 1	
<b>Nombre Caso de Prueba:</b>	Registrarse como nuevo usuario
<b>Descripción:</b>	Accedemos a la página de ingreso del sistema e ingresamos a la página de registro de usuarios nuevos, en ella llenaremos los datos que nos soliciten y presionamos el botón siguiente. A continuación en otra página llenaremos los datos del nuevo usuario y presionamos el botón registrar.
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	nombre: "Juan Perez" , ci: "6543210", tipo: "estudiante", nombre de usuario: "juan123", contraseña: "maria", correo electrónico: "correodejuan@yahoo.es"
<b>Resultado esperado:</b>	El usuario es registrado correctamente y sus datos se almacenan en la Base de Datos sin ningún problema
<b>Evaluación:</b>	La prueba fue superada con éxito

Tabla 9: Caso de prueba "registrarse como nuevo usuario"  
Fuente: [Elaboración propia]

Caso de Prueba N° 2	
<b>Nombre Caso de Prueba:</b>	Ingresar al sistema
<b>Descripción:</b>	Accedemos a la página de ingreso del sistema, introducimos nuestros datos de usuario e ingresamos al sistema
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "juan123", contraseña: "maria", ingresar como: "estudiante"
<b>Resultado esperado:</b>	El usuario ingresa al sistema correctamente y su configuración personal se carga sin problemas
<b>Evaluación:</b>	La prueba fue superada con éxito

Tabla 10: Caso de prueba "ingresar al sistema"  
Fuente: [Elaboración propia]

Caso de Prueba N° 3	
<b>Nombre Caso de Prueba:</b>	Leer anuncios administrativos
<b>Descripción:</b>	Ingresamos al sistema como Administrativos, accedemos a los

	anuncios administrativos, introducimos un nuevo anuncio y luego lo eliminamos
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", tipo: "administrativo", título: "este es un anuncio de prueba", mensaje: "se comunica a todos los estudiantes regularizar su inscripción"
<b>Resultado esperado:</b>	El anuncio es publicado correctamente y al ser eliminado no ocurre ningún problema
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 11: Caso de prueba "leer anuncios administrativos"**

Fuente: [Elaboración propia]

<b>Caso de Prueba N° 4</b>	
<b>Nombre Caso de Prueba:</b>	Configurar perfil de usuario
<b>Descripción:</b>	Ingresamos al sistema como estudiantes, accedemos al módulo de configuración, cambiamos nuestra contraseña y elegimos un nuevo estilo
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "juan123", contraseña: "maria", nueva contraseña: "123456", estilo: "estilo 2"
<b>Resultado esperado:</b>	La contraseña del usuario es cambiada correctamente, el nuevo estilo es elegido y cambiado inmediatamente
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 12: caso de prueba "configurar perfil de usuario"**

Fuente: [Elaboración propia]

<b>Caso de Prueba N° 5</b>	
<b>Nombre Caso de Prueba:</b>	utilizar los foros de un curso
<b>Descripción:</b>	Ingresamos al sistema como administrativos, seleccionamos un curso e ingresamos a sus foros. Creamos un nuevo foro e ingresamos a este, enviamos un nuevo mensaje, lo eliminamos y salimos del foro. Finalmente eliminamos el foro que creamos para la prueba.
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", nombre del nuevo foro: "foro de prueba", comentario: "este es un foro de prueba", título del nuevo mensaje: "primer mensaje", mensaje: "este es un mensaje de prueba"
<b>Resultado esperado:</b>	El nuevo foro es creado sin ningún problema, el mensaje es enviado y almacenado. Posteriormente el mensaje es eliminado correctamente al igual que el foro
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 13: Caso de prueba "utilizar los foros de un curso"**

Fuente: [Elaboración propia]

Caso de Prueba N° 6	
<b>Nombre Caso de Prueba:</b>	acceder a los contenidos de un curso
<b>Descripción:</b>	Ingresamos al sistema como administrativos, seleccionamos un curso e ingresamos a sus contenidos. Enviamos un nuevo contenido y accedemos a este. Finalmente eliminamos el contenido que enviamos para la prueba.
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", nombre del nuevo contenido: "contenido de prueba", archivo:"nuevo documento de texto.txt", descripción: "este es un contenido de prueba"
<b>Resultado esperado:</b>	El contenido es enviado sin ningún problema. Se accede al contenido y este se muestra en el navegador. Posteriormente el contenido es eliminado correctamente
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 14: Caso de prueba "acceder a los contenidos de un curso"**  
Fuente: [Elaboración propia]

Caso de Prueba N° 7	
<b>Nombre Caso de Prueba:</b>	acceder a los comunicados de un curso
<b>Descripción:</b>	Ingresamos al sistema como administrativos, seleccionamos un curso e ingresamos a sus comunicados. Ingresamos un nuevo comunicado y lo eliminamos
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", título del comunicado: "nuevo comunicado", mensaje:"este es el primer comunicado del curso"
<b>Resultado esperado:</b>	El comunicado es publicado y eliminado sin problemas
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 15: Caso de prueba "acceder a los comunicados de un curso"**  
Fuente: [Elaboración propia]

Caso de Prueba N° 8	
<b>Nombre Caso de Prueba:</b>	Administrar usuarios
<b>Descripción:</b>	Ingresamos al sistema como administrativos, accedemos al módulo de administración e ingresamos a la administración de usuarios. Adicionamos un nuevo usuario, modificamos sus datos y lo eliminamos del sistema
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", nombre del nuevo usuario: "david", contraseña:"123", tipo:"estudiante", estado:"0", estilo:"estilo_1", correo electrónico:"david@yahoo.es"
<b>Resultado esperado:</b>	El usuario es adicionado correctamente, posteriormente sus datos son modificados y finalmente se lo elimina del sistema
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 16: Caso de prueba "administrar usuarios"**  
Fuente: [Elaboración propia]

<b>Caso de Prueba N° 9</b>	
<b>Nombre Caso de Prueba:</b>	administrar estudiantes
<b>Descripción:</b>	Ingresamos al sistema como administrativos, accedemos al módulo de administración e ingresamos a la administración de estudiantes. Luego adicionamos un estudiante, modificamos sus datos y lo eliminamos del sistema.
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", id nuevo estudiante: "2106073", nombre:"Pablo Torrez Torrez", carrera:"comun"
<b>Resultado esperado:</b>	El nuevo estudiante es adicionado correctamente, luego sus datos son modificados y finalmente este estudiante es eliminado del sistema sin problemas
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 17: Caso de prueba "administrar estudiantes"**  
Fuente: [Elaboración propia]

<b>Caso de Prueba N° 10</b>	
<b>Nombre Caso de Prueba:</b>	administrar docentes
<b>Descripción:</b>	Ingresamos al sistema como administrativos, accedemos al módulo de administración e ingresamos a la administración de docentes. Luego adicionamos un nuevo docente, modificamos sus datos y lo eliminamos del sistema.
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", id nuevo docente: "1234567", nombre:"Freddy Paredes"
<b>Resultado esperado:</b>	El nuevo docente es adicionado correctamente, luego sus datos son modificados y finalmente este docente es eliminado del sistema sin problemas
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 18: Caso de prueba "administrar docentes"**  
Fuente: [Elaboración propia]

<b>Caso de Prueba N° 11</b>	
<b>Nombre Caso de Prueba:</b>	administrar cursos
<b>Descripción:</b>	Ingresamos al sistema como administrativos, accedemos al módulo de administración e ingresamos a la administración de cursos. Luego adicionaremos un curso que posteriormente modificaremos. adicionamos un estudiante, lo eliminamos del curso y finalmente eliminaremos todo el curso
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: "tony", sigla del nuevo curso: "inf-111", paralelo:"A", carrera:"Telecomunicaciones", materia:"introducción a la programación", id docente:"1234567", id estudiante del curso:"6543210"

<b>Resultado esperado:</b>	El curso es creado correctamente, el estudiante es adicionado y posteriormente eliminado del curso. Finalmente el curso es eliminado sin presentar ningún problema
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 19: Caso de prueba “administrar cursos”**  
Fuente: [Elaboración propia]

Caso de Prueba N° 12	
<b>Nombre Caso de Prueba:</b>	administrar estilos
<b>Descripción:</b>	Ingresamos al sistema como administrativos, accedemos al módulo de administración e ingresamos a la administración de estilos. Luego adicionaremos un estilo que después eliminaremos
<b>Condiciones de Ejecución:</b>	Servidor activo, cliente con sistema operativo Windows XP, navegador Internet Explorer 5
<b>Entradas:</b>	usuario: “tony”, nombre del estilo: “estilo nuevo”, archivo del estilo:”estilo_nuevo.css”, archivo de vista previa:”estilo_nuevo.bmp”
<b>Resultado esperado:</b>	El nuevo estilo es adicionado al sistema correctamente, y al eliminarlo no se presentó ningún problema
<b>Evaluación:</b>	La prueba fue superada con éxito

**Tabla 20: Caso de prueba “administrar estilos”**  
Fuente: [Elaboración propia]

### 3.1.1.5 PRUEBAS DE SEGURIDAD

Las pruebas de seguridad se llevaron a cabo tomando en cuenta los criterios de seguridad observados en el Capítulo II, como resultado de estas pruebas se propusieron las políticas de seguridad que se muestran en el punto 3.3.2.

### 3.1.1.6 PRUEBAS DE CONFIGURACIÓN

La siguiente actividad consistió en implementar la aplicación Web en diferentes configuraciones, utilizando diferentes sistemas operativos y navegadores Web. Los resultados de esta prueba se los puede apreciar en la siguiente tabla:

SISTEMA OPERATIVO	NAVEGADOR	RESULTADOS
Windows XP Service Pack 2	Internet Explorer 6.0	No se presentó ningún problema
	Netscape Navigator 9.0.0.6	Se encontraron problemas de presentación, pero se corrigieron rápidamente. Estos problemas fueron: <ul style="list-style-type: none"> <li>- las palabras acentuadas no se mostraban correctamente</li> <li>- problema con JavaScript al manipular propiedades de los formularios</li> </ul>
	Mozilla Firefox 2.0.0.14	Se presentaron los mismos problemas que con el navegador de Netscape, y estos se corrigieron satisfactoriamente

Windows Vista Home Premium	Internet Explorer 7.0.6	Esta prueba se la realizo luego de corregir los errores anteriores y no se presentó ningún problema.
Linux RedHat 9	Mozilla 1.2.1	No se presento ningún problema
Linux Ubuntu 7.04	Mozilla Firefox 2.0.0.3	Se mostraron alertas de seguridad, que advertirán el envío de información no cifrada

**Tabla 21: Resultados de la implementación del sistema en diferentes configuraciones**  
Fuente: [Elaboración propia]

### 3.1.1.7 PRUEBA DE IMPLEMENTACIÓN

Como última actividad se probó la aplicación Web con una población de usuarios finales: estudiantes, docentes y administrativos. Esta actividad se desarrolló entre el 4 y el 14 de junio del 2008 para la materia de Electrónica I. Con la colaboración del docente de la materia se elaboraron los contenidos que se ingresaron al sistema instalado en un servidor de intranet ubicado en la sala de Internet de la biblioteca del instituto.

### 3.3.2 Políticas de seguridad

Como parte de la etapa de cierre del proyecto se propusieron las siguientes políticas de seguridad para el sistema, estas se tomaron en base a los requerimientos de la institución y a la bibliografía revisada. Las siguientes recomendaciones corresponden a la seguridad física del sistema:

- Se recomienda mantener al servidor Web en un lugar físico seguro, para que los usuarios no autorizados no puedan tener acceso a él.
- Se debe recomendar a los usuarios mantener su contraseña y nombre de usuario como un secreto y guardarlos de forma segura. En caso de olvido, vía correo electrónico puede solicitar al administrador del sistema que le recuerde sus datos de usuario. En el caso de que se sospeche una suplantación el usuario puede solicitar la eliminación de su cuenta y registrarse nuevamente con un nombre de usuario y contraseña diferentes.

También se ponen a consideración del administrador del Servidor Web las siguientes recomendaciones que son parte de la seguridad lógica del sistema:

- Realizar copias de seguridad a la base de datos 3 veces por semestre, se sugiere que esta actividad se desarrolle previo a las épocas de exámenes, pues se prevé que en estas fechas la afluencia de usuarios será mayor.

- Se debe proteger el servidor Web y todos los demás equipos de la misma red con contraseñas rigurosas.
- Recordar a los usuarios que al enviar ficheros al servidor debe tomarse en cuenta el tamaño máximo (3Mb) y se debe llenar los datos correctamente para evitar problemas en la presentación de los mismos.
- Establecer el tamaño máximo para el envío de archivos a 3Mb, esto se logra configurando el archivo php.ini.
- Se recomienda al administrador del sistema que la carpeta de administrativos funcione fuera del alcance de otros usuarios, y que los administrativos puedan acceder al sistema mediante intranet.
- Validar frecuentemente la carpeta de contenidos, eliminando aquellos archivos que no se encuentran registrados en la base de datos.
- Proteger la Base de Datos, ejecutando el servidor de Base de Datos únicamente con los privilegios necesarios.
- Cerrar los puertos que no se utilizan y desactivar los servicios no usados.
- Ejecutar en el Servidor un programa Firewall que controle el flujo de información y el envío de archivos hacia el Servidor.

### 3.3.3 Métricas de la aplicación Web

Luego de concluir con las pruebas a la aplicación Web, se procedió a realizar las siguientes mediciones:

*i) Para un estudiante:*

Número de páginas Web estáticas	NPE = 4
Número de páginas Web dinámicas	NPD = 14
Número de vínculos internos de la página	NVI = 40
Número de objetos de datos persistentes	NOP = 11

*ii) Para un docente:*

Número de páginas Web estáticas	NPE = 4
Número de páginas Web dinámicas	NPD = 23
Número de vínculos internos de la página	NVI = 47
Número de objetos de datos persistentes	NOP = 11

iii) Para un administrativo:

Número de páginas Web estáticas	NPE = 5
Número de páginas Web dinámicas	NPD = 52
Número de vínculos internos de la página	NVI = 80
Número de objetos de datos persistentes	NOP = 11

En base a estas medidas se obtuvieron las siguientes métricas:

$$\text{Índice de personalización de un estudiante} = \text{NPD} / (\text{NPE} + \text{NPD}) = 0.77$$

$$\text{Índice de personalización de un docente} = 0.85$$

$$\text{Índice de personalización de un administrativo} = 0.91$$

Se puede observar claramente que el grado de personalización de un administrativo es el más alto pues este cuenta con más privilegios que otros usuarios. En el otro extremo se encuentra el estudiante que tiene un grado de personalización inferior pero aceptable.

$$\text{Grado de acoplamiento para un estudiante} = \text{NVI} / (\text{NPE} + \text{NPD}) = 1.42$$

$$\text{Grado de acoplamiento para un docente} = 1.74$$

$$\text{Grado de acoplamiento para un administrativo} = 1.40$$

Como se puede observar el grado de acoplamiento para los tres tipos de usuario es superior a 1, lo que significa un buen diseño de la navegación en la aplicación Web. Esta métrica también nos muestra que existe entre uno a dos vínculos por cada una de las páginas Web.

$$\text{Grado de complejidad para un estudiante} = (\text{NPE} + \text{NPD}) / \text{NOP} = 1.63$$

$$\text{Grado de complejidad para un docente} = 2.45$$

$$\text{Grado de complejidad para un administrativo} = 5.19$$

Como el número de objetos de datos persistentes se tomaron a partir del número de objetos del modelo conceptual, es el mismo para cualquier tipo de usuario, se puede observar una clara diferencia en cuanto a la complejidad del sistema para cada tipo de usuario. Un estudiante tiene entre uno y dos páginas Web asignadas a cada objeto persistente. Un docente tiene entre dos y tres páginas Web por un objeto persistente. Y un administrativo cuenta con 5 páginas Web por cada objeto de contenido.

### 3.3.4 Diseño de la ayuda

El diseño de la ayuda se lo realizó en base a los casos de uso del sistema. La estrategia para implementar esta ayuda fue construir un manual de usuario, el cual se publicó en el Ambiente Educativo Virtual como un curso disponible para todos los usuarios.

### 3.3.5 Implementación

Debido a la poca población estudiantil que existe actualmente en el instituto, no es necesario hacer una gran inversión para implementar el sistema. Las características necesarias para la implementación de un servidor que aloje al Ambiente Educativo Virtual son las siguientes:

- Microprocesador Intel Core2Duo de 2.5Ghz o superior
- Memoria RAM de 2 Gb
- Disco duro de 80 Gb
- Tarjeta de red Fast Ethernet

Al mismo tiempo el servidor debe tener instalado los siguientes programas:

- Sistema operativo Linux Ubuntu 7.04, u otro que no tenga costo de licencia
- Servidor de paginas Web Apache 2.0.59 o superior
- Servidor de base de datos MySQL 5.0.24a o superior
- Lenguaje de programación PHP 5.1.6 o superior

Estas características se obtuvieron a partir de las siguientes mediciones:

**Perfil de uso (para 1 usuario):** CPU = 21.32 Mhz, Memoria = 7804 Kb

Tomando en cuenta que son 200 estudiantes y 15 docentes, la carga máxima para el servidor sería:

$$\text{CPU} = 21.32\text{Mhz} * 215 = 4583.8 \text{ Mhz}$$

$$\text{Memoria} = 7804\text{Kb} * 215 = 1677860 \text{ Kb}$$

Para el cálculo de la capacidad de disco se tomo en cuenta la capacidad de almacenamiento de cada curso que es de 100Mb. Sabiendo que en el instituto se dictan 74 materias, y como máximo existen 2 paralelos por materia, la capacidad necesaria de disco sería:

$$74 * 2 * 100\text{Mb} = 14800\text{Mb}$$

Posteriormente se calculo la velocidad de transmisión que debería tener la tarjeta de red. En caso de que 215 usuarios accedieran al mismo tiempo a un contenido de 3Mb, con un tiempo de espera máximo de 1 minuto, la velocidad necesaria es:

$$(215 * 3\text{Mb}) / 60 \text{ seg} = 10.75 \text{ Mbps}$$



# CAPÍTULO IV MODELADO DEL SISTEMA

## 4.1 MODELADO DE LOS REQUERIMIENTOS

Para modelar los requerimientos de los usuarios y describir el comportamiento de la aplicación se elaboraron los siguientes diagramas de casos de uso:

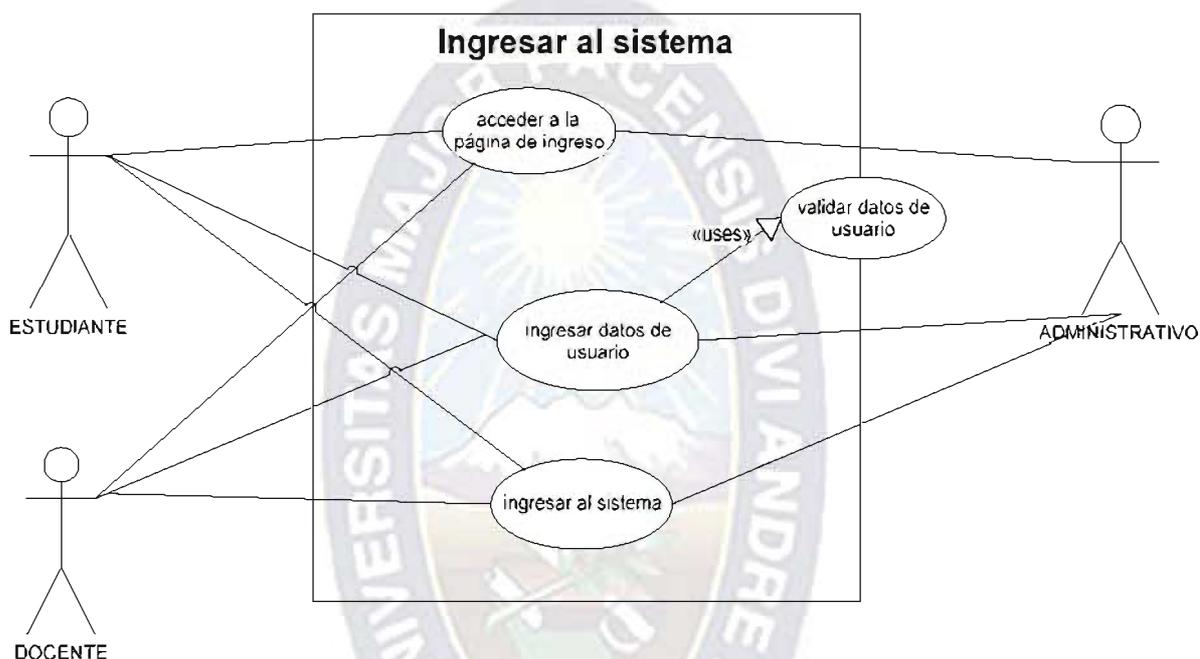


Figura 15: Diagrama de casos de uso “registrarse como nuevo usuario”  
Fuente: elaboración propia

Caso de uso	Registrarse como nuevo usuario
Actores	Estudiante, Docente
Descripción	En este caso de uso el usuario (estudiante o docente) se registra como nuevo usuario del Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario accede a la página de ingreso del AEV</li> <li>- el usuario ingresa al módulo de registro</li> <li>- el usuario ingresa sus datos personales</li> <li>- el sistema verifica la validez de estos datos</li> <li>- el usuario ingresa su nombre de usuario y contraseña</li> <li>- el sistema verifica si no existe otro usuario con el mismo nombre, y almacena la información en la base de datos.</li> </ul>

Flujo alternativo	<ul style="list-style-type: none"> <li>- el nombre de usuario ya existe, entonces se muestra un mensaje de error</li> <li>- en caso de ser un administrativo el registro se debe realizar personalmente con el administrador del sistema</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario sea miembro de la institución, estudiante o docente</li> <li>- el usuario no ha sido registrado anteriormente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- el nuevo usuario está correctamente registrado</li> </ul>

**Tabla 22: Descripción de casos de uso “registrarse como nuevo usuario”**  
Fuente: [Elaboración propia]



**Figura 16: Diagrama de casos de uso “ingresar al sistema”**  
Fuente: [Elaboración propia]

Caso de uso	Ingresar al sistema
Actores	Estudiante, Docente, Administrativo
Descripción	Este caso de uso el usuario muestra cómo un usuario ingresa al sistema
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario accede a la página de ingreso del sistema</li> <li>- el usuario ingresa sus datos de usuario (nombre de usuario, contraseña y tipo de usuario)</li> <li>- el sistema valida los datos del usuario</li> <li>- si los datos son correctos el usuario ingresa al sistema</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- el nombre de usuario o la contraseña son incorrectos, entonces se muestra un mensaje de error</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario se haya registrado correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- el usuario ingresa al sistema correctamente</li> </ul>

**Tabla 23: Descripción de casos de uso “ingresar al sistema”**  
Fuente: [Elaboración propia]

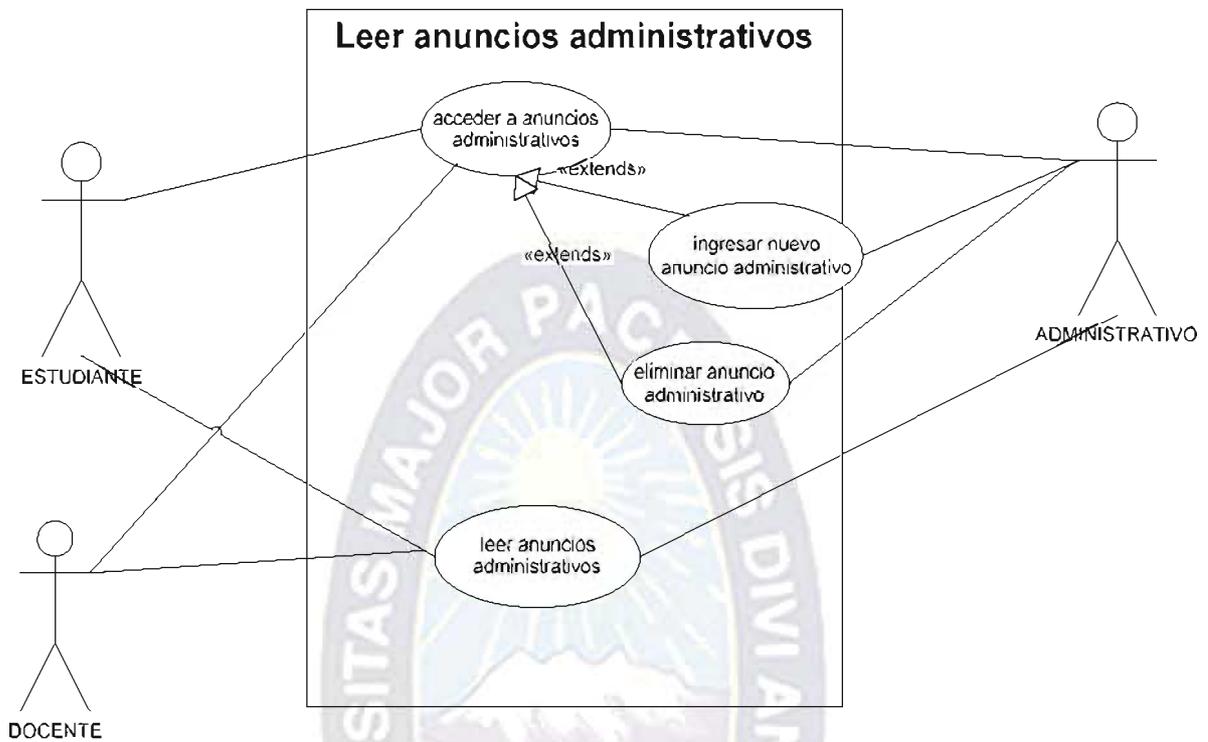
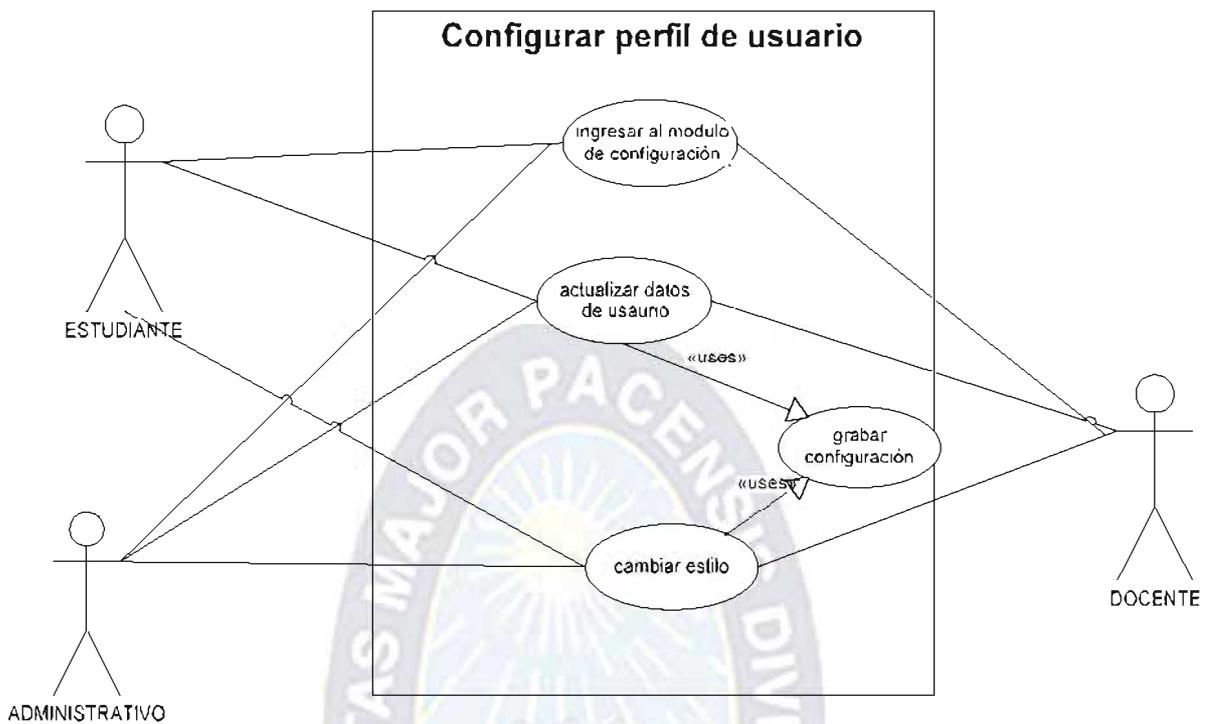


Figura 17: Diagrama de casos “leer anuncios administrativos”  
Fuente: [Elaboración propia]

Caso de uso	Leer anuncios administrativos
Actores	Estudiante, Docente, Administrativo
Descripción	En este caso de uso se muestra como un usuario puede acceder a los anuncios administrativos.
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de anuncios administrativos</li> <li>- el usuario observa los anuncios publicados</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- si el usuario es un administrativo, también puede ingresar nuevos anuncios y eliminar anuncios antiguos</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- ninguna</li> </ul>

Tabla 24: Descripción de caso de uso “leer anuncios administrativos”  
Fuente: [Elaboración propia]



**Figura 18: Diagrama de casos de uso “configurar perfil de usuario”**  
**Fuente: [Elaboración propia]**

Caso de uso	Configurar perfil de usuario
Actores	Estudiante, Docente, Administrativo
Descripción	En este caso de uso el usuario configura su perfil, ya sea actualizando sus datos de usuario o cambiando su estilo
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de configuración</li> <li>- el usuario actualiza sus datos de usuario</li> <li>- el usuario cambia su estilo</li> <li>- el sistema guarda los cambios de la nueva configuración e informa al usuario</li> </ul>
Flujo alternativo	- ninguno
Precondiciones	- el usuario ingresa al sistema correctamente
Poscondiciones	- los cambios se almacenaron correctamente

**Tabla 25: Descripción de caso de uso “configurar perfil de usuario”**  
**Fuente: [Elaboración propia]**

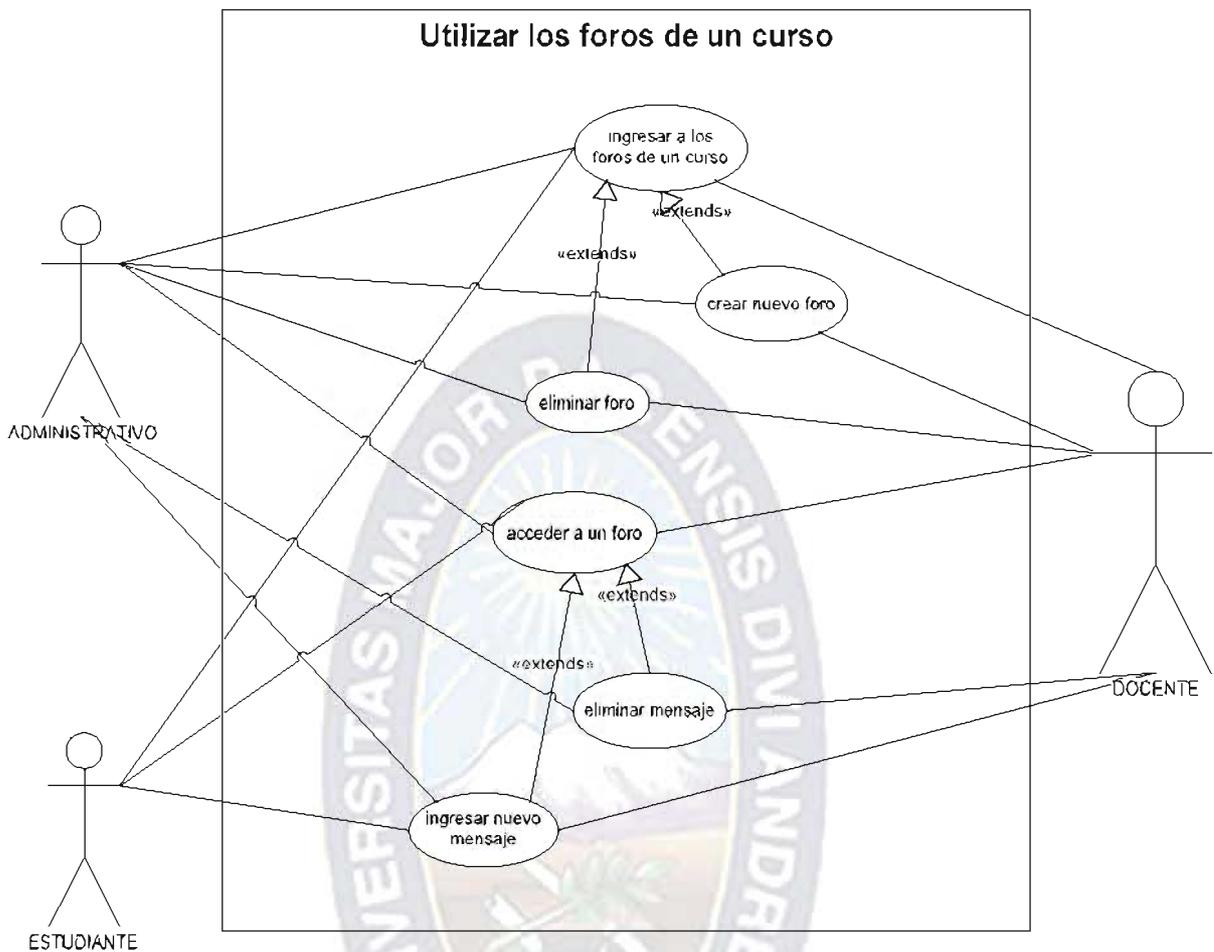


Figura 19: Diagrama de casos de uso “utilizar los foros de un curso”

Fuente: [Elaboración propia]

Caso de uso	utilizar los foros de un curso
Actores	Estudiante, Docente, Administrativo
Descripción	En este caso de uso se muestra cómo el usuario utiliza los foros de un determinado curso
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa a los foros de un determinado curso</li> <li>- el usuario selecciona un foro y accede a este</li> <li>- el usuario observa los mensajes publicados en el foro</li> <li>- el usuario escribe un nuevo mensaje</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- si el usuario es un docente o un administrativo este puede además crear y eliminar un foro</li> <li>- si el usuario es un docente o un administrativo también puede eliminar los mensajes que considere</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> <li>- que el usuario haya ingresado al módulo de cursos</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- ninguna</li> </ul>

Tabla 26: Descripción de caso de uso “utilizar los foros de un curso”

Fuente: [Elaboración propia]

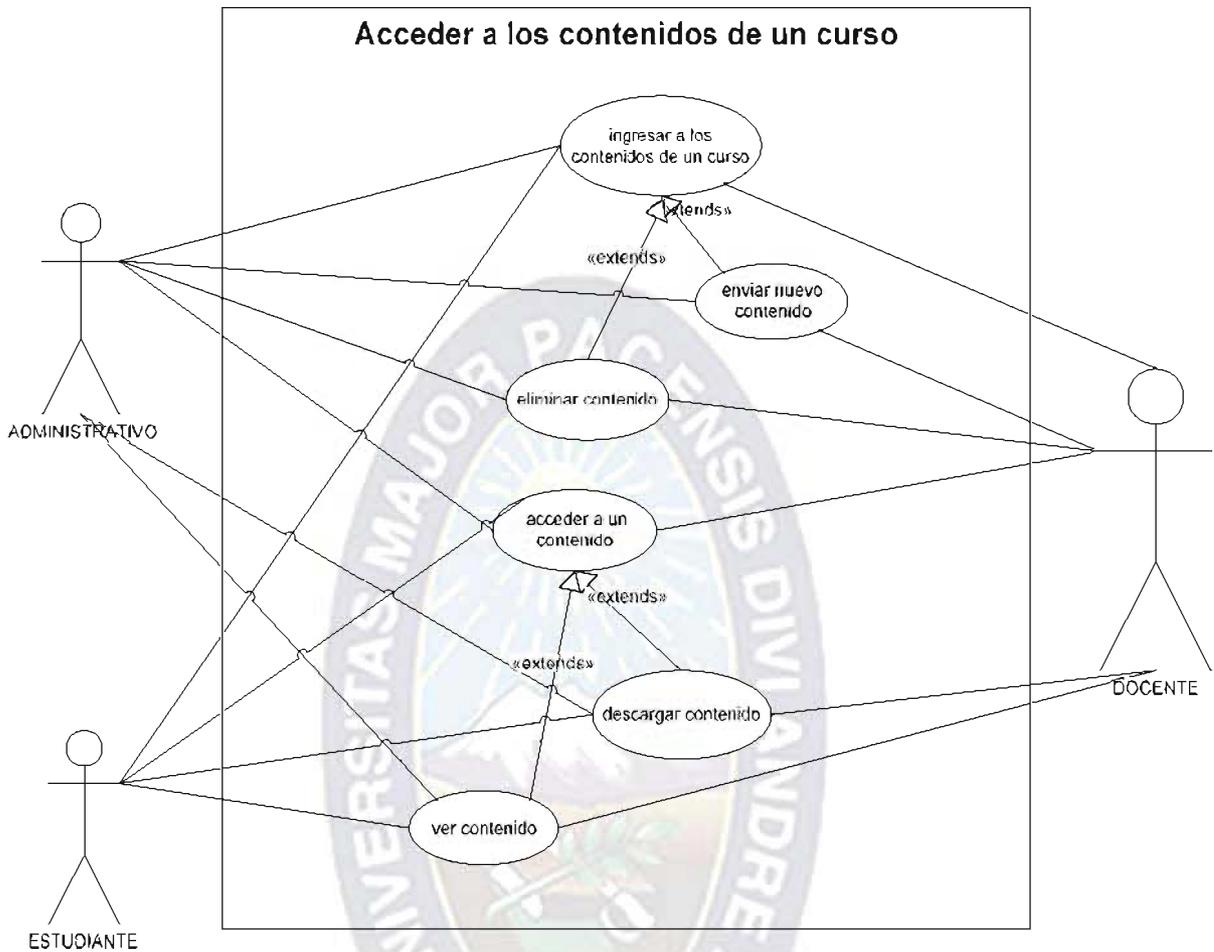


Figura 20: Diagrama de casos de uso “acceder a los contenidos de un curso”  
Fuente: [Elaboración propia]

Caso de uso	acceder a los contenidos de un curso
Actores	Estudiante, Docente, Administrativo
Descripción	En este caso de uso se puede observar como un usuario accede a los contenidos de determinado curso
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa a los contenidos de un determinado curso</li> <li>- el usuario selecciona un contenido y accede a este</li> <li>- el usuario observa el contenido o lo descarga para verlo en otra ocasión</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- si el usuario es un docente o un administrativo este puede además enviar y eliminar los contenidos que considere</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> <li>- que el usuario haya ingresado al módulo de cursos</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- ninguna</li> </ul>

Tabla 27: Descripción de caso de uso “acceder a los contenidos de un curso”  
Fuente: [Elaboración propia]

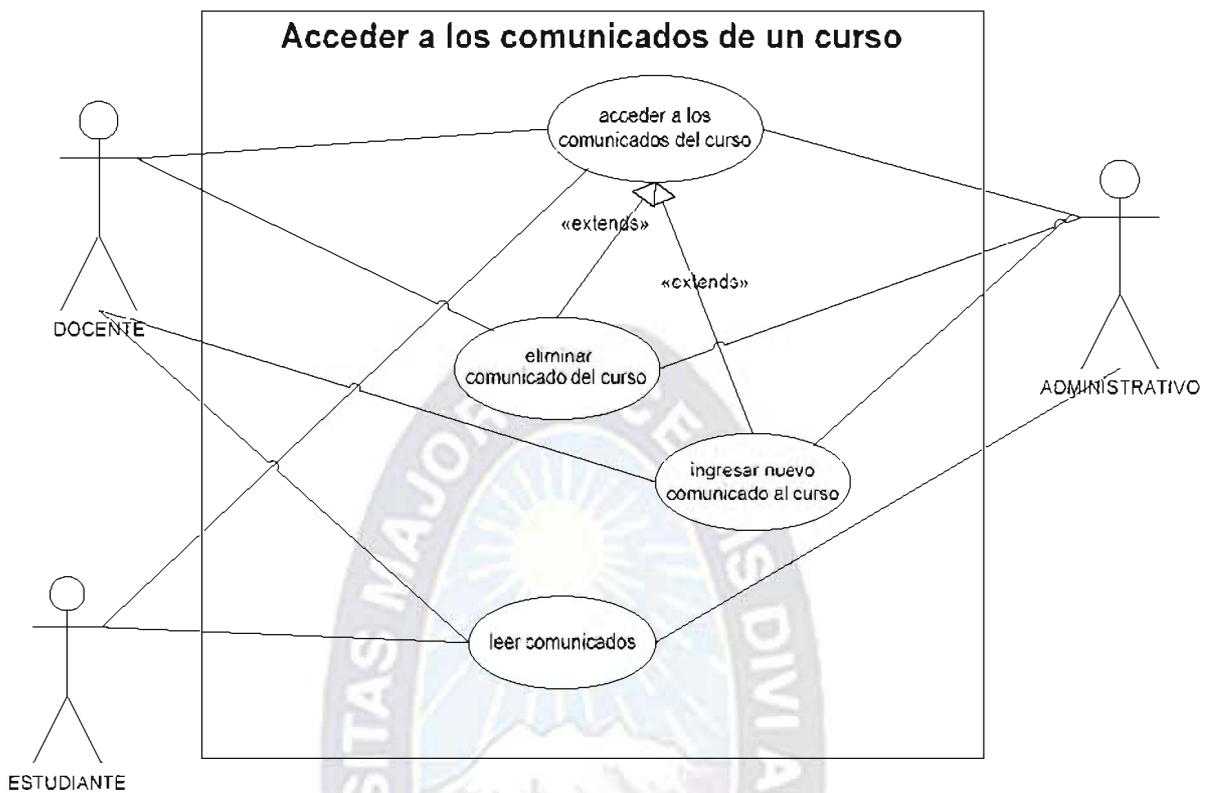


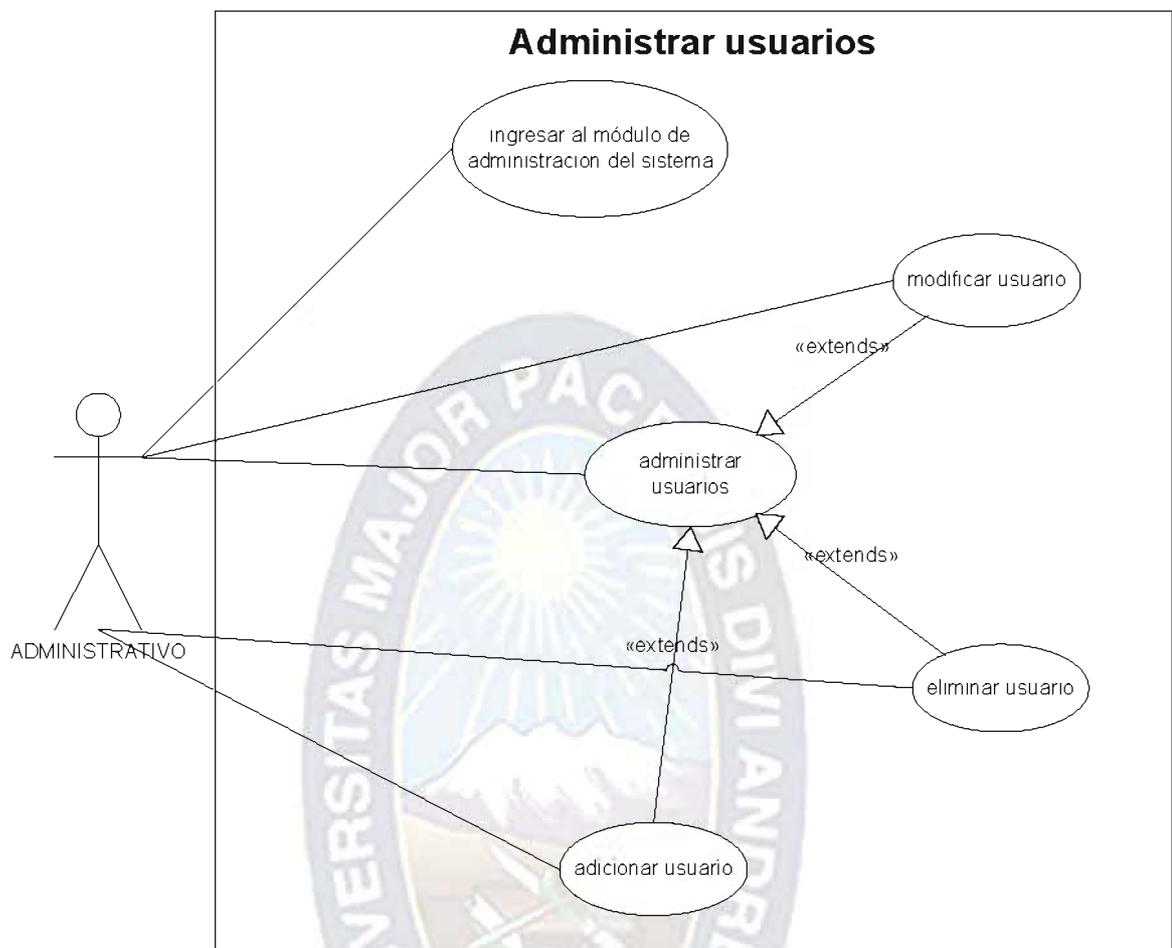
Figura 21: Diagrama de casos de uso “acceder a los comunicados de un curso”

Fuente: [Elaboración propia]

Caso de uso	acceder a los comunicados de un curso
Actores	Estudiante, Docente, Administrativo
Descripción	En este caso de uso se observa como un usuario utiliza los comunicados de un determinado curso
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario accede a los comunicados de un determinado curso</li> <li>- el usuario lee los comunicados publicados en el curso</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- si el usuario es un docente o un administrativo, además puede ingresar o eliminar comunicados</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> <li>- que el usuario haya ingresado al módulo de cursos</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- ninguno</li> </ul>

Tabla 28: Descripción de caso de uso “acceder a los comunicados de un curso”

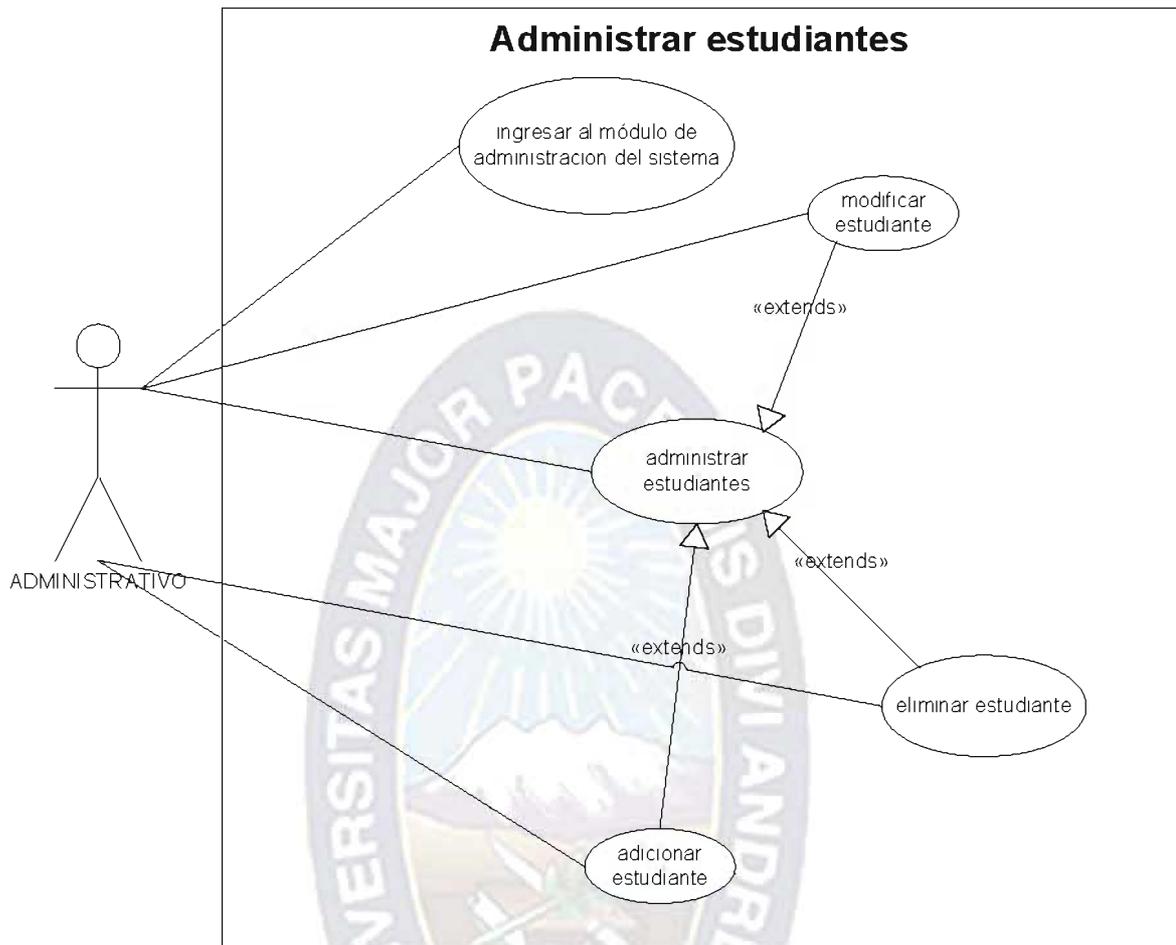
Fuente: [Elaboración propia]



**Figura 22: Diagrama de casos de uso “administrar usuarios”**  
Fuente: [Elaboración propia]

Caso de uso	Administrar usuarios
Actores	Administrativo
Descripción	En este caso de uso el administrativo gestiona a los usuarios del Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de administración del sistema</li> <li>- el usuario accede a la administración de los usuarios</li> <li>- el usuario adiciona, modifica o elimina un usuario</li> <li>- el sistema almacena los cambios en la base de datos</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- ninguno</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- los cambios se almacenan correctamente</li> </ul>

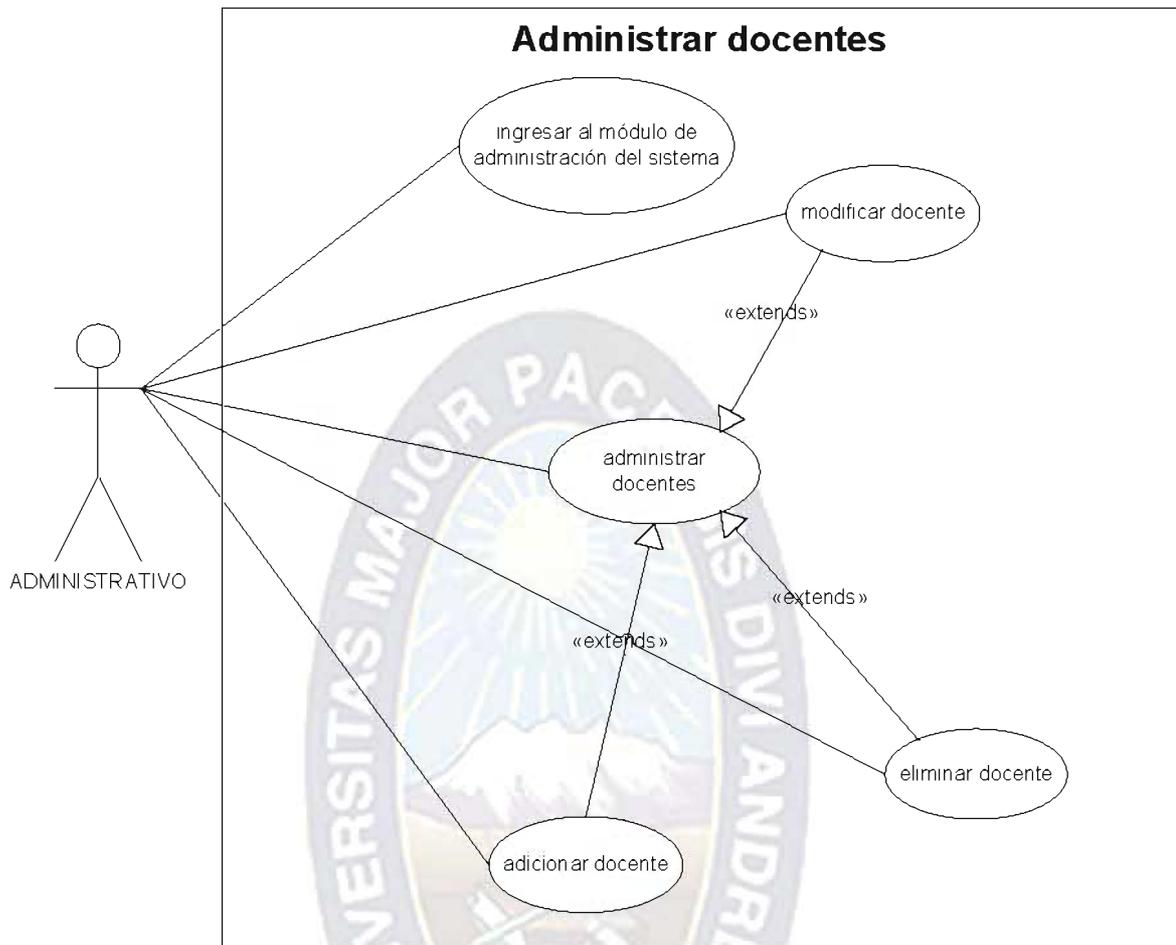
**Tabla 29: Descripción de caso de uso “administrar usuarios”**  
Fuente: [Elaboración propia]



**Figura 23: Diagrama de casos de uso “administrar estudiantes”**  
 Fuente: [Elaboración propia]

Caso de uso	administrar estudiantes
Actores	Administrativo
Descripción	En este caso de uso el administrativo gestiona a los estudiantes que pertenecen al Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de administración del sistema</li> <li>- el usuario accede a la administración de estudiantes</li> <li>- el usuario adiciona, modifica o elimina estudiantes</li> <li>- el sistema almacena los cambios en la base de datos</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- ninguno</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- los cambios se almacenan correctamente</li> </ul>

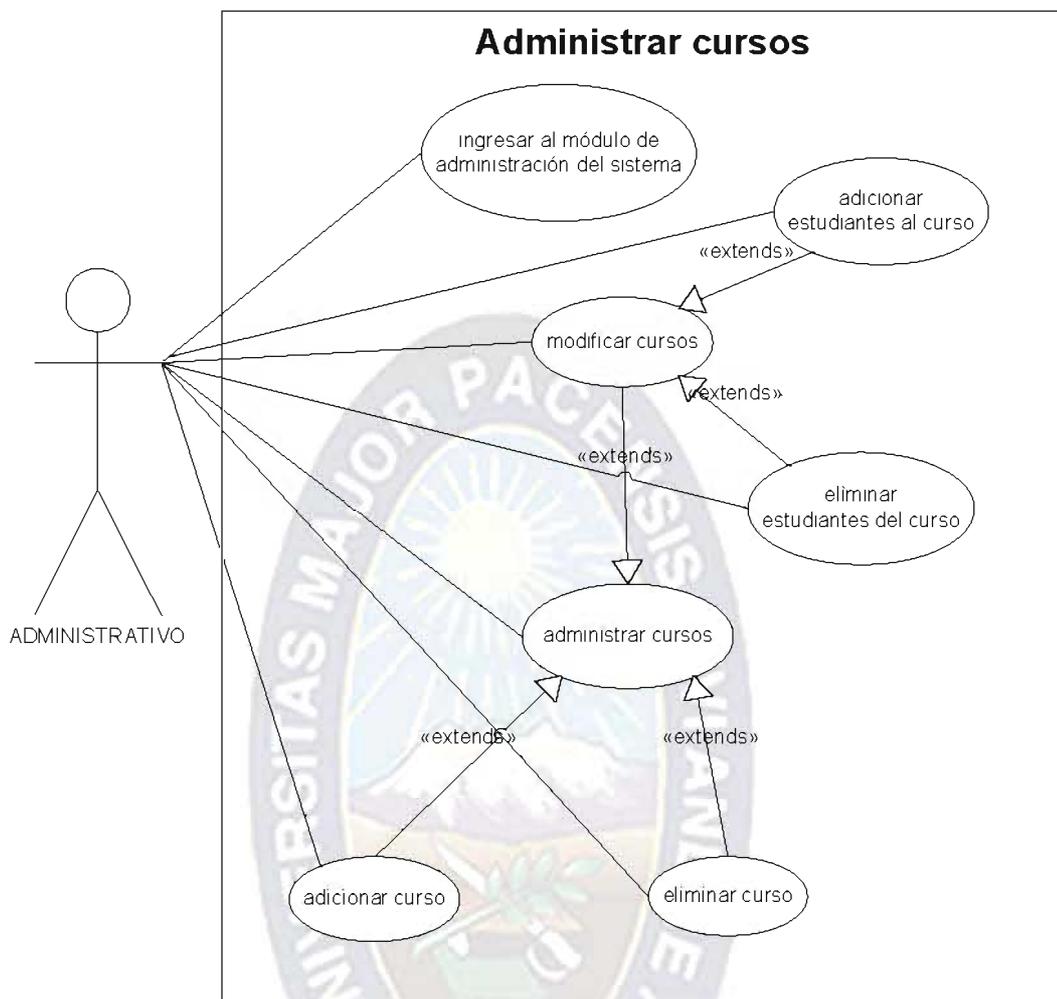
**Tabla 30: Descripción de caso de uso “administrar estudiantes”**  
 Fuente: [Elaboración propia]



**Figura 24: Diagrama de casos de uso “administrar docentes”**  
 Fuente: [Elaboración propia]

Caso de uso	administrar docentes
Actores	Administrativo
Descripción	En este caso de uso el administrativo gestiona a los docentes que pertenecen al Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de administración del sistema</li> <li>- el usuario accede a la administración de los docentes</li> <li>- el usuario adiciona, modifica o elimina un docente</li> <li>- el sistema almacena los cambios en la base de datos</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- ninguno</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- los cambios se almacenan correctamente</li> </ul>

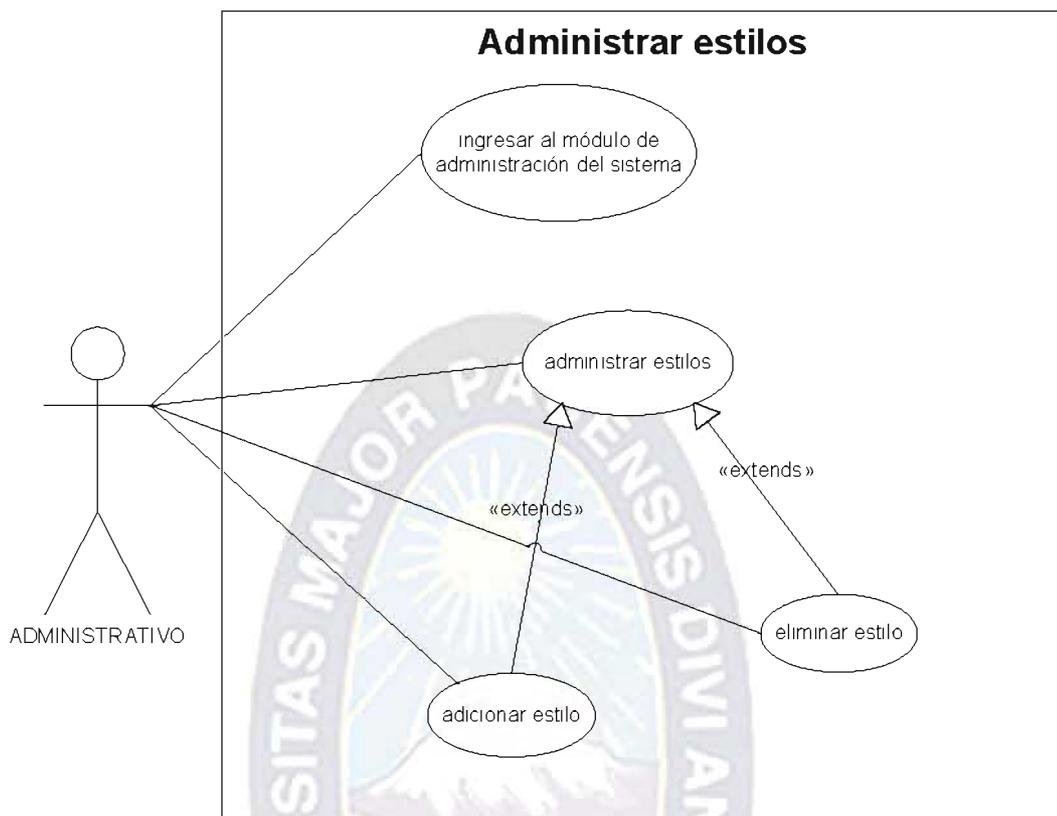
**Tabla 31: Descripción de caso de uso “administrar docentes”**  
 Fuente: [Elaboración propia]



**Figura 25: Diagrama de casos de uso “administrar cursos”**  
 Fuente: [Elaboración propia]

Caso de uso	Administrar cursos
Actores	Administrativo
Descripción	En este caso de uso el administrativo gestiona los cursos del Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de administración del sistema</li> <li>- el usuario accede a la administración de los cursos</li> <li>- el usuario adiciona, modifica o elimina un curso</li> <li>- el sistema almacena los cambios en la base de datos</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- el usuario además puede adicionar y eliminar los estudiantes que pertenecen a un curso</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- los cambios se almacenan correctamente</li> </ul>

**Tabla 32: Descripción de caso de uso “administrar cursos”**  
 Fuente: [Elaboración propia]



**Figura 26: Diagrama de casos de uso “administrar estilos”**

Fuente: [Elaboración propia]

Caso de uso	administrar estilos
Actores	Administrativo
Descripción	En este caso de uso el actor administra las hojas de estilo que se usarán en el Ambiente Educativo Virtual
Flujo de eventos básico	<ul style="list-style-type: none"> <li>- el usuario ingresa al módulo de administración del sistema</li> <li>- el usuario accede a la administración de estilos</li> <li>- el usuario adiciona o elimina una hoja de estilos</li> <li>- el sistema almacena los cambios en la base de datos</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>- ninguno</li> </ul>
Precondiciones	<ul style="list-style-type: none"> <li>- que el usuario haya ingresado al sistema correctamente</li> </ul>
Poscondiciones	<ul style="list-style-type: none"> <li>- los cambios se almacenan correctamente</li> </ul>

**Tabla 33: Descripción de caso de uso “administrar estilos”**

Fuente: [Elaboración propia]

## 4.2 MODELO CONCEPTUAL

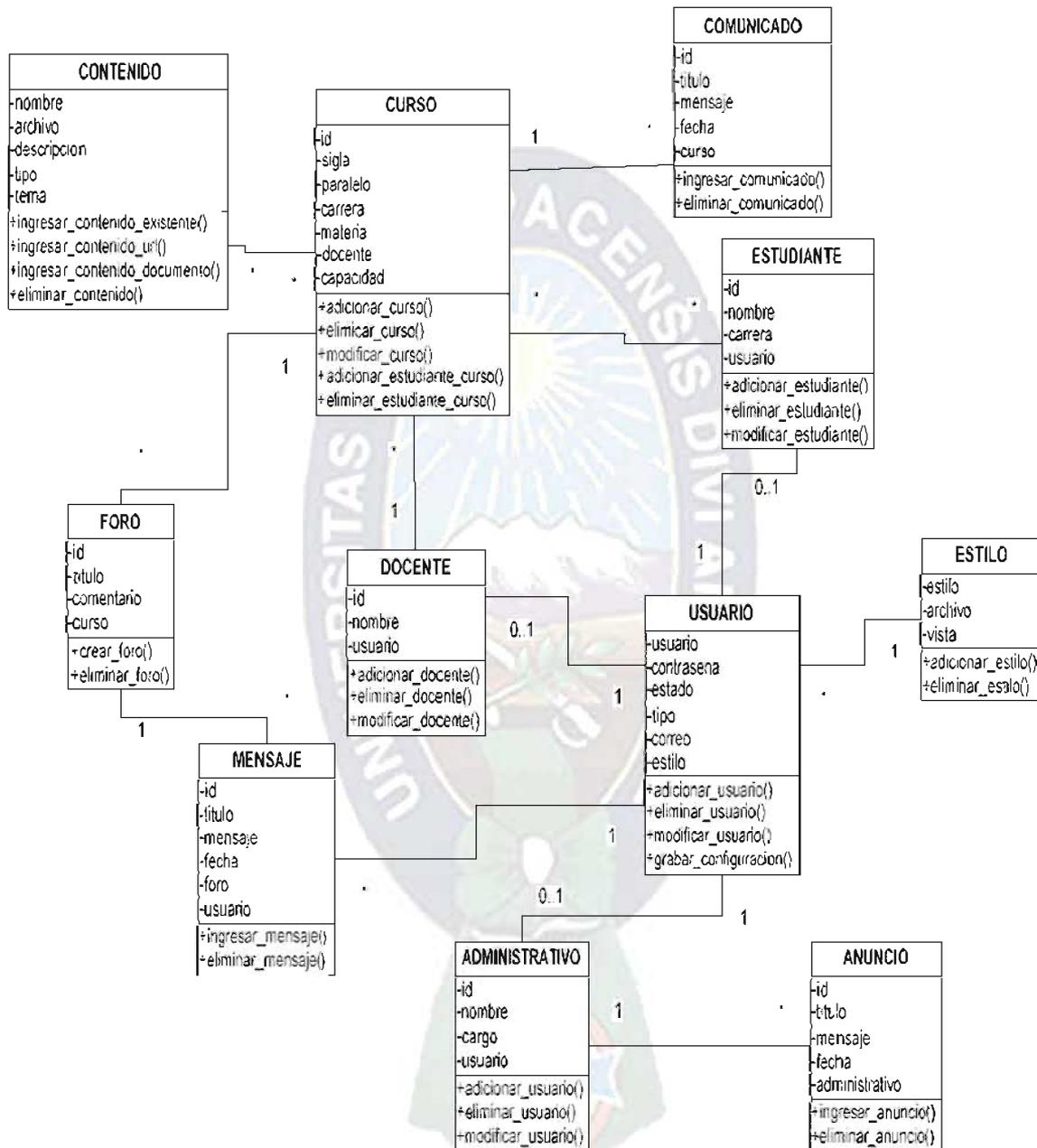


Figura 27: Diagrama de clases del modelo conceptual  
Fuente: [Elaboración propia]

### 4.3 MODELO DE NAVEGACIÓN

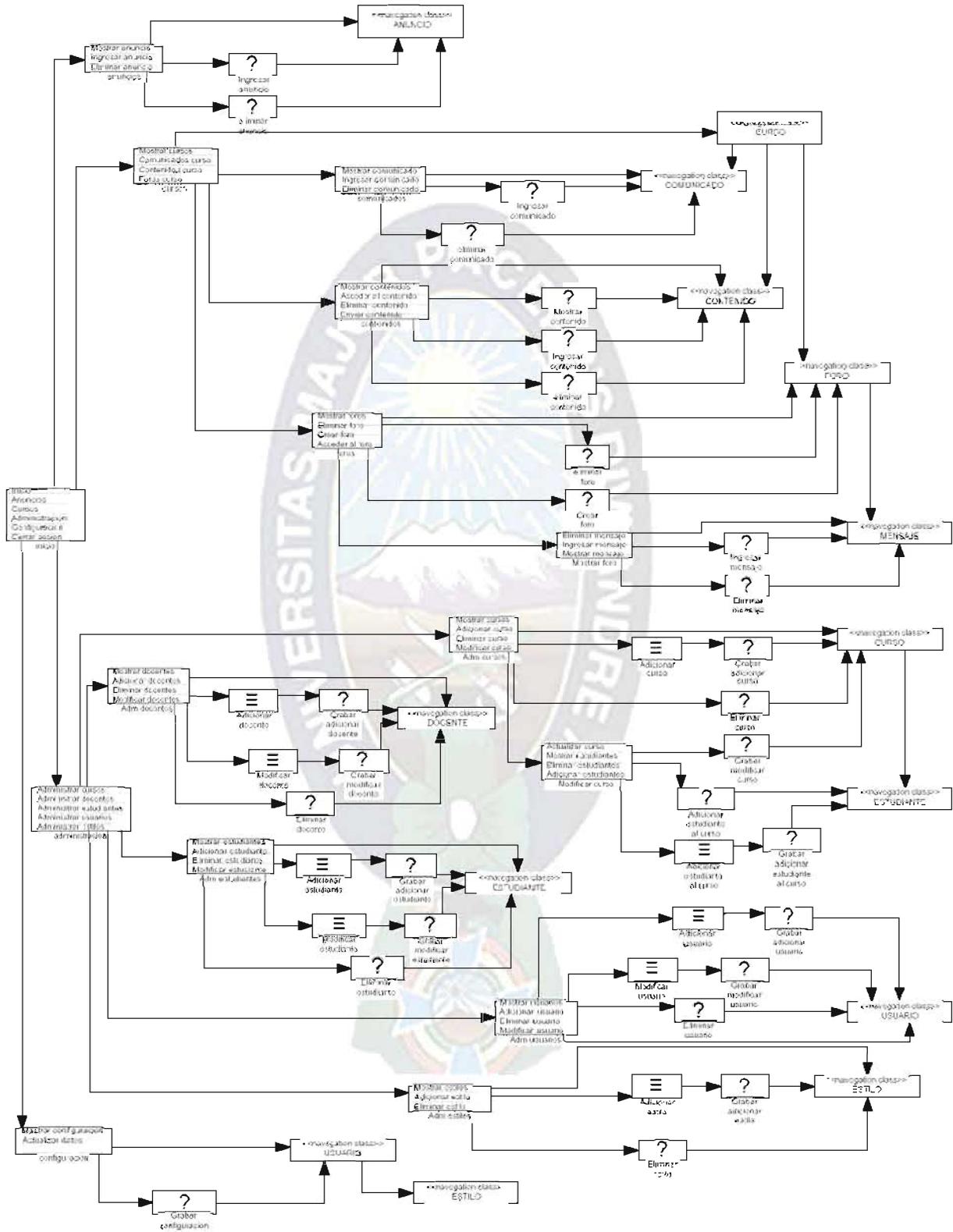
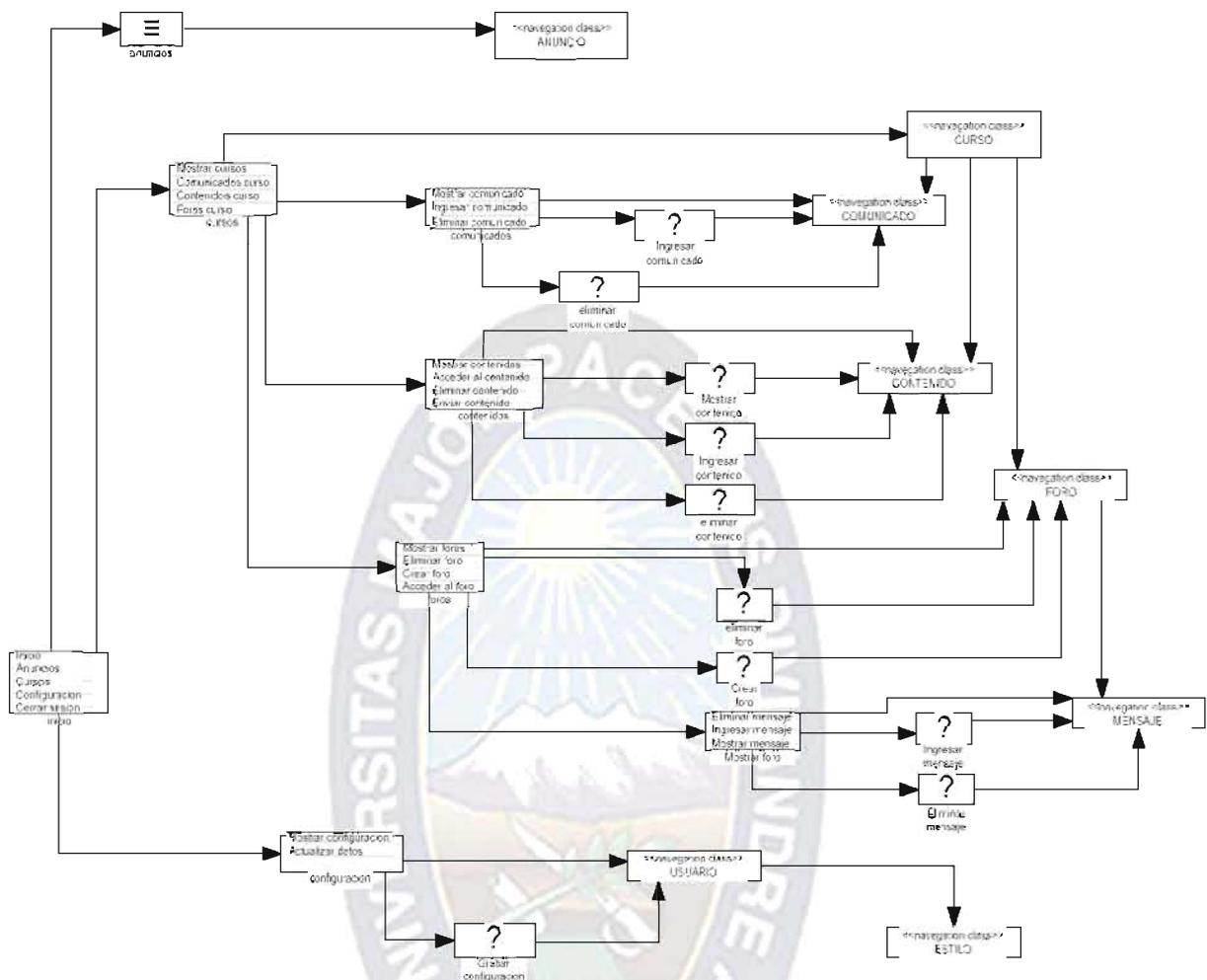


Figura 28: Modelo de navegación para un Administrativo  
Fuente: [Elaboración propia]



**Figura 29: Modelo de navegación para un Docente**  
Fuente: [Elaboración propia]

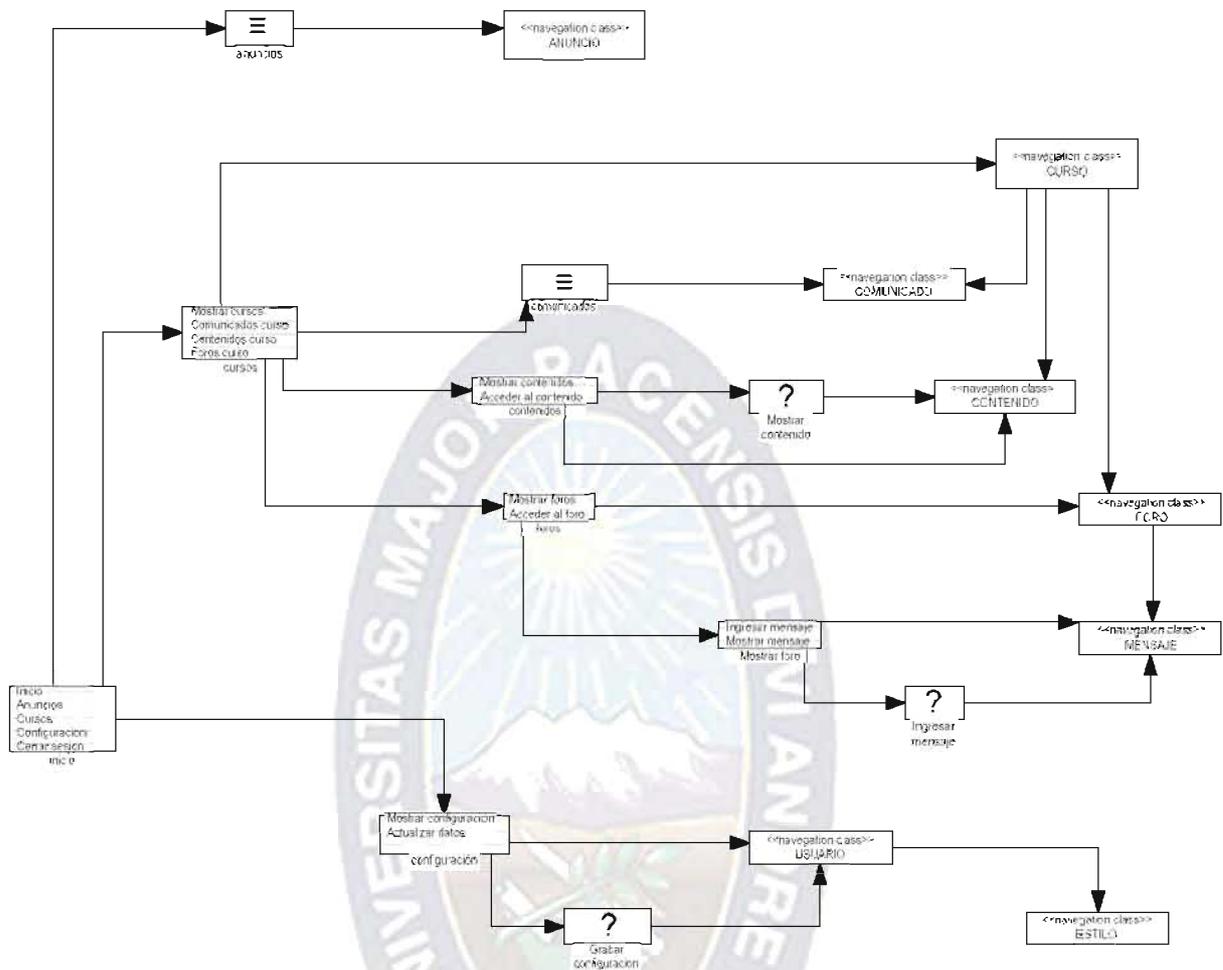


Figura 30: Modelo de navegación para un Estudiante  
Fuente: [Elaboración propia]

#### 4.4 MODELO DE PRESENTACIÓN

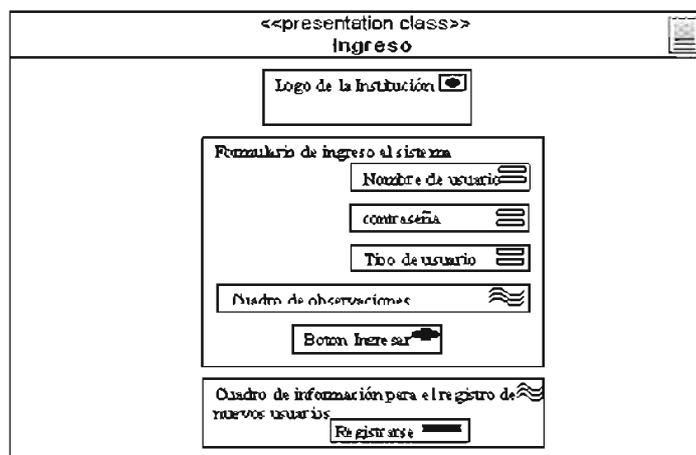


Figura 31: Interfaz de ingreso al AEV  
Fuente: [Elaboración propia]

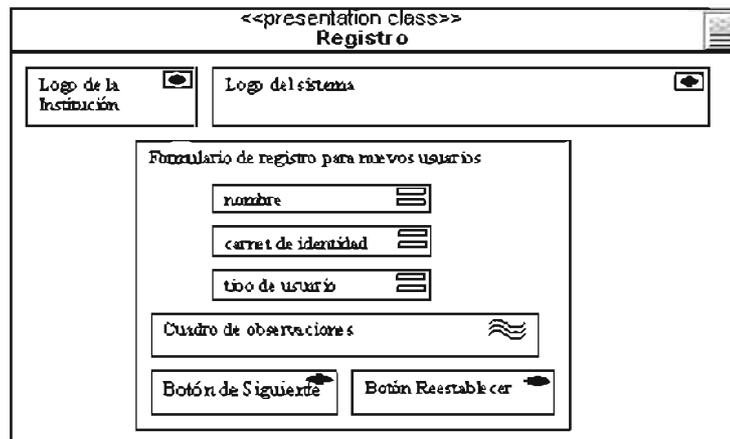


Figura 32: Interfaz de registro para usuarios nuevos  
Fuente: [Elaboración propia]

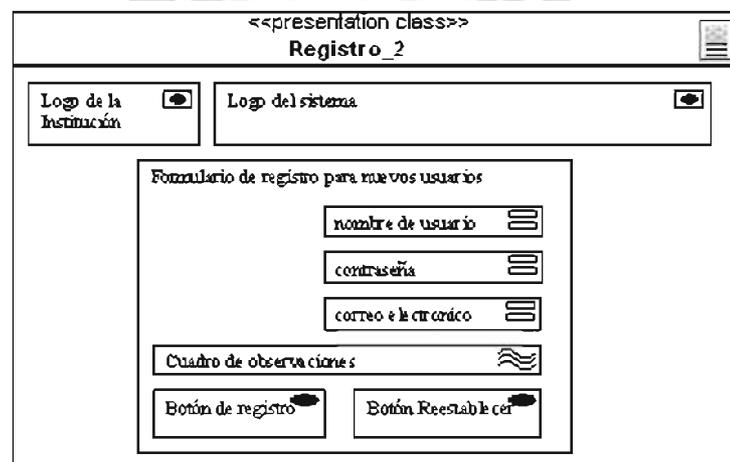


Figura 33: Interfaz de registro para usuarios nuevo (segundo paso)  
Fuente: [Elaboración propia]

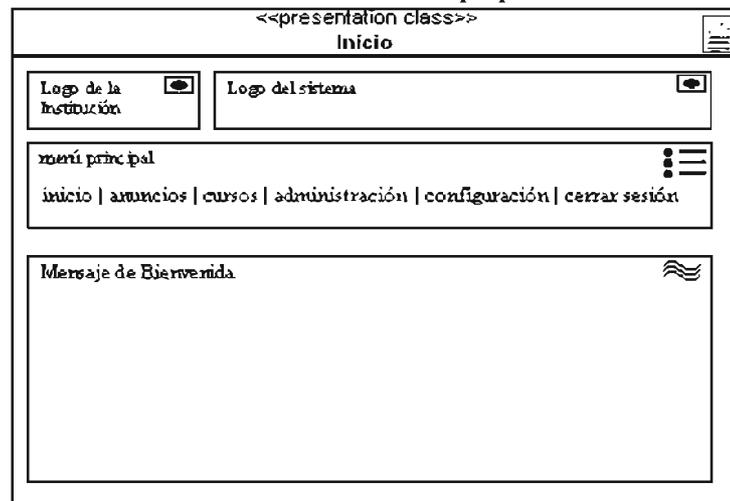


Figura 34: Interfaz de inicio  
Fuente: [Elaboración propia]

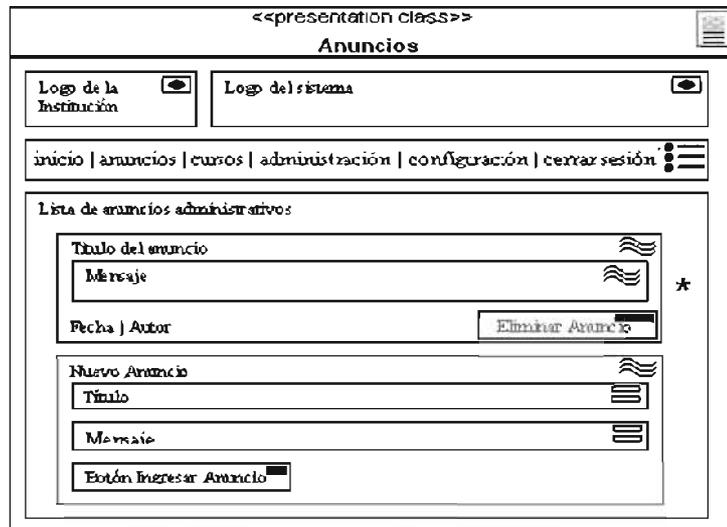


Figura 35: Interfaz para los anuncios administrativos  
Fuente: [Elaboración propia]

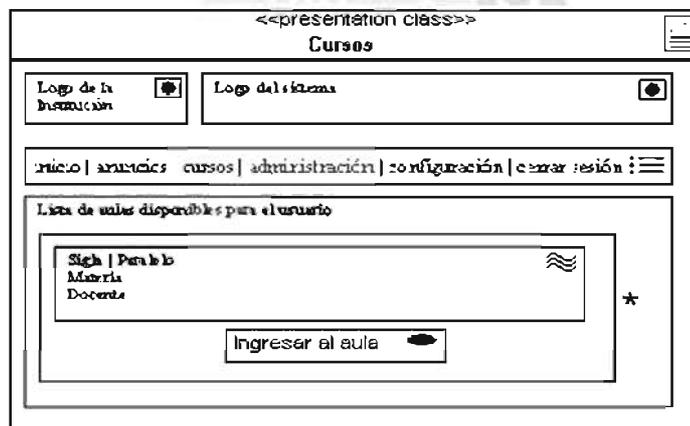


Figura 36: Interfaz de los cursos  
Fuente: [Elaboración propia]

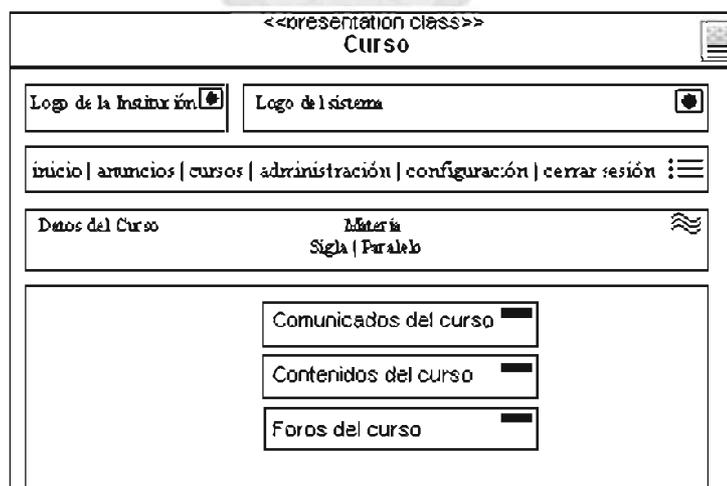


Figura 37: Interfaz dentro de un curso  
Fuente: [Elaboración propia]

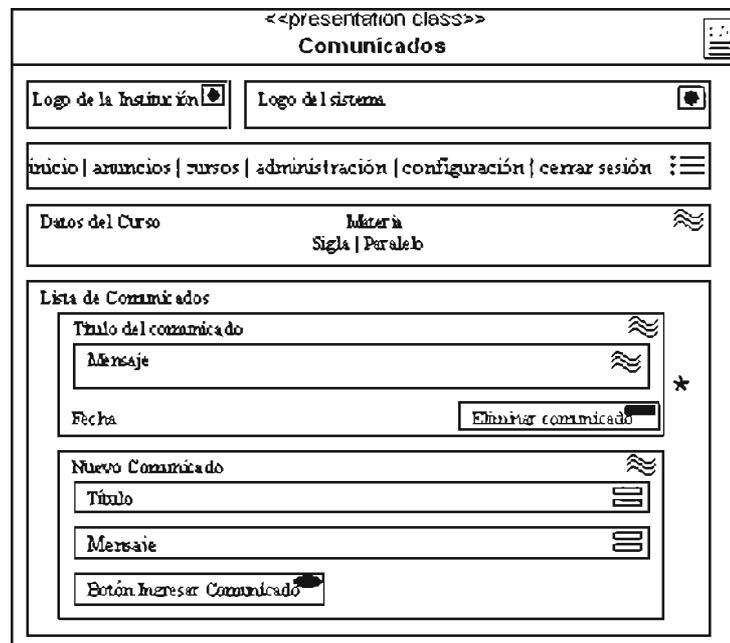


Figura 38: Interfaz de los comunicados  
Fuente: [Elaboración propia]

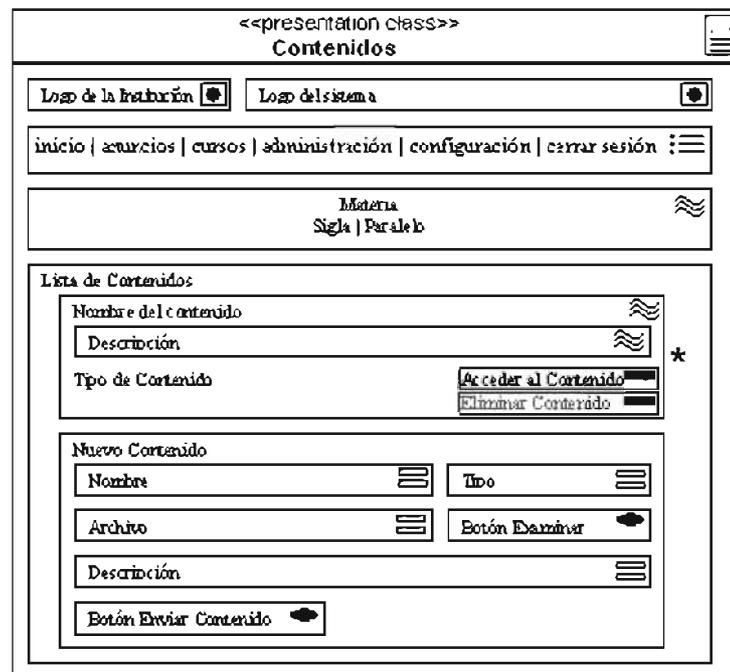


Figura 39: Interfaz de los contenidos  
Fuente: [Elaboración propia]

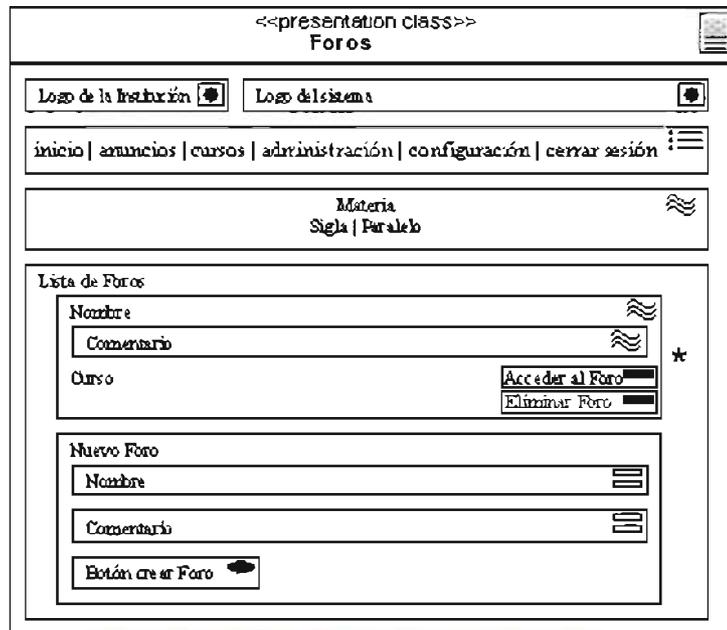


Figura 40: Interfaz de los foros  
Fuente: [Elaboración propia]

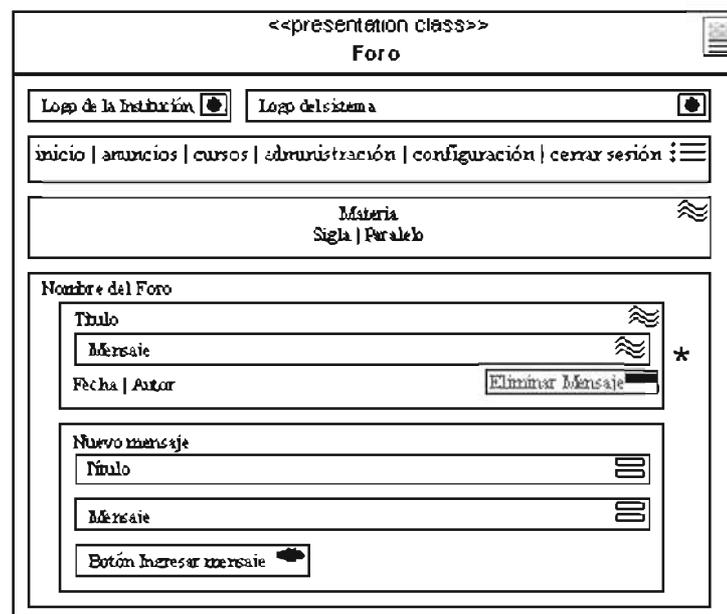


Figura 41: Interfaz dentro de un foro  
Fuente: [Elaboración propia]

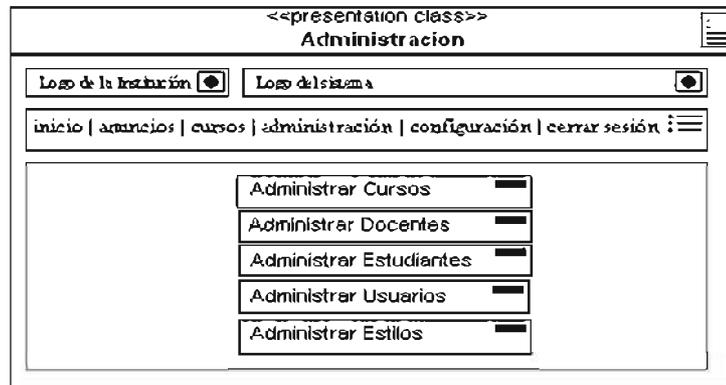


Figura 42: Interfaz de la administración  
Fuente: [Elaboración propia]

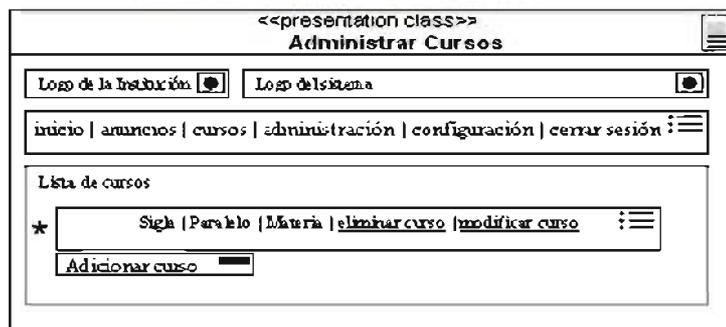


Figura 43: Interfaz para administrar cursos  
Fuente: [Elaboración propia]

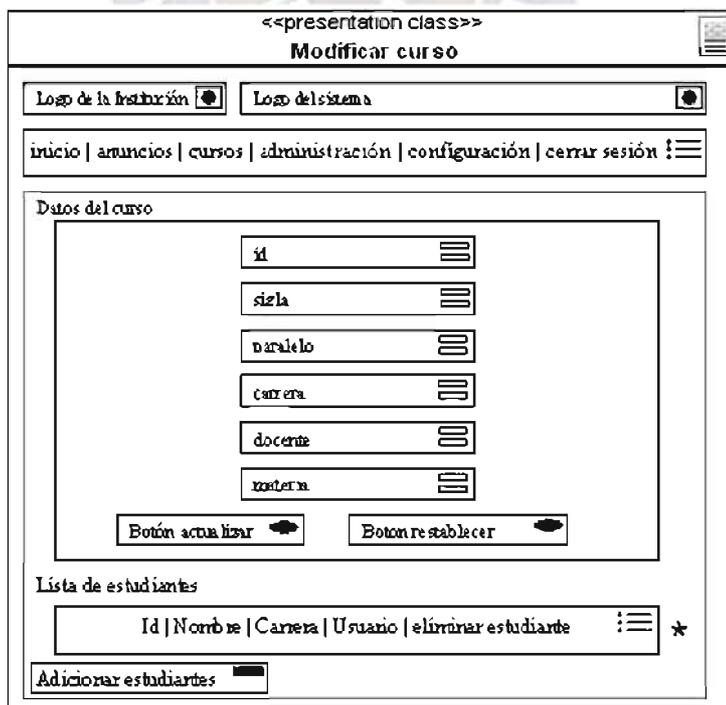


Figura 44: Interfaz para modificar un curso  
Fuente: [Elaboración propia]

<<presentation class>>  
Adicionar estudiante curso

Log de la institución  Log de sistema

inicio | anuncios | cursos | administración | configuración | cerrar sesión ☰

Datos del curso

Id curso

materia

paralelo

Datos del estudiante

Id estudiante

Botón adicionar ➡ Botón restablecer ➡

Figura 45: Interfaz para adicionar estudiantes a un curso  
Fuente: [Elaboración propia]

<<presentation class>>  
Adicionar curso

Log de la institución  Log de sistema

inicio | anuncios | cursos | administración | configuración | cerrar sesión ☰

Datos del curso

sigla

paralelo

carrera

materia

id docente

Botón Adicionar ➡ Botón restablecer ➡

Figura 46: Interfaz para adicionar un curso  
Fuente: [Elaboración propia]

<<presentation class>>  
Administrar docentes

Log de la institución  Log de sistema

inicio | anuncios | cursos | administración | configuración | cerrar sesión ☰

Lista de docentes

*	Id	Nombre	Usuario	eliminar docente	modificar docente	☰

Adicionar docente

Figura 47: Interfaz para administrar docentes  
Fuente: [Elaboración propia]

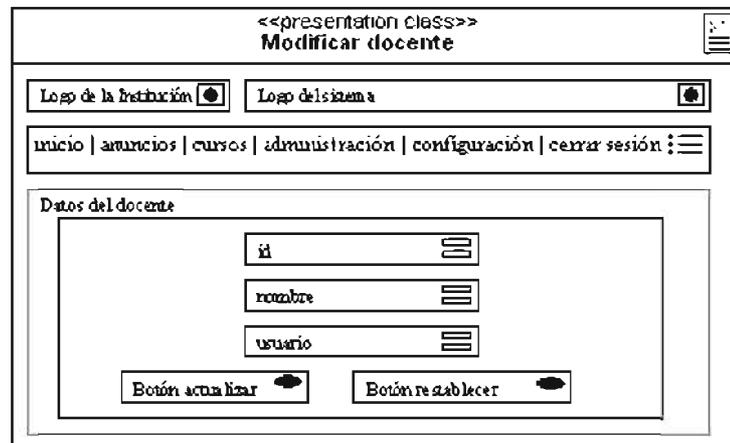


Figura 48: Interfaz para modificar docentes  
Fuente: [Elaboración propia]

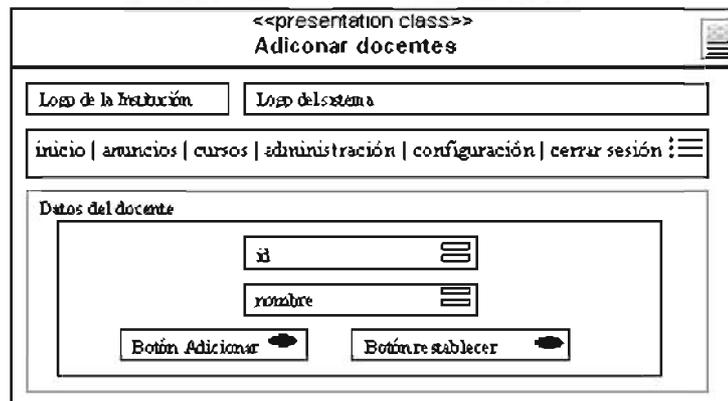


Figura 49: Interfaz para adicionar un docente  
Fuente: [Elaboración propia]

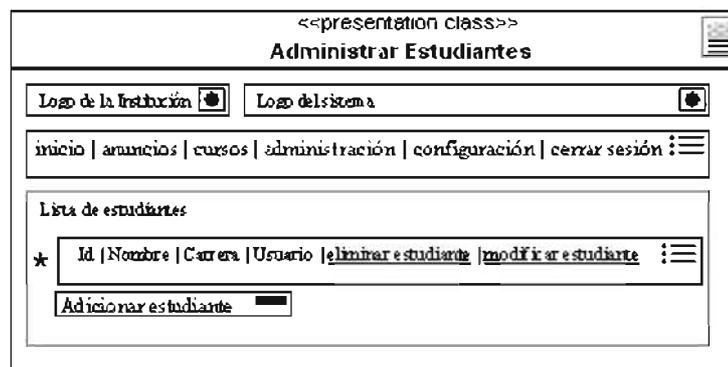


Figura 50: Interfaz para administrar estudiantes  
Fuente: [Elaboración propia]

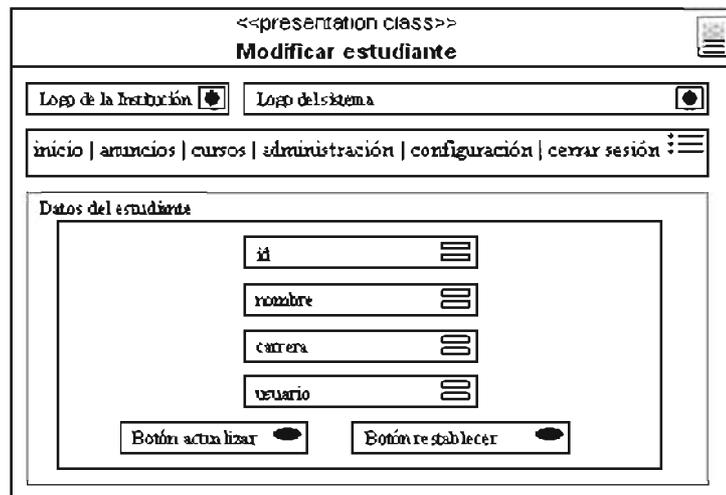


Figura 51: Interfaz para modificar estudiantes  
Fuente: [Elaboración propia]

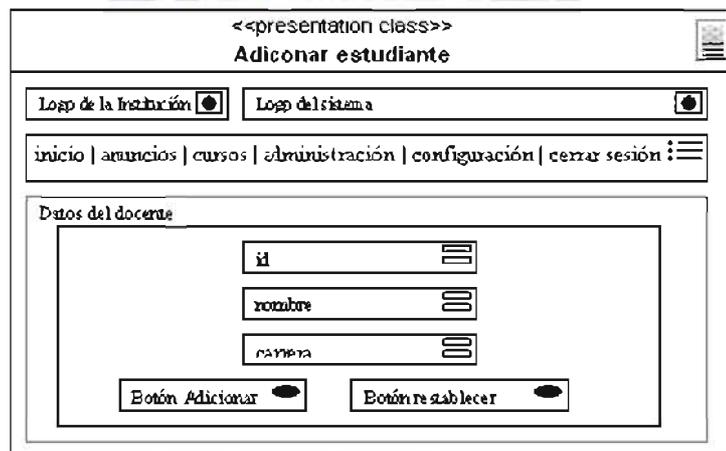


Figura 52: Interfaz adicionar estudiantes  
Fuente: [Elaboración propia]

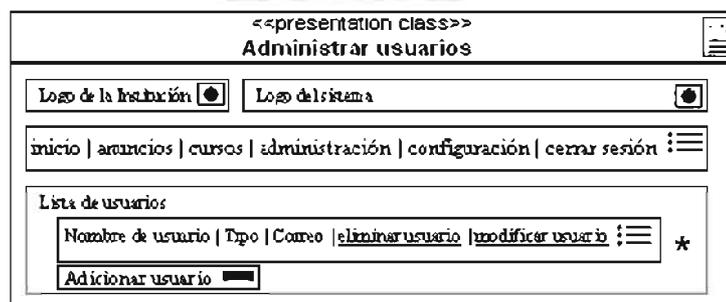


Figura 53: Interfaz administrar usuarios  
Fuente: [Elaboración propia]

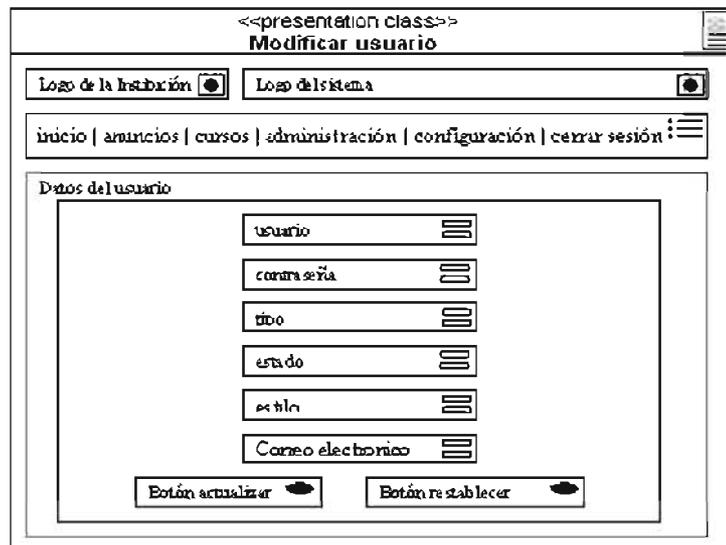


Figura 54: Interfaz modificar usuarios  
Fuente: [Elaboración propia]

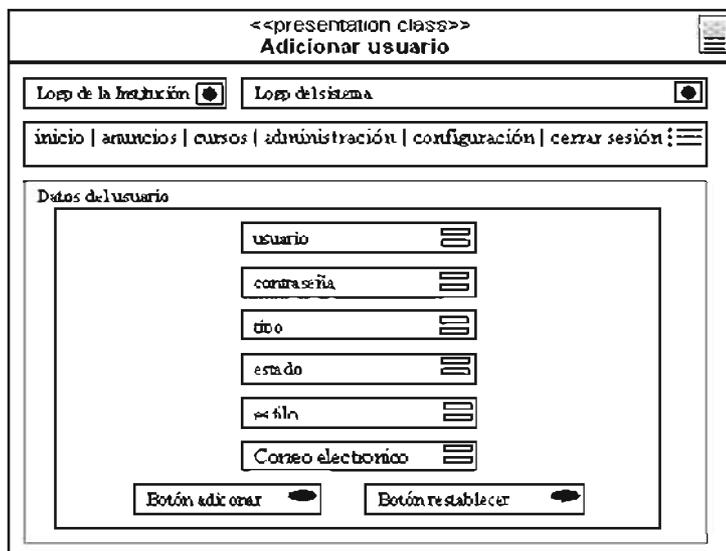


Figura 55: Interfaz para adicionar usuarios  
Fuente: [Elaboración propia]

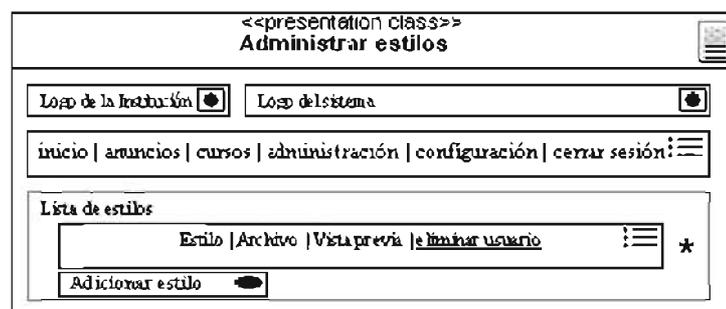


Figura 56: Interfaz para administrar estilos  
Fuente: [Elaboración propia]

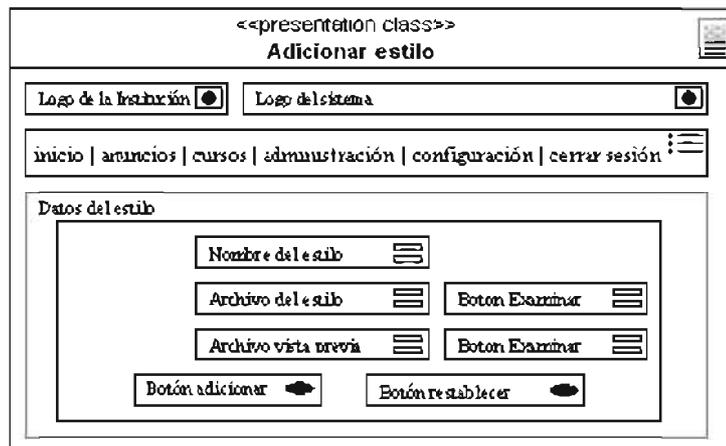


Figura 57: Interfaz para adicionar estilos  
Fuente: [Elaboración propia]

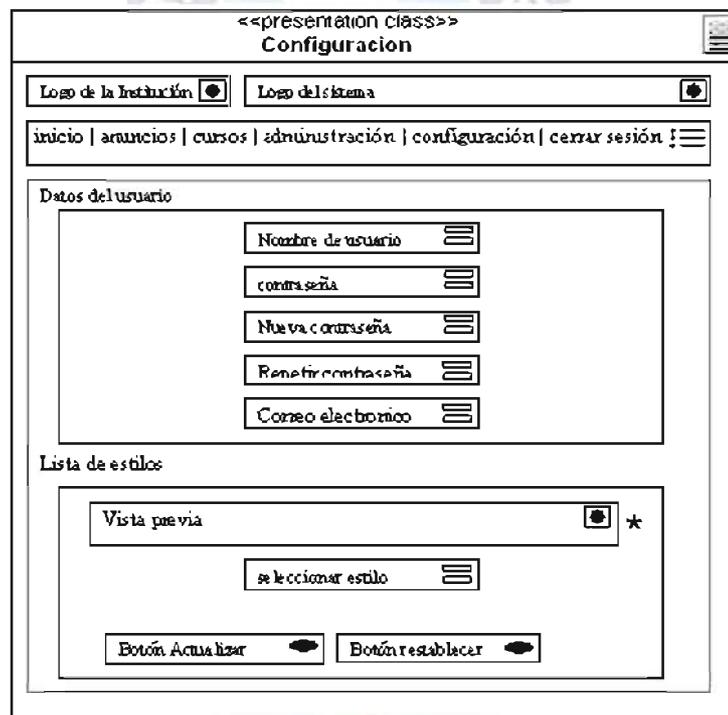


Figura 58: Interfaz para configuración de usuarios  
Fuente: [Elaboración propia]

## 4.5 MODELO ENTIDAD RELACIÓN

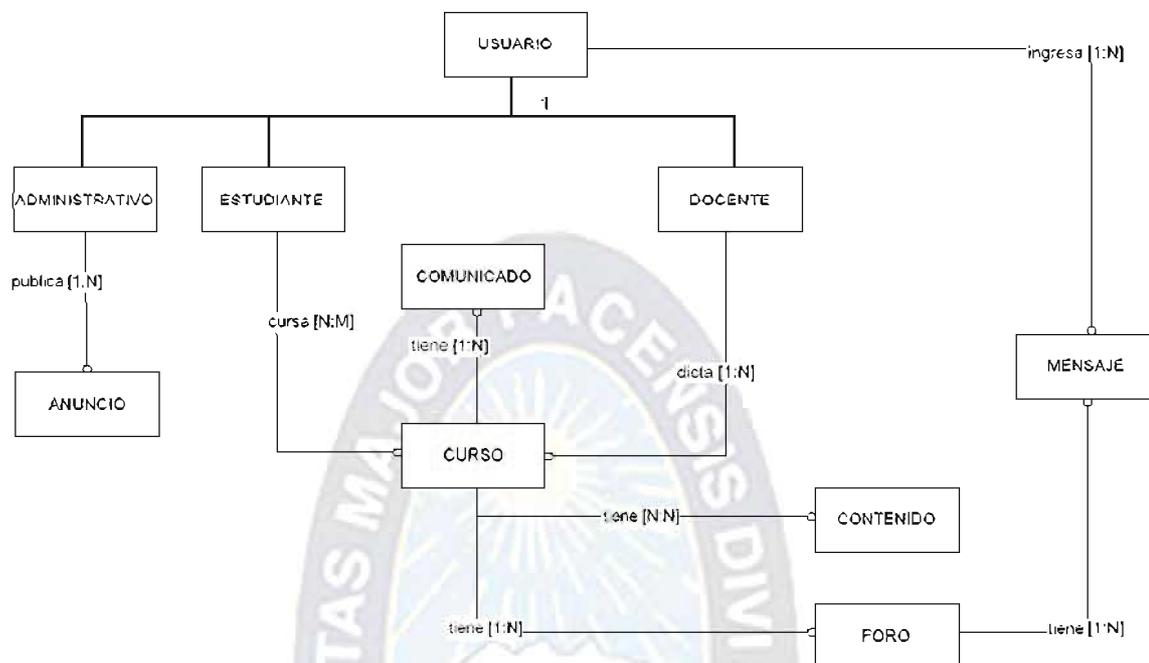


Figura 59: Diagrama entidad relación  
Fuente: [Elaboración propia]

<b>USUARIO</b>	Persona que tiene acceso al sistema, puede ser un estudiante, docente o administrativo
Usuario	Nombre que el usuario tiene para acceder al sistema, también es la clave principal
Contraseña	Clave de acceso del usuario
Estado	Sirve para permitir (1) o denegar (0) el acceso al sistema a un usuario
Estilo	Es un atributo multivaluado, que contiene los siguientes datos: - estilo, es el nombre que tiene el estilo - archivo, es el archivo en el que se encuentra la hoja de estilos (*.css) - vista, es el archivo para la vista previa del estilo
Tipo	Tipo de usuario, ya sea estudiante, docente o administrativo
Correo	Correo electrónico del usuario
<b>ESTUDIANTE</b>	Persona que realiza sus estudios en el instituto
Id	Clave principal del estudiante, puede ser su cedula de identidad
Nombre	Nombre completo del estudiante, con el siguiente formato: <i>ApellidoPaterno ApellidoMaterno Nombres</i>
Carrera	Carrera a la que pertenece el estudiante
<b>DOCENTE</b>	Persona que imparte clases a los estudiantes
Id	Clave principal del docente, puede ser su cedula de identidad
Nombre	Nombre completo del docente
<b>ADMINISTRATIVO</b>	Persona que ocupa un cargo administrativo dentro de la institución
Id	Clave principal del administrativo, puede ser su cedula de identidad

Nombre	Nombre completo del administrativo
Cargo	Cargo que desempeña el administrativo
<b>CURSO</b>	Clase que se imparte en la institución, ya sea un seminario o una materia semestral
<b>Id</b>	Clave principal que identifica al curso
Sigla	Código asignado a la materia
Paralelo	Código que sirve para distinguir dos cursos de la misma materia
Carrera	Carrera a la que pertenece el curso
Materia	Materia que se imparte en el curso
Capacidad	Espacio restante(en bytes) para el almacenamiento de contenidos, la capacidad máxima para un curso es 100000000bytes
<b>COMUNICADO</b>	Mensaje que se publica para un curso
<b>Id</b>	Clave principal del comunicado
Título	Encabezado del mensaje
Mensaje	Información que se publica para un curso
Fecha	Fecha y hora en la que se publicó el comunicado
<b>CONTENIDO</b>	Material educativo (en formato digital) que será utilizado en un curso
<b>Id</b>	Clave principal del contenido
Nombre	Nombre del contenido
Archivo	Archivo o dirección URL del contenido
Descripción	Breve descripción acerca del contenido
Tipo	Tipo de contenido, ya sea un contenido en forma de documento o un contenido SCORM (este tipo no es soportado por el sistema actual)
Curso	Curso al que pertenece el contenido
Tema	Tema o capítulo en el que será utilizado el contenido
<b>FORO</b>	Medio de comunicación que es utilizado en un curso
<b>Id</b>	Clave principal del foro
Título	Encabezado del foro
Comentario	Breve descripción acerca del foro
<b>MENSAJE</b>	Información que es publicada dentro de un foro
<b>Id</b>	Clave principal del mensaje
Título	Encabezado del mensaje
Mensaje	Cuerpo del mensaje
Fecha	Fecha y hora en la que se publicó el mensaje
<b>ANUNCIO</b>	Mensaje que publica un administrativo para toda la institución
<b>Id</b>	Clave principal del anuncio
Título	Encabezado del mensaje
Mensaje	Mensaje administrativo
Fecha	Fecha y hora en la que se publicó el mensaje

Tabla 34: Diccionario de datos del Ambiente Educativo Virtual  
Fuente: [Elaboración propia]

## 4.6 MODELADO DE LA BASE DE DATOS

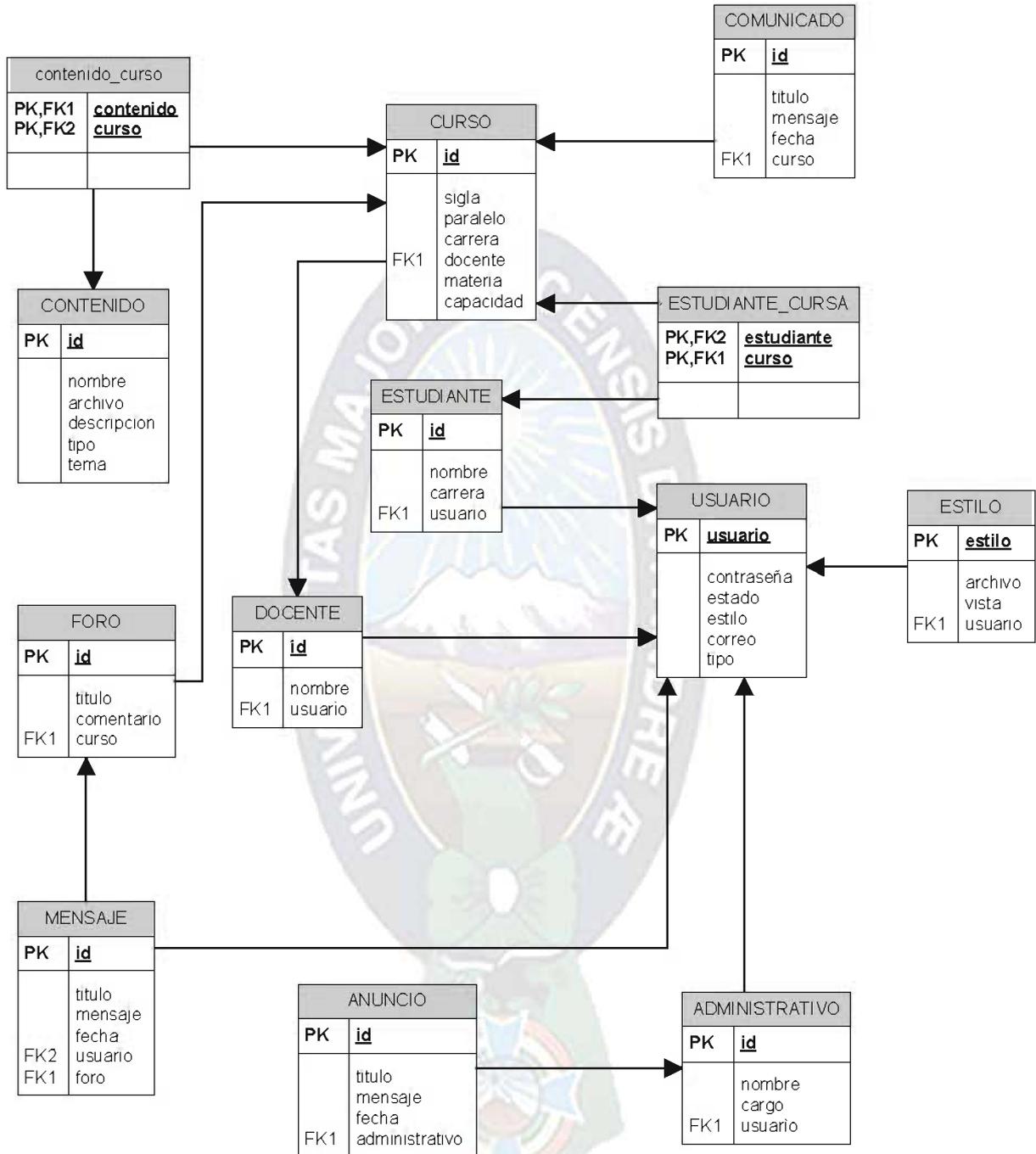


Figura 60: Diagrama Relacional de la Base de Datos AEV<sup>5</sup>  
Fuente: elaboración propia

<sup>5</sup> Como se puede observar la base de datos AEV se encuentra en forma normal de Boyce-Codd, pues todo determinante es parte de una clave primaria

## CAPÍTULO V

### 5.1 CONCLUSIONES

Después de terminar con el desarrollo del proyecto, tomando en cuenta la problemática inicial y los objetivos planteados para este Proyecto de Grado se puede afirmar que se han superado satisfactoriamente las metas trazadas llegando a las siguientes conclusiones:

- El Ambiente Educativo Virtual desarrollado supone un gran aporte para la solución de la problemática en la institución respecto al uso de tecnologías educativas. Pues contribuye a mejorar el proceso educativo en el Instituto Superior de Electrónica Informática y Telecomunicaciones “Santo Toribio de Mogrovejo”.
- El sistema alberga a tres tipos de usuarios: estudiantes, docentes y administrativos, satisfaciendo al mismo tiempo sus necesidades y cumpliendo con los requerimientos de la institución.
- La seguridad de todo sistema supone un coste económico y de eficiencia considerable, tomando en cuenta estos factores se ha tenido las consideraciones necesarias para buscar la seguridad en el sistema, cumpliendo con los criterios de seguridad y proponiendo políticas de seguridad para el sistema.
- Desde un principio se busco cumplir con los estándares existentes para el e-learning, sin embargo no se desarrolló el soporte para contenidos SCORM como estaba previsto en un principio, pues debido a las limitaciones de tiempo se tuvo que prescindir de este módulo, ya que además no se encontraba dentro de las necesidades principales de la institución. Sin embargo se ha propuesto la implementación de este estándar en un futuro próximo.
- Se ha diseñado el sistema de tal forma que exista una independencia entre el contenido y la presentación. Esto permite que el sistema pueda ser utilizado como base para el desarrollo de proyectos futuros, como la implementación de contenidos SCORM.

Finalmente se concluye diciendo que las nuevas tecnologías para la educación no suponen una ruptura con las anteriores, ya que por más bueno que sea un contenido o un sistema de educación virtual, no podrá despertar en el estudiante la misma motivación que un buen docente, sin embargo estas tecnologías se presentan como un complemento necesario para la educación tradicional.

## 5.2 RECOMENDACIONES

En base a las políticas de seguridad propuestas y a las observaciones realizadas durante las pruebas de implementación se elaboraron las siguientes recomendaciones. Estas recomendaciones van dirigidas al Director Académico del Instituto Superior de Electrónica Informática y Telecomunicaciones “Santo Toribio de Mogrovejo”.

- Mantener un control estricto acerca de la seguridad física del Ambiente Educativo Virtual.
- La seguridad lógica debe estar a cargo del administrador del servidor Web. Debe incluir el uso de contraseñas, la administración de puertos y servicios del servidor, el control de los archivos y contenidos, la ejecución de un Firewall y la administración de la Base de Datos.
- En caso de que se quiera conectar el Ambiente Educativo Virtual con la Base de Datos del Sistema Académico, debe redoblar la seguridad en el sistema, en este sentido se debe considerar la implementación de un Servidor Web seguro, como es el caso de Apache 2.

## BIBLIOGRAFÍA

- [Angulo] Raquel Ivonné Jalil Angulo, “*Un Nuevo Paradigma en Sistemas de Aprendizaje Basados en la Web*”, Universidad Autónoma Juan Misael Saracho.
- [AISL, 2007] Asociación de Investigación en Software Inteligente, “*Diseño y Construcción de Aplicaciones Ágiles con el Método Scrum*”, Mayo del 2007.
- [BSE, 2007] Baufest Software Engineering, “*¿Qué es Scrum?*”, Recuperado octubre del 2007, de: <http://www.baufest.com>
- [Cordon y Anaya] Oscar Cordon, Karina Anaya, “*Enseñanza Virtual: Fundamentos, perspectivas actuales y visión de la Universidad de Granada*”, Centro de Enseñanzas Virtuales de la Universidad de Granada.
- [Foix y Zavando, 2002] Cristian Foix, Sonia Zavando, “*Estándares e-learning: Estado del Arte*”, Centro de Tecnologías de Información de INTEC, julio del 2002.
- [Gomez y Gabardina, 2006] Ing. Mariana Gómez, Ing. Juan Gabardina, “*Introducción SCRUM*”, Facultad de Ingeniería Universidad de Buenos Aires, octubre del 2006.
- [Graells, 2007] Dr. Pere Marquès Graells, “*El impacto de la Sociedad de la Información en el mundo educativo*”, Recuperado en Septiembre del 2007, de: <http://dewey.uab.es/pmarques/siyedu.htm>
- [Gonzalez] José Mariano González Romano, “*Seguridad en Aplicaciones Web*”, Curso de PHP.

- [INSCERVANTES] Departamento de Tecnología y Proyectos Lingüísticos Instituto Cervantes, “*El Aula Virtual de Español (AVE)*”, Recuperado en Septiembre del 2007, de: [www.formatex.org/micte2005/148.pdf](http://www.formatex.org/micte2005/148.pdf)
- [ISEIT, 2007] Instituto Superior de Electronica, Informatica y Telecomunicaciones, “*Antecedentes Históricos del UPM Santo Toribio*”, Recuperado en Agosto del 2007, de: <http://www.fevalegria.org>
- [Koch, 2000] Nora Parcus de Koch, “*Software Engineering for Adaptive Hypermedia Systems. Reference Model*”. Ludwig-Maximilians-Universität München, Alemania.
- [Letelier y Penadés] Patricio Letelier y M. Carmen Penadés, “*Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*”, Universidad Politécnica de Valencia.
- [Pedruelo, 2004] Miguel Rebollo Pedruelo, “*El estándar SCORM para EaD*”, Tesina del Máster en Enseñanza y Aprendizaje Abiertos y a Distancia Universidad Nacional de Educación a Distancia, Diciembre 2004.
- [Pressman, 2005] Roger S. Pressman, “*Ingeniería del Software*” 5º Edición, Editorial Concepción Fernández Madrid, 2005.
- [Pressman, 2006] Roger S. Pressman, “*Ingeniería del Software*” 6º Edición, Editorial McGraw-Hill, 2007.
- [Rosario, 2007] Jimmy Rosario, “*La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual*” Recuperado en Septiembre del 2007, de: <http://www.cibersociedad.net/archivo/articulo.php?art=218>
- [UPM] Universidad Politécnica de Madrid, Gabinete de tele-educación, “*Manual Moodle*”, Recuperado en Septiembre del 2007, de:

<http://www.agricolas.INSCERVANTES.es/cvirtual/MANUAL%20DE%20MOODLE-protegido1.pdf>

- [SA1, 2007] “*Ambiente Educativo Virtual*”, de Wikipedia, la enciclopedia libre, Recuperado en Septiembre del 2007, de: [http://es.wikipedia.org/wiki/Ambiente\\_Educativo\\_Virtual](http://es.wikipedia.org/wiki/Ambiente_Educativo_Virtual)
- [SA2, 2007] “*Educación a distancia*”, de Wikipedia, la enciclopedia libre, Recuperado en Septiembre del 2007, de: [http://es.wikipedia.org/wiki/Educaci%C3%B3n\\_a\\_distancia](http://es.wikipedia.org/wiki/Educaci%C3%B3n_a_distancia)
- [SA3, 2007] “*E-learning*”, de Wikipedia, la enciclopedia libre, Recuperado en Septiembre del 2007, de: <http://es.wikipedia.org/wiki/E-learning>
- [SA4, 2007] “*Educación virtual*”, De Wikipedia, la enciclopedia libre, Recuperado en Octubre del 2007, de: [http://es.wikipedia.org/wiki/Educaci%C3%B3n\\_virtual](http://es.wikipedia.org/wiki/Educaci%C3%B3n_virtual)
- [SA5, 2008] “*Guía de diseño de páginas en internet*”, Recuperado enero 2008, de: <http://www.geocities.com/SiliconValley/Heights/1779>
- [SA6, 2008] “*Cliente-Servidor*”, Recuperado enero 2008, de: <http://es.wikipedia.org/wiki/Cliente-servidor>
- [Holzner, 2005] Steven Holzner, “*Manual avanzado de PHP 5*”, Editorial Anaya Multimedia, 2005.
- [VWD, 2008] Visual Web Developer, “*Procedimientos de seguridad básicos para aplicaciones Web*”, Recuperado marzo 2008, de: <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbconBestSecurityPracticesForWebApplications.asp>



**ANEXO B DOCUMENTACIÓN**

