

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERIA EN SISTEMAS E INFORMÁTICA

Desarrollo del portal web del departamento de ciencias de la  
computación de la ESPE utilizando herramientas open  
Source

Previa a la obtención de Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

Carlos Eduardo Anchundia Valencia  
JUAN PABLO BALDA ANDRADE

SANGOLQUÍ, diciembre de 2006

## **CERTIFICACION**

Certifico que el presente trabajo fue realizado en su totalidad por los Señores CARLOS EDUARDO ANCHUNDIA VALENCIA y JUAN PABLO BALDA ANDRADE como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA

Diciembre del 2006

---

Ing. Edison Lascano

## **DEDICATORIA**

Esta tesis está dedicada a nuestros padres, quienes con amor y sabiduría, con paciencia y fortaleza nos han sabido guiar a lo largo de nuestra carrera universitaria que hemos culminado con éxito.

**Carlos Eduardo Anchundia Valencia**

**Juan Pablo Balda Andrade**

## **Agradecimientos**

Agradecemos a Dios por brindarnos día tras día la fortaleza, la sabiduría, la salud y el sustento económico y así darnos la oportunidad de finalizar con éxito nuestros estudios superiores y emprender nuestra vida profesional.

A nuestros padres por ser nuestro soporte y animarnos a perseverar ante los obstáculos que se presentaron en esta trayectoria. Por inculcarnos sus principios y valores sembrando en nosotros los ideales y la humildad necesaria para desarrollarnos como personas de bien.

A las autoridades, docentes y personal administrativo que forman parte del Departamento de Ciencias de la Computación que nos apoyaron a largo de estos cinco años brindándonos el conocimiento y las experiencias vividas, que nos han servido para ampliar nuestras capacidades, habilidades y pensamiento.

**Carlos Eduardo Anchundia Valencia**

**Juan Pablo Balda Andrade**

## *ÍNDICE DE CONTENIDO*

<b>RESUMEN</b>	<b>1</b>
<hr/>	
<b>CAPÍTULO I: INTRODUCCIÓN</b>	<b>2</b>
<hr/>	
<b>1.1. TEMA</b>	<b>2</b>
<b>1.2. ANTECEDENTES</b>	<b>2</b>
<b>1.3. OBJETIVOS</b>	<b>3</b>
1.3.1. OBJETIVO GENERAL	3
1.3.2. OBJETIVOS ESPECÍFICOS	3
<b>1.4. ALCANCE</b>	<b>3</b>
<b>1.5. JUSTIFICACIÓN</b>	<b>4</b>
<b>CAPÍTULO II: MARCO TEÓRICO DE REFERENCIA</b>	<b>6</b>
<hr/>	
<b>2.1. CONCEPTOS BÁSICOS</b>	<b>6</b>
2.1.1. PORTAL, SITIO Y PÁGINA WEB	6
2.1.1.1. Página Web	6
2.1.1.2. Sitio Web	7
2.1.1.3. Portal Web	7
2.1.2. ARQUITECTURA CLIENTE – SERVIDOR	8
2.1.2.1. Cliente – Servidor	8
2.1.2.2. Servicios de Internet	9
Correo Electrónico	10
FTP (File Transfer Protocol)	10
Foro	10
Chat	11
Otros Servicios	11
2.1.3. TECNOLOGÍAS WEB	11
2.1.3.1. Tecnología del lado del Cliente	12
HTML	12
DHTML	12
JavaScript	13
VbScript	13
WML	13
2.1.3.2. Tecnología del lado del Servidor	14
CGI/Perl	14
ASP y ASPX	14
JSP	15
PHP	16
2.1.3.3. Comparación entre las tecnologías	17
2.1.4. SISTEMAS DE ADMINISTRACIÓN DE BASE DE DATOS	18
2.1.4.1. PostgreSQL	18
2.1.4.2. MySQL	20
2.1.4.3. Oracle	20
2.1.4.4. Cuadro comparativo	21
2.1.5. PROTOCOLOS	25
2.1.5.1. TCP/IP	25
2.1.5.2. HTTP	27

2.1.5.3.	FTP	28
2.1.6.	PLATAFORMAS	29
2.1.6.1.	UNIX	29
2.1.6.2.	Linux	30
2.1.6.3.	Windows	31
2.1.7.	ADMINISTRADOR DE CONTENIDOS	32
<b>2.2.</b>	<b>METODOLOGÍA RUP PARA EL DESARROLLO DEL PROYECTO</b>	<b>32</b>
2.2.1.	UML	32
2.2.1.1.	Notas	33
2.2.1.2.	Diagramas de Caso de Uso	33
	Actores y Caso de Uso	34
	Usos y Extends	34
2.2.1.3.	Diagramas de Clase	36
	Asociación	36
	Navegabilidad	37
	Atributos	38
	Operación	38
	Generalización	39
	Reglas de Restricción	40
	Agregación y Composición	40
	Interfaz y Clase abstracta	41
	Clase de Asociación	43
2.2.1.4.	Diagramas de Iteración	43
	Diagrama de Secuencia	43
	Diagrama de Colaboración	44
2.2.1.5.	Diagramas de Paquetes	45
2.2.1.6.	Diagramas de Estado	46
2.2.1.7.	Diagramas de Actividad	47
2.2.1.8.	Diagramas de Componentes	48
2.2.2.	RATIONAL UNIFIED PROCESS	49
2.2.2.1.	Desarrollo iterativo	49
2.2.2.2.	Administración de requerimientos	50
2.2.2.3.	El uso de arquitectura basada en componentes	50
2.2.2.4.	Software de modelamiento visual	50
2.2.2.5.	Verificación continua de la calidad del software	50
2.2.3.	ARQUITECTURA	51
2.2.4.	CONCEPTOS	52
2.2.4.1.	Artefactos	52
2.2.4.2.	Actividades	52
2.2.4.3.	Roles	53
2.2.5.	DISCIPLINAS O FLUJOS DE TRABAJO	53
2.2.5.1.	Modelado del negocio	54
2.2.5.2.	Requisitos	55
2.2.5.3.	Análisis y Diseño	56
2.2.5.4.	Implementación	57
2.2.5.5.	Pruebas	58
2.2.5.6.	Entrega	58
2.2.5.7.	Administración de proyecto	59
2.2.5.8.	Configuración y control de cambios	60
2.2.5.9.	Entorno	60
2.2.6.	FASES	61
2.2.6.1.	Inicio	61
2.2.6.2.	Elaboración	62
2.2.6.3.	Construcción	63

2.2.6.4. Transición	64
2.2.7. RESUMEN	65
<b>2.3. LA SITUACIÓN ACTUAL</b>	<b>67</b>

---

## **CAPÍTULO III: DESARROLLO DEL PORTAL** **68**

<b>3.1. INICIO</b>	<b>68</b>
3.1.1. ÁMBITO DEL PROYECTO	68
3.1.2. MODELADO DEL NEGOCIO	68
3.1.3. REQUERIMIENTOS	71
3.1.4. GLOSARIO DE TÉRMINOS	75
3.1.5. INTERFAZ DE USUARIO	76
<b>3.2. ELABORACIÓN</b>	<b>77</b>
3.2.1. MODELO DE ANÁLISIS	77
3.2.1.1. Administrar Materias	77
3.2.1.2. Editar Variables del Sistema	78
3.2.1.3. Administrar Encuestas	78
3.2.1.4. Administrar Contenido de Actividades, Banners y Noticias	78
3.2.1.5. Administrar Secciones y Páginas	79
3.2.1.6. Administrar Cuentas de Usuario	80
3.2.1.7. Editar Currículo	80
3.2.1.8. Administrar Zona de Materias	81
3.2.1.9. Validar Usuarios	82
3.2.1.10. Navegar en Zona de Materias	82
3.2.1.11. Navegar en el Portal	83
3.2.2. ARQUITECTURA DEL SOFTWARE	83
3.2.2.1. Cliente	84
3.2.2.2. Presentación	84
3.2.2.3. Negocio	84
3.2.2.4. Acceso a datos	86
3.2.3. DIAGRAMAS DE SECUENCIA	87
3.2.3.1. Administrar Materias	87
3.2.3.2. Editar Variables del Sistema	88
3.2.3.3. Administrar Encuestas	89
3.2.3.4. Administrar Contenido de Actividades, Banners y Noticias	90
3.2.3.5. Administrar Secciones y Páginas	93
3.2.3.6. Administrar Cuentas de Usuario	95
3.2.3.7. Editar Currículo	96
3.2.3.8. Administrar Zona de Materias	97
3.2.3.9. Validar Usuarios	99
3.2.3.10. Navegar en Zona de Materias	99
3.2.3.11. Navegar en el Portal	100
3.2.4. DIAGRAMAS DE CLASE DE DISEÑO	101
3.2.5. MODELO ENTIDAD/RELACIÓN	102
<b>3.3. CONSTRUCCIÓN</b>	<b>103</b>
3.3.1. CASOS DE PRUEBA	103
3.3.2. DIAGRAMA DE COMPONENTES	115
<b>3.4. TRANSICIÓN</b>	<b>115</b>
3.4.1. ENTREGA	115
3.4.2. MANUAL DE INSTALACIÓN	116
3.4.3. MANUAL DE USUARIO	116

<b>CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES</b>	<b>117</b>
3.5. CONCLUSIONES	117
3.6. RECOMENDACIONES	119
<b>BIBLIOGRAFÍA</b>	<b>120</b>
<b>ANEXO A: ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</b>	<b>121</b>
<b>ANEXO B: MANUAL DE INSTALACIÓN</b>	<b>132</b>
<b>ANEXO C: MANUAL DE USUARIO</b>	<b>152</b>
<b>BIOGRAFÍAS</b>	<b>192</b>
CARLOS EDUARDO ANCHUNDIA VALENCIA	192
JUAN PABLO BALDA ANDRADE	192



## ***LISTADO DE TABLAS***

Tabla 1 Comparación de Bases de Datos .....	17
Tabla 2 Comparación de Bases de Datos .....	24
Tabla 3 Arquitectura TCP/IP .....	26
Tabla 4 Multiplicidad de asociaciones .....	36
Tabla 5 Resumen General de la Metodología RUP .....	66
Tabla 6 Procesos del Departamento .....	68
Tabla 7 Descripción de Caso de Uso Administrar Materias .....	73
Tabla 8 Descripción de Caso de Uso Editar Variables del Sistema .....	73
Tabla 9 Descripción de Caso de Uso Administrar Encuestas .....	73
Tabla 10 Descripción de Caso de Uso Administrar Materias .....	73
Tabla 11 Descripción de Caso de Uso Administrar Secciones y Páginas .....	74
Tabla 12 Descripción de Caso de Uso Administrar Cuentas de Usuario .....	74
Tabla 13 Descripción de Caso de Uso Editar Currículo .....	74
Tabla 14 Descripción de Caso de Uso Administrar Zona de Materias .....	74
Tabla 15 Descripción de Caso de Uso Validar Usuarios .....	75
Tabla 16 Descripción de Caso de Uso Navegar en Zona de Materias .....	75
Tabla 17 Descripción de Caso de Uso Navegar en el Portal .....	75
Tabla 18 Glosario de Términos .....	75
Tabla 19 Caso de Prueba: Inicio de sesión .....	103
Tabla 20 Caso de Prueba: Modificar Variables del Sistema .....	104
Tabla 21 Caso de Prueba: Respalidar Base .....	104
Tabla 22 Caso de Prueba: Reiniciar Datos .....	104
Tabla 23 Caso de Prueba: Modificar Página de Error .....	105
Tabla 24 Caso de Prueba: Reporte de Bitácora .....	105
Tabla 25 Caso de Prueba: Modificar Bloque de Bienvenida .....	105
Tabla 26 Caso de Prueba: Administrar Secciones .....	106
Tabla 27 Caso de Prueba: Administrar Páginas .....	107
Tabla 28 Caso de Prueba: Administrar Encuestas .....	108
Tabla 29 Caso de Prueba: Administrar Banners .....	109
Tabla 30 Caso de Prueba: Bloque de Eventos .....	110
Tabla 31 Caso de Prueba: Administrar Noticias .....	110
Tabla 32 Caso de Prueba: Administrar Actividades .....	111
Tabla 33 Caso de Prueba: Administrar Usuarios .....	112
Tabla 34 Caso de Prueba: Administrar Materias .....	113
Tabla 35 Listado de Errores .....	114

## ***LISTADO DE FIGURAS***

Figura 1 Nota.....	33
Figura 2 Actor.....	34
Figura 3 Uses.....	35
Figura 4 Extends.....	36
Figura 5 Asociación.....	37
Figura 6 Navegabilidad.....	37
Figura 7 Operación.....	39
Figura 8 Generalización.....	39
Figura 9 Restricciones.....	40
Figura 10 Agregación y Composición.....	41
Figura 11 Clase Abstracta.....	42
Figura 12 Interface.....	42
Figura 13 Clase de Asociación.....	43
Figura 14 Diagrama de Secuencia.....	44
Figura 15 Diagrama de Colaboración.....	45
Figura 16 Diagrama de Paquetes.....	46
Figura 17 Diagrama de Estado.....	47
Figura 18 Diagrama de Actividad.....	48
Figura 19 Diagrama de Componentes.....	49
Figura 20 Arquitectura de RUP y sus Dimensiones.....	51
Figura 21 Modelo del Negocio.....	70
Figura 22 Modelo de Casos de Uso (Requerimientos Funcionales).....	72
Figura 23 Esquema de interfaz.....	76
Figura 24 Administrar Materias.....	77
Figura 25 Editar Variables del Sistema.....	78
Figura 26 Administrar Encuestas.....	78
Figura 27 Administrar Contenido de Actividades, Banners y Noticias.....	79
Figura 28 Administrar Secciones y Páginas.....	79
Figura 29 Administrar Cuentas de Usuario.....	80
Figura 30 Editar Currículo.....	80
Figura 31 Administrar Zona de Materias.....	81
Figura 32 Validar Usuarios.....	82
Figura 33 Navegar en Zona de Materias.....	82
Figura 34 Navegar en Zona de Materias.....	83
Figura 35 Arquitectura del Software.....	84
Figura 36 Clases Controladores.....	85
Figura 37 Clases Controladores (Continuación).....	86
Figura 38 Administrar Materias.....	87
Figura 39 Editar Variables del Sistema.....	88
Figura 40 Administrar Encuestas.....	89
Figura 41 Administrar Contenido (Actividades).....	90
Figura 42 Administrar Contenido (Banner).....	91
Figura 43 Administrar Contenido (Noticias).....	92
Figura 44 Administrar Secciones y Páginas (Secciones).....	93
Figura 45 Administrar Secciones y Páginas (Páginas).....	94
Figura 46 Administrar Cuentas de Usuario.....	95
Figura 47 Editar Currículo.....	96

Figura 48 Administrar Zona de Materias .....	97
Figura 49 Administrar Zona de Materias (Continuación) .....	98
Figura 50 Validar Usuarios .....	99
Figura 51 Navegar en Zona de Materias .....	99
Figura 52 Navegar en el Portal .....	100
Figura 53 Diagrama de Clases de Diseño .....	101
Figura 54 Modelo Entidad / Relación .....	102
Figura 55 Diagrama de Componentes.....	115

## Resumen

El presente Proyecto de Tesis consiste en el desarrollo de un Portal Web para el Departamento de Ciencias de la Computación (DCC) de la ESPE cuya principal característica es la facilidad de edición de información gracias a un Sistema de Administración de Contenido. El Portal además incluye un servicio que permite una comunicación Docente – Alumno, que le brinda a los docentes la posibilidad de publicar mensajes y subir archivos que pueden ser accedidos por los alumnos.

Para el desarrollo de todo proyecto de software es necesario apoyarse en una metodología que brinde las pautas a seguir. Rational Unified Process, por sus características y flexibilidad fue la metodología escogida durante este proyecto. RUP se basa en iteraciones y es evolutiva, lo cual permite obtener una mejora del producto luego de cada iteración.

A más de la metodología es importante seleccionar las herramientas. En este proyecto se utilizaron herramientas Open Source tanto para el desarrollo como para la implementación, las cuales se enfocaron a la utilización de la tecnología Java, entre las que tenemos Netbeans y Sun Java System Application Server. La tecnología de Java se la usa extensamente a nivel mundial por sus características y robustez.

El principal motivo de la elaboración del presente proyecto es la de mejorar las funciones de comunicación entre la institución y los usuarios, y así mismo que sirva como base para incorporar nuevas aplicaciones y servicios que contribuyan al mejoramiento de procesos internos y para con los estudiantes.

# Capítulo I: Introducción

## 1.1. Tema

Desarrollo del Portal Web del Departamento de Ciencias de la Computación de la ESPE utilizando herramientas Open Source.

## 1.2. Antecedentes

El presente Proyecto de Tesis se desarrolló durante un momento de cambio de la estructura organizacional de la Escuela Politécnica del Ejército, en el cual han desaparecido las Facultades y se han formado nuevas unidades administrativas denominadas Departamentos. Por esta razón, la ex Facultad de Ingeniería de Sistemas e Informática ha desaparecido, siendo ahora el Departamento de Ciencias de la Computación, de aquí en adelante DCC, la unidad encargada de coordinar las actividades académicas, la investigación y la extensión que eran responsabilidad de la FISI.

Actualmente no cuenta con un portal en Internet apropiado y capaz de manejar la información completa y necesaria hacia docentes, estudiantes y público en general, consecuentemente se tiene poca información disponible para los usuarios y posibles aspirantes que deseen conocer sobre las carreras que ofrece el Departamento de Ciencias de la Computación en la ESPE; lo cual la sitúa en desventaja competitiva con respecto a las demás carreras afines en otras universidades.

La necesidad de contar con este tipo de servicios para el usuario se hace evidente e indispensable debido a la tendencia de las organizaciones de publicar portales con el fin de brindar mejores prestaciones y accesibilidad a sus clientes.

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

Desarrollar el Portal Web para el Departamento de Ciencias de la Computación de la ESPE que ofrezca información y servicios a los docentes, estudiantes y público en general, mediante el manejo de un Administrador de Contenidos construido con herramientas Open Source.

#### **1.3.2. Objetivos específicos**

- Analizar la situación actual entorno a los servicios Web que tiene el Departamento.
- Utilizar una metodología de Desarrollo de Software.
- Analizar y diseñar el Portal Web
- Desarrollar el Portal Web
- Implantar y realizar las pruebas respectivas

### **1.4. Alcance**

Desarrollar el Portal Web para el Departamento de Ciencias de la Computación de manera que mediante la utilización de un Administrador de Contenidos se pueda ofrecer información dinámica referente al Departamento y además manejar un servicio interactivo Docente – Alumno.

La siguiente estructura constituirá la base en la que se desarrollará el proyecto:

- Información Organizacional del Departamento
  - Contempla información sobre la misión, la visión, objetivos, definición de la carrera y perfil profesional, reseña histórica, y otro tipo de

información como el listado del personal docente (tiempo completo y parcial) y autoridades.

- Información Académica
  - Consta de la malla curricular y los horarios correspondientes al período en curso.
- Servicios
  - Incluye servicios como el acceso al correo electrónico, enlaces a páginas de interés, noticias de actualidad, ingreso a una zona de materias para docentes y alumnos.
- Administrador de contenidos
  - Permite administrar el contenido disponible para quienes visiten el Portal.
- Administrador de Usuarios
  - Permite administrar los usuarios del sistema.
- Administrador de Materias
  - Permite administrar las diferentes materias que se imparten en en las Carreras del Departamento

### **1.5. Justificación**

La utilización de un Portal Web permite que la información sea difundida y accesible cuando sea requerida; es decir que la información puede ser consultada en cualquier momento y desde cualquier lugar diferente a las oficinas del Departamento.

Mediante la implantación de tecnología Web se puede brindar una mejor atención a las personas involucradas con el Departamento, integrando no solo información sino también servicios que beneficiarán el desempeño de las actividades y el trabajo académico.

Además de contar con un mejor servicio, la imagen corporativa del Departamento incursionaría en un campo más amplio como lo es el de la Internet, haciendo que el Departamento sea más difundido no sólo a nivel local sino también a nivel internacional debido al uso masivo que esta tecnología tiene actualmente.

Dentro de las características principales del Portal destacan:

- Información Organizacional del Departamento
- Información Académica
- Servicios
- Administrador de contenidos



## Capítulo II: Marco Teórico de Referencia

En un mundo globalizado es necesario que las empresas puedan ofrecer mejores prestaciones, a fin de poder ser competitivas y expandir sus horizontes. En tal virtud, las empresas reconocen el poder de la Internet y como esta tecnología puede mejorar las prestaciones en varios ámbitos como la productividad, el marketing, ventas, entre otras.

La Internet brinda una gran gama de servicios, entre ellos el principal es el WWW (World Wide Web), que permite publicar y explorar información en la red, así como acceder a los recursos que ofrece, todo esto de forma transparente para el usuario. En este campo existe una gran variedad de conceptos y elementos que se aclararán a continuación.

### 2.1. Conceptos Básicos

Para mayor comprensión del proyecto, en esta sección se aclararán ciertos términos y conceptos, los cuales se utilizarán a largo del desarrollo de todo el proyecto.

#### 2.1.1. Portal, Sitio y Página Web

Estos tres términos provocan confusión debido a que a simple vista son similares, pero en realidad difieren tanto en su propósito como en su magnitud. A continuación se describen cada uno de los términos.

##### 2.1.1.1. *Página Web*

Una *Página Web* es un documento multimedia que puede llevar incorporado texto, imágenes, sonido y animaciones, mostrando la información que una persona u

organización quiere publicar en la red; generalmente está escrito en un lenguaje denominado HTML (HyperText Markup Language).

#### 2.1.1.2. *Sitio Web*

Un *Sitio Web* o *Website* juega un papel importante en cuanto a los servicios que pueden ser implantados, los mismos que pueden llegar a ofrecer muchas ventajas a empresas u organizaciones que precisen darse a conocer, incluso en algunos casos llegando a complementar la actividad que desarrolla la empresa pero en un entorno distinto como lo es en la Internet. Los servicios permiten ampliar las prestaciones ofrecidas a través de contenidos que interactúan con el usuario y que despliegan información útil como por ejemplo la consulta de saldos de cuentas bancarias a través de la Web o compras en línea.

El *Website* está conformado por un grupo de Páginas Web orientadas a un solo tema como por ejemplo: información sobre una compañía para promocionar un producto; permitiendo el acceso a todo tipo de usuarios, tanto internos como externos a una organización.

#### 2.1.1.3. *Portal Web*

Un *Portal Web* es un Sitio Web especializado que consta de una Página Web diseñada para servir como punto de partida en la navegación. Suelen incorporar distintos servicios como correo electrónico, foros, buscador, noticias, FTP, etc.; además su contenido es dinámico e interactúa con el usuario.

El *Portal Web*, debido a sus características necesariamente requiere un tipo de administración especial que sea capaz de mantener su contenido actualizado y ofrecer una

manera sencilla de edición. Esto se puede lograr integrando un Administrador de Contenidos, facilitando así el trabajo del Webmaster.

### 2.1.2. Arquitectura Cliente – Servidor

En cuanto surgieron las redes computacionales, la idea era poder compartir recursos como por ejemplo: información, dispositivos de E/S, dispositivos de almacenamiento, etc.

Para ello los computadores involucrados deben iniciar un proceso de comunicación entre ellas para poder efectuar un intercambio de información a lo que se denomina *Transacción*. En este proceso existen lo que se denomina cliente, quien es el que inicia el diálogo o solicita los recursos; y servidor, quien responde a las solicitudes.

#### 2.1.2.1. Cliente – Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.

- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.
- Reglas del negocio
- Acceso a datos

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer el funcionamiento del servidor, sólo como comunicarse con él.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.
- Un Cliente Web realiza transacciones con un Servidor Web, el Cliente hace un requerimiento a un servicio que se encuentra en un Servidor, el mismo puede ser un Servidor Jakarta, Apache, Internet Information Server, SJSAS, etc., los cuales acceden a los datos, los filtran y aplican las reglas del negocio y retornan una página HTML como resultado al Cliente.

#### 2.1.2.2. *Servicios de Internet*

Los Portales Web ofrecen *Servicios* con el fin de satisfacer las necesidades de los usuarios, además de darle un valor agregado y mayor funcionalidad al Portal como la capacidad de prestar servicios a los usuarios y poder obtener un beneficio intangible y/o tangible para la empresa. Estos son algunos de los principales *Servicios*:

### ***Correo Electrónico***

El *Correo Electrónico* o *E-mail* es un servicio de Internet por medio del cual se puede enviar y recibir mensajes escritos desde y hacia cualquier lugar del mundo, siempre que se tenga acceso a Internet.

Este servicio no solo funciona a través de Internet, también se puede configurar en una red interna como un sistema de comunicación dentro de una empresa para agilizar procesos.

### ***FTP (File Transfer Protocol)***

Este servicio permite la transferencia de archivos desde y hacia un servidor *FTP*. La arquitectura de un *FTP* es Cliente – Servidor, el servidor posee una estructura de directorios o carpetas en donde se encuentran alojados los archivos de texto, gráficos, etc. y el cliente accede mediante un software de *FTP* o por línea de comandos para descargar archivos a su PC o enviarlos al servidor.

### ***Foro***

Un *Foro* es un servicio que permite comunicarse y compartir preguntas y respuestas sobre diversos temas mediante la publicación de estos mensajes en una Página Web para que pueda ser vista por cualquiera; de tal manera que ayuden a resolver problemas habituales o particulares, o simplemente permitir opinar y armar una discusión sobre algún tema propuesto por algún usuario. Además, también permite a los usuarios publicar sus perfiles con el objetivo de poder conocer personas con intereses similares e intercambiar conocimientos. Un foro trabaja offline, es decir fuera de línea, de modo que la información a publicar debe ser subida (upload) al servidor por lo que no se necesita que quien publicó la información esté conectado todo el tiempo.

## ***Chat***

El *Chat* es un sistema mediante el cual dos o más personas pueden comunicarse a través de Internet, en forma simultánea, es decir en tiempo real (en línea, online), por medio de texto, audio y hasta video, sin importar si se encuentra en diferentes ciudades o países.

## ***Otros Servicios***

Los Portales Web, además de brindar servicios como los mencionados anteriormente, también se pueden crear servicios específicos a la organización. En el caso del Departamento pueden existir servicios como:

- Ofertas de trabajo y pasantías,
- Préstamo de equipos,
- Préstamo de libros,
- Noticias, etc.

### **2.1.3. Tecnologías Web**

Por *Tecnologías Web* se entiende al lenguaje utilizado en el desarrollo de aplicaciones Web, el cual está constituido por una serie de etiquetas y/o comandos que dan el formato al contenido y definen el comportamiento de una Página Web. Estas etiquetas o comandos son procesados por el Navegador del lado del cliente o por el Servidor (dependiendo del tipo de Tecnología), para finalmente mostrar el contenido al usuario en una forma más amigable y entendible.

### 2.1.3.1. *Tecnología del lado del Cliente*

En las *Tecnologías del lado del Cliente* la información es procesada por éste y el servidor se encarga únicamente de la respuesta de la petición. El cliente necesita tener un navegador o browser capaz de interpretar las respuestas que le da el servidor. Generalmente se usa este tipo de tecnologías para validaciones dentro de formularios, tomar acciones ante un evento del usuario, etc. Como tecnologías que se ejecutan del lado del cliente tenemos las siguientes.

#### ***HTML***

*HTML* significa Lenguaje de Marcado de Hipertexto (HyperText Markup Language). Es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que se presenta el texto y otros elementos de la página.

*HTML* se creó sin pensar en todos los posibles usos que se le iba a dar, sin embargo, pese a esta limitación, se han ido incorporando modificaciones con el tiempo, los cuales se convirtieron estándares del *HTML*. Gracias a Internet, *HTML* se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

#### ***DHTML***

Debido a las limitaciones que presenta el HTML, apareció *DHTML*, que significa HTML Dinámico (Dynamic HTML). *DHTML* es una extensión de HTML que permite, entre otras, incluir pequeñas animaciones y menús dinámicos en páginas Web. El código *DHTML* utiliza hojas de estilo (CSS) y JavaScript.

## ***JavaScript***

Es un lenguaje script utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje del lado del cliente más utilizado.

Con *JavaScript* se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones *JavaScript* y ejecutarlas para realizar estos efectos e interactividades.

## ***VbScript***

*VbScript* es un lenguaje script del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización no es tan masiva en comparación con Javascript.

Está basado en Visual Basic, por lo que, tanto en su sintaxis como en la manera de trabajar son muy similares. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en *VbScript*, pues este último es una versión reducida del primero.

El modo de funcionamiento de *VbScript* para construir efectos especiales en páginas Web es muy similar al utilizado en JavaScript y los recursos a los que se puede acceder también son los mismos.

## ***WML***

*WML* significa Lenguaje de Marcado Inalámbrico (Wireless Markup Language). En este lenguaje se escriben las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDAs) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos.



### 2.1.3.2. *Tecnología del lado del Servidor*

Las *Tecnologías del lado de Servidor* procesan la información antes de ser enviada al cliente. Esto brinda la ventaja de que un cliente no necesita tener instalado ciertos programas o aplicaciones para ver el contenido de la información. Para usar estas tecnologías es necesario contar con un servidor potente capaz de soportar la carga de procesamiento de peticiones de numerosos clientes debido a que el servidor es el que realiza la mayor parte del trabajo. Entre las *Tecnologías del lado del Servidor* tenemos las siguientes:

#### ***CGI/Perl***

Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente es poco utilizado por diversas razones, entre las que destacan la dificultad con la que se desarrollan los programas y la carga que supone para el servidor que los ejecuta.

Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos.

#### ***ASP y ASPX***

*ASP* (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor.

El tipo de servidores que emplea este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows usando el servidor Web que está pensado para esta tecnología, el Internet Information Server (IIS).

Con *ASP* se puede realizar muchos tipos de aplicaciones distintas. Permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor.

Actualmente existe la nueva versión de *ASP*, el *ASP.NET (ASPX)*, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. Sin duda, *ASPX* es mucho mejor que el *ASP* tradicional, ya que trae diversas mejoras entre las cuales se destacan:

**Rendimiento:** la aplicación se compila en una sola vez al lenguaje nativo, y luego, en cada petición tiene una compilación Just In Time, es decir se compila desde el código nativo, lo que permite un mejor rendimiento. También permite el almacenamiento del caché en el servidor

**Rapidez en programación:** mediante diversos controles, se puede con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.

**Servicios Web:** trae herramientas para compartir datos e información entre distintos sitios.

**Seguridad:** tiene diversas herramientas que garantizan la seguridad de las aplicaciones.

## ***JSP***

*JSP* (Java Server Pages) es una tecnología de servidor orientada a crear páginas Web con programación en Java. Con *JSP* se pueden crear aplicaciones Web que se ejecuten en varios servidores Web de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma.

En *JSP* se crean las páginas de manera parecida a como se crean en ASP o PHP. Se generan archivos con extensión *.jsp* que incluyen las sentencias Java a ejecutar en el servidor dentro de la estructura de etiquetas HTML. Antes de que sean funcionales los archivos, el motor *JSP* lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo *.class* (Byte codes de Java). Esta fase de traducción se lleva a

cabo habitualmente cuando se recibe la primera solicitud de la página .jsp, aunque existe la opción de precompilar el código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

Es necesario instalar un servidor de aplicaciones como Tomcat o Sun Java System Application Server (SJSAS), los cuales permiten procesar las páginas .jsp.

## ***PHP***

*PHP* (Hiptertext Preprocesor) es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

*PHP* se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como la de ser gratuito, independiente de plataforma, rápido y seguro.

Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web. Esto hace que sea cual sea el sistema, éste pueda ser compatible con el lenguaje, permitiendo portar el sitio desarrollado en *PHP* de un sistema a otro sin prácticamente ningún trabajo.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

### 2.1.3.3. Comparación entre las tecnologías

Tabla 1 Comparación de Bases de Datos

Parámetros	PHP	JSP	ASPX	Perl
Multiplataforma	✓	✓	Mono	✓
Compilador con análisis Semántico		✓	✓	
Lenguaje Pseudo Interpretado		✓	✓	
Tratamiento Avanzado de excepciones Nativo	✓	✓	✓	
Orientada a Objetos	Limitado	✓	✓	✓
Soporte	✓✓	✓	✓	✓
Licenciamiento (Free)	✓	✓		✓

La comparación entre las principales tecnologías es muy relativa debido a que en general ofrecen las mismas prestaciones en cuanto a rendimiento y características. Entre los principales parámetros de comparación tenemos:

- **Multiplataforma:** Esto quiere decir que puede ser ejecutada sobre diferentes Sistemas Operativos. El caso de Asp.Net, se ejecuta bajo plataformas Microsoft, aunque con el desarrollo del proyecto Mono, esta limitación sobre sistemas basados en Unix desaparecería.
- **Compilador con Análisis Semántico:** Esto quiere decir que el compilador no solo verifica los errores de sintaxis sino que además revisan la concordancia de los procesos. Con esto, se pueden detectar errores que puedan surgir durante etapas posteriores del desarrollo.
- **Lenguaje Pseudo Interpretado:** Los lenguajes pseudos interpretados son aquellos que necesitan un framework para poder ser ejecutados. Esto hace que puedan ser independientes de la arquitectura en que se están ejecutando.
- **Tratamiento Avanzado de Excepciones:** Usa el Try – Catch – Finally para manejo de excepciones.

- ***Orientada a Objetos:*** Manejo de clases y demás propiedades relacionadas con POO.
- ***Soporte:*** Facilidad de encontrar información disponible en Internet, foros, libros, artículo, etc.
- ***Licenciamiento:*** Tipo de licencias necesarias para el uso y desarrollo de aplicaciones.

#### 2.1.4. Sistemas de Administración de Base de Datos

Los *Sistemas de Administración de Base de Datos* (DBMS) son un conjunto de programas que se encargan de manejar todos los accesos a las bases de datos, así como su creación y mantenimiento. Básicamente estos sistemas se componen de un lenguaje de definición de datos (DDL - Data Definition Language), de un lenguaje de manipulación de datos (DML - Data Manipulation Language) y de un lenguaje de consulta (SQL - Structured Query Language).

##### 2.1.4.1. PostgreSQL

PostgreSQL es un servidor de base de datos relacional libre. Es una alternativa a otros sistemas de bases de datos de código abierto como MySQL, Firebird y MaxDB, así como sistemas propietarios como Oracle, SQL Server o DB2.

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características:

- **DBMS Objeto-Relacional:** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.
- **Altamente Extensible:** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL.
- **MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control),** es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios.
- **Cliente/Servidor:** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- **Write Ahead Logging (WAL):** La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos.

Debido a que se trata de una herramienta gratuita, existe una buena cantidad de información especialmente en la Internet, además se distribuye con sistemas Linux como Red Hat lo que aumenta su uso.

#### 2.1.4.2. *MySQL*

El servidor de bases de datos *MySQL* es la base de datos de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar.

El servidor *MySQL* fue desarrollado originalmente para manejar bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor *MySQL* ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad y seguridad hacen de *MySQL* un servidor bastante apropiado para ser usado como base de datos en Internet.

Actualmente la cantidad de información que se maneja en los sistemas modernos y a la arquitectura simple de *MySQL* hace que el procesamiento de datos se vuelva lento y la base inestable al manejar grandes volúmenes de información, limitando su uso a aplicaciones pequeñas.

El uso extenso que tiene *MySQL* en aplicaciones Web hace que exista una gran cantidad de manuales, tutoriales, foros, artículos, libros, etc. sobre la misma lo que facilita su desarrollo y favorece su creciente éxito

#### 2.1.4.3. *Oracle*

*Oracle* es básicamente una herramienta para la gestión de Bases de Datos pero a diferencia de las dos anteriores, ésta no es gratuita, es decir su adquisición conlleva la compra de licencias respectivas. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y

multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo, como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo: Access, MySQL, SQL Server, PostgreSQL etc.

*Oracle* es actualmente la base de datos más robusta en el mercado, mostrando gran estabilidad y velocidad con grandes volúmenes de información; además cuenta con un amplio soporte mediante manuales, tutoriales, libros y, a diferencia de las anteriores, la empresa *Oracle* puede proporcionar ayuda por la compra del producto.

#### 2.1.4.4. Cuadro comparativo

A pesar de que existen diversas bases de datos, para escoger entre ellas se debe realizar un análisis comparativo para determinar cual es la más conveniente. No siempre la más potente o poderosa es la mejor opción, sino que se debe establecer parámetros con los cuales se pueda medir el cumplimiento o no de necesidades del proyecto.

Entre las necesidades básicas del proyecto tenemos:

- El proyecto debe ser Open Source
- La base de datos tiene que ser escalable
- Base de datos rápida y liviana

En base a estos requerimientos, los parámetros se detallan a continuación:

- **Almacenamiento de Datos:** MySQL tiene diferentes mecanismos de almacenamiento disponibles. Existen algunos mecanismos de almacenamiento pero el uso de tablas InnDBO implica tener características más avanzadas y mejores prestaciones. Lo más importante es que InnoDB y BDB son los únicos tipos de tablas de transacción segura. PostgreSQL, por otro lado, usa solamente un mecanismo de almacenamiento llamado Postgres Storage System.



- **Integridad de Datos:** Una de las características críticas de cualquier motor de base de datos es la integridad de la información. ACID (Atomic, Consistent, Isolated, Durable) asegura la integridad de datos; esencialmente significa que cuando una transacción es llevada a cabo dentro de una base de datos, o la transacción es exitosa y es almacenada o nada es escrito. PostgreSQL y MySQL soportan ACID. Ambas también soportan rollbacks parciales de transacciones además de manejar los bloqueos (deadlocks). EL nivel de bloqueo que usa MySQL bloqueo a nivel de fila (row-level), por otro lado PostgreSQL usa Multy Version Concurrency Control (MVCC) por defecto, las nuevas versiones soportan bloqueo a nivel de fila. MVCC toma una foto instantánea de la base para luego serializarla.
- **Características Avanzadas:** PostgreSQL tiene muchas característica como las que Oracle, DB2, o MS-SQL tienen, como triggers, vistas, herencias, secuencias, stored procedures, cursores y tipos de datos definidos por el usuario. MySQL versión 5.0, soporta vistas, stored procedures y cursores. MySQL's versión 5.1, soportará triggers pero a diferencia de PostgreSQL, MySQL soporta características avanzadas como partición de datos.
- **Índices:** Oracle emplea sofisticadas estrategias de indexación a diferencia que las usadas en bases de datos open-source las cuales son mucho más primitivas. PostgreSQL y MySQL soportan indexación single column, multi-column, unique y primary key.
- **Tipos de Datos:** PostgreSQL y MySQL soportan la mayoría de los estándares de tipos de datos así como objetos. PostgreSQL soporta tipos de datos definidos por el usuario, a diferencia de MySQL que no lo hace.

- **Replication:** MySQL y PostgreSQL soportan escenarios de replicación single-master y multi-slave. PostgreSQL adicionalmente soporta replicación multi-master, multi-slave de terceros así como adicionales métodos de replicación.
- **Plataformas:** Oracle y DB2 corren sobre multiples plataformas, Microsoft SQL Server esta limitada a Windows. MySQL y PostgreSQL soportan plataformas como Windows, Linux, FreeBSD y MacOSX.
- **Metodos de Interface:** PostgreSQL y MySQL soportan ODBC y JDBC para conectividad en red los cuales proveen acceso en texto plano y altos niveles de seguridad como métodos de encriptación SSL. Los métodos de autenticación son importantes; MySQL usa un método simple en la que almacena toda la información de autenticación en una tabla. Cuando un usuario intenta acceder, MySQL compara sus credenciales con las de la tabla, verificando desde que máquina se puede conectar el usuario y a que recursos puede acceder. PostgreSQL puede usar un método similar pero también tiene otros como el de usar archivos remotos para definir que usuario puede acceder a una base de datos específica. También puede usar la autenticación de propio sistema operativo así como el de la base de datos. Así mismo ambas soportan acceso mediante C/C++, Java, Perl, Python, and PHP.
- **Backups:** Las bases de datos open-source pueden no llenar a cabalidad las necesidades de respaldo. PostgreSQL y MySQL tienen scripts que facilitan un contenedor para texto simple y sus esquemas y ofrecen una solución para hacer un respaldo de la base en caliente o respaldar la base sin necesidad de bajarla o apagarla. Estas bases pueden ser monitoreadas y respaldadas mediante software de terceros como Vertias NetBackup o Tivoli TSM mediante agentes que proveen respaldos en línea, pero son pocos los vendedores que crean agentes para PostgreSQL y MySQL. PostgreSQL usa un sistema llamado Write Ahead Logging

que provee chequeo de la consistencia de los datos. MySQL lo tiene también pero únicamente bajo tablas tipo InnoDB.

- **GUIs:** Existen en el mercado diversas herramientas con interfaz grafica que permiten administrar las bases de datos que van desde aplicaciones sobre el sistema operativo hasta interfaces Web.
- **Migración de Datos:** MySQL y PostgreSQL tienen utilidades que facilitan la migración desde base de datos comerciales y están disponibles por terceros, comerciales u open source. PostgreSQL incluye utilidades para migrar datos desde Oracle y MySQL.

**Tabla 2 Comparación de Bases de Datos**

	PgSQL	MySQL	Oracle
<b>Integridad de datos</b>			
ACID	✓	✓	✓
Row-level bloqueo	✓	✓	✓
Rollbacks parciales	✓	✓	✓
<b>Características avanzadas</b>			
Stored procedures	✓	✓	✓
Vistas	✓	✓	✓
Triggers	✓	✓	✓
Secuencias	✓	✓	✓
Cursores	✓	✓	✓
User-defined data types	✓		✓
<b>Índices</b>			
Single column	✓	✓	✓
Multi-column	✓	✓	✓
Primary key	✓	✓	✓
Full text	✓	✓	✓
<b>Replicación</b>			
Single-master	✓*	✓	✓
Multi-master	✓*		
<b>Interfaces</b>			
ODBC/JDBC	✓	✓	✓
C/C++, Java	✓	✓	✓

\* No disponible en versión gratuita

### 2.1.5. Protocolos

Los protocolos definen reglas que permiten que los dispositivos de una red se comuniquen entre sí. Existe gran variedad de protocolos, cada uno diseñado para alguna aplicación o propósito, como por ejemplo para transporte de datos, corrección de errores, correo electrónico, transferencia de archivos, etc. A continuación se describen algunos de estos protocolos.

#### 2.1.5.1. *TCP/IP*

Transfer Control Protocol/Internet Protocol o TCP/IP es el nombre común para referirse a un conjunto de protocolos desarrollados en el DoD (Departamento de Defensa) de los EE.UU. en los años '70 para soportar el desarrollo de las redes a nivel mundial. TCP e IP son los dos protocolos más conocidos del conjunto.

IP, o Internet Protocol, es la especificación que determina hacia donde son encaminados los paquetes, en función de su dirección de destino. TCP, o Transmission Control Protocol, se asegura de que los paquetes lleguen correctamente a su destino. Si TCP determina que un paquete no ha sido recibido, intentará volver a enviarlo hasta que sea recibido correctamente. Su arquitectura se representa en cuatro Capas, y cada capa maneja protocolos diferentes como:

**Tabla 3 Arquitectura TCP/IP**

Capa	Protocolos
Aplicación	TFTP (Trivial File Transfer Protocol)
	FTP (File Transfer Protocol)
	DNS (Domain Name Service)
	Telnet
Transporte	SNMP (Simple Network Management Protocol)
	SMTP (Simple Mail Transport Protocol)
	TCP (Transfer Control Protocol)
	UDP (User Datagram Protocol)
Internet	IP (Internet Protocol)
	ICMP (Internet Control Message Protocol)
	ARP (Address Resolution Protocol)
	RARP (Reverse Address Resolution Protocol)
Acceso a Red	Ethernet, FastEthernet
	PPP (Point to Point Protocol)
	SLIP (Serial Line Internet Protocol)
	FDDI (Fiber Distributed Data Interface)
	ATM (Asynchronous Transfer Mode)
	Frame Relay
	ARP (Address Resolution Protocol)
	RARP (Reverse Address Resolution Protocol)

Algunos de los motivos de su popularidad son:

- Independencia del fabricante
- Soporta múltiples tecnologías
- Puede funcionar en máquinas de cualquier tamaño
- Estándar de EEUU desde 1983

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador
- Conectividad Universal a través de la red
- Reconocimientos de extremo a extremo
- Protocolos estandarizados

El protocolo TCP/IP está diseñado para enrutar y tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores Web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

Un inconveniente de TCP/IP es que es más difícil de configurar y de mantener que NetBEUI o IPX/SPX; además es algo más lento en redes con un volumen de tráfico medio bajo. Sin embargo, puede ser más rápida en redes con un volumen de tráfico grande donde haya que enrutar un gran número de tramas.

El protocolo TCP/IP se puede utilizar en grandes redes empresariales como por ejemplo en campus universitarios o en complejos empresariales en donde utilizan muchos enrutadores y conexiones a servidores. También se puede utilizar en redes pequeñas en donde 100 ó 200 estaciones de trabajo acceden a servicios de intranet o internet mediante un servidor que ofrezca servicios Web.

#### 2.1.5.2. *HTTP*

El Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol) funciona con la World Wide Web, que es la parte de crecimiento más rápido y más utilizada de Internet. Una de las principales razones de este crecimiento sorprendente de la Web es la facilidad con la que permite acceder a la información.

Este protocolo permite usar una serie de métodos para indicar la finalidad de la petición. Se basa en otros conceptos y estándares como Uniform Resource Identifier (URI), Uniform Resource Location (URL) y Uniform Resource Name (URN), para indicar el recurso al que hace referencia la petición. Los mensajes se pasan con un formato similar al usado por el Internet Mail y el Multipurpose Internet Mail Extensions (MIME).

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, archivo multimedia o aplicación CGI) es conocido por su URL.

Generalmente es el cliente el que inicia la comunicación HTTP y consiste en la petición de un recurso del servidor.

#### 2.1.5.3. *FTP*

FTP es un servicio confiable orientado a conexión que utiliza TCP para transferir archivos entre sistemas que admiten FTP. El propósito principal de FTP es transferir archivos desde un computador hacia otro copiando y moviendo archivos desde los servidores hacia los clientes, y desde los clientes hacia los servidores. Cuando los archivos se copian de un servidor, FTP primero establece una conexión de control entre el cliente y el servidor, luego se establece una segunda conexión, que es un enlace entre los computadores a través del cual se transfieren los datos. La transferencia de datos se puede realizar en modo ASCII o en modo binario. Estos modos determinan la codificación que se usa para el archivo de datos que, en el modelo OSI, es una tarea de la capa de presentación. Cuando termina la transferencia de archivos, la conexión de datos se termina automáticamente. Una vez que se ha completado toda la sesión para copiar y trasladar archivos, el vínculo de comandos se cierra cuando el usuario se desconecta y finaliza la sesión.

TFTP es un servicio no orientado a conexión que usa el Protocolo de Datagramas del Usuario (UDP). TFTP está diseñado para ser pequeño y fácil de implementar. Por lo tanto, carece de la mayoría de las características de FTP. TFTP puede leer o escribir archivos hacia o desde un servidor remoto pero no puede enlistar directorios y actualmente no proporciona autenticación de usuarios. Es útil en algunas LAN porque opera más rápidamente que FTP y, en un entorno estable, funciona de forma confiable.

#### 2.1.6. Plataformas

El Sistema operativo es el software básico que controla una computadora. Es un programa muy especial, quizá el más complejo e importante en una computadora. Se encarga de iniciar la computadora y hace que reconozca el CPU, la memoria, el teclado, el sistema de vídeo, las unidades de disco, etc.

Además, proporciona la facilidad para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación. El sistema operativo tiene tres grandes funciones: coordina y manipula el hardware de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el mouse; organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas, y gestiona los errores de hardware y la pérdida de datos.

##### 2.1.6.1. *UNIX*

*UNIX* es un sistema operativo multitarea, multiusuario, creado en 1969 por los investigadores Thompson y Ritchie de los Laboratorios Bell, en Estados Unidos. Las primeras versiones fueron escritas en assembler, pero muy pronto fue reescrito en lenguaje C.



En sus primeros años, no se lo utilizó comercialmente, sino que se lo usaba para proyectos de investigación en laboratorios y se distribuía gratuitamente en las universidades, donde tuvo mucha aceptación.

En 1975, Bell decidió comercializarlo. Dado que el sistema se vendía con una licencia que permitía modificarlo y redistribuirlo, a lo largo del tiempo fueron surgiendo una gran variedad de sistemas derivados del *UNIX* original. Los más conocidos, actualmente, son: Solaris, AIX, HPUX, SCO, BSD. La principal característica de los sistemas operativos mencionados es que solo funcionan con los servidores para los que fueron diseñados; por ejemplo, Solaris solo funciona con los servidores Sun, AIX con los IBM, etc.

Debido a la aparición de esta diversidad de sistemas *UNIX*, se hizo necesario normalizarlos para que cumplan con determinadas normas (POSIX, UNIX95, etc), para permitir la compatibilidad entre los diferentes sistemas.

#### 2.1.6.2. *Linux*

*Linux* es una implementación de libre distribución UNIX para computadoras personales, servidores, y estaciones de trabajo. Como sistema operativo, *Linux* es muy eficiente y tiene un excelente diseño.

Linux es multitarea, multiusuario, multiplataforma y multiprocesador; en las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado

pero puede usar los de los otros sistemas; y soporta redes tanto en TCP/IP como en otros protocolos.

### 2.1.6.3. *Windows*

Existen varios sistemas de operativos Windows desarrollador por Microsoft como las versiones 3.1, 9x, Millenium y XP destinadas para usuarios domésticos; además, existen versiones como la 3.11, NT, 2000 y 2003 destinadas para múltiples propósitos que son capaces de manejar una gran gama de funciones de servidor, tanto de manera centralizada como distribuida.

Windows NT representó la primera propuesta de Microsoft para desarrollar un sistema operativo orientado a la administración de redes, capaz de competir con sistemas clásicos como UNIX. A partir de NT se fue evolucionando hasta la versión actual Windows Server 2003.

Windows Server 2003 ha mostrado mejoras en seguridad y administración, dándole mayor robustez al sistema. Entre los principales servicios que puede brindar están:

- Servidor de archivos e impresión
- Servidor Web y aplicaciones Web
- Servidor de correo
- Terminal Server
- Servidor de acceso remoto/red privada virtual (VPN)
- Servicio de directorio, Sistema de dominio (DNS), y servidor DHCP.

### 2.1.7. Administrador de Contenidos

Un *administrador de contenidos* o también llamado editor de contenidos, no es más que un sistema que permite administrar Páginas Web dentro de un Sitio o Portal mediante la edición de su contenido. Este sistema permite el mantenimiento de Portales o Sitios de una forma más sencilla, fácil y rápida debido a que permiten alterar la información que muestra una Página Web sin necesidad de editar o reescribir el código.

Un Portal generalmente está ligado a un *Administrador de Contenidos* debido a que la cantidad de información manejada podría ser muy grande como para ser mantenida por un Administrador o WebMaster mediante la edición del código de cada página.

El Administrador de Contenido consiste en el desarrollo de una interfaz que permita a un Administrador o WebMaster cambiar contenidos o insertar objetos en las páginas, evitando así las confusiones que puede darse manipulando códigos, etiquetas o comandos.

## 2.2. Metodología RUP para el desarrollo del proyecto

### 2.2.1. UML

Unified Modeling Language o UML es un lenguaje de notaciones utilizada para el desarrollo de sistemas por medio de conceptos orientados a objetos. UML unifica una serie de métodos de análisis y diseño orientada a objetos (OOA&D) como los métodos de Booch, Rumbaugh (OMT) y Jacobson; pero cabe recalcar que es un lenguaje de modelado mas no un método ni metodología.

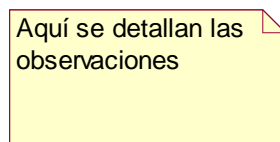
UML define una notación y un metamodelo. La *notación* es la parte gráfica que se ve en los modelos, es decir la sintaxis del lenguaje del modelado, por ejemplo las asociaciones, multiplicidades, clases, etc. Un *metamodelo* es la aplicación de varias notaciones para poder expresar un todo a través de diagramas como diagramas de clases,

etc. Existen diversos tipos de diagramas que representan ciertas facetas de un proceso, estos son:

- Diagramas de Casos de Uso
- Diagramas de Clase
- Diagramas de Iteración
- Diagramas de Paquetes
- Diagramas de Estado
- Diagramas de Actividad
- Diagramas de Emplazamiento

#### 2.2.1.1. *Notas*

Una *Nota* es un símbolo que puede ser usado en todos los tipos de diagramas, su función es que dentro de ellas se puedan realizar anotaciones, comentarios o aclaraciones con respecto a un diagrama, parte de él o un objeto específico. Su simbología es:



**Figura 1 Nota**

#### 2.2.1.2. *Diagramas de Caso de Uso*

Un *diagrama de Caso de Uso* representa la forma en que interactúa el usuario con un sistema de cómputo.

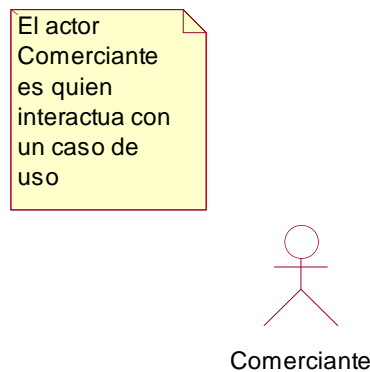
- Un caso de uso capta las principales funciones visibles para el usuario
- Un caso de uso puede ser pequeño o grande

- Un caso de uso logra un objetivo discreto para el usuario

Por lo general la obtención de la información para desarrollar un caso de uso proviene de la recopilación mediante charlas con el usuario y analizando distintos aspectos de lo que desean hacer con el sistema. Tiene diversos componentes como los casos de uso, los actores, y definiciones como uses y extends.

### ***Actores y Caso de Uso***

Como *Actor* se entiende al usuario cuando esta desempeñando un papel ante el sistema. Los actores efectúan o llevan a cabo un caso de uso o varios de ellos, y de igual manera un caso de uso puede llevarse a cabo por varios actores. Un actor no necesariamente es una persona, sino que puede ser un sistema externo que interactúa con el sistema.



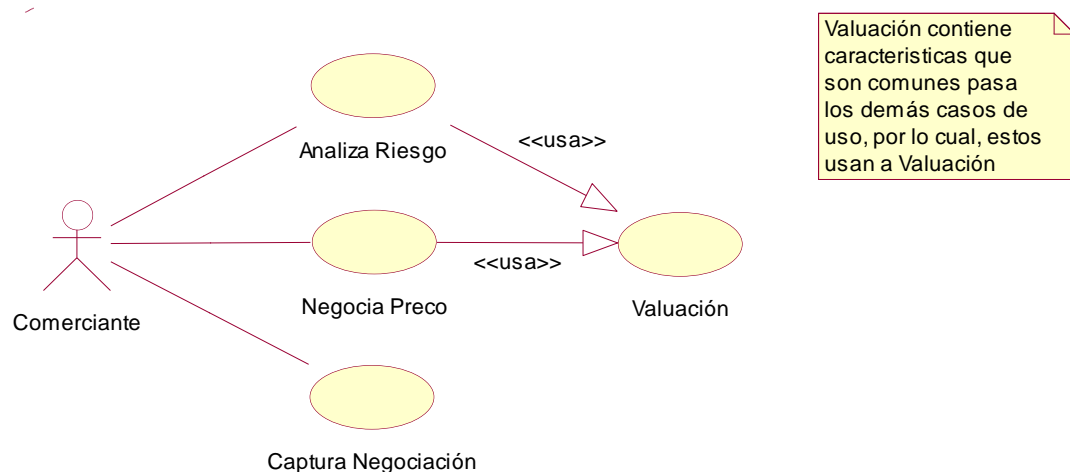
**Figura 2 Actor**

### ***Uses y Extends***

*Uses* y *Extends* representan las relaciones entre los casos de uso. La relación *Extend* se usa entre casos de uso que hacen las mismas funciones pero una es una extensión de la otra, tiene otras características que ayudan a realizar otras funciones, por ejemplo, un caso de uso puede tener una función determinada para realizar algo normalmente, pero si esa acción falla existe otro caso de uso que extiende las capacidades del caso de uso normal

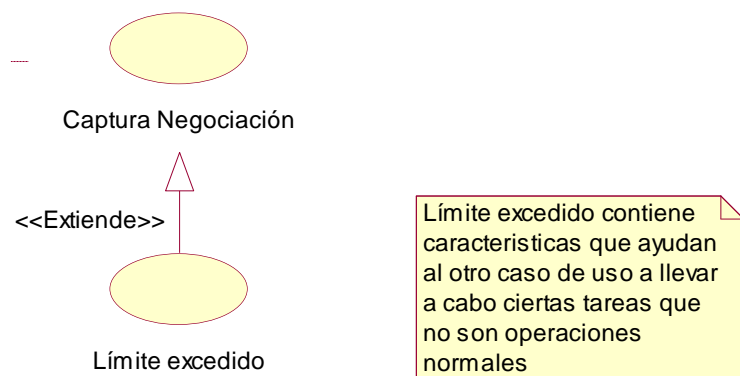
pero para controlar dichas fallas. Se puede contemplar un caso de uso en la fase de inicio, pero en la fase de elaboración puede dividir el caso de uso complejo en varios casos de uso y varias extensiones.

La relación *Uses* ocurre cuando se tiene una porción de comportamiento similar en más de un caso de uso y no se requiere copiar la descripción de tal conducta en otro caso de uso, por ejemplo, varios casos de uso pueden realizar una actividad que debe ser complementada con otra, entonces estos casos de uso usan la característica de otra.



**Figura 3 Uses**

En general se usa *Extend* para describir variaciones de una conducta normal y se usa *Uses* cuando existen repeticiones de conductas en varios casos de uso y se desea evitar las repeticiones.



**Figura 4 Extends**

### 2.2.1.3. Diagramas de Clase

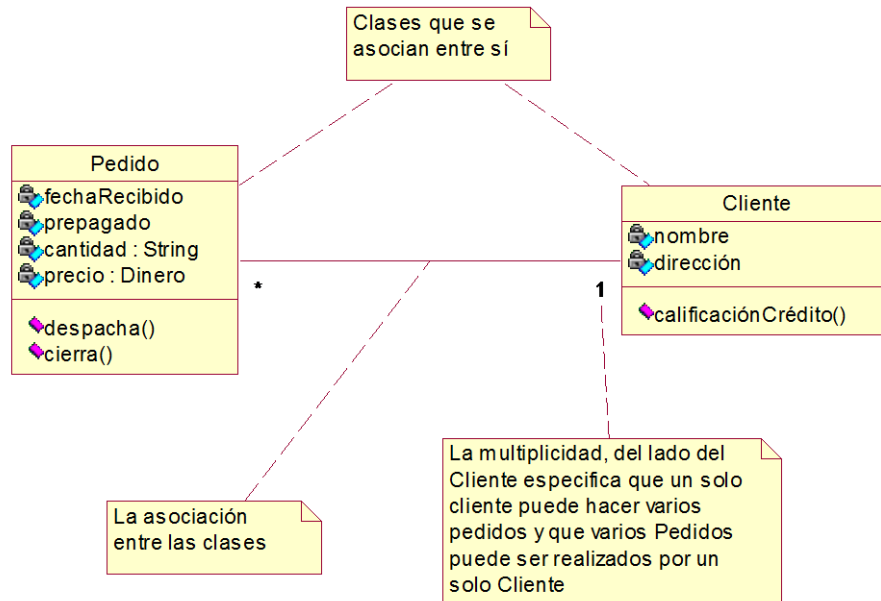
Un *Diagrama de Clase* describe los tipos de objetos que existen en el sistema y las diversas clases de relaciones estáticas que existen entre ellos. Estas son Asociaciones y los Subtipos. En un *Diagrama de Clase* se muestran los atributos, operaciones y restricciones de una clase.

#### **Asociación**

Representan la relación entre instancias de clase, esta relación tiene una multiplicidad, la cual define la cantidad de objetos que participan en la relación y pueden ser:

**Tabla 4 Multiplicidad de asociaciones**

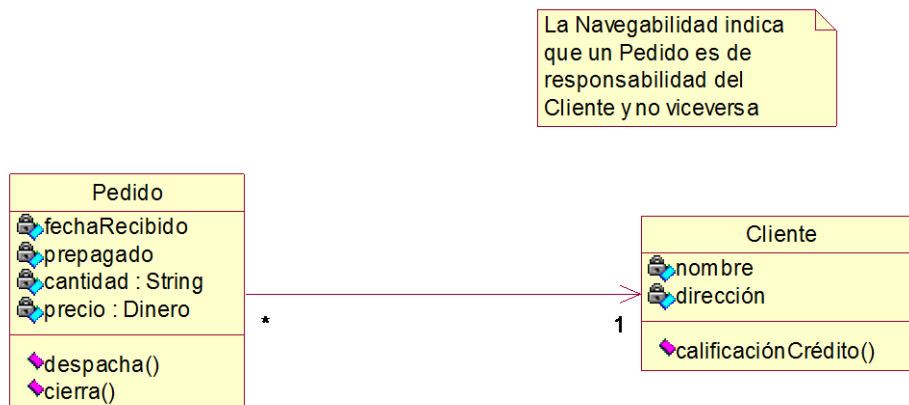
Multiplicidad	Descripción
1	Solo 1
1..*	Su asociación puede ser entre 1 e infinito
1..n	Su asociación puede ser entre 1 y <i>n</i>
*	Su asociación puede ser entre 0 e infinito



**Figura 5 Asociación**

### *Navegabilidad*

Son flechas en las asociaciones, pueden indicar la responsabilidad o la capacidad correspondiente para iniciar una actividad. Pueden ser unidireccionales o bidireccionales pero cuando no se define navegabilidad es porque se desconoce o porque es bidireccional. Generalmente son denominadas con un verbo con lo que se puede formar una oración para entender la relación, pero pueden ser un sustantivo para determinar cierto papel como para representar responsabilidades u operaciones.



**Figura 6 Navegabilidad**



## ***Atributos***

Son las características de la clase. Su sintaxis es:

*visibilidad nombre: tipo = valor por omisión*

La visibilidad puede ser:

- + → Public
- # → Protect
- - → Private

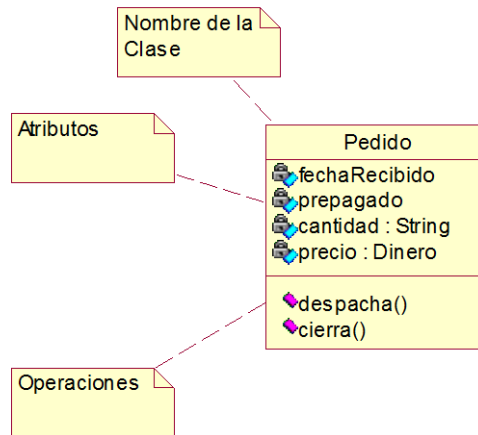
## ***Operación***

Son los procesos que puede llevar a cabo una clase, por lo general no se muestran las *operaciones* que únicamente manipulan los valores de atributos, su sintaxis es:

*visibilidad nombre (lista\_parámetros): expresión-tipo\_dato a regresar {cadena\_propiedades}*

- La lista\_parámetros pueden ser opcionales y su sintaxis es igual a la de los atributos
- expresión-tipo\_dato es una especificación opcional dependiente del lenguaje
- cadena\_propiedades indica valores de propiedad que se aplican a la operación dada

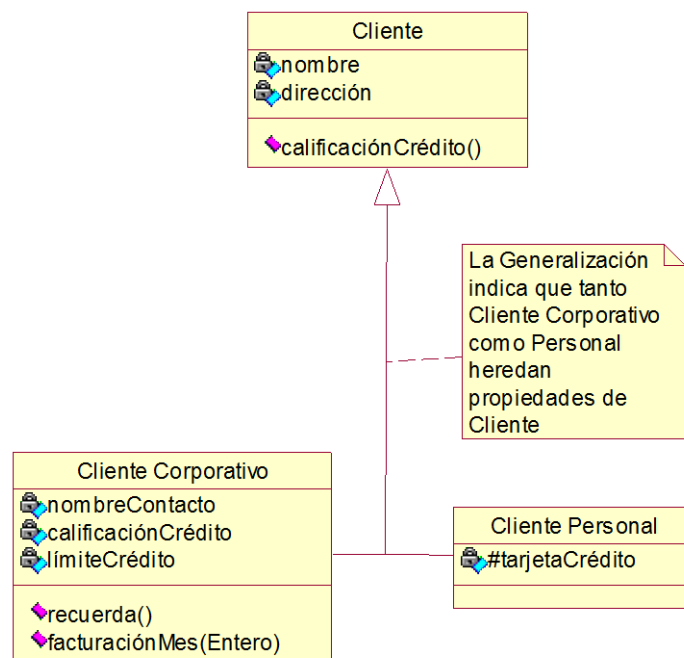
Una operación es algo que se invoca sobre un objeto, mientras que un método es el cuerpo de la operación, esta diferencia se aplica cuando se habla de polimorfismo.



**Figura 7 Operación**

### **Generalización**

En la *Generalización* existen una clase denominada *Supertipo* que es la clase general, y otra clase denominada *Subtipo*, las cuales tienen similitudes. Se lo puede entender como si se tratase de una herencia ya que como concepto clave se entiende que todo lo que se siga de una clase Supertipo es aplicable también para una clase Subtipo.



**Figura 8 Generalización**

## Reglas de Restricción

Las *restricciones* indican las limitantes en una clase. No existe en UML una sintaxis para definir las a más de que tienen que ser escritas entre llaves y puede ser escrita en lenguaje informal.

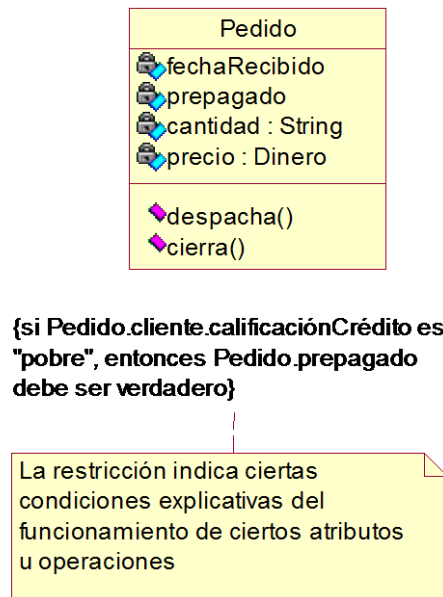


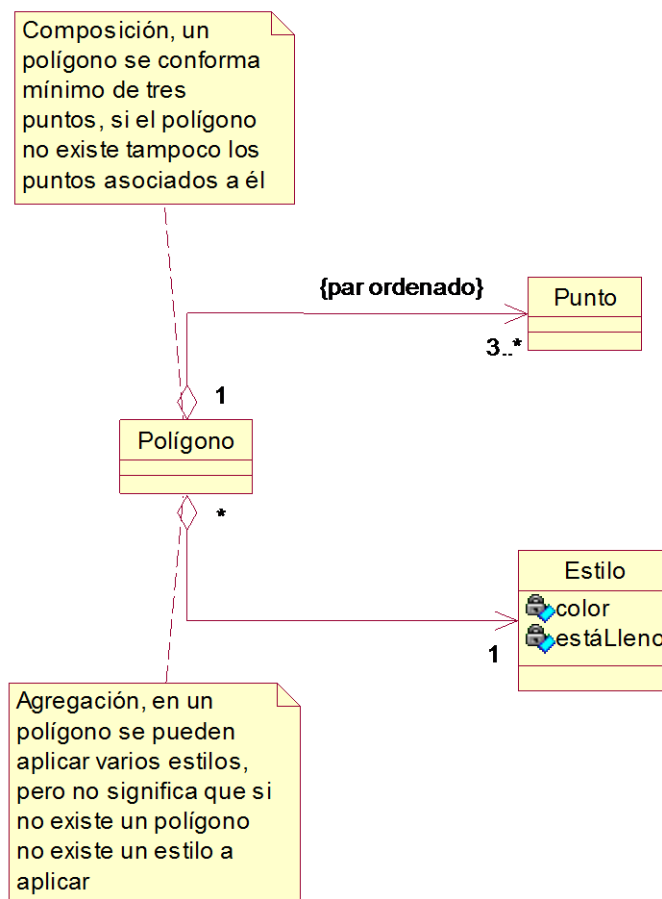
Figura 9 Restricciones

## Agregación y Composición

La *Agregación* y la *Composición* son un tipo de asociación especial, ya que da a entender entre otras cosas la forma en que trabajan y sobre la multiplicidad.

La *composición* se refiere a que una clase se conforma necesariamente de otras clases, por ejemplo, un reloj se compone del mecanismo, el horero, el minuterero básicamente. Un reloj sin el horero no tiene sentido, y la manecilla sin estar en el reloj no tiene caso. Se simboliza con un rombo relleno y la multiplicidad en este lado es de 1.

La *agregación* por otro lado es otro tipo de componente pero no tan esencial, como la forma del reloj, los colores que usa, etc. Se representa con un rombo sin relleno y su multiplicidad es \*.

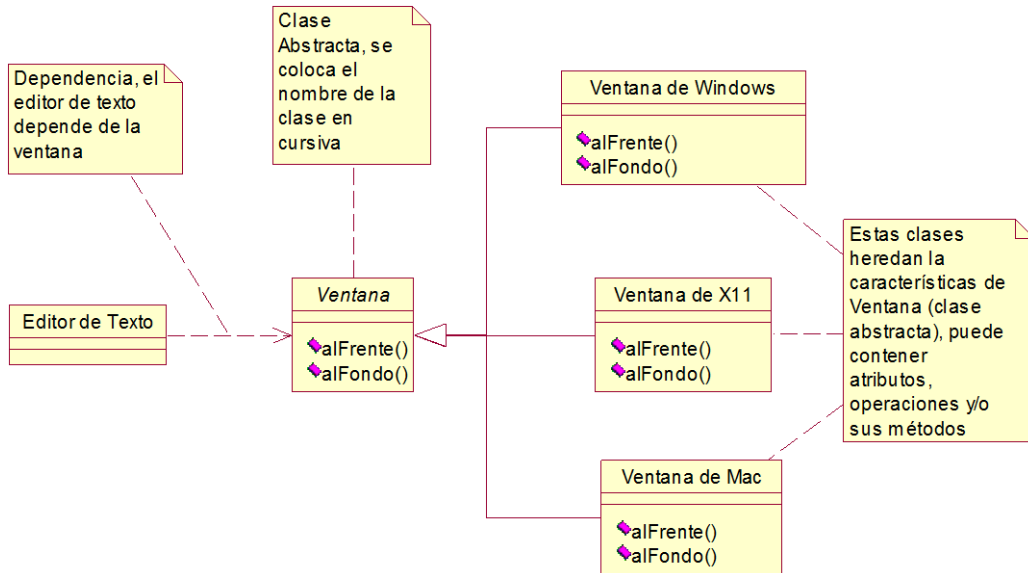


**Figura 10 Agregación y Composición**

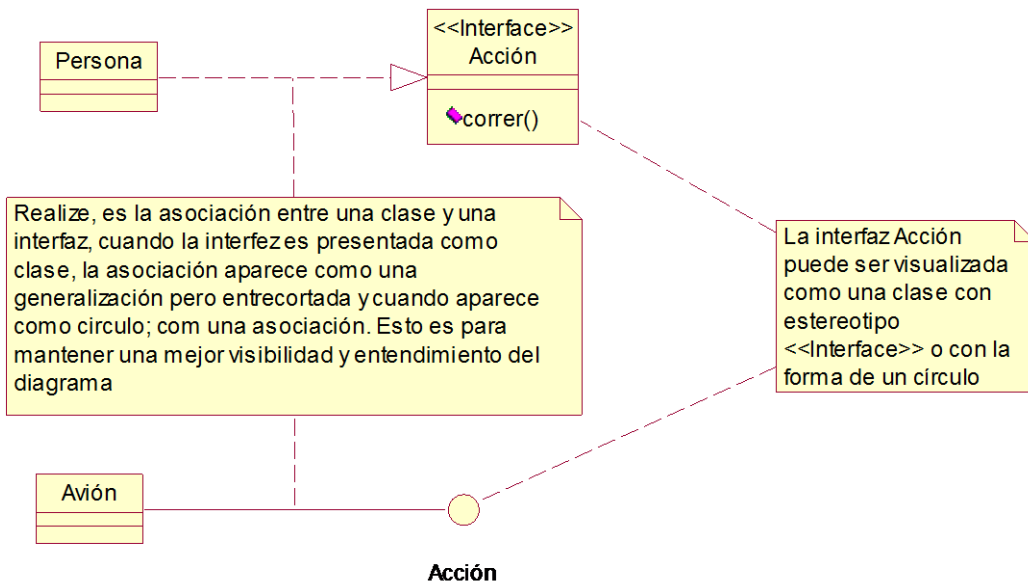
### ***Interfaz y Clase abstracta***

Una *clase abstracta* define a clases que no se implementan, únicamente sirven para generalizar características que puede ser heredadas por otras clases. Una interfaz es una clase abstracta pero que sirve para especificar el comportamiento que deben tener las clases que la implementen.

Una *clase abstracta* puede contener atributos y operaciones con o sin sus métodos, pero una interfaz no contiene atributos ni la implementación de las operaciones, es decir los métodos.



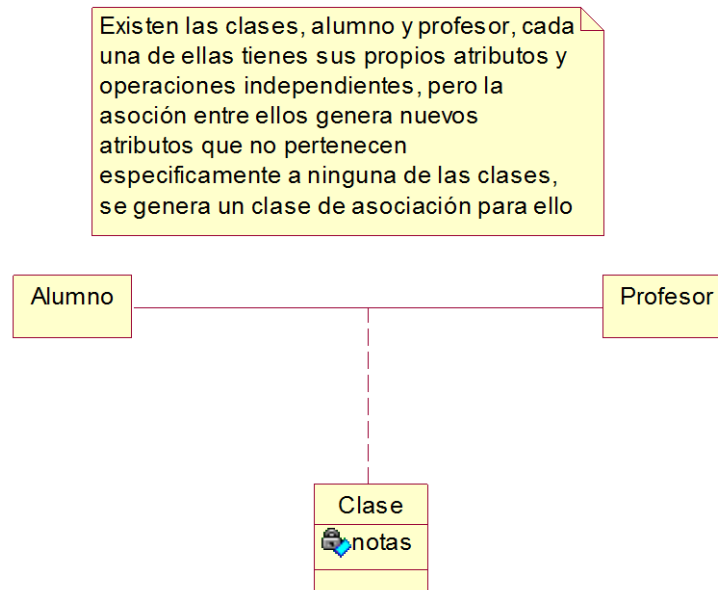
**Figura 11 Clase Abstracta**



**Figura 12 Interface**

## ***Clase de Asociación***

Una *Clase de Asociación* es un tipo de clase que permite añadir características a una asociación entre dos clases, esto quiere decir que se pueden tener dos clases asociadas entre sí, pero existen atributos entre ellas, pero que necesariamente no se pueden especificar en ninguna de las clases sino a la asociación.



**Figura 13 Clase de Asociación**

### ***2.2.1.4. Diagramas de Iteración***

Los *Diagramas de Iteración* describen la manera en que los grupos de objetos colaboran entre sí; mostrando cierto número de ejemplos de objetos y mensaje que se pasan entre los objetos dentro de un diagrama de casos de uso.

### ***Diagrama de Secuencia***

Estos diagramas son de mucha utilidad ya que permiten visualizar en detalle como, cuando y por qué interactúan los objetos.

En este tipo de diagramas los objetos son ilustrados como cajas y su línea de vida por la línea punteada que sale de cada caja. Los mensajes son flechas entre las líneas de vida y

especifican el orden en que estos transcurren hacia abajo. Dentro de los mensajes, existen varios indicadores útiles como los son la *Condición* y el *Marcado de Iteración*, estas se escriben entre corchetes “[ ]”, la condición sólo se envía cuando es verdadera y el marcado de iteración se antepone un asterisco “\*”. Además existen *Líneas de Regreso* cuando un mensaje existente retorna, no uno nuevo y aparecen como líneas punteadas.

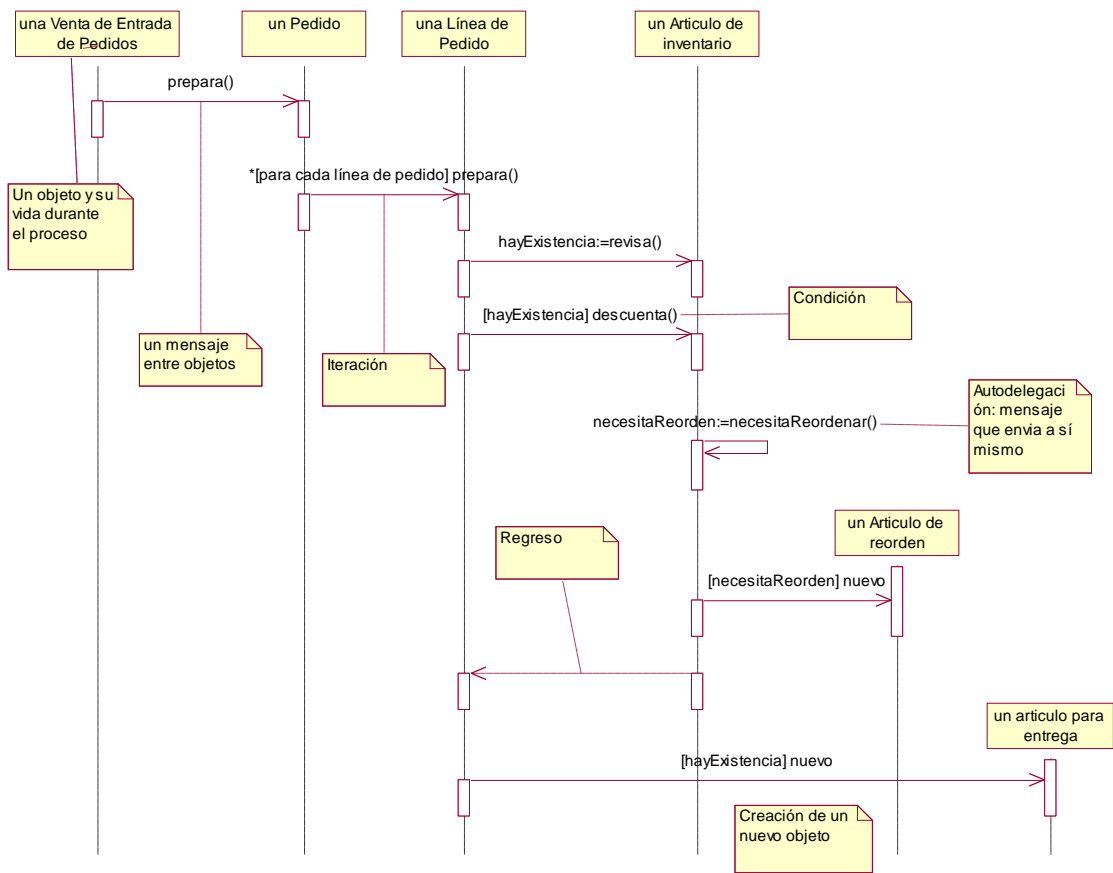
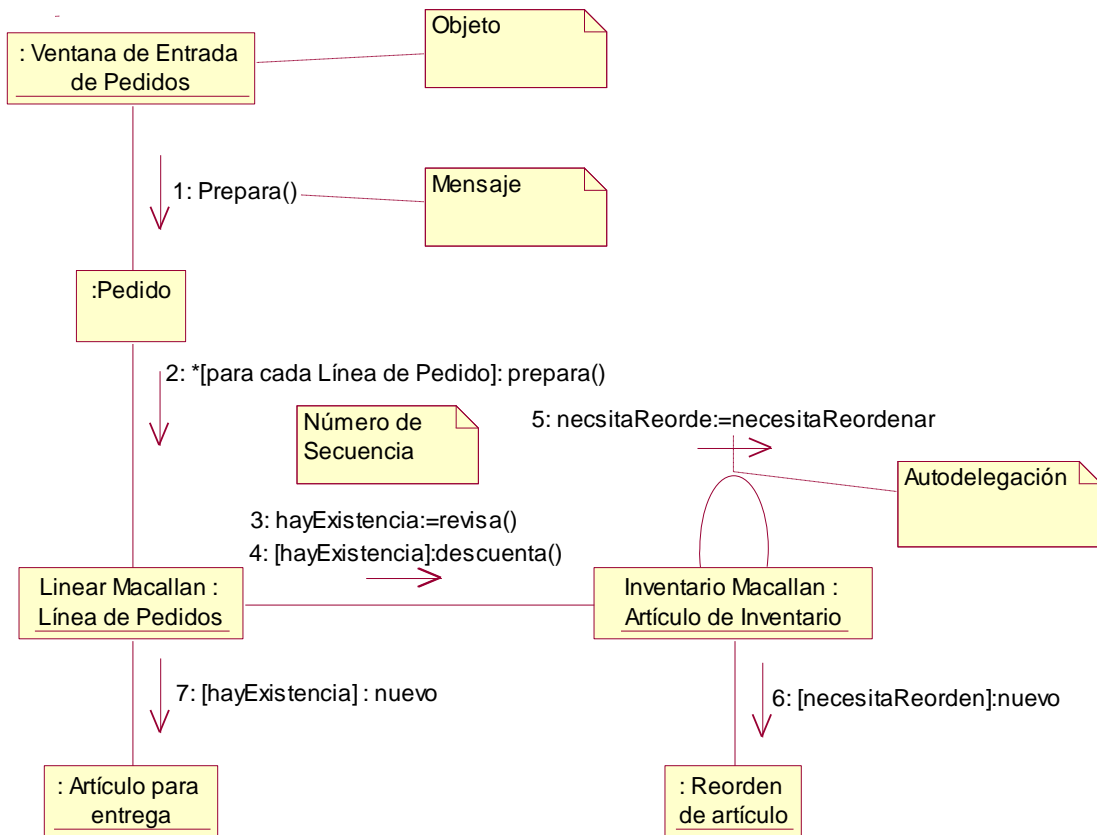


Figura 14 Diagrama de Secuencia

### Diagrama de Colaboración

El *diagrama de colaboración* es otro tipo de esquema en el que se visualiza el orden de ocurrencia de las actividades y utiliza los mismos marcadores que el diagrama de secuencia.

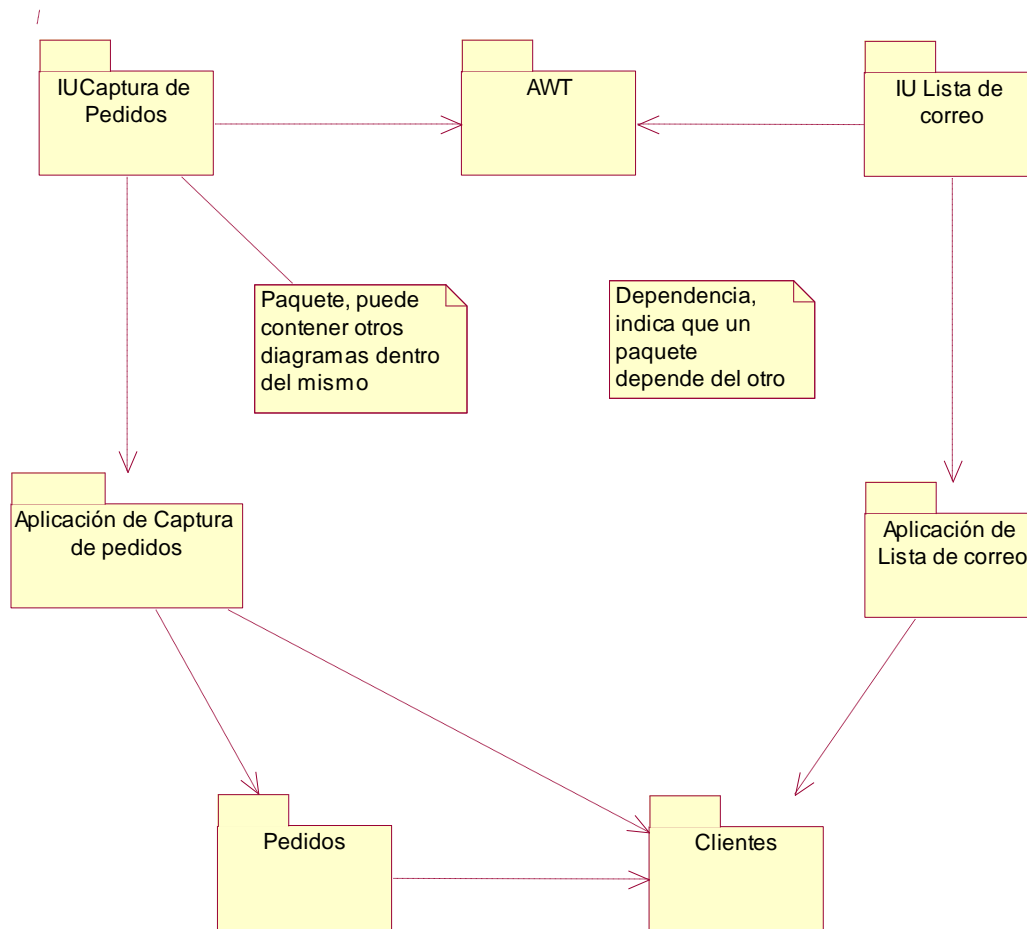


**Figura 15 Diagrama de Colaboración**

### 2.2.1.5. Diagramas de Paquetes

Los *paquetes* se usan para fragmentar un sistema en varios más pequeños, la idea es agrupar las clases o cualquier elemento en unidades de niveles más altos llamado paquete. Esta agrupación es arbitraria, pero se pueden agrupar los elementos de acuerdo a las dependencias entre ellos.





**Figura 16 Diagrama de Paquetes**

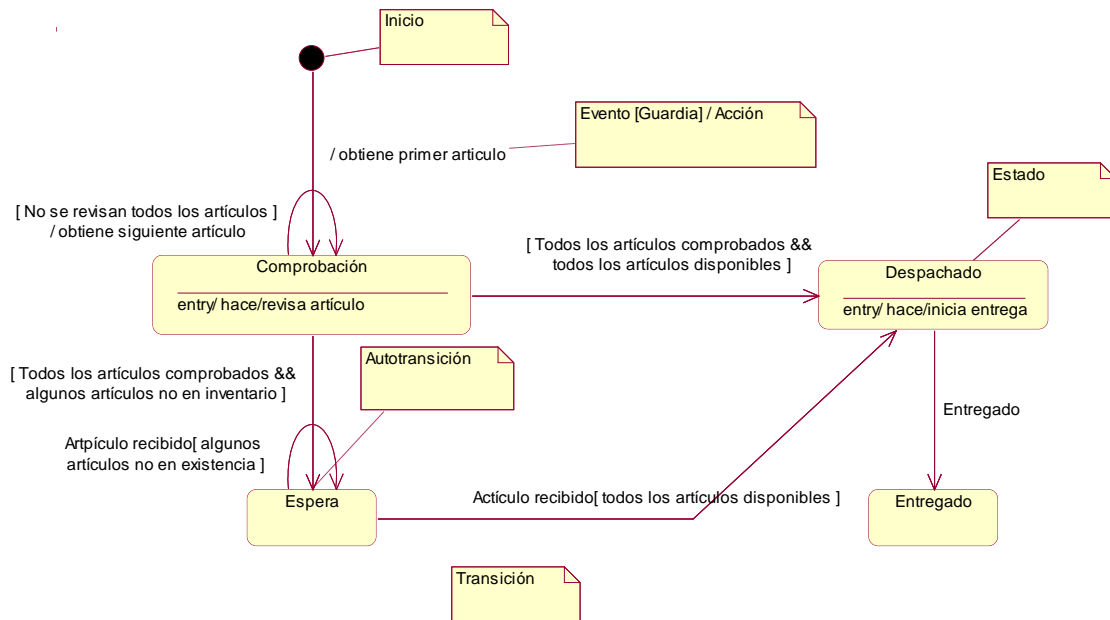
### 2.2.1.6. Diagramas de Estado

Con los *diagramas de estado* se pueden definir el comportamiento de un sistema a través de posibles estados de los objetos del sistema y la manera en que estos cambian como resultado de eventos que se producen en dichos objetos.

En los *diagramas de estado* intervienen un punto de inicio, actividades, estados y transiciones.

La sintaxis para un transición o actividad es: Evento [Guardia] / Acción, en donde evento indica algo que ocurre sobre un estado, el guardia es una condición lógica que retorna verdadero o falso y permite un cambio de estado cuando es verdadero y la acción

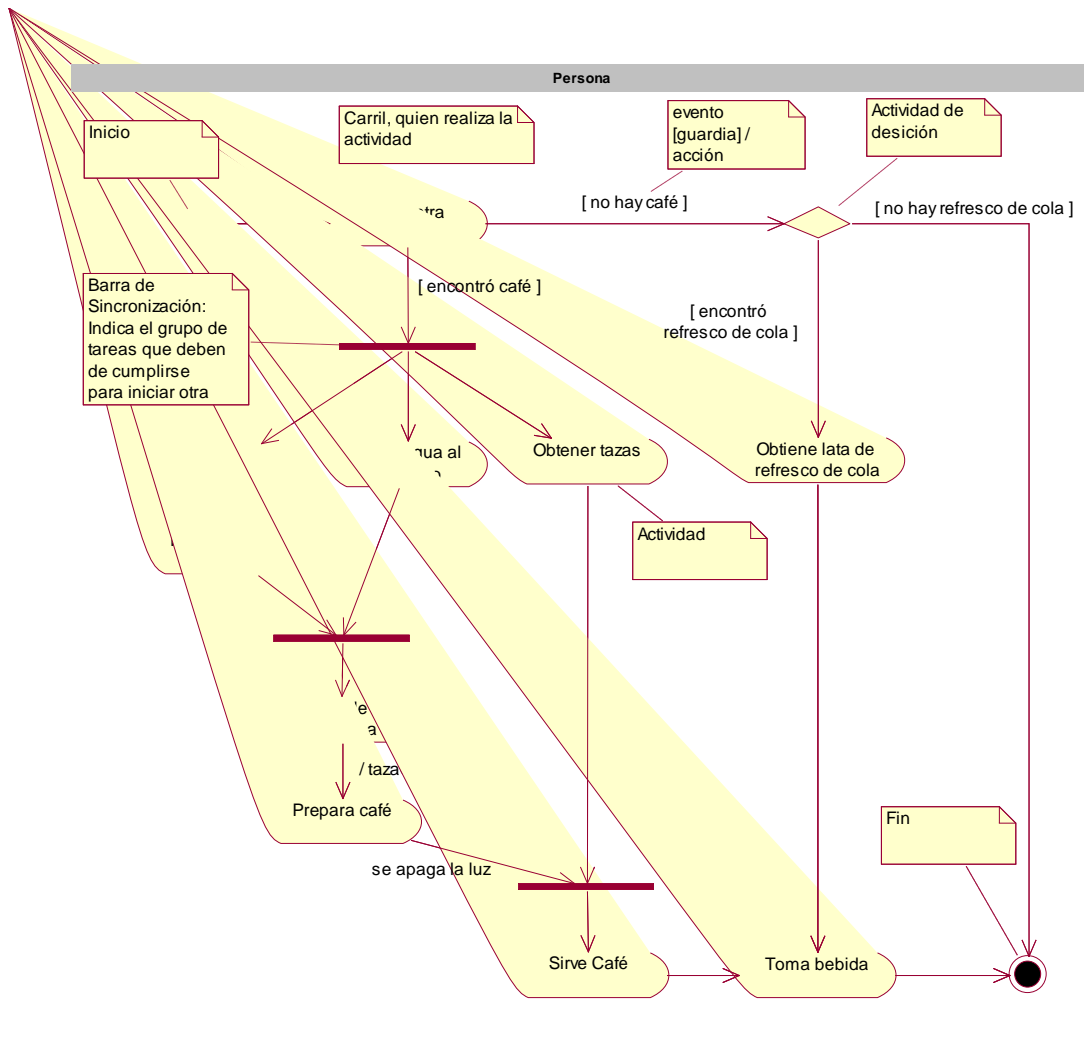
es lo que se ejecuta cuando cambia de estado. Cuando no se especifica una acción, indica que el cambio del estado ocurre apenas finaliza una actividad.



**Figura 17 Diagrama de Estado**

### 2.2.1.7. Diagramas de Actividad

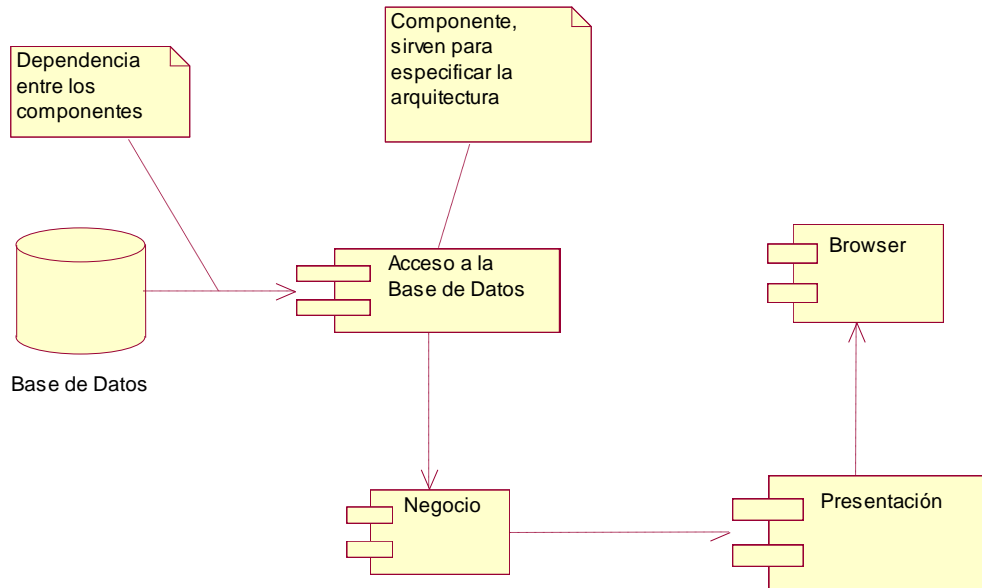
Los *diagramas de actividad* son en las conexiones de los flujos de trabajo, además para la descripción del comportamiento para los procesos en paralelo. La sintaxis para las transiciones es parecida al diagrama de estado.



**Figura 18 Diagrama de Actividad**

### 2.2.1.8. Diagramas de Componentes

El *Diagrama de Componentes* muestra la relación entre componentes de software, sus dependencias, su comunicación, su ubicación y otras condiciones. Los componentes también pueden exponer las interfaces, las cuales son los puntos de entrada o los servicios que un componente está ofreciendo y dejando disponibles a otros componentes de software y clases. Generalmente, un componente está compuesto por numerosas clases y paquetes de clases internos.



**Figura 19 Diagrama de Componentes**

### 2.2.2. Rational Unified Process

*Rational Unified Process (RUP)* constituye una arquitectura que utiliza el Unified Modeling Language como lenguaje estándar gráfico para visualizar, especificar, construir y documentar un sistema de software.

*RUP* es un marco de trabajo que provee una disciplina metodológica para el desarrollo de software. El objetivo es el de producir un software de alta calidad que reúna todas las necesidades del usuario final dentro de un cronograma y presupuesto dado.

Su utilización provee de las “Mejores Prácticas” que incluyen:

#### 2.2.2.1. Desarrollo iterativo

La funcionalidad del sistema debe ser entregada en unas series sucesivas de versiones, anulando los riesgos críticos y obteniendo una retroalimentación en cada versión.

#### 2.2.2.2. *Administración de requerimientos*

*La administración de requerimientos* es un acercamiento sistemático para producir una propuesta, organizar comunicar y dirigir los cambios en los requerimientos de un sistema o aplicación; lo que permite un mejor control sobre la satisfacción del cliente, presupuesto y cronograma.

#### 2.2.2.3. *El uso de arquitectura basada en componentes*

RUP provee de una vía metódica y sistemática para diseñar, desarrollar y validar una arquitectura. Con esto se obtiene flexibilidad y reusabilidad de código.

#### 2.2.2.4. *Software de modelamiento visual*

Los *modelos* son simplificaciones de la realidad que nos ayudan a entender y graficar un problema y la solución de un sistema como un todo. RUP usa Unified Modeling Language (UML) como un *lenguaje estándar gráfico* para visualizar, especificar, construir y documentar un sistema.

#### 2.2.2.5. *Verificación continua de la calidad del software*

La *calidad* es responsabilidad de todos los miembros involucrados en el desarrollo de una organización y debe ser mantenida durante todo el ciclo de vida hasta su implementación para conseguir *calidad* tanto en cada proceso como en el producto final.

### 2.2.3. Arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La *arquitectura* de RUP posee dos dimensiones:

- La dimensión horizontal representa el tiempo y muestra las fases e iteraciones que ocurren sobre la vida del proyecto
- La dimensión vertical representa el contenido o agrupaciones lógicas conocidas como disciplinas.

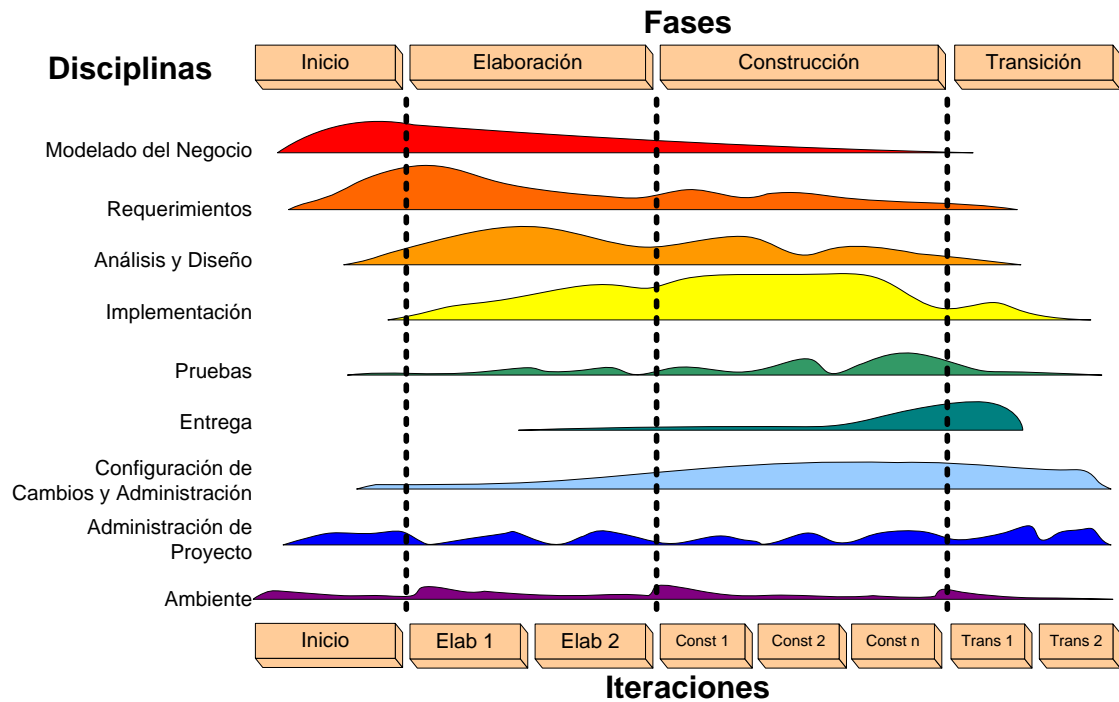


Figura 20 Arquitectura de RUP y sus Dimensiones<sup>1</sup>

<sup>1</sup> Gráfico tomado de Rational Unified Process ([http://en.wikipedia.org/wiki/Rational\\_Unified\\_Process](http://en.wikipedia.org/wiki/Rational_Unified_Process)), sección *Discipline and Workflow*.

El énfasis en las disciplinas se lleva en todas y cada unas de las fases y en cada iteración, por ejemplo de esta manera se mantiene el control sobre los nuevos requerimientos en todo momento.

#### 2.2.4. Conceptos

Los conceptos son estructuras que se manejan en un estándar Meta – Modelo que describen los procesos del software e incluyen:

##### 2.2.4.1. *Artefactos*

Es lo que se produce y pueden tener varias figuras o formas, tal como:

- Un modelo, tal como un caso de uso o diseño del modelo. Contiene elementos del modelo (sub artefactos) como clases, casos de uso y diseños de subsistemas.
- Base de Datos u otros tipos de repositorios de información tabular como hojas de cálculo.
- Código fuente o ejecutables.
- Varios tipos de documentos, como especificaciones de requerimientos, un plan o plan de desarrollo de software.

##### 2.2.4.2. *Actividades*

Se refiere a como desempeñar un trabajo. Una *actividad* es desempeñada por un rol. Usualmente define una serie de pasos que involucra la creación o actualización de uno o más artefactos. Algunos ejemplos de *actividades* son:

- Buscar actores o casos de uso: Una *actividad* desempeñada por el rol del analista de sistema para identificar los niveles de requerimiento funcional en términos de actor o casos de uso.
- Describir una distribución: Una *actividad* desempeñada por el rol del arquitecto de software para describir la distribución del software a través de múltiples procesadores.

#### 2.2.4.3. Roles

Es quien desempeña un trabajo. Define un comportamiento o responsabilidades de un individuo, puede parametrizar trabajos individuales entre sí como un equipo dentro del contexto de la ingeniería de software en la organización.

Los *roles* no son individuales, sino que describe responsabilidades. Un individuo puede tomar varios *roles* en un tiempo determinado y frecuentemente cambiará de *roles* durante la duración del proyecto.

#### 2.2.5. Disciplinas o Flujos de trabajo

Un *flujo de trabajo* describe cuales son las actividades que son desarrolladas entre sí y el orden en que éstas son desarrolladas. Estas actividades son una secuencia que muestra como y en que orden trabajar, a esto se le denomina un grupo de trabajo.

RUP describe para cada disciplina un *flujo de trabajo*, los *flujos de trabajo* definidos en RUP son: Modelado del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas y Entrega, las cuales son los principales para el desarrollo de un sistema.

Con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así



como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables. A continuación se dará una explicación de cada flujo de trabajo.

#### 2.2.5.1. *Modelado del negocio*

El flujo de trabajo que resulta de esta disciplina es la de mostrar el funcionamiento de la organización donde se implantará una determinada aplicación. Los objetivos del modelado de negocio son:

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado (organización objetivo).
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.
- Derivar los requisitos del sistema necesarios para apoyar a la organización objetivo.

Para lograr estos objetivos, el modelo de negocio describe como desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de Casos de Uso del negocio y un Modelo de Objetos del Negocio. Complementario a estos modelos, se desarrollan otras especificaciones tales como un Glosario.

### 2.2.5.2. *Requisitos*

Este flujo de trabajo establece lo que debe de hacer el sistema. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos. Los objetivos del flujo de datos Requisitos son:

- Establecer y mantener un acuerdo entre clientes y coordinadores de IT sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requisitos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

En este flujo de trabajo, y como parte de los requisitos de facilidad de uso, se diseña la interfaz gráfica de usuario. Para ello habitualmente se construyen prototipos de la interfaz gráfica de usuario que se contrastan con el usuario final.

#### 2.2.5.3. *Análisis y Diseño*

En este flujo de trabajo se traducir los requisitos a una especificación que describe cómo implementar el sistema. Los objetivos del análisis y diseño son:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento.

El análisis consiste en obtener una visión del sistema a través de los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes. Además si el sistema usará una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas.

De igual manera se obtiene la documentación de la arquitectura de software, que captura varias vistas arquitectónicas del sistema.

#### 2.2.5.4. *Implementación*

En este flujo de trabajo se implementan las clases y objetos en archivos fuente, binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

En cada iteración habrá que hacer lo siguiente:

- Planificar qué subsistemas deben ser implementados y en que orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en que orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se prueban los subsistemas individualmente.
- Se integra el sistema siguiendo el plan.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

#### 2.2.5.5. Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Esta disciplina brinda soporte a las otras disciplinas. Sus objetivos son:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Las actividades de este flujo comienzan pronto en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de la prueba en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

El desarrollo del flujo de trabajo consistirá en planificar que es lo que hay que probar, diseñar cómo se va a hacer, implementar lo necesario para llevarlos a cabo, ejecutarlos en los niveles necesarios y obtener los resultados, de forma que la información obtenida nos sirva para ir refinando el producto a desarrollar.

#### 2.2.5.6. Entrega

El objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.

- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Este flujo de trabajo se desarrolla con mayor intensidad en la fase de transición, ya que el propósito del flujo es asegurar una aceptación y adaptación sin complicaciones del software por parte de los usuarios. Su ejecución inicia en fases anteriores, para preparar el camino, sobre todo con actividades de planificación, en la elaboración del manual de usuario y tutoriales.

#### 2.2.5.7. *Administración de proyecto*

La *Administración del proyecto* es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

Los objetivos de este flujo de trabajo son:

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

#### 2.2.5.8. *Configuración y control de cambios*

La finalidad de este flujo de trabajo es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

#### 2.2.5.9. *Entorno*

La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir. En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

El principal artefacto que se usa en este flujo de trabajo es el caso de desarrollo que especifica para el proyecto actual en concreto, como se aplicará el proceso, que productos se van a utilizar y como van a ser utilizados. Además se tendrán que definir las guías para los distintos aspectos del proceso, como pueden ser el modelado del negocio y los Casos de Uso, para la interfaz de usuario, el diseño, la programación, el manual de usuario.

## 2.2.6. Fases

Las fases proveen a RUP una manera de organizar y distribuir las actividades que a través de iteraciones proveen de convergencias y progreso de una solución. Las fases son:

### 2.2.6.1. *Inicio*

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de proyecto para determinar que recursos deben ser asignados al proyecto.

Los objetivos de esta fase son:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar:

- Todos los interesados en el proyecto coinciden en la definición del ámbito del sistema y las estimaciones de agenda.
- Entendimiento de los requisitos, como evidencia de la fidelidad de los Casos de Uso principales.
- Las estimaciones de tiempo, coste y riesgo son creíbles.
- Comprensión total de cualquier prototipo de la arquitectura desarrollado.
- Los gastos hasta el momento se asemejan a los planeados.



Si el proyecto no pasa estos criterios hay que plantearse abandonarlo o repensarlo profundamente.

#### 2.2.6.2. *Elaboración*

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio.

Los objetivos de esta fase son:

- Presentar un modelo de Casos de Uso completo al menos hasta el 90%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- Recopilar requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.
- Describir de la arquitectura software.
- Desarrollar un prototipo de la Interfaz de usuario.
- Actualizar Plan de Proyecto (de ser necesario).
- Un manual de usuario preliminar (opcional).

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes. En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.
- La arquitectura es estable.

- El Plan de proyecto debe estar claro y definido para la fase de construcción.
- Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.
- Los gastos hasta ahora son aceptables, comparados con los previstos.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

### 2.2.6.3. *Construcción*

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos concretos incluyen:

- Actualizar diagramas de Caso de Uso (de ser necesario).
- Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.
- Presentar la versión preliminar del Manual de Usuario (con suficiente detalle).
- Realizar pruebas del producto.

Los criterios de evaluación de esta fase son los siguientes:

- El producto es estable y maduro como para ser entregado a la comunidad de usuario para ser probado.

- Todos los usuarios expertos están listos para la transición en la comunidad de usuarios.
- Son aceptables los gastos actuales versus los gastos planeados.

#### 2.2.6.4. *Transición*

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

Los objetivos de esta fase son:

- Probar de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios.
- Implantar el proyecto en paralelo con los sistemas que serán sustituidos por el proyecto.
- Entrenar a los usuarios y técnicos de mantenimiento.
- Traspasar el producto a los equipos de marketing, distribución y venta.
- Presentar documentos legales.
- Actualizar el Manual de Usuario.
- Entregar producto final.

Los criterios de evaluación de esta fase son los siguientes:

- El usuario se encuentra satisfecho.
- Son aceptables los gastos actuales versus los gastos planificados.

### 2.2.7. Resumen

La siguiente tabla muestra cada una de las fases que comprenden el desarrollo del presente proyecto especificando las actividades y documentación a realizar en cada una de ellas.

**Tabla 5 Resumen General de la Metodología RUP**

Fases	Principales Productos de cada Fase
Inicio	<ul style="list-style-type: none"> <li>• Plan de proyecto: Se definen los objetivos, alcance, las actividades, responsabilidades, cronograma del proyecto.</li> <li>• Modelado del negocio: Una visión general del funcionamiento y características principales del negocio.               <ul style="list-style-type: none"> <li>○ Descripción general del negocio: Visión macro de las principales funciones desempeñadas por el Departamento de Ciencias de la Computación y descripción de la situación actual de los procesos que intervendrán en este proyecto (procesos a automatizar).</li> <li>○ Casos de uso del negocio: Modelado de los procesos involucrados en el proyecto a través de casos de uso.</li> </ul> </li> <li>• Requerimientos: Recopilación de requerimientos del sistema               <ul style="list-style-type: none"> <li>○ Modelo de Casos de uso: modelado de los requerimientos expresados mediante casos de uso.</li> </ul> </li> <li>• Un glosario inicial: Terminología usadas en el negocio.</li> <li>• Interfaz de usuario: Establecer un esquema inicial de las interfaces y los lineamientos que definen la estructura de las páginas.</li> </ul>
Elaboración	<ul style="list-style-type: none"> <li>• Modelo de RUP: Se desarrollan los diagramas listados a continuación.               <ul style="list-style-type: none"> <li>○ Realización de Caso de Uso (Diagrama de clases de análisis)</li> <li>○ Diagramas de Secuencia</li> </ul> </li> <li>• Descripción de la arquitectura software: Se determina la arquitectura con la que funcionará el sistema.</li> <li>• Desarrollo de Diagrama de clases: Se realiza la diagramación respectiva con las clases que intervendrán en la programación del sistema.</li> <li>• Modelo Entidad Relación: Se diagrama la Base de Datos.</li> <li>• Interfaz de usuario: Modelo de navegabilidad del Portal.</li> </ul>
Construcción	<ul style="list-style-type: none"> <li>• Codificación: Se programan las clases, funcionalidad, conexión a base de datos, etc.</li> <li>• Prototipo con capacidad operativa (beta): Se presenta el sistema de prueba con funciones operativas.</li> <li>• Modelo de RUP: Se desarrollan los diagramas listados a continuación.               <ul style="list-style-type: none"> <li>○ Diagrama de Componentes</li> </ul> </li> <li>• Desarrollo del Manual de Usuario: De acuerdo al sistema y sus avances paralelamente se desarrolla el respectivo manual.</li> <li>• Casos de Pruebas: Se buscan ambigüedades y fallas en el sistema.</li> </ul>
Transición	<ul style="list-style-type: none"> <li>• Versión final del Manual de Usuario: Se presenta el manual completo sobre el sistema.</li> <li>• Producto final: Se presenta el sistema completo y funcional</li> <li>• Lanzamiento y entrega del producto: Se realiza la entrega y la implementación en el servidor correspondiente</li> </ul>

- Las iteraciones de la fase de construcción están dirigidas normalmente a conseguir una nueva versión tanto del producto final, del manual y de las pruebas de verificación en cada versión.

### **2.3. La Situación Actual**

El Departamento de Ciencias de la Computación actualmente cuenta con un sitio en Internet que muestra muy poca información y que además sólo presta un servicio de correo electrónico. En lo referente al hardware, el Departamento apenas posee un computador clon que funciona como Servidor Web y Servidor de Correo Electrónico, lo cual no es recomendable.

Claramente se pueden apreciar las deficiencias del sitio, siendo éstas las razones por las que se tiene poca información sobre el Departamento hacia los usuarios y posibles aspirantes que deseen conocer sobre la carrera en la ESPE; lo cual la sitúa en desventaja competitiva con respecto a carreras afines en otras universidades.

La necesidad de contar con este tipo de servicios para el usuario se hace evidente e indispensable debido a la tendencia de las organizaciones de publicar portales con el fin de brindar mejores prestaciones y accesibilidad a sus clientes.

Actualmente la ESPE se encuentra en la transición a la nueva estructura organizacional por Departamentos y Carreras. Según esta estructura el Departamento de Ciencias de la Computación (DCC) tiene bajo su responsabilidad la carrera de Ingeniería en Sistemas e Informática, bajo la modalidad presencial, y la Tecnología en Computación, en la modalidad a distancia

## Capítulo III: Desarrollo del Portal

### 3.1. Inicio

#### 3.1.1. Ámbito del Proyecto

El proyecto abarca la ejecución de todas las fases descritas en la metodología RUP. Una vez desarrollado el sistema, éste será instalado en el servidor respectivo bajo un ambiente Linux junto con los componentes necesarios para su funcionamiento, como el motor de base de datos y el script inicial (datos básicos), el servidor de aplicaciones y el ambiente de ejecución de aplicaciones Java.

#### 3.1.2. Modelado del Negocio

Las principales funciones desempeñadas por el Departamento de Ciencias de la Computación se detallan en la siguiente tabla:

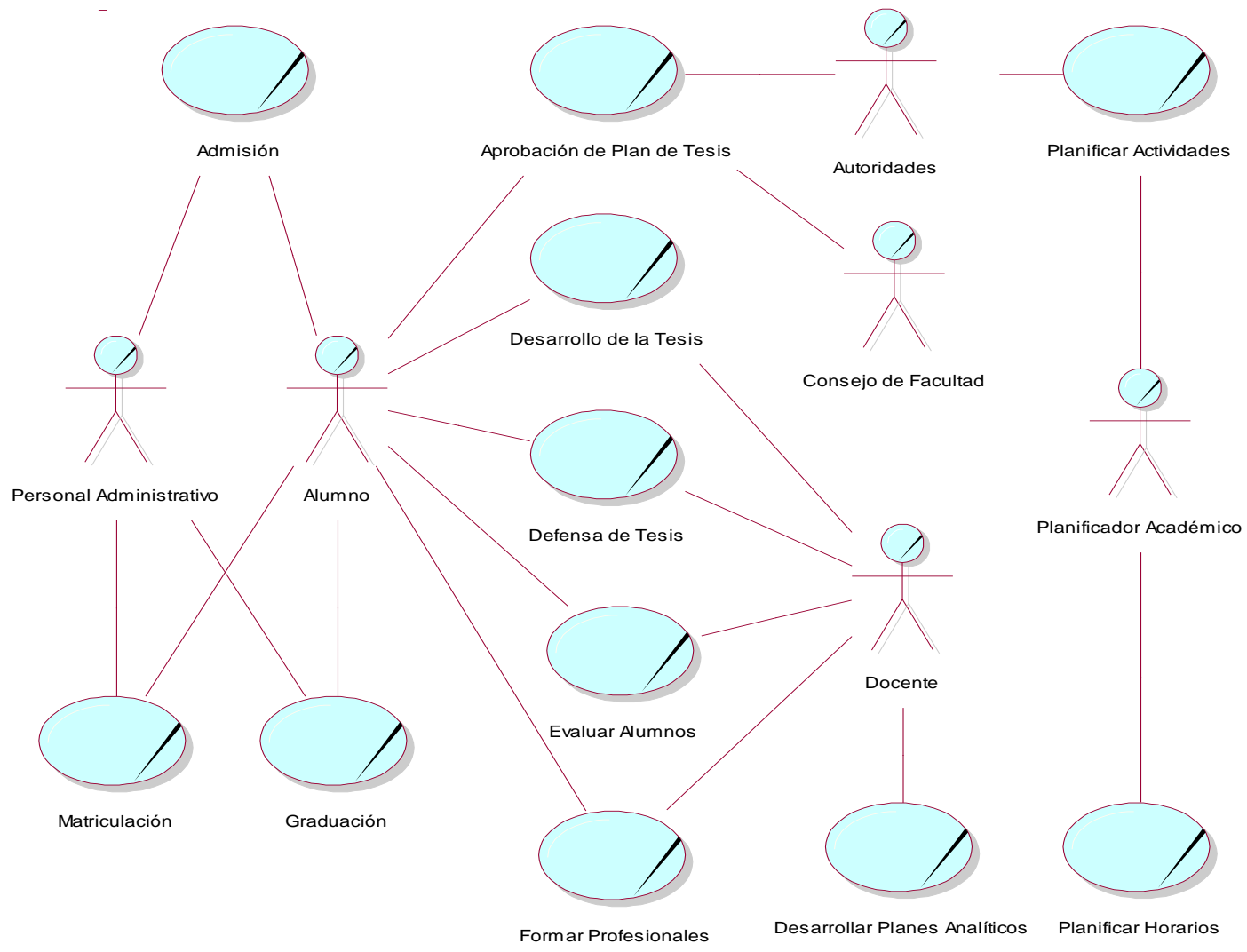
**Tabla 6 Procesos del Departamento**

Proceso	Responsables	Descripción
Admisión	Personal Administrativo Alumno	Consiste en la revisión de los requisitos de un estudiante para poder formar parte del alumnado de las Carreras.
Matriculación	Personal Administrativo Alumno	Resume los procesos donde el alumno se registra en ciertas materias, se validan, se separa un cupo, y emite comprobantes para que se cancelen los valores respectivos en el banco.
Planificar Actividades	Autoridades Planificador Académico	Se planifican todas las actividades curriculares y extracurriculares que se presentarán en el periodo académico respectivo.
Planificar Horarios	Planificador Académico	Se analiza y se definen los horarios de clases disponibles para cada paralelo con la asignación de su respectivo docente.

Desarrollar Planes Analíticos	Docentes Alumno	Se desarrolla el p�nsum de estudio para cada materia.
Formar Profesionales	Docentes Alumno	Resume actividades y proceso de ense�anza – aprendizaje.
Evaluar Alumnos	Docentes Alumno	Consiste en la forma de evaluar y presentar los resultados en las fechas fijadas para definir si un estudiante cumple con la aprobaci�n de una materia
Aprobaci�n de Plan de Tesis	Autoridades Consejo de Facultad	Se define el proceso en el que uno y dos estudiantes proponen un tema de tesis el cual debe ser revisado por el personal respectivo para su aprobaci�n o no.
Desarrollo de la Tesis	Docentes Alumno	Son las actividades como reuniones, revisiones y presentaciones de avances sobre un tema de tesis aprobado.
Defensa de Tesis	Docentes Alumno	Consiste en el proceso en el que se defiende el plan en presencia del personal respectivo para definir si los alumnos pueden continuar con su graduaci�n.
Graduaci�n	Personal Administrativo Alumno	Es el proceso en el cual se legaliza los t�tulos de aquellos estudiantes que han cumplido con todos los requisitos necesarios.

El siguiente Diagrama de Casos de Uso representa el Modelo del Negocio:





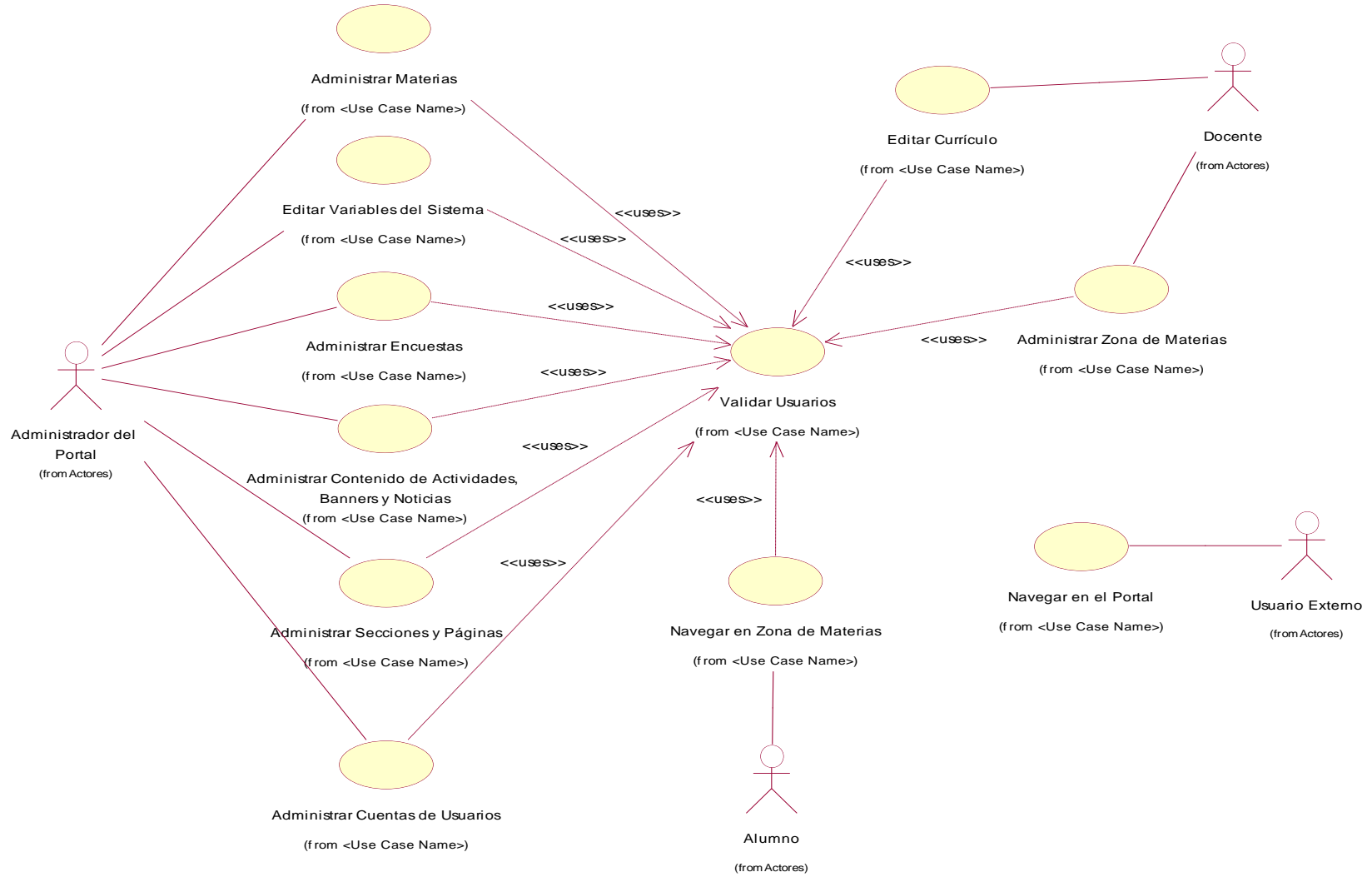
**Figura 21 Modelo del Negocio**

### 3.1.3. Requerimientos

Los requerimientos del proyecto son la base sobre la cual se construirá la aplicación, ya que éstos representan la funcionalidad.

Para el Desarrollo del Portal del Departamento de Ciencias de la Computación se han recopilado los requerimientos de acuerdo a las necesidades planteadas por el personal del mismo, con la ayuda de la utilización del Estándar IEEE 830, cuyo documento completo se encuentra en el Anexo A (Especificación de Requisitos de Software para el desarrollo el Portal del Departamento de Ciencias de la Computación) que fue presentado, revisado y firmado por las autoridades involucradas.

El siguiente Diagrama de Casos de Uso representa el resumen de los requerimientos funcionales definidos en el Anexo A.



**Figura 22 Modelo de Casos de Uso (Requerimientos Funcionales)**

A partir del modelo de casos de uso tenemos:

**Tabla 7 Descripción de Caso de Uso Administrar Materias**

Nombre	Administrar Materias
Actor	Administrador del Portal
Propósito	Permite administrar las Materias existentes en la malla curricular.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	Si una materia es eliminada, se eliminarán sus relaciones con docentes, estudiantes y materiales.

**Tabla 8 Descripción de Caso de Uso Editar Variables del Sistema**

Nombre	Editar Variables del Sistema
Actor	Administrador del Portal
Propósito	Edita parámetros generales del sistema como Nombre de la Organización, Unidad Administrativa, Período Académico y Datos del Webmaster.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	

**Tabla 9 Descripción de Caso de Uso Administrar Encuestas**

Nombre	Administrar Encuestas
Actor	Administrador del Portal
Propósito	Permite administrar encuestas que serán publicadas en el Portal con sus respectivas opciones.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	Si una encuesta es eliminada se borrarán también sus opciones en la tabla respectiva,

**Tabla 10 Descripción de Caso de Uso Administrar Materias**

Nombre	Administrar Contenido de Actividades, Banners y Noticias
Actor	Administrador del Portal
Propósito	Permite administrar las actividades del presente período académico, los Banners Publicitarios y los enlaces a Noticias de interés y actualidad.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	

**Tabla 11 Descripción de Caso de Uso Administrar Secciones y Páginas**

<b>Nombre</b>	<b>Administrar Secciones y Páginas</b>
Actor	Administrador del Portal
Propósito	Permite administrar las páginas y su respectiva información, además de agruparlas por secciones.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	Si una sección es eliminada, las páginas que pertenecen a ésta se eliminarán también. Si una página que contiene otras es eliminada, éstas se eliminarán.

**Tabla 12 Descripción de Caso de Uso Administrar Cuentas de Usuario**

<b>Nombre</b>	<b>Administrar Cuentas de Usuario</b>
Actor	Administrador del Portal
Propósito	Permite administrar la información de los usuarios, sus permisos de administración y su perfil entre los cuales se encuentran Alumno, Docente y Administrativo.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	Si un docente es eliminado se eliminarán los materiales que haya cargado.

**Tabla 13 Descripción de Caso de Uso Editar Currículo**

<b>Nombre</b>	<b>Editar Currículo</b>
Actor	Docente
Propósito	Permite editar los campos de estudios realizados, experiencia profesional y cargar su respectiva foto.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	

**Tabla 14 Descripción de Caso de Uso Administrar Zona de Materias**

<b>Nombre</b>	<b>Administrar Zona de Materias</b>
Actor	Docente
Propósito	Permite al Docente organizar un contenedor de información que desee compartir con los alumnos divididos por las materias que imparte.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	Si un material es eliminado, los archivos cargados se eliminarán. Un alumno no tendrá acceso a ese archivo.

**Tabla 15 Descripción de Caso de Uso Validar Usuarios**

Nombre	Validar Usuarios
Actor	Sistema
Propósito	Es el control del ingreso de los usuarios.
Precondición	
Postcondición	Si el acceso del usuario es aceptado, se cargan sus permisos para determinar a que lugares tiene acceso. Si es rechazado, se muestra un mensaje mostrando una advertencia.

**Tabla 16 Descripción de Caso de Uso Navegar en Zona de Materias**

Nombre	Navegar en Zona de Materias
Actor	Alumno
Propósito	Permite al Alumno revisar el contenido que ha sido publicado por un Docente en su respectiva Materia.
Precondición	El usuario externo debe iniciar su sesión. El usuario es validado para determinar si tiene los permisos necesarios.
Postcondición	

**Tabla 17 Descripción de Caso de Uso Navegar en el Portal**

Nombre	Navegar en el Portal
Actor	Usuario Externo
Propósito	Consiste en el control de navegación por todas las páginas que conforman el Portal.
Precondición	
Postcondición	

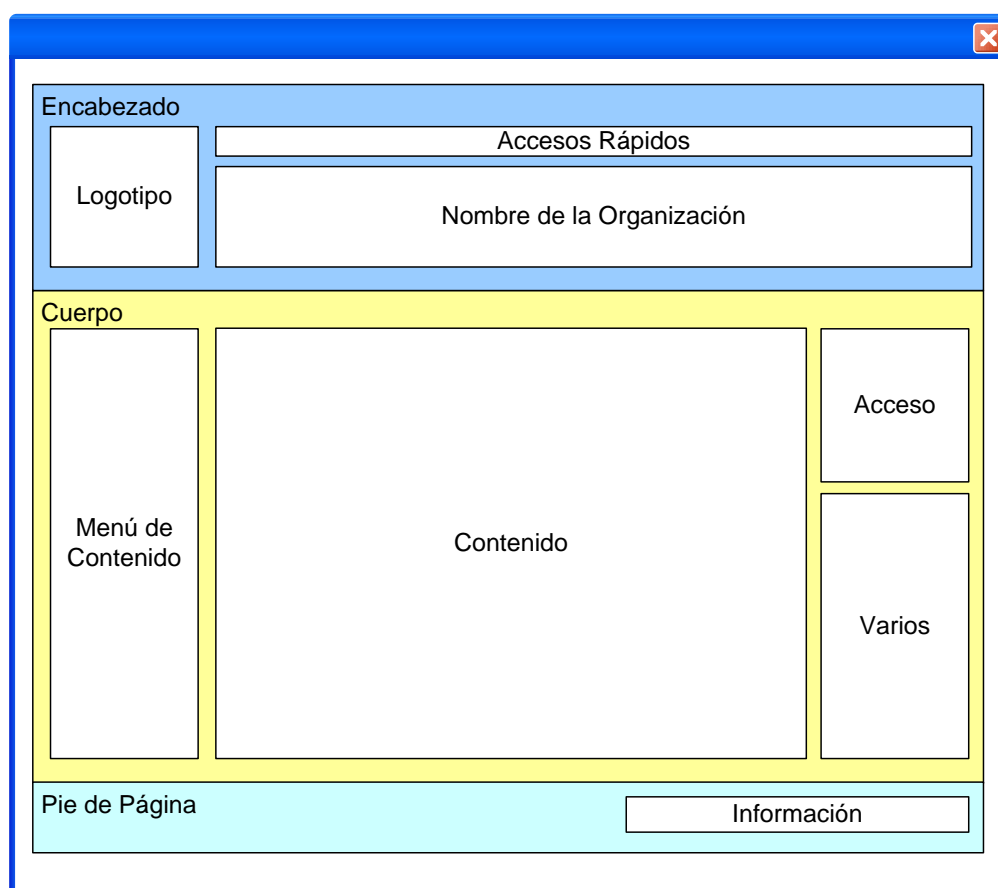
### 3.1.4. Glosario de Términos

**Tabla 18 Glosario de Términos**

Término	Significado
DCC	Departamento de Ciencias de la Computación
Administrar	Este término implica las acciones de agregar, modificar, buscar y eliminar información dentro del Sistema.
Zona de Materias	Es el espacio administrado por el Docente que permite al Alumno acceder a los contenidos de una materia en particular.
Usuario Externo	Es un usuario que no ha sido iniciado una sesión en el Sistema.

### 3.1.5. Interfaz de Usuario

El siguiente gráfico representa el esquema de la interfaz que será utilizada como plantilla en las páginas del Portal.



**Figura 23 Esquema de interfaz**

Este esquema se encuentra dividido básicamente en tres secciones, las cuales se describen a continuación:

- **Encabezado**
  - Logotipo: imagen o icono representativo de la organización.
  - Accesos rápido: vínculos usados para facilitar la navegación dentro del Portal.

- Nombre de la Organización: comprende el nombre del Departamento y de la Universidad.
- **Cuerpo**
  - Menú de Contenido: contiene enlaces o vínculos que permiten explorar el contenido ofrecido por el Portal, los mismos que se encuentran agrupados por secciones.
  - Contenido: muestra la información que el usuario desea visualizar.
  - Acceso: permite ingresar a la zona privada del Portal a los usuarios registrados.
  - Varios: espacio que contendrá encuestas y banners publicitarios.
- **Pie de Página**
  - Información: despliega datos del Departamento como dirección, teléfonos y derechos de autor.

## 3.2. Elaboración

### 3.2.1. Modelo de Análisis

La realización de casos de uso del Modelo de Negocio se detalla a continuación:

#### 3.2.1.1. Administrar Materias



**Figura 24 Administrar Materias**



### 3.2.1.2. Editar Variables del Sistema

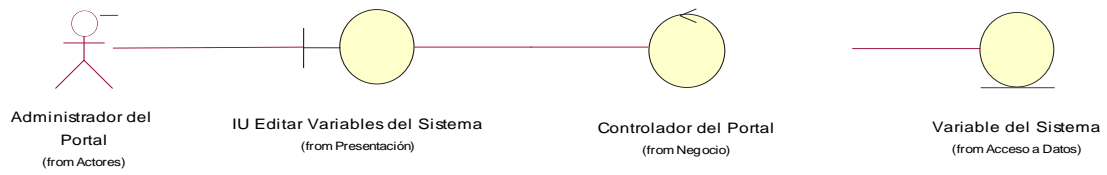


Figura 25 Editar Variables del Sistema

### 3.2.1.3. Administrar Encuestas

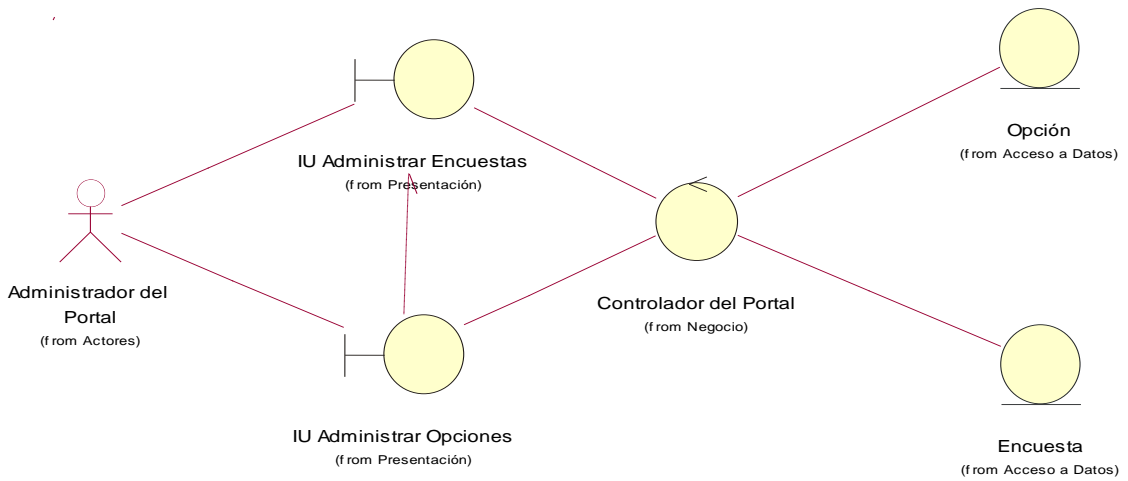
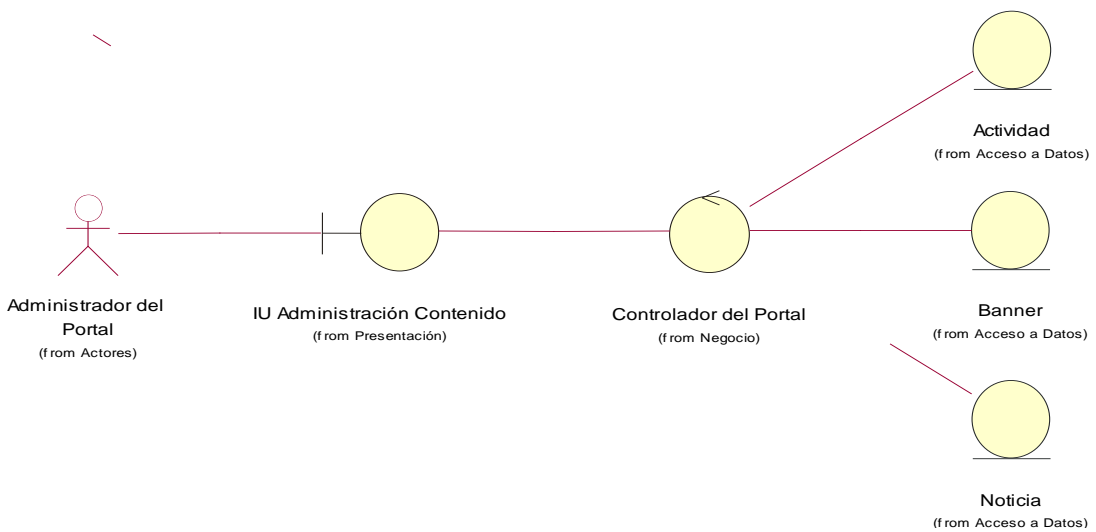


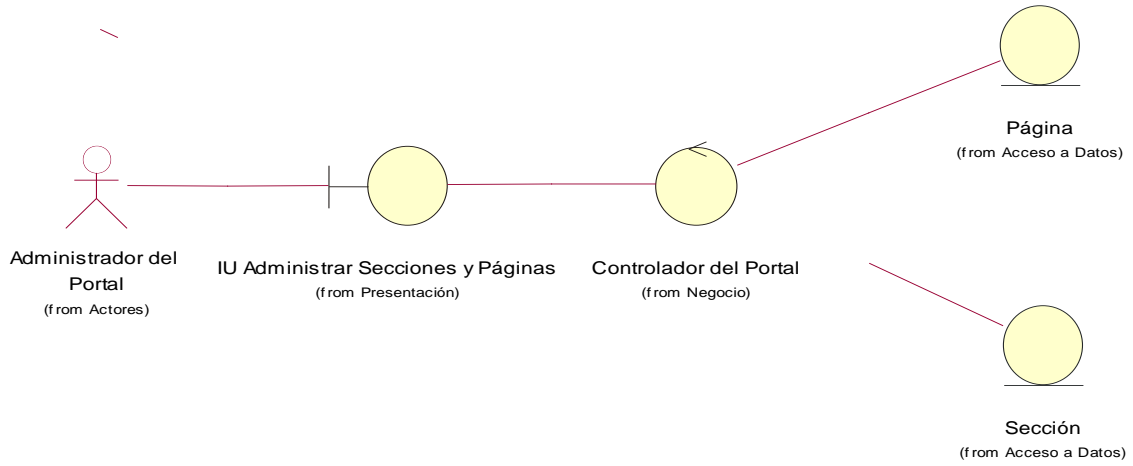
Figura 26 Administrar Encuestas

### 3.2.1.4. Administrar Contenido de Actividades, Banners y Noticias



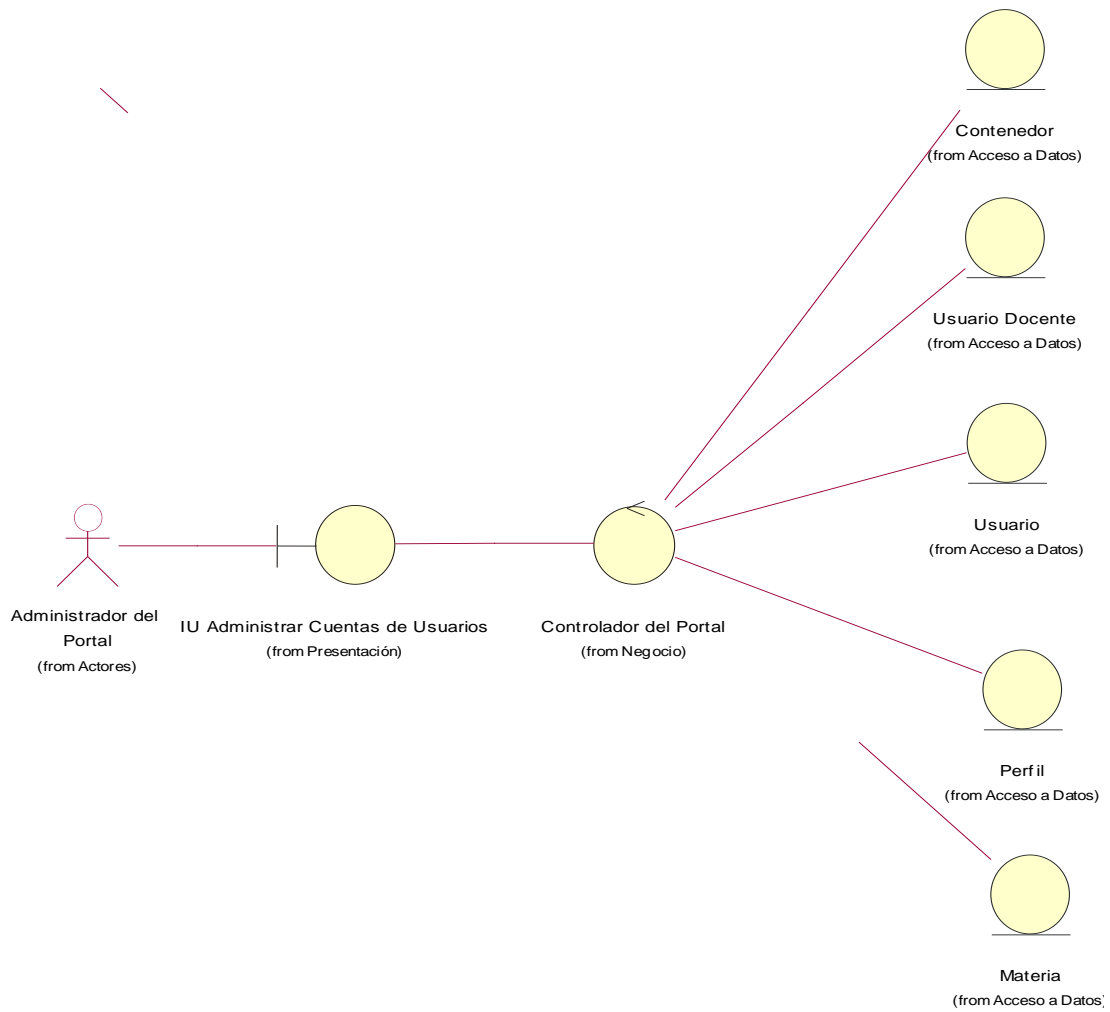
**Figura 27 Administrar Contenido de Actividades, Banners y Noticias**

**3.2.1.5. Administrar Secciones y Páginas**



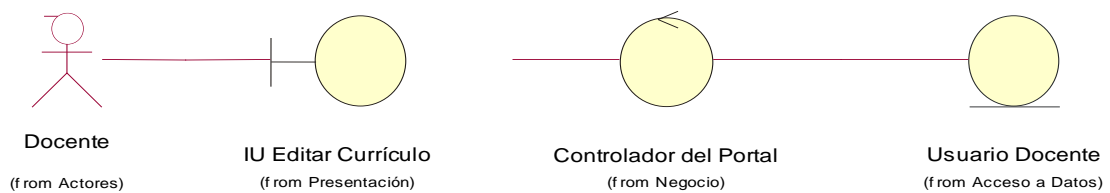
**Figura 28 Administrar Secciones y Páginas**

### 3.2.1.6. Administrar Cuentas de Usuario



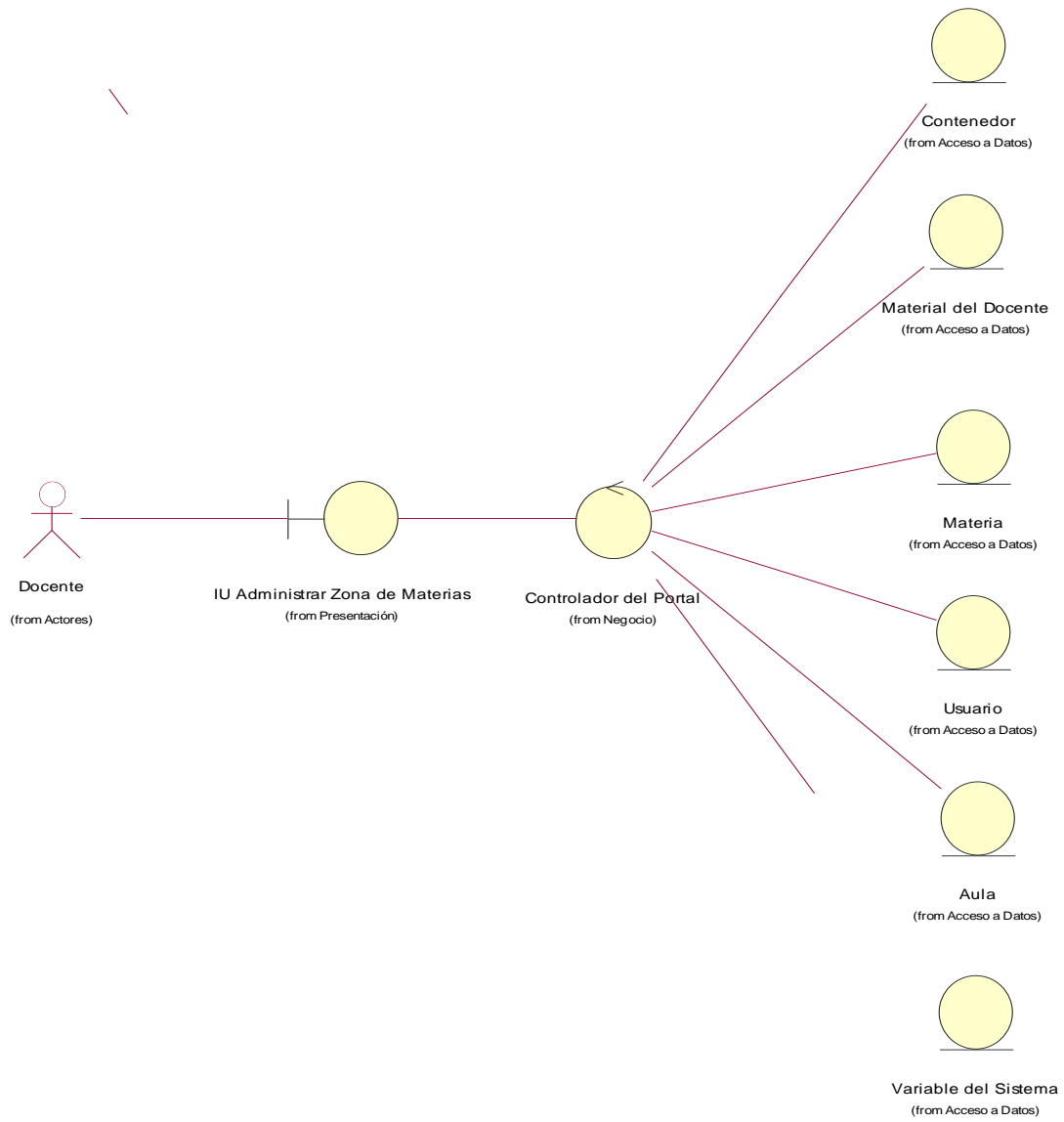
**Figura 29 Administrar Cuentas de Usuario**

### 3.2.1.7. Editar Currículo



**Figura 30 Editar Currículo**

### 3.2.1.8. Administrar Zona de Materias



**Figura 31 Administrar Zona de Materias**

### 3.2.1.9. Validar Usuarios

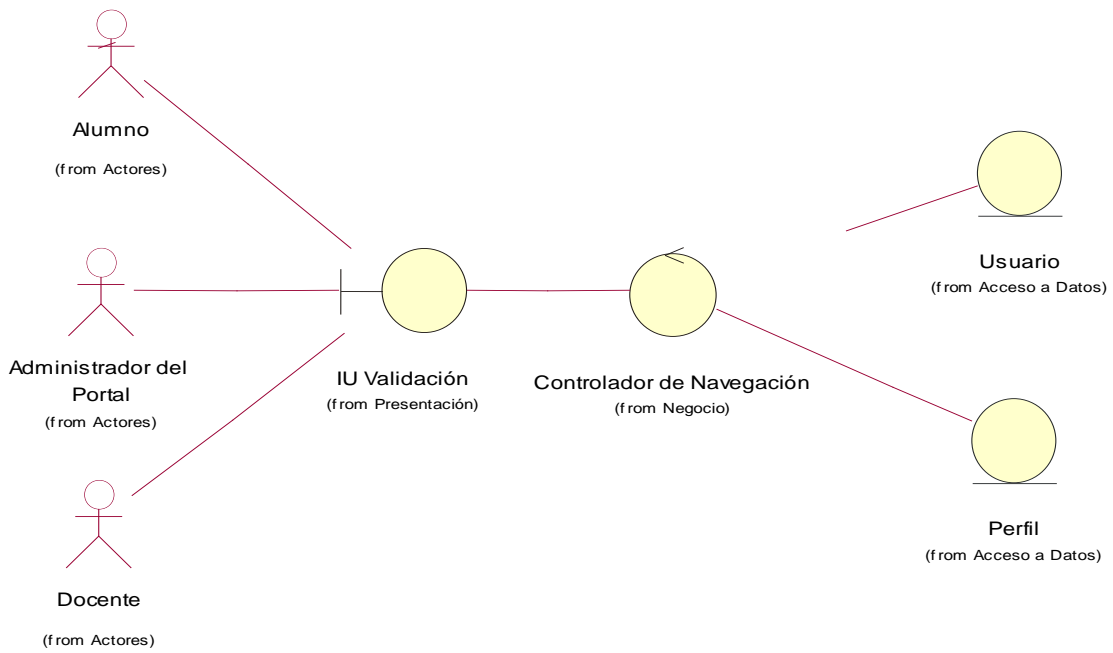


Figura 32 Validar Usuarios

### 3.2.1.10. Navegar en Zona de Materias

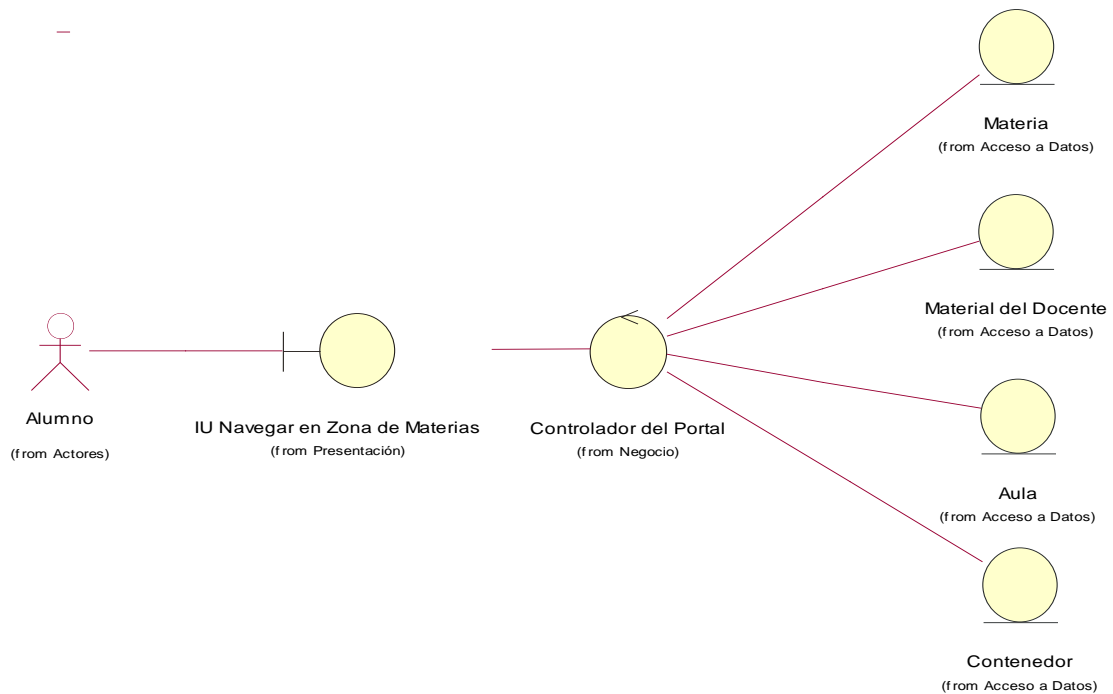


Figura 33 Navegar en Zona de Materias

### 3.2.1.11. Navegar en el Portal

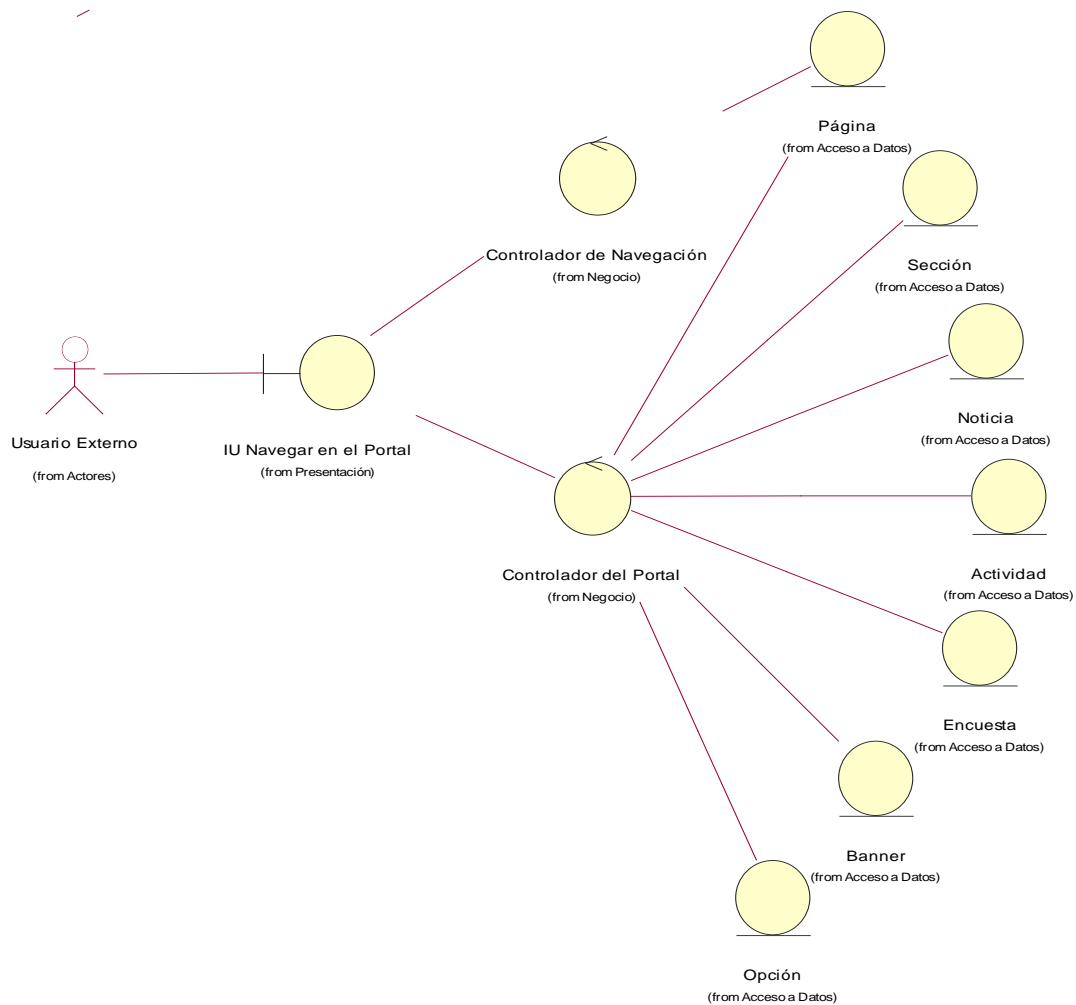
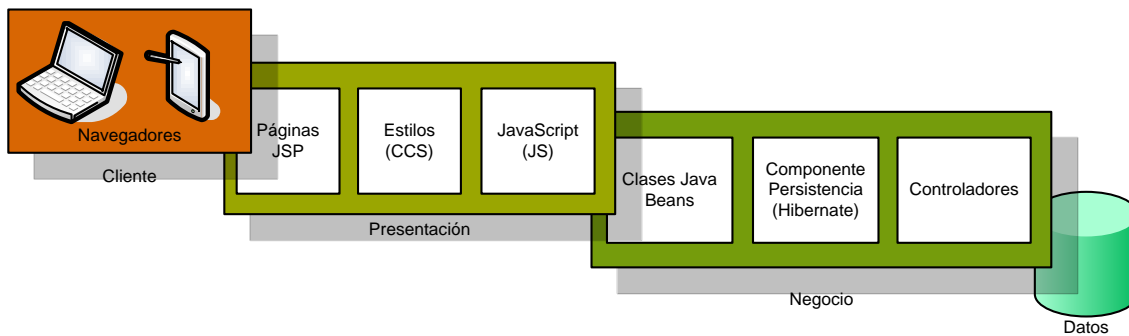


Figura 34 Navegar en Zona de Materias

### 3.2.2. Arquitectura del Software

A partir de la realización de los casos de uso, se obtuvieron los diagramas de clases de análisis. Estos nos permiten estructurar las diferentes clases en varias capas y definir las clases del modelo de diseño.

De acuerdo al análisis realizado y las tecnologías a utilizar el diagrama siguiente define la arquitectura del software.



**Figura 35** Arquitectura del Software

### 3.2.2.1. *Cliente*

En esta se encuentran los Navegadores Web como Internet Explorer, Netscape o Firefox, que son los encargados de mostrar la información al usuario.

### 3.2.2.2. *Presentación*

La conforman las páginas JSP que contienen la información, la cual es presentada con formatos almacenados en los CSS (Hojas de Estilo en Cascada). Además también se complementa con JavaScript para realizar operaciones en el cliente.

### 3.2.2.3. *Negocio*

Contiene las clases desarrolladas en Java como los Beans y los Controladores que controlan la lógica del sistema. A continuación se presentan las clases de negocio:

ControladorNavegacion
<ul style="list-style-type: none"> <li>◆comprobarIdPagina(IdPagina : Long) : Pagina</li> <li>◆validarUsuario(login : String, pass : String, sesion : HttpSession) : Boolean</li> <li>◆crearSesion(sesion : HttpSession, objUsr : Usuario)</li> <li>◆cerrarSesion(sesion : HttpSession)</li> <li>◆md5(pass : String) : String</li> <li>◆ruta(idPagina : Long, idOrig : Long) : String</li> </ul>

ControladorPortal
<ul style="list-style-type: none"> <li>◆agregarActividad(descripcion : String, fechaInicio : Date, fechaFin : Date)</li> <li>◆agregarAula(idUsuario : Long, idContenedor : Long, paralelo : Char)</li> <li>◆agregarPagina(idSeccion : Long, idPaginaPadre : Long, nombre : String, contenido : String, descripcion : String, posicion : Long, contenedor : Boolean, er)</li> <li>◆agregarSeccion(nombre : String, posicion : Long)</li> <li>◆agregarBanner(ruta : String, req : HttpServletRequest, nombre : String)</li> <li>◆agregarBitacora(usuario : String, tabla : String, accion : String)</li> <li>◆agregarContenedor(idUsuario : Long, materias : String [])</li> <li>◆agregarEncuesta(pregunta : String)</li> <li>◆agregarMateria(nombre : String, creditos : Int)</li> <li>◆agregarMaterial(ruta : String, req : HttpServletRequest, idUsuario : String)</li> <li>◆agregarNoticia(titular : String, enlace : String)</li> <li>◆agregarOpcion(opcion : String, idEncuesta : Long)</li> <li>◆agregarUsuario(idPerfil : Long, nombres : String, apellidos : String, cedula : String, login : String, correo : String, permisos : String, ruta : String) : Long</li> <li>◆agregarUsuarioDocente(idUsuario : Long, tipo : String)</li> <li>◆bloquearUsuario(idUsuario : Long)</li> <li>◆buscarActividad(idActividad : Long) : Actividad</li> <li>◆buscarAlumno(cedula : String) : Usuario</li> <li>◆buscarAula(idUsuario : Long, idContenedor : Long) : Aula</li> <li>◆buscarBanner(idBanner : Long) : Banner</li> <li>◆buscarContenedor(idUsuario : Long, idMateria : Long) : Contenedor</li> <li>◆buscarContenedorAlumno(idUsuario : Long, idMateria : Long) : Contenedor</li> <li>◆buscarDocente(idUsuario : Long) : UsuarioDocente</li> <li>◆buscarEncuesta(idEncuesta : Long) : Encuesta</li> <li>◆buscarMateria(idMateria : Long) : Materia</li> <li>◆buscarMaterial(idMaterial : Long) : MaterialDocente</li> <li>◆buscarNoticia(idNoticia : Long) : Noticia</li> <li>◆buscarOpcion(idOpcion : Long) : Opcion</li> <li>◆buscarPagina(idPagina : Long) : Pagina</li> <li>◆bloquearPagina(idPagina : Long)</li> <li>◆bloquearPaginasHijo(idPagina : Long)</li> <li>◆buscarUsuario(idUsuario : Long) : Usuario</li> <li>◆buscarUsuario(login : String) : Usuario</li> <li>◆buscarUsuarios(búsqueda : String, inicial : Int, cant : Int) : List</li> <li>◆buscarUsuarios(búsqueda : String) : List</li> <li>◆buscarVariableSistema(nombre : String) : VariableDelSistema</li> <li>◆listarActividades(mes : Int, anio : Int) : List</li> <li>◆listarActividades() : List</li> <li>◆listarAlumnos(idContenedor : Long, paralelo : Char) : List</li> <li>◆listarArbol(idPagina : Long, nivel : Int) : String</li> <li>◆listarContenidos() : Collection</li> <li>◆listarPaginas(idSeccion : Long) : List</li> <li>◆listarPaginas(búsqueda : String) : List</li> <li>◆listarPaginas() : List</li> <li>◆listarPaginasHijo(idPagina : Long, nivel : Int) : String</li> <li>◆listarPaginasHijo(idPagina : Long) : List</li> <li>◆listarPaginasTodas(idSeccion : Long) : List</li> <li>◆buscarPerfil(idPerfil : Long) : Perfil</li> <li>◆buscarSeccion(idSeccion : Long) : Seccion</li> <li>◆modificarBanner(idBanner : Long, nombre : String, enlace : String)</li> </ul>

**Figura 36 Clases Controladores**



```

*eliminarBanner(idBanner : Long, ruta : String) : List
*listarBanners() : List
*listarBitacora(fechaInicial : String, fechaFinal : String, busqueda : String) : List
*listarBitacora(inicial : Int, cant : Int) : List
*listarBitacora() : List
*listarDocentes(opcion : Int) : List
*listarEncuestas() : List
*listarMateriales(idContenedor : Long, paralelo : Char) : List
*listarMateriales(idContenedor : Long) : List
*listarMaterias(idUsuario : Long, bandera : Boolean) : List
*listarMaterias() : List
*listarMateriasAlumno(idUsuario : Long) : List
*listarNoticias() : List
*listarOpciones(id : Long) : List
*listarPerfiles() : List
*listarSecciones() : List
*listarUsuarios() : List
*listarUsuarios(inicial : Int, cant : Int) : List
*listarUsuarios(idPerfil : Long) : List
*listarVariablesSistema() : List
*eliminarActividad(idActividad : Long)
*eliminarAlumno(idUsuario : Long, idContenedor : Long)
*eliminarPagina(idPagina : Long)
*eliminarSeccion(idSeccion : Long)
*eliminarBanner(idBanner : Long, ruta : String)
*eliminarEncuesta(idEncuesta : Long)
*eliminarMateria(idMateria : Long)
*eliminarMaterial(idMaterial : Long, idUsuario : String, ruta : String)
*eliminarNoticia(idNoticia : Long)
*eliminarOpcion(idOpcion : Long)
*eliminarUsuario(idUsuario : Long, ruta : String)
*modificarActividad(idActividad : Long, descripcion : String, fechaInicio : Date, fechaFin : Date)
*modificarPagina(idPagina : Long, idSeccion : Long, idPaginaPadre : Long, nombre : String, contenido : String, descripcion : String, posicion : Long)
*modificarOpcion(idOpcion : Long, opcion : String, votos : Long)
*modificarNoticia(idNoticia : Long, titular : String, enlace : String)
*modificarMaterial(idMaterial : Long, contenido : String, paralelo : String)
*modificarMateria(idMateria : Long, nombre : String, creditos : Int)
*modificarEncuesta(idEncuesta : Long, pregunta : String)
*modificarDocente(idUsuario : Long, estudios : String, experiencia : String, tipo : String)
*modificarContenedor(idUsuario : Long, materias : String [])
*modificarUsuario(idUsuario : Long, nombres : String, apellidos : String, cedula : String, login : String, correo : String, permisos : String)
*modificarVariableSistema(objListaNueva : Iterator)
*reordenarPaginas(idSeccion : Long, idPaginaPadre : Long, posicion : Long)
*reordenarSecciones(antiguaPos : Long, nuevaPos : Long)
*resetearPasswordUsuario(idUsuario : Long)
*reordenarSecciones(posicion : Long, uno : Int)
*reordenarPaginas(idSeccion : Long, idPaginaPadre : Long, antiguaPos : Long, nuevaPos : Long)
*reordenarPaginas(id : String, posicion : Long)
*modificarSeccion(idSeccion : Long, nombre : String, posicion : Long)
*modificarPassword(idUsuario : Long, password : String)

```

**Figura 37 Clases Controladores (Continuación)**

#### 3.2.2.4. Acceso a datos

Consiste en el motor de base de datos que permite administrar la información de forma persistente de manera que puedan ser consultados cuando se requieran. La base de datos utilizada para el Portal es MySQL.

### 3.2.3. Diagramas de Secuencia

Los diagramas de secuencia a continuación detallan la interacción de las clases de análisis.

#### 3.2.3.1. Administrar Materias

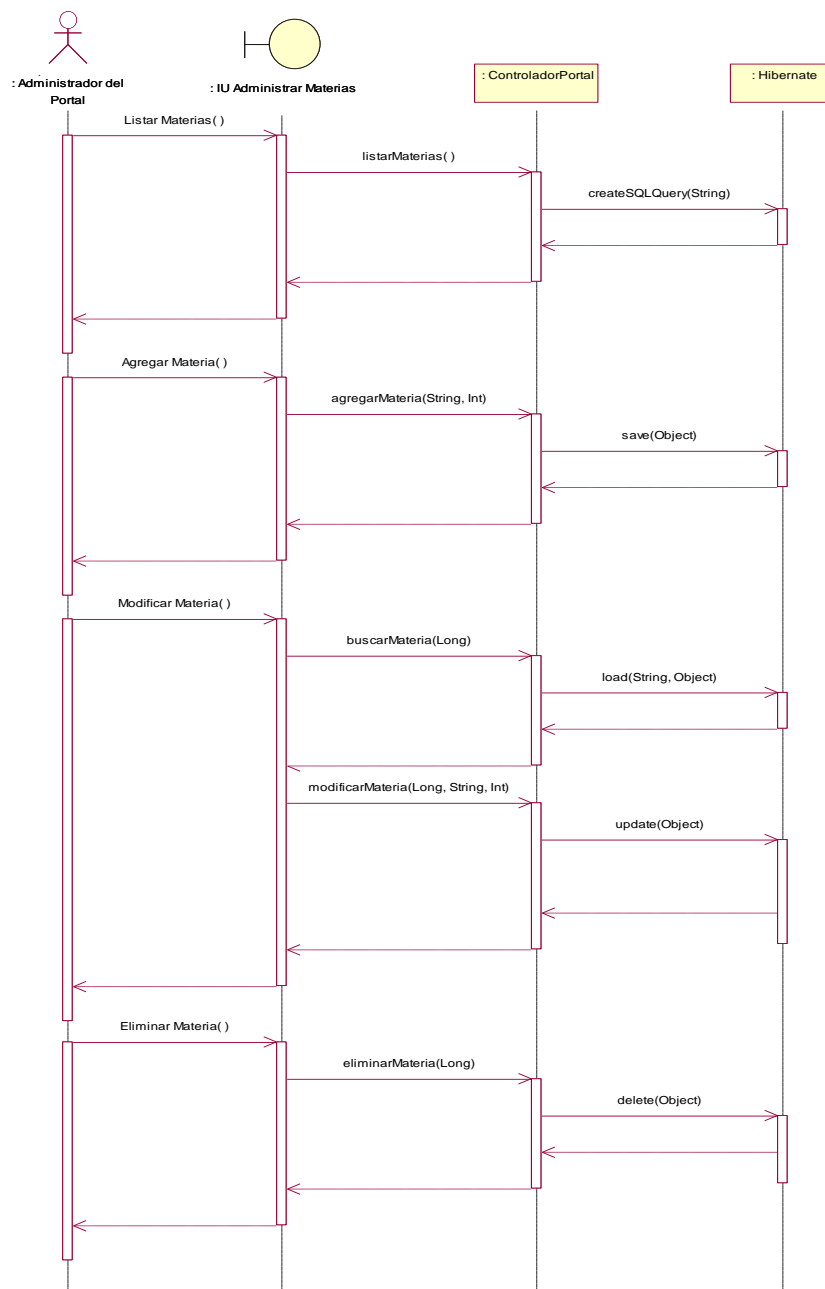


Figura 38 Administrar Materias

### 3.2.3.2. Editar Variables del Sistema

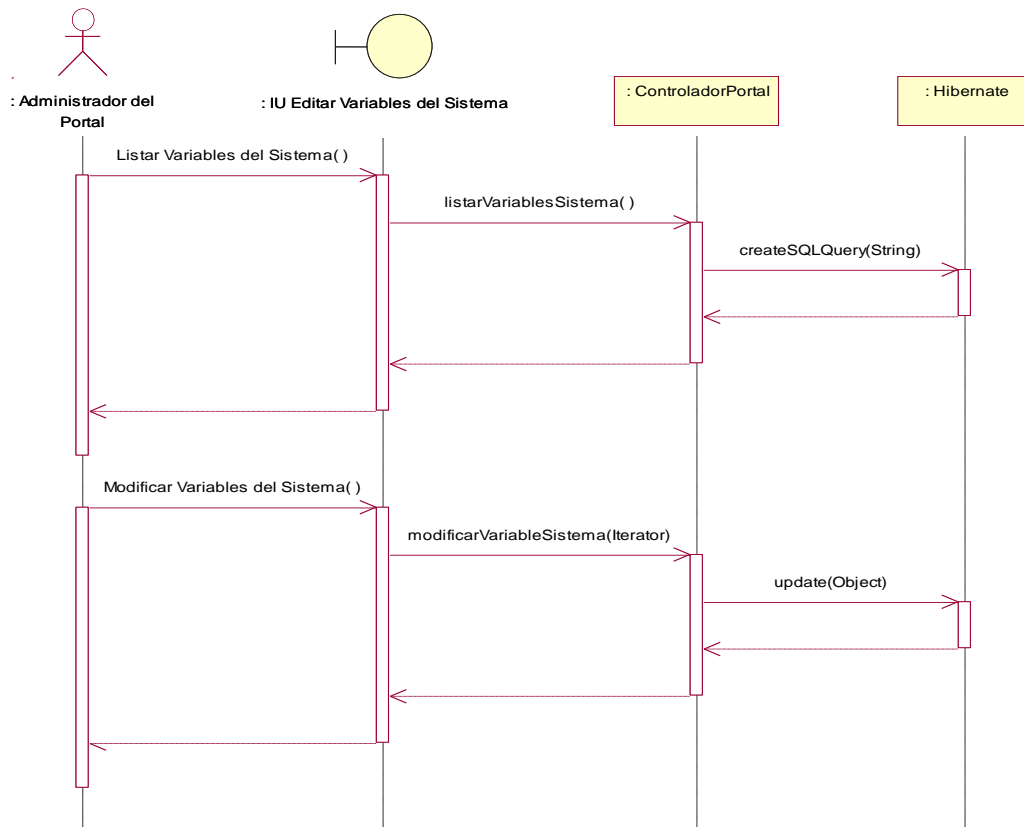


Figura 39 Editar Variables del Sistema

### 3.2.3.3. Administrar Encuestas

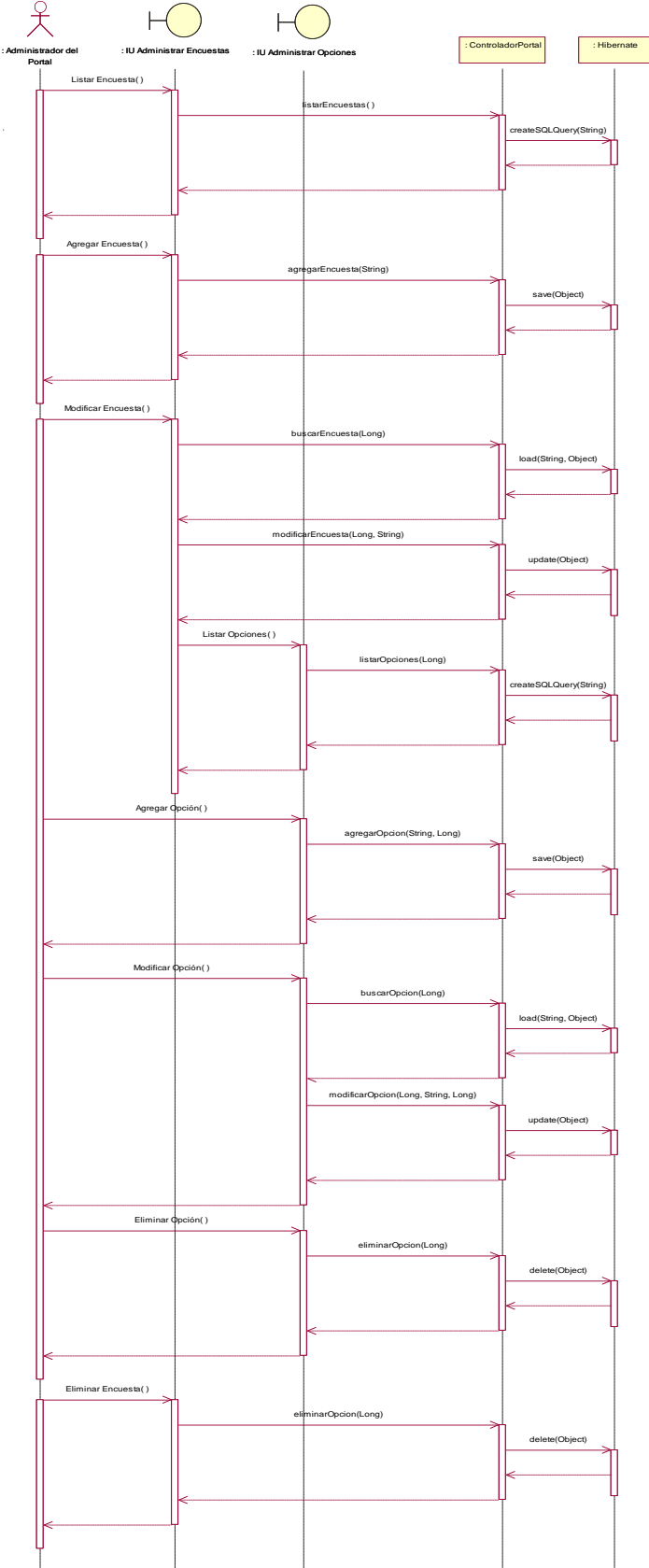


Figura 40 Administrar Encuestas  
89

### 3.2.3.4. Administrar Contenido de Actividades, Banners y Noticias

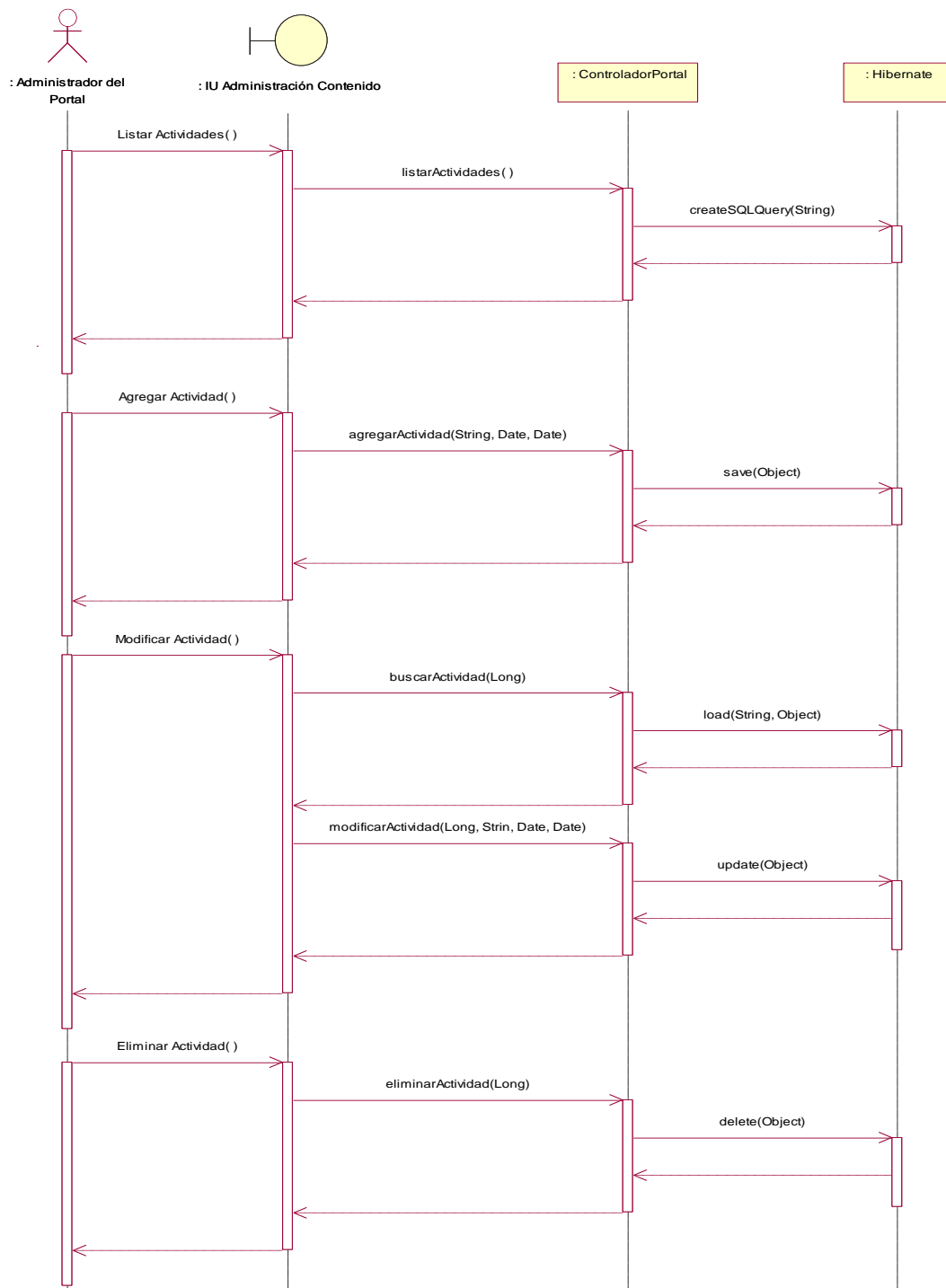
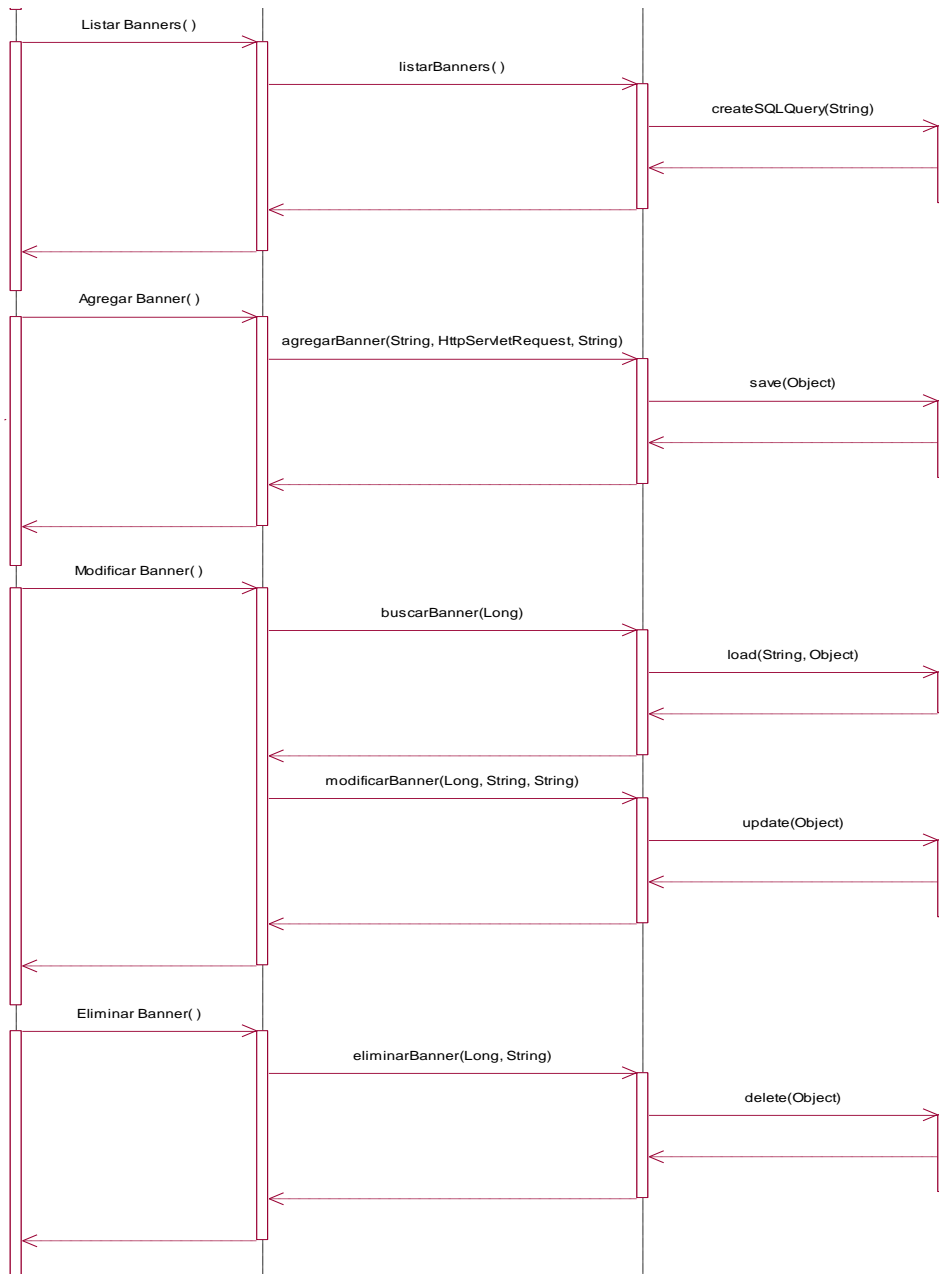
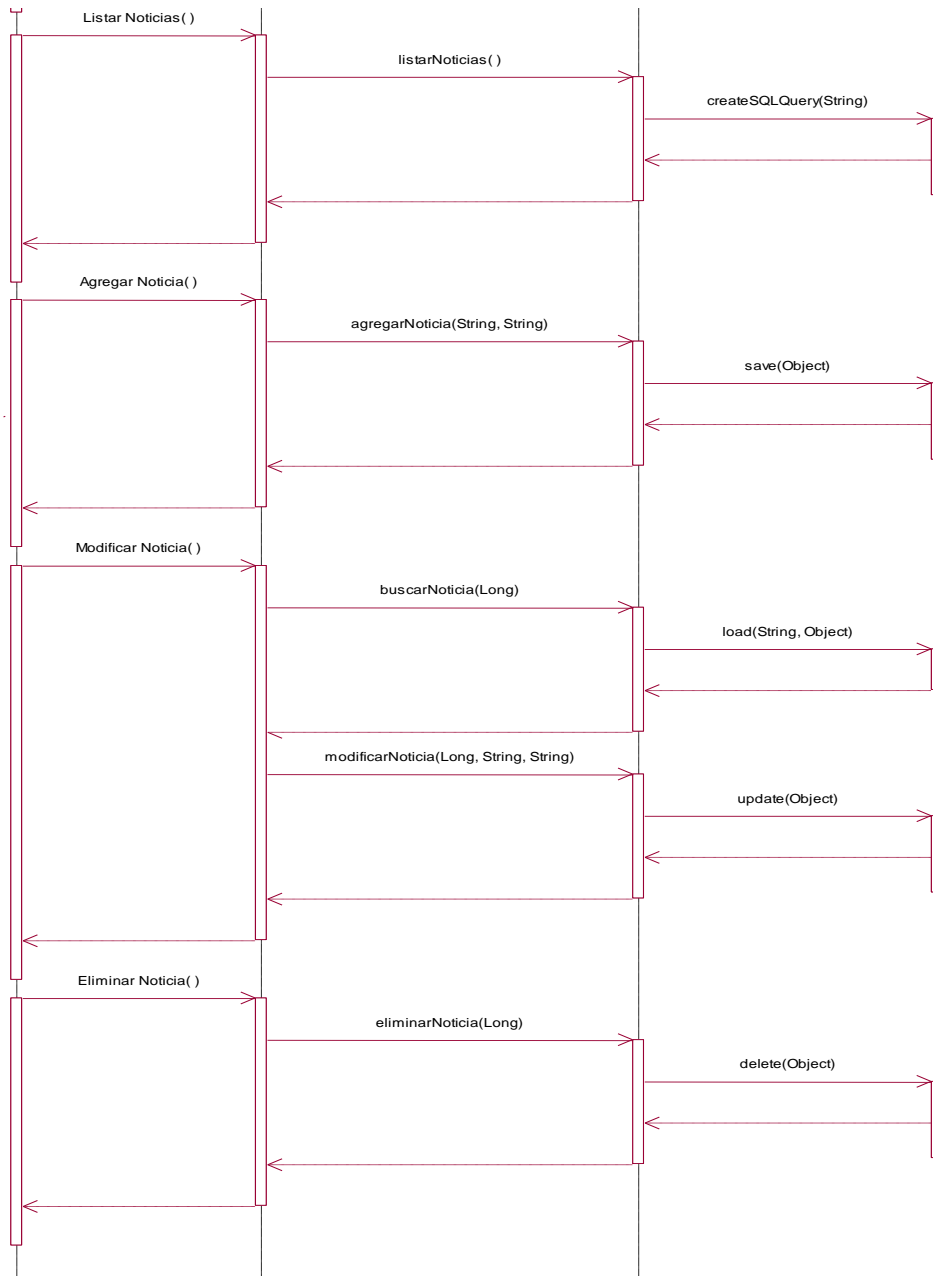


Figura 41 Administrar Contenido (Actividades)



**Figura 42 Administrar Contenido (Banner)**



**Figura 43 Administrar Contenido (Noticias)**

### 3.2.3.5. Administrar Secciones y Páginas

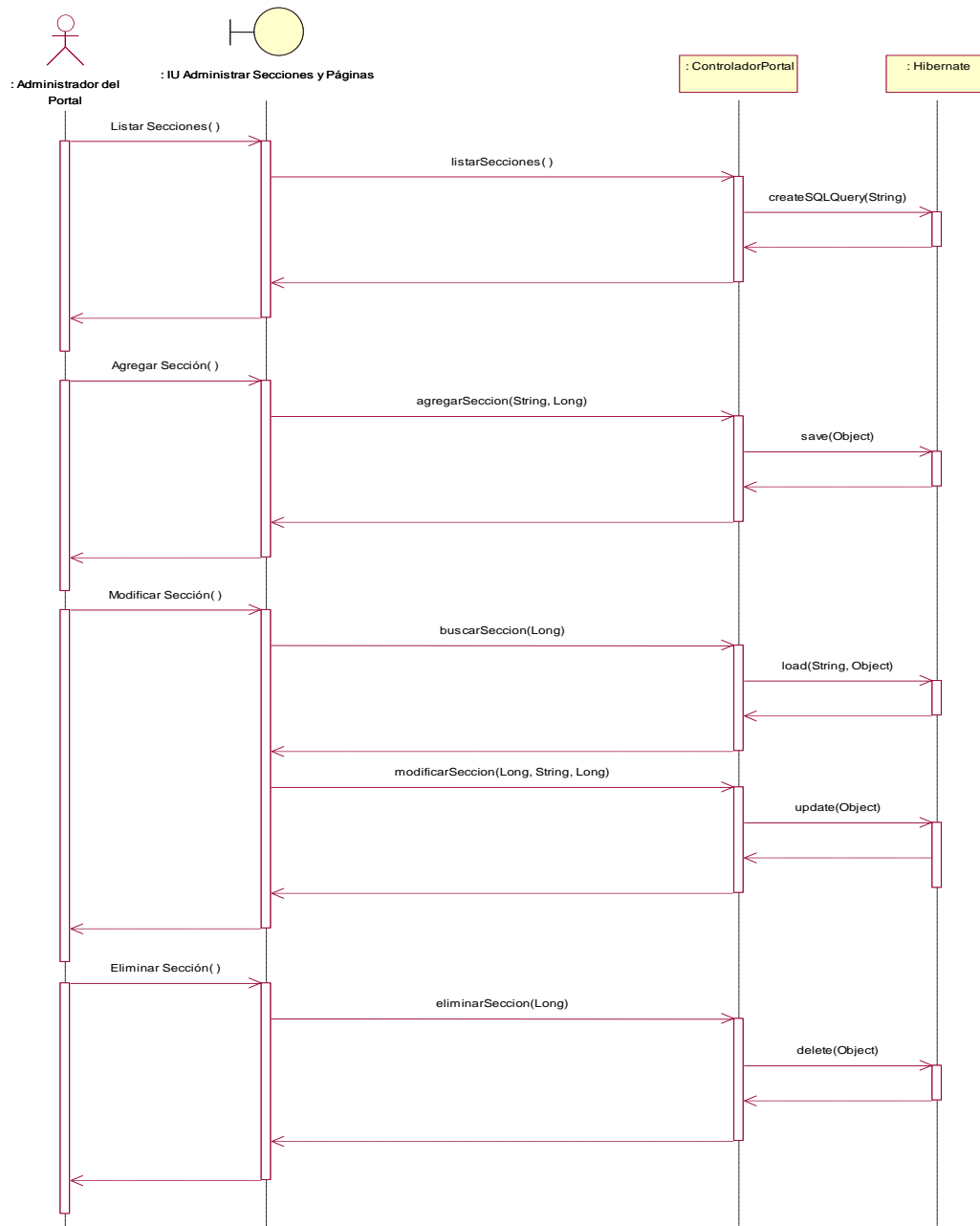


Figura 44 Administrar Secciones y Páginas (Secciones)



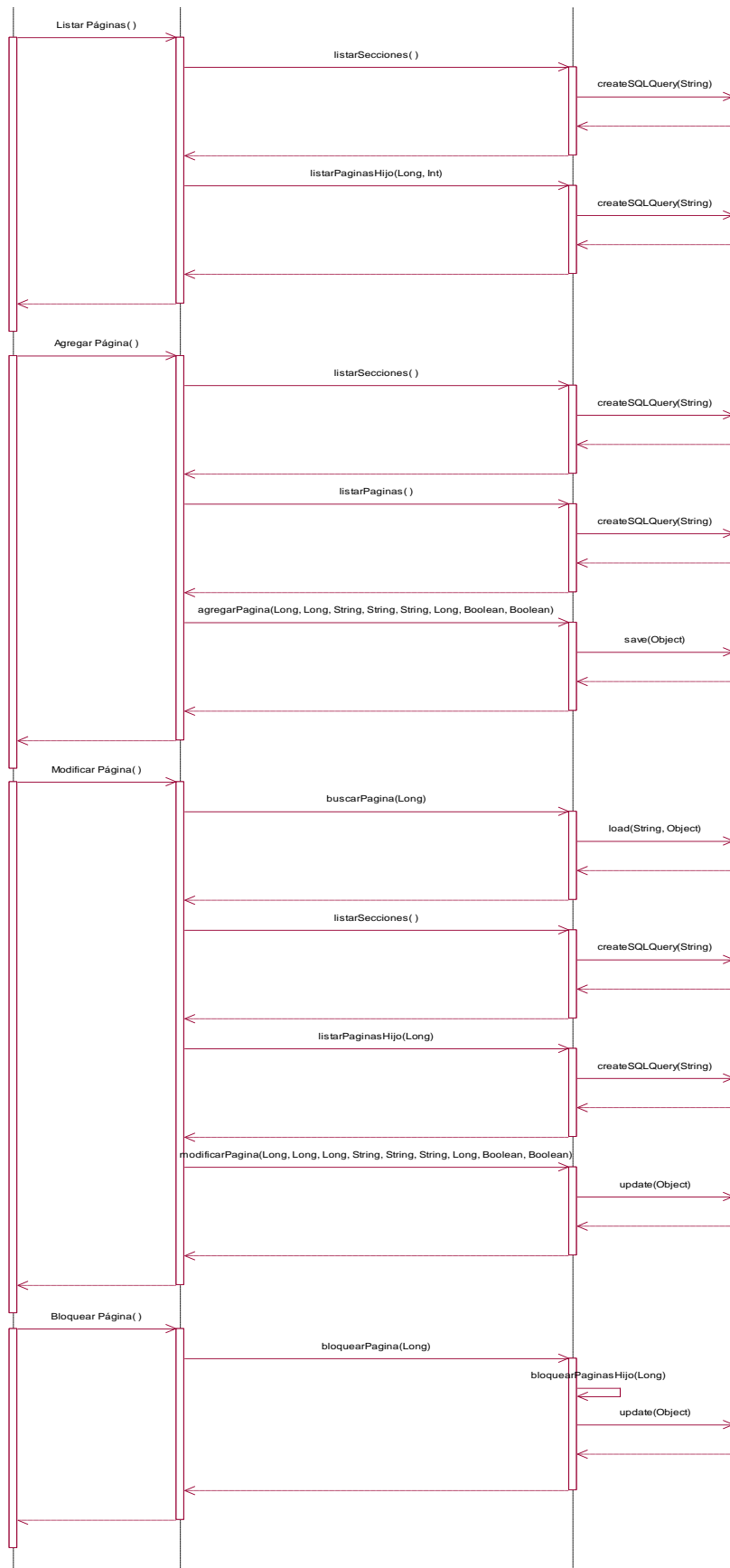


Figura 45 Administrar Secciones y Páginas (Páginas)

### 3.2.3.6. Administrar Cuentas de Usuario

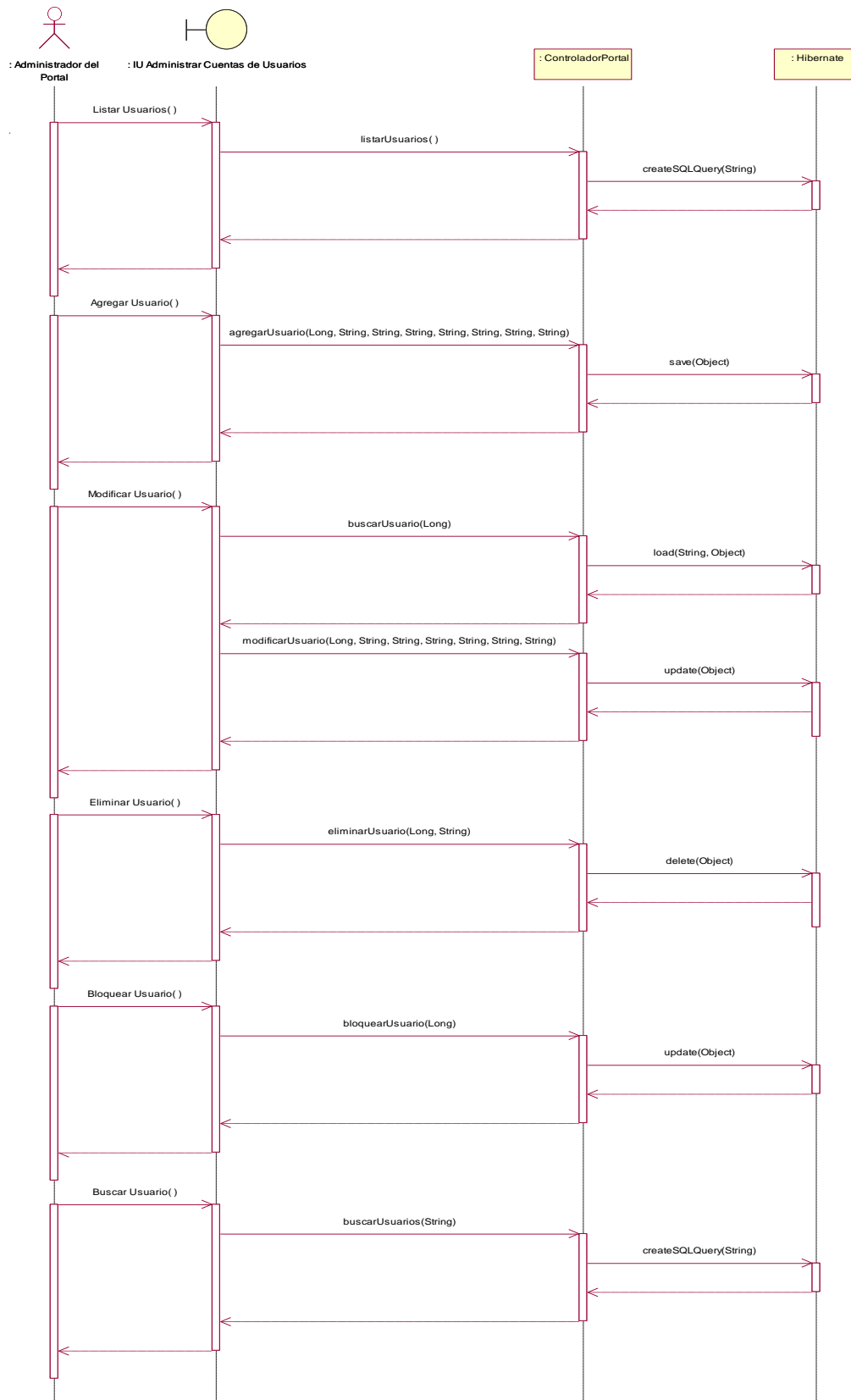


Figura 46 Administrar Cuentas de Usuario

### 3.2.3.7. Editar Currículo

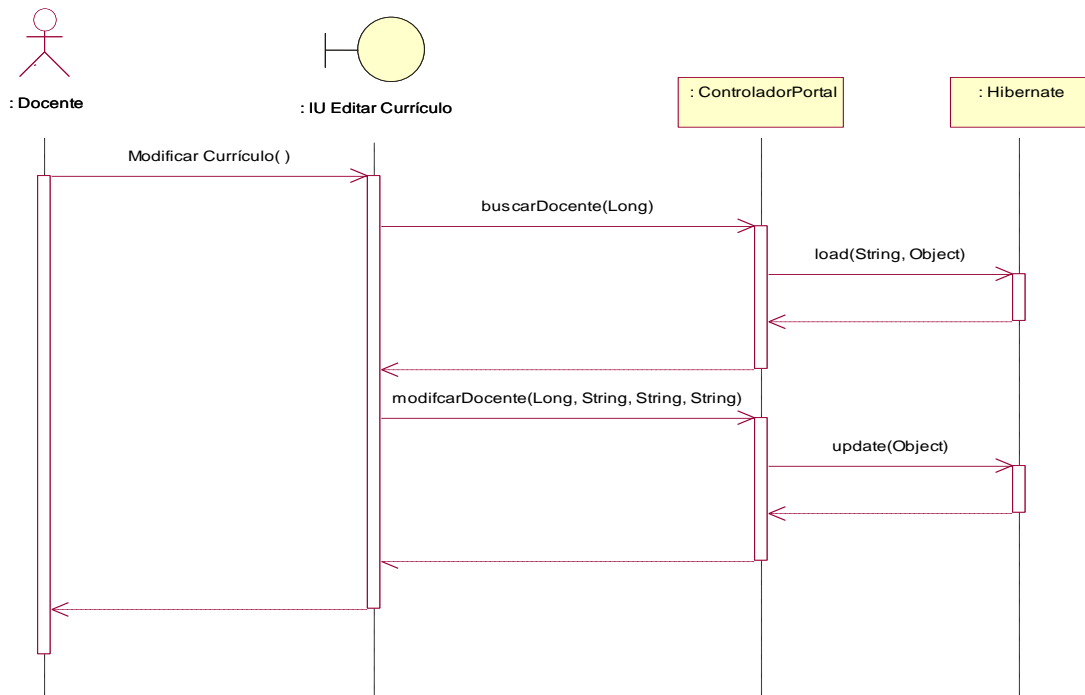


Figura 47 Editar Currículo

### 3.2.3.8. Administrar Zona de Materias

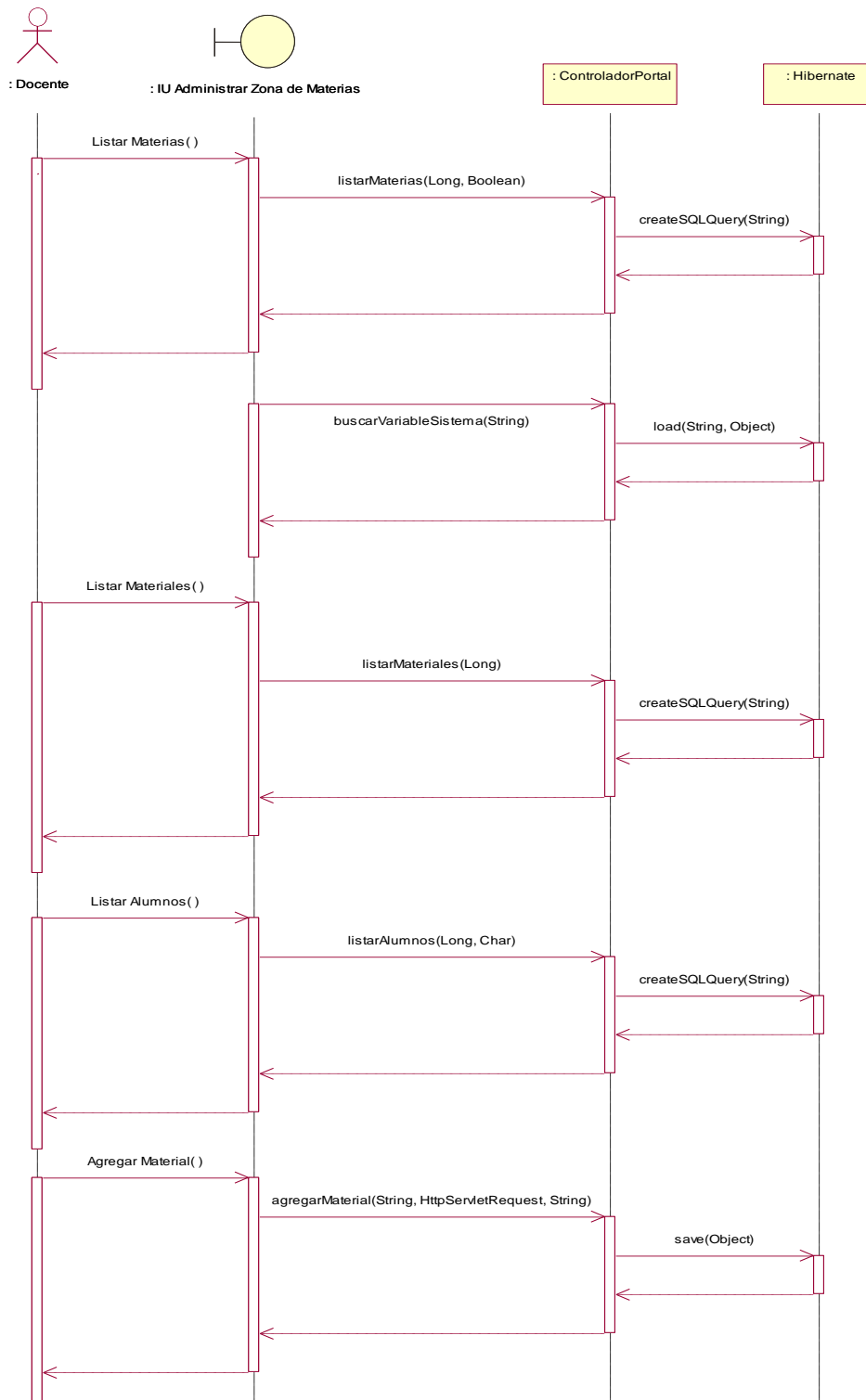
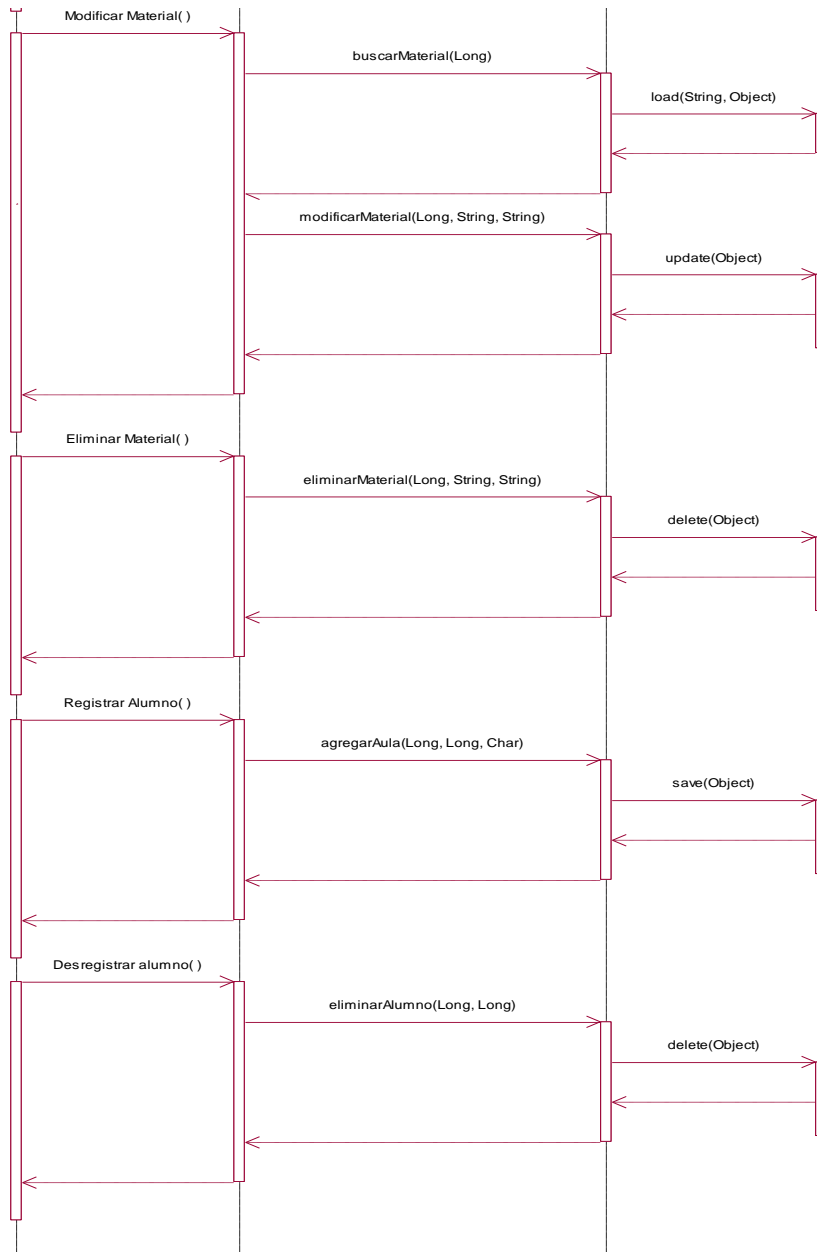


Figura 48 Administrar Zona de Materias



**Figura 49 Administrar Zona de Materias (Continuación)**

### 3.2.3.9. Validar Usuarios

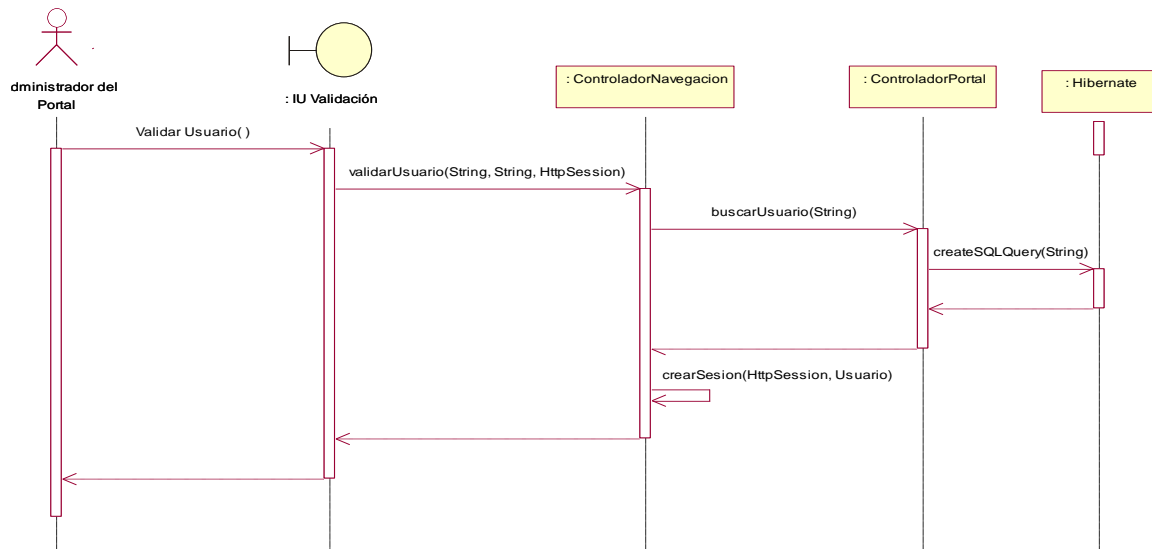


Figura 50 Validar Usuarios

### 3.2.3.10. Navegar en Zona de Materias

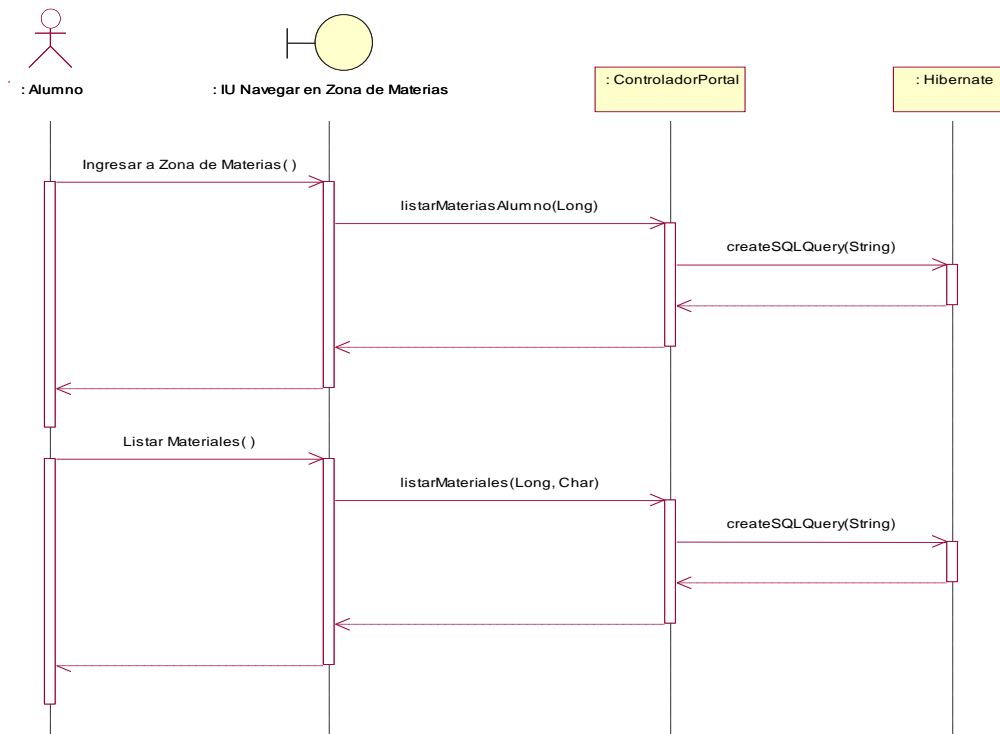


Figura 51 Navegar en Zona de Materias

### 3.2.3.11. Navegar en el Portal

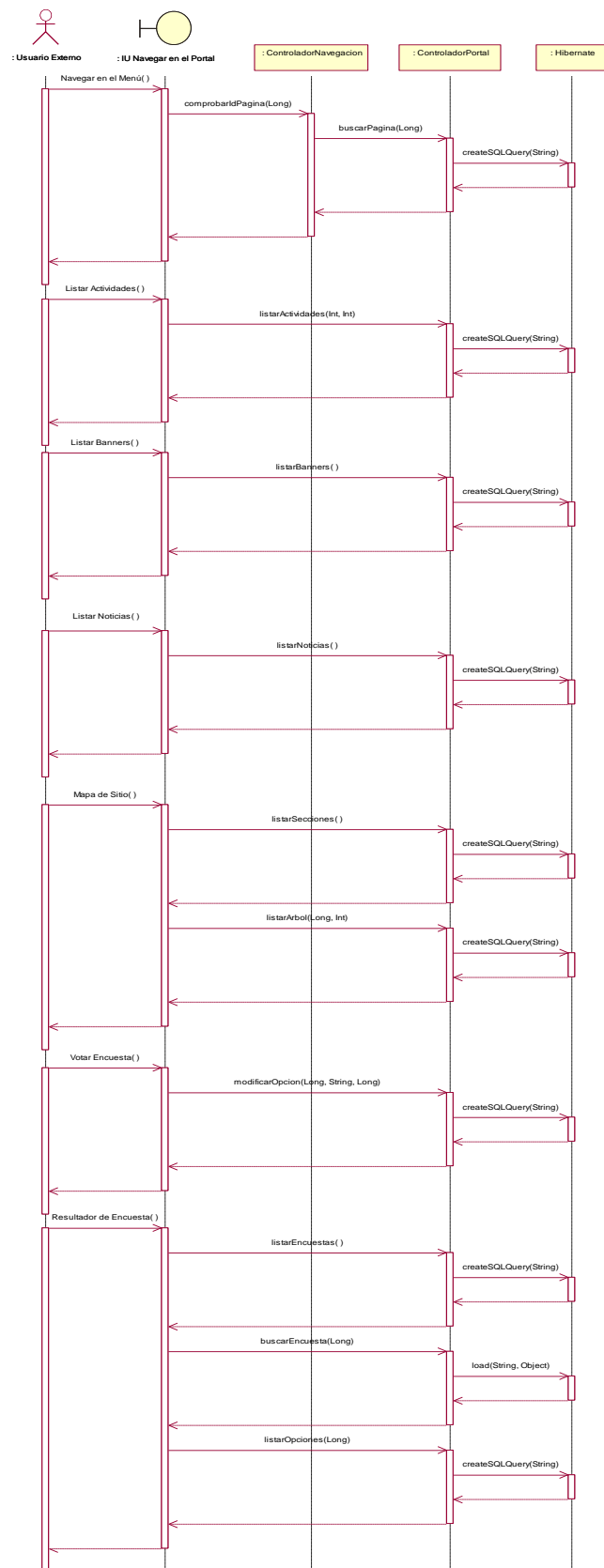


Figura 52 Navegar en el Portal

### 3.2.4. Diagramas de Clase de Diseño

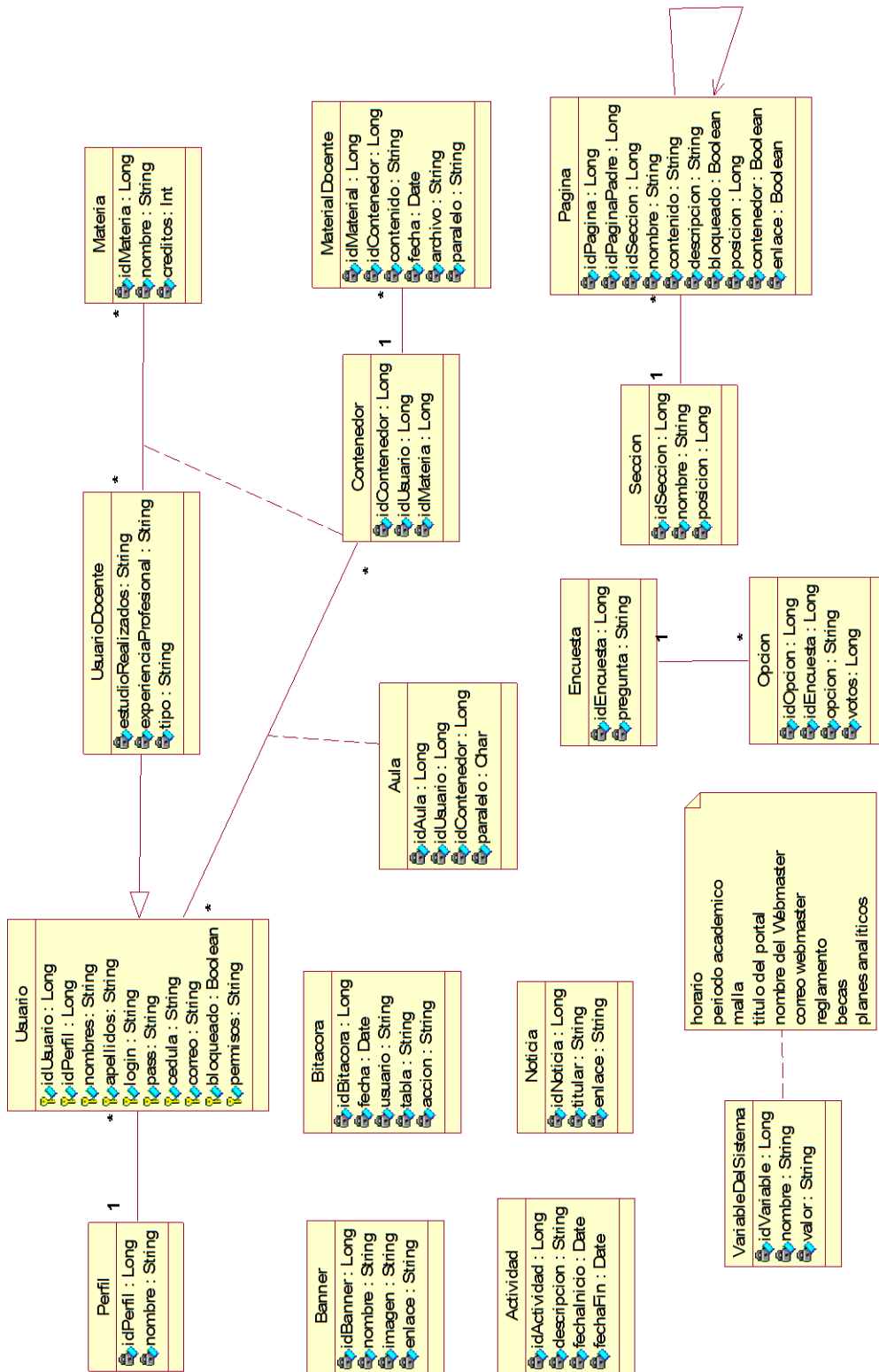


Figura 53 Diagrama de Clases de Diseño



### 3.2.5. Modelo Entidad/Relación

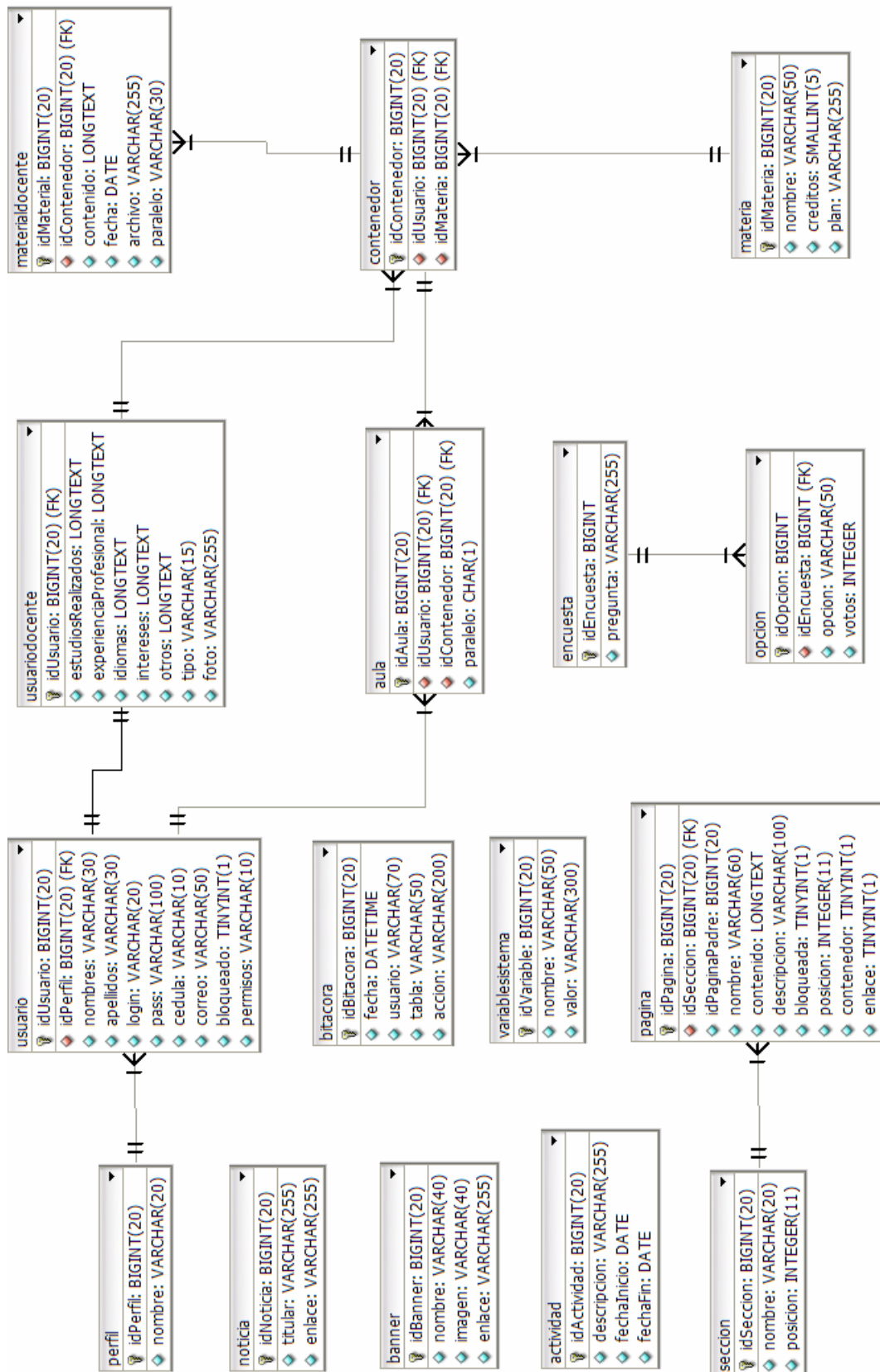


Figura 54 Modelo Entidad / Relación

### 3.3. Construcción

Durante esta fase se ha utilizado las siguientes aplicaciones y herramientas para el desarrollo del Portal:

- NetBeans 5.0
- JDK 1.5
- Sun Java System Application Server 9 Update 1
- MySql 5.0

Con la utilización de lo mencionado anteriormente se ha generado el código fuente respectivo, así como el script de la base de datos y los parámetros de configuración del servidor de aplicaciones.

Para verificar los resultados de la implementación, a fin de que la versión final del sistema sea entregada sin error, es necesario diseñar e implementar pruebas, las mismas que se detallan a continuación.

#### 3.3.1. Casos de Prueba

Las tablas que se encuentran a continuación describen las pruebas realizadas sobre el sistema para determinar posibles fallos y corregirlos. El listado de los errores corregidos se encuentran en la Tabla 35.

**Tabla 19 Caso de Prueba: Inicio de sesión**

Inicio de sesión	
Propósito	Comprobar que un usuario pueda ingresar al sistema
Datos de Prueba	usuario: {elascano, jbalda, chinojosa} password: {elascano, miclave, chinojosa}
Pasos	1. Ingresar a la página principal del Portal 2. Introducir el usuario 3. Introducir el password 4. Hacer clic en Ingresar

**Tabla 20 Caso de Prueba: Modificar Variables del Sistema**

<b>Modificar Variables del Sistema</b>	
Propósito	Verificar que las variables se modifiquen y que se reflejen los cambios
Datos de Prueba	-
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de General</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en General</li> <li>4. Realizar los cambios en las variables</li> <li>5. Hacer clic en Guardar</li> <li>6. Hacer clic en Regresar al Portal</li> <li>7. Verificar los cambios</li> </ol>

**Tabla 21 Caso de Prueba: Respaldo Base**

<b>Respaldo Base</b>	
Propósito	Comprobar respaldo de la base de datos
Datos de Prueba	-
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de General</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en General</li> <li>4. Hacer clic en Respaldo Base</li> <li>5. Guardar el archivo</li> <li>6. Abrir el archivo para comprobar el script</li> </ol>

**Tabla 22 Caso de Prueba: Reiniciar Datos**

<b>Reiniciar Datos</b>	
Propósito	Verificar que se realice el borrado de los datos, archivos y carpetas
Datos de Prueba	-
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de General</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en General</li> <li>4. Hacer clic en Reiniciar Datos</li> <li>5. Seleccionar todas la tablas</li> <li>6. Hacer clic en Reiniciar</li> <li>7. Verificar que los datos hayan sido borrados</li> </ol>

**Tabla 23 Caso de Prueba: Modificar Página de Error**

Modificar Página de Error	
Propósito	Verificar la página de error se pueda modificar
Datos de Prueba	contenido={Esta es la página de error}
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de General</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Página de Error</li> <li>4. Modificar el contenido</li> <li>5. Hacer clic en Modificar</li> <li>6. Ingresar un URL erróneo en el browser para comprobar los cambios</li> </ol>

**Tabla 24 Caso de Prueba: Reporte de Bitácora**

Reporte de Bitácora	
Propósito	Comprobar que se genere el reporte
Datos de Prueba	fechaInicio={2006-09-10} fechaFin={2006-11-02} cadena={creación}
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de General</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Bitácora</li> <li>4. Hacer clic en Reporte</li> <li>5. Introducir las fechas</li> <li>6. Introducir la cadena</li> <li>7. Hacer clic en Generar</li> </ol>

**Tabla 25 Caso de Prueba: Modificar Bloque de Bienvenida**

Modificar Bloque de Bienvenida	
Propósito	Comprobar que se modifique la bienvenida
Datos de Prueba	contenido={<h3>Bienvenidos al Departamento de Ciencias de la Computaci&oacute;n</h3><p>El Departamento de Ciencias de la Computaci&oacute;n&nbsp;les da la m&aacute;s cordial bienvenida al Portal Web.</p><p>Te invitamos a recorrer las distintas secciones, las cuales muestran lo que somos y lo que ofrecemos.</p><p>Nuestra excelencia depende de t&iacute;aacute;.</p>}
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Secciones y Páginas</li> <li>2. Hacer clic en Administrar Portal</li> </ol>

3. Hacer clic en Bloque de Bienvenida
4. Modificar el contenido
5. Hacer clic en Modificar
6. Hacer clic en Regresar al Portal para verificar el cambio

**Tabla 26 Caso de Prueba: Administrar Secciones**


Administrar Secciones	
Propósito	Comprobar que se pueda agregar, modificar y eliminar una sección
Datos de Prueba	<p>Para nueva sección:  nombre={Académico}  posición={2}</p> <p>Para modificar sección:  nombre={Educación}  posición={3}</p>
Pasos	<p>Para nueva sección:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Secciones y Páginas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Secciones</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir nombre</li> <li>6. Escoger posición</li> <li>7. Hacer clic en Crear</li> <li>8. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol> <p>Para modificar sección:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Secciones y Páginas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Secciones</li> <li>4. Hacer clic en la sección Académico</li> <li>5. Modificar nombre y posición</li> <li>6. Hacer clic en Modificar</li> <li>7. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol> <p>Para eliminar sección:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Secciones y Páginas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Secciones</li> <li>4. Haga clic en el icono  que se encuentra al lado de la sección Educación</li> <li>5. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol>

Tabla 27 Caso de Prueba: Administrar Páginas


Administrar Páginas	
Propósito	<p>Comprobar que se pueda agregar, modificar y eliminar una página</p> <hr/> <p>Para nueva página:</p> <pre> nombre={Horarios} pertenece={Académico} posición={3} descripción={horarios clases período académico} contenido={&lt;p&gt;Haga clic en el siguiente enlace para descargar&amp;nbsp;los horarios correspondientes al período actual de la carrera de Ingeniería en Sistemas e Informática.&lt;/p&gt;&lt;p&gt;&lt;a href=" ../archivos/File/horariooct06-mar07.pdf"&gt;Semestre Oct-2006&amp;nbsp;Mar-2007&lt;/a&gt;&lt;/p&gt;&lt;p&gt;&lt;a href=" ../archivos/File/horariohumanisticasoct06-mar07.pdf"&gt;Humanísticas Semestre Oct-2006&amp;nbsp;Mar-2007&lt;/a&gt;&lt;/p&gt;} </pre>
Datos de Prueba	<p>Para modificar página:</p> <pre> nombre={Horario} Pertenece={Académico} posición={2} descripción={ horario clases período académico} contenido={&lt;p&gt;Haga clic en el siguiente enlace para descargar&amp;nbsp;los horarios correspondientes al período actual de la carrera de Ingeniería en Sistemas e Informática.&lt;/p&gt;&lt;p&gt;&lt;a href=" ../archivos/File/horariooct06-mar07.pdf"&gt;Semestre Oct-2006&amp;nbsp;Mar-2007&lt;/a&gt;&lt;/p&gt;&lt;p&gt;&lt;a href=" ../archivos/File/horariohumanisticasoct06-mar07.pdf"&gt;Humanísticas Semestre Oct-2006&amp;nbsp;Mar-2007&lt;/a&gt;&lt;/p&gt;} </pre>
Pasos	<p>Para nueva página:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Secciones y Páginas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Páginas y Contenido</li> <li>4. Hacer clic en Nueva</li> <li>5. Introducir nombre</li> <li>6. Escoger página a la que pertenece</li> <li>7. Escoger posición</li> <li>8. Introducir descripción</li> <li>9. Introducir contenido</li> <li>10. Hacer clic en Crear</li> </ol>

11. Hacer clic en Regresar al Portal para verificar el cambio

Para modificar página:


1. Ingresar con un usuario con permiso de administración de Secciones y Páginas
2. Hacer clic en Administrar Portal
3. Hacer clic en Páginas y Contenido
4. Hacer clic en la página Horarios
5. Modificar nombre, posición, descripción y contenido
6. Hacer clic en Modificar
7. Hacer clic en Regresar al Portal para verificar el cambio

Para eliminar página:

1. Ingresar con un usuario con permiso de administración de Secciones y Páginas
2. Hacer clic en Administrar Portal
3. Hacer clic en Páginas y Contenido
4. Haga clic en el icono  que se encuentra al lado de la página Horario
5. Hacer clic en Regresar al Portal para verificar el cambio

**Tabla 28 Caso de Prueba: Administrar Encuestas**


Administrar Encuestas	
Propósito	Comprobar que se pueda agregar, modificar y eliminar una encuesta
Datos de Prueba	<p>Para nueva encuesta:  pregunta={¿Qué opina sobre el nuevo portal del Departamento?}  opciones={Excelente,Bueno,Malo}</p> <p>Para modificar encuesta:  pregunta={¿Qué opina sobre el nuevo portal?}  opciones={Excelente,Bueno,Malo}</p>
Pasos	<p>Para nueva encuesta:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Encuestas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Encuestas</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir pregunta</li> <li>6. Hacer clic en Crear</li> <li>7. Crear opciones</li> </ol> <p>Para modificar encuesta:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Encuestas</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Encuestas</li> </ol>

4. Hacer clic en la encuesta ¿Qué opina sobre el nuevo portal del Departamento?
  5. Modificar pregunta
  6. Hacer clic en Modificar
- Para eliminar encuesta:
1. Ingresar con un usuario con permiso de administración de Encuestas
  2. Hacer clic en Administrar Portal
  3. Hacer clic en Encuestas
  4. Haga clic en el icono  que se encuentra al lado de la encuesta ¿Qué opina sobre el nuevo portal?
- 

**Tabla 29 Caso de Prueba: Administrar Banners**

Administrar Banners	
Propósito	Comprobar que se pueda agregar, modificar y eliminar un banner
Datos de Prueba	<p>Para nuevo banner:</p> <p>nombre={ Célula .Net }</p> <p>enlace={ http://www.microsoft.com/latam/educacion/carrerascomputacion/netcells/default.asp }</p> <p>imagen={ celula.jpg }</p> <p>Para modificar banner:</p> <p>nombre={ Célula .Net Microsoft }</p>
Pasos	<p>Para nuevo banner:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Banners</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Banners</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir nombre</li> <li>6. Introducir enlace</li> <li>7. Escoger imagen</li> <li>8. Hacer clic en Crear</li> <li>9. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol> <p>Para modificar banner:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Banners</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Banners</li> <li>4. Hacer clic en el Banner Célula .Net</li> <li>5. Modificar nombre</li> <li>6. Hacer clic en Modificar</li> <li>7. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol> <p>Para eliminar banner:</p>



1. Ingresar con un usuario con permiso de administración de Banners
2. Hacer clic en Administrar Portal
3. Hacer clic en Banners
4. Haga clic en el icono  que se encuentra al lado del banner Célula .Net Microsoft

**Tabla 30 Caso de Prueba: Bloque de Eventos**

Modificar Bloque de Eventos	
Propósito	Comprobar que se modifiquen los eventos
Datos de Prueba	<pre> contenido={&lt;h4&gt;Elecci&amp;oacute;n de la Reina&lt;br /&gt;&lt;/h4&gt;26 de octubre del 2006. Asiste al evento en el Sal&amp;oacute;n 2000 y apoya a tu candidata favorita.&lt;br /&gt;} </pre>
Pasos	<ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Eventos</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Bloque de Eventos</li> <li>4. Modificar el contenido</li> <li>5. Hacer clic en Modificar</li> <li>6. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol>

**Tabla 31 Caso de Prueba: Administrar Noticias**


Administrar Noticias	
Propósito	Comprobar que se pueda agregar, modificar y eliminar una noticia
Datos de Prueba	<p>Para nueva noticia:</p> <pre> titular={ Yahoo se asocia con eBay para competir con Google } enlace={ http://www.100cia.com/ultima_hora/google_lidera_b_squedas_y_yahoo_el_correo_electronico_gratuito.html } </pre> <p>Para modificar noticia:</p> <pre> titular={ Yahoo se asocia con eBay para competir con el gigante Google } </pre>
Pasos	<p>Para nueva noticia:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Noticias</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Noticias</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir titular</li> <li>6. Introducir enlace</li> <li>7. Hacer clic en Crear</li> </ol>

8. Hacer clic en Regresar al Portal para verificar el cambio

Para modificar noticia:

1. Ingresar con un usuario con permiso de administración de Noticias
2. Hacer clic en Administrar Portal
3. Hacer clic en Noticias
4. Hacer clic en la noticia Yahoo se asocia con eBay para competir con Google
5. Modificar titular
6. Hacer clic en Modificar
7. Hacer clic en Regresar al Portal para verificar el cambio

Para eliminar banner:


5. Ingresar con un usuario con permiso de administración de Banners
6. Hacer clic en Administrar Portal
7. Hacer clic en Banners
8. Haga clic en el icono  que se encuentra al lado del banner Célula .Net Microsoft

**Tabla 32 Caso de Prueba: Administrar Actividades**

Administrar Actividades	
Propósito	Comprobar que se pueda agregar, modificar y eliminar una actividad
Datos de Prueba	<p>Para nueva actividad:</p> <p>descripción={Entrega de notas del primer parcial}  fecha inicio={2006-10-23}  Fecha finalización={2006-10-27}</p> <p>Para modificar actividad:</p> <p>descripción={Entrega de notas del primer parcial}  fecha inicio={2006-10-30}  fecha finalización={2006-11-03}</p>
Pasos	<p>Para nueva actividad:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Actividades</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Actividades</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir descripción</li> <li>6. Introducir fecha de inicio y finalización</li> <li>7. Hacer clic en Crear</li> <li>8. Hacer clic en Regresar al Portal para verificar el cambio</li> </ol> <p>Para modificar actividad:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Actividades</li> </ol>

2. Hacer clic en Administrar Portal
3. Hacer clic en Actividades
4. Hacer clic en la actividad Entrega de notas del primer parcial
5. Modificar fechas
6. Hacer clic en Modificar
7. Hacer clic en Regresar al Portal para verificar el cambio

Para eliminar actividad:


1. Ingresar con un usuario con permiso de administración de Actividades
2. Hacer clic en Administrar Portal
3. Hacer clic en Actividades
4. Haga clic en el icono  que se encuentra al lado de la actividad Entrega de notas del primer parcial

**Tabla 33 Caso de Prueba: Administrar Usuarios**

Administrar Usuarios	
Propósito	Comprobar que se pueda agregar, modificar y eliminar un usuario
Datos de Prueba	<p>Para nuevo usuario:</p> <ul style="list-style-type: none"> <li>perfil={ Alumno }</li> <li>nombres={ Córdoba Díaz }</li> <li>apellidos={ Carlos Andrés }</li> <li>cedula={ 1212121212 }</li> <li>login={ ccordoba }</li> </ul> <p>Para modificar usuario:</p> <ul style="list-style-type: none"> <li>perfil={ Alumno }</li> <li>nombres={ Córdoba Díaz }</li> <li>apellidos={ Carlos Xavier }</li> <li>cedula={ 1212121212 }</li> <li>login={ ccordoba }</li> </ul>
Pasos	<p>Para nuevo usuario:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Usuarios</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Usuarios</li> <li>4. Hacer clic en Nuevo</li> <li>5. Escoger perfil</li> <li>6. Introducir nombres</li> <li>7. Introducir apellidos</li> <li>8. Introducir cédula</li> <li>9. Introducir login</li> <li>10. Hacer clic en Crear</li> </ol> <p>Para modificar usuario:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Usuarios</li> </ol>

2. Hacer clic en Administrar Portal
3. Hacer clic en Usuarios
4. Hacer clic en el usuario ccordoba
5. Modificar datos
6. Hacer clic en Modificar

Para eliminar actividad:

1. Ingresar con un usuario con permiso de administración de Usuarios
2. Hacer clic en Administrar Portal
3. Hacer clic en Usuario
4. Haga clic en el icono  que se encuentra al lado del usuario ccordoba

**Tabla 34 Caso de Prueba: Administrar Materias**

Administrar Materias	
Propósito	Comprobar que se pueda agregar, modificar y eliminar una materia
Datos de Prueba	<p>Para nueva materia:</p> <p>nombre={Cálculo I}</p> <p>créditos={5}</p> <p>plan={planCall.pdf}</p> <p>Para modificar usuario:</p> <p>nombre={Cálculo I}</p> <p>créditos={6}</p>
Pasos	<p>Para nueva materia:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Materias</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Materias</li> <li>4. Hacer clic en Nuevo</li> <li>5. Introducir nombre</li> <li>6. Introducir créditos</li> <li>7. Escoger plan</li> <li>8. Hacer clic en Crear</li> </ol> <p>Para modificar materia:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Materias</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Materias</li> <li>4. Hacer clic en la materia Cálculo I</li> <li>5. Modificar datos</li> <li>6. Hacer clic en Modificar</li> </ol> <p>Para eliminar materias:</p> <ol style="list-style-type: none"> <li>1. Ingresar con un usuario con permiso de administración de Materias</li> <li>2. Hacer clic en Administrar Portal</li> <li>3. Hacer clic en Materias</li> </ol>

4. Haga clic en el icono  que se encuentra al lado de la materia Cálculo I

**Tabla 35 Listado de Errores**

Errores	Solución
No se puede eliminar banner	Añadir permiso para eliminar en el archivo server.policy
Campo Nombre muy pequeño en agregar y modificar materia	Se aumentó el tamaño del campo
Problema de validación en Variables del sistema	Se reverenciaron bien los campos para que valide bien
No se puede ingresar nuevo usuario	Se arregló el algoritmo de validación de la cédula
No se puede respaldar base	Aumentar tiempo del refresh para que se alcance a generar el archivo
Error al eliminar usuario	Añadir permiso de ejecución en el archivo server.policy
No se muestra los banners creados	Se corrigió la ruta donde se encuentran los banners
No se pueden poner tildes al añadir o modificar secciones	Corregir la expresión regular para permitir tildes
Las tildes no se guardan correctamente al crear banners	Recibir los datos con el tipo de codificación adecuado
La bitácora no muestra los últimos cambios	Modificar la consulta de la base para que muestre los datos en orden descendente
No se muestra la página de error personalizada	Corregir el URL de la página de error en el archivo web.xml
Los resultados de las encuestas siempre muestran cero	Modificar el proceso de votación para que contabilice adecuadamente los votos
Se eliminó la encuesta que se mostraba en el portal y generó un error	Se prohibió la eliminación de la encuesta que está seleccionada para ser mostrada
No se puede cambiar el orden de las páginas	Arreglar el algoritmo que ordena las páginas
Los alumnos no pueden descargar archivos subidos por docentes	Corregir la ruta de acceso a los archivos subidos
Nombres y Apellidos intercambiados al cargar lista de docentes con un archivo de excel	Modificar el código para asignar correctamente los campos

### 3.3.2. Diagrama de Componentes

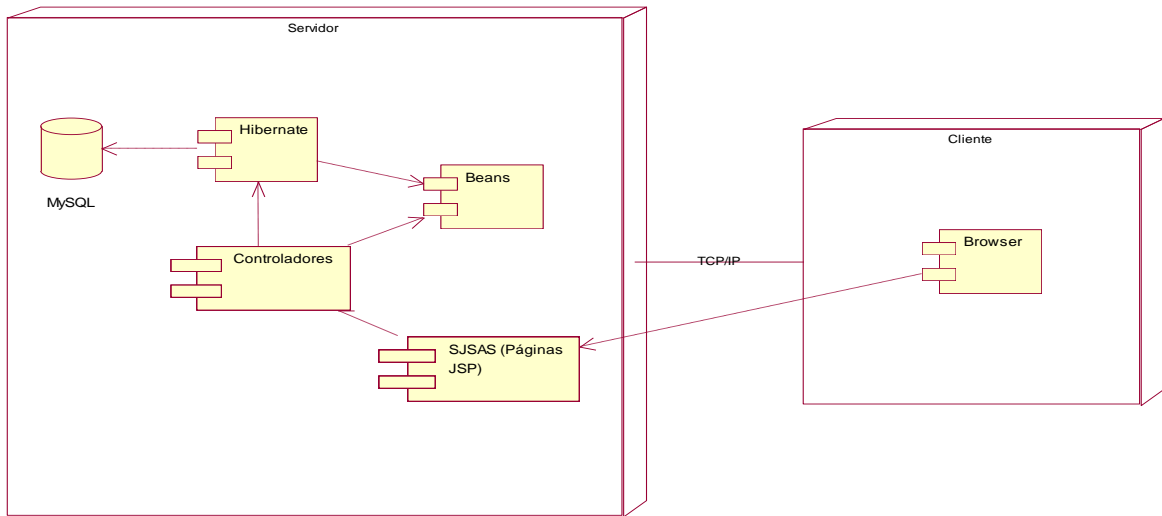


Figura 55 Diagrama de Componentes

## 3.4. Transición

### 3.4.1. Entrega

Adjunto a esta documentación se entregarán los respectivos Manuales de Instalación y de Usuario así como un CD en el que se encuentra la aplicación y todo el software necesario para que el Sistema pueda funcionar.

El CD contiene los siguiente:

- Aplicación
  - El Sistema, que es un archivo con extensión .WAR
  - El Script Inicial de la Base de Datos que permite crear la Base con la información respectiva.
- Software necesario para ejecutar la aplicación
  - Instaladores de la Base de Datos y de herramientas de administración

- El instalador del Servidor de Aplicaciones, para cargar la aplicación
- Aplicaciones complementarias
  - Extensión para el Servidor de Aplicaciones para que ejecute PHP
  - Cliente de Correo
  - Herramienta de Administración del Servidor
  - Navegadores Web

### 3.4.2. Manual de Instalación

El manual de Instalación del Sistema resume las características mínimas de hardware y que aplicaciones son necesarias para la ejecución del mismo. De igual manera se explican los pasos para la instalación de esas aplicaciones y como configurarlas.

El manual de Instalación del Sistema se encuentra en el Anexo B.

### 3.4.3. Manual de Usuario

El manual de Usuario resume las características funcionales que se pueden realizar en el Sistema. De igual manera se explican las distintas interfaces y los campos que éstas contienen.

El manual de Usuario se encuentra en el Anexo C.

## Capítulo IV: Conclusiones y Recomendaciones

### 3.5. Conclusiones

- El Portal se desarrolló e implementó con herramientas Open Source tales como Netbeans, Sun Java System Application Server, MySql, JDK y DBDesigner, las cuales se ejecutan bajo un sistema Linux, brindando así la ventaja de reducir costos en licencias, tanto para el desarrollo como para el uso de las aplicaciones, que finalmente encarecen el precio total de un producto.
- Rational Unified Process (RUP) es un marco de referencia que sirve de guía para el desarrollo de software y debido a su flexibilidad permitió adaptar sus modelos y diagramas para la construcción del Portal, siendo ésta una aplicación orientada a la Web.
- Mediante el uso de un sistema de Administración de Contenidos en el Portal se facilita la tarea de edición de contenido para el administrador del Portal (Webmaster), de tal manera que se pueda ofrecer la información actualizada a los visitantes por la rapidez con la que se pueden crear, modificar o eliminar páginas.
- Es necesario para cualquier institución, organización o empresa tener un espacio en la Web para darse a conocer, para mantener una comunicación con sus clientes y para asegurar una ventaja competitiva sobre los demás. La importancia de los conceptos de infocomunicación y el uso de sistemas informáticos es muy amplia en países desarrollados y no debe pasar desapercibida por empresas nacionales.
- La incorporación de un Portal en el Departamento, permite añadir servicios que mejoren las prestaciones internas y externas, situándolo en ventaja



respecto a otras unidades afines. Estos servicios pueden ser automatizaciones de muchas de las actividades del negocio como en el caso del DCC, préstamos de libros, equipos audiovisuales, administración de pasantías, etc.

- En el ambiente del intercambio de información de una aplicación Web es muy importante su seguridad. Es necesario tener conceptos bastantes sólidos para asegurar la confiabilidad, integridad, disponibilidad e interoperabilidad de la información.
- El principal problema al desarrollar una aplicación Web es la gran variedad de tecnologías y herramientas (servidores Web, navegadores, plataformas) existentes que al momento de interactuar e integrarse ocasionan muchos inconvenientes de compatibilidad. Por ejemplo, una página puede verse distinta en diferentes navegadores, debido a que no todos siguen completamente los estándares establecidos por la W3C. Una aplicación diseñada para una determinada tecnología puede presentar problemas de incompatibilidad al tratar de ejecutarla sobre otra tecnología.
- El levantamiento de Requerimientos es un proceso muy importante que requiere de la participación activa tanto de los usuarios como de quien realiza el levantamiento porque constituye la base sobre la que se construye una aplicación que debe satisfacer las necesidades de aquellos usuarios. Si los usuarios no están prestos a colaborar, el levantamiento puede resultar un fracaso, tornándose en una tarea ardua y complicada que puede retrasar un proyecto.

### **3.6. Recomendaciones**

- Se recomienda adquirir un certificado de seguridad por medio de una empresa con ese fin como Verisign, ya que estos certificados tienen la ventaja de que garantizan que la información viaje de forma segura y confiable a través de Internet, evitando así la manipulación de la misma.
- Desarrollar otras aplicaciones que se incorporen al Portal para que ayuden a mejorar los procesos del Departamento y así brindar un servicio de mayor calidad.
- Implementar un entorno apropiado para el servidor, que incorpore el equipamiento necesario para asegurar la funcionalidad del mismo durante el mayor tiempo posible del año. Es decir un espacio que incorpore el medio ambiente adecuado, las redundancias de fuentes de energía y de comunicación con el mundo exterior.
- A pesar de la flexibilidad de RUP, es recomendable usar una metodología orientada a la Web porque ofrecen otras características específicas para este tipo de aplicaciones que facilitan y mejoran su desarrollo.
- Se recomienda difundir el uso del portal y sus servicios prestados a docentes y estudiantes para que justifique la presencia del mismo y creen necesidades que puedan ser desarrolladas e implantadas a futuro.
- Es de vital importancia mantener al día la información mostrada por el Portal, designando responsables de actualiza las diferentes secciones que existen en la Aplicación, y así mostrar información útil y oportuna a los visitantes del Portal.

## Bibliografía

- JACOBSON Ivar; BOOCH Grady, RUMBAUGH James (2000). El Proceso Unificado de Desarrollo de Software. Primera Edición. España. Addison Wesley.
- PRESSMAN, Roger (2001). Ingeniería de Software: Un enfoque práctico. Quinta Edición. McGraw-Hill.
- IBM. Redbooks Paper: Rational Software IBM. Disponible en Web: <http://www.redbooks.ibm.com/redpapers/pdfs/redp3877.pdf>
- Wikipedia. Rational Unified Process. Disponible en Web: [http://en.wikipedia.org/wiki/Rational\\_Unified\\_Process](http://en.wikipedia.org/wiki/Rational_Unified_Process)

## **Anexo A: Especificación de Requerimientos de Software**

## **Especificación de Requisitos Software para el desarrollo del Portal de la FISI**

### **1. Introducción**

Este documento es una Especificación de Requisitos Software (ERS) para el desarrollo del Portal de Facultad de Sistemas e Informática de la Escuela Politécnica del Ejército. Todo su contenido ha sido elaborado teniendo en cuenta las necesidades que presenten la Facultad. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar IEEE 830

#### **1.1. Propósito**

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto a los desarrolladores del Portal, como a los directivos de la Facultad y a la comunidad de posibles usuarios finales. Después de ser aprobado, este documento constituirá la base sobre la cual quedaran establecidas de forma definitiva las características funcionales y requerimientos de Hardware y Software, siendo esta la guía del proceso de desarrollo del Portal de la FISI.

#### **1.2. Alcance**

El Portal es una aplicación orientada a la Web la cual tendrá como principal objetivo dar a conocer a la FISI, su historia, el personal docente y administrativo con el que cuenta, así como también los laboratorios y todo lo que un usuario externo necesita saber sobre la facultad; también permitir una mejor comunicación entre el Personal Docente, Administrativo y los Estudiantes.

Con esto algunos procesos internos serán más ágiles siendo un beneficio no solo para los estudiantes sino también para todo el personal de la facultad al momento de implantar el Portal.

### **1.3. Definiciones, acrónimos y abreviaturas**

- ERS: Especificación de Requerimientos de Software
- FISI: Facultad de Ingeniería de Sistemas e Informática
- ESPE: Escuela Politécnica del Ejército

### **1.4. Visión global**

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del Portal, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

## **2. Descripción**

### **2.1. Perspectiva del producto**

Este Portal, es el primero que va a tener la FISI ya que actualmente cuenta con una página Html que provee servicio de e – mail, además el sistema por su naturaleza tiene una estrecha relación con el sistema académico, pero con el fin de precautelar la seguridad de la información académica se ha previsto contar con archivos planos.

La aplicación será un sistema que utilizará programación modular y un administrador de contenidos lo cual la hará escalable, ya que se pueden añadir nuevas funciones mediante módulos que pueden ser realizados independientemente y crear la cantidad de páginas que el Administrador crea necesario de forma rápida y sencilla.

### **2.2. Funciones del producto**

- Proveer información relevante para el estudiante, docente y personal administrativo de la FISI
- Brindar los servicios de foros, chat, e – mail y ftp
- Permitir que el personal docente, administrativo y estudiantes de la FISI puedan tener acceso a la información relevante de la facultad
- Bloc de Docentes, donde los estudiantes pueden consultar el plan analítico de la materia, ver información, recursos y bibliografía de la misma, además de poder subir o presentar los deberes al profesor de su materia
- Permitir que los estudiantes puedan acceder a información acerca del proceso de graduación y oportunidades de trabajo así como también las pasantías que estén disponibles
- Permitir una fácil administración mediante la implementación de un editor de contenidos
- Presentar información de la Facultad a los usuarios y público en general
- Uso de Páginas HTTP Seguras y cuentas de usuario con las cuales se implementarán sesiones para asegurar que la información sea íntegra y confiable

### 2.3. Características del usuario

Se puede definir 4 tipos de usuarios los cuales se detallan a continuación:

- *Administrador*, Es la persona encargada de subir y crear nuevas páginas así como también definir y restringir el acceso
- *Docente*, Este usuario tendrá acceso al Bloc de Docente permitiéndole subir información concerniente a sus materias además de poder ver la información de la facultad y usar los servicios del Portal
- *Estudiante*, Podrá tener acceso a su respectiva materia y además de poder ver la información concerniente a la facultad y hacer uso de los servicios del portal
- *Otros*, Estos usuarios solo podrán acceder a la información considerada pública referente a la Facultad y podrán acceder al servicio de foros

### 2.4. Restricciones generales

- El Portal va a obtener la información de los estudiantes mediante un archivo plano que nos va a proporcionar el área de Organización y Sistemas de la ESPE al momento de iniciar el periodo académico
- El Cliente necesita un Navegador Web conectado al Internet para poder acceder al Portal (como mínimo, Internet Explorer v4.0, Mozilla Firefox v1.0 y Netscape v6.0)
- El servicio de ftp va a ser provisto por la plataforma de implantación



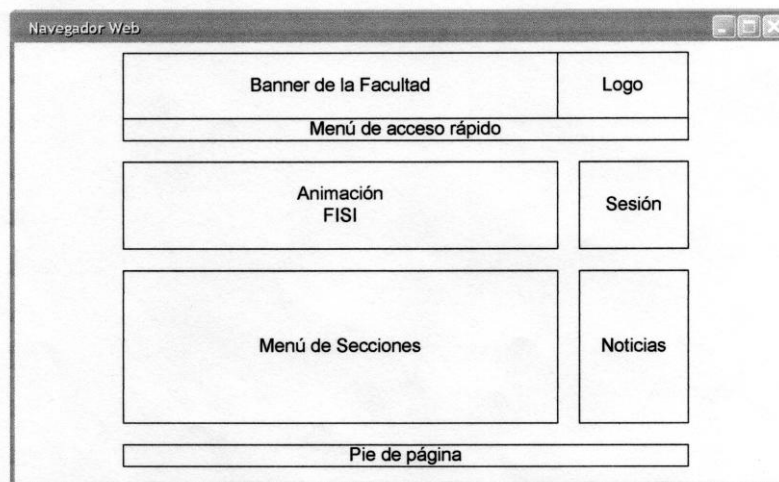


### 3. Requerimientos específicos

#### 3.1. Requerimiento de interfaces externas

##### 3.1.1. Interfaces de usuario

###### *Página principal*

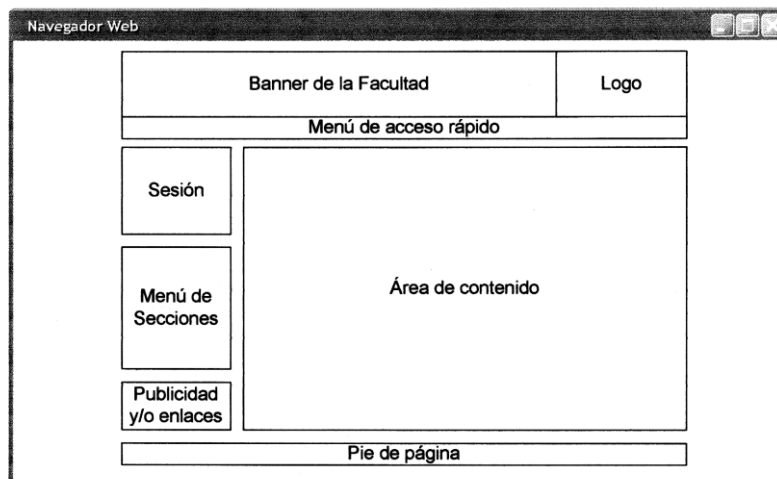


Objeto	Descripción	Tamaño [pixel]
Página	Página principal, permite el ingreso al sistema, contiene los enlaces a las diferentes secciones y otro tipo de información general.	Ancho = 800
Banner de la Facultad	Imagen con el nombre de la Escuela Politécnica del Ejército y el de la Facultad de Ingeniería de Sistemas e Informática	Ancho = 570 Alto = 100
Logo	Animación que muestra el escudo de la FISI y la ESPE	Ancho = 90 Alto = 90
Menú de acceso rápido	Contiene acceso al Home Page, mapa del sitio, portal de la ESPE y búsqueda	Ancho = 800



Animación FISI	Animación que muestra las áreas de aplicación en la que los estudiantes pueden desenvolverse en el campo laboral	Ancho = 569 Alto = 175
Sesión	Estado de la sesión: Iniciar, Cerrar y datos	Ancho = 172
Menú de Secciones	Muestra los enlaces a las diferentes páginas que ofrece el portal agrupadas por secciones	Ancho = 569
Noticias	Muestra las principales noticias	Ancho = 172
Pie de página	Muestra los derechos de autor	Ancho = 800

### ***Páginas secundarias***



Objeto	Descripción	Tamaño [píxel]
Página	Diseño de páginas secundarias, template usadas para mostrar el contenido	Ancho = 800
Banner de la Facultad	Imagen con el nombre de la Escuela Politécnica del Ejército y el de la Facultad de Ingeniería de Sistemas e Informática	Ancho = 570 Alto = 100
Logo	Animación que muestra el escudo de la FISI	Ancho = 90

	y la ESPE	Alto = 90
Menú de acceso rápido	Contiene acceso al Home Page, mapa del sitio, portal de la ESPE y búsqueda	Ancho = 800
Sesión	Estado de la sesión: Iniciar, Cerrar y datos	Ancho = 150
Menú de Secciones	Muestra los links a otras páginas agrupados por secciones a los que podemos acceder desde el portal	Ancho = 150
Publicidad y/o enlaces	Publicidad que la facultad requiera mostrar (Empresas, eventos, etc.) y/o enlaces externos	Ancho = 150
Área de contenido	Presenta el contenido, con el título de la página, las sub secciones y su respectivo contenido, varía de acuerdo a las necesidades que requiera la página a mostrar	Ancho = 609
Pie de página	Muestra los derechos de autor	Ancho = 800

### 3.1.2. Interfaces de Hardware

El Sistema se va a comunicar con el Hardware de la siguiente manera:

- Para el ftp se va a usar el puerto 21
- Para https el puerto 443
- Para el e – mail el puerto 25
- Para http el puerto 80
- Para el chat el puerto 194

### 3.1.3. Interfaces de Software

- El software que va a ser utilizado como Sistema Operativo es LINUX Red Hat 9, el cual debe estar configurado para ejecutar JSP (Jakarta Tomcat 5.5.9 y JDK v1.5.0) y PostgreSQL
- La Base de Datos va a ser PostgreSQL v 8.0

### **3.1.4. Interfaces de comunicación**

El Portal va a ser una aplicación orientada a la Web para lo cual la FISI ya cuenta con una IP Real y un nombre de Dominio que es el <http://fisi.espe.edu.ec>.

Por tal motivo solo se debe realizar el sistema con las características y funciones mencionadas anteriormente y subirlo para que pueda ser accedido por cualquier usuario, sea este de la Facultad o público en general.

## **3.2. Requerimientos funcionales**

### **3.2.1. Editor de Contenidos**

El Editor de Contenidos nos va a permitir:

- Administración de las páginas para el Portal
- Administración de Home Page de Cada Docente
  - Foto del Docente
  - Curriculum
  - Datos Generales
  - Materias que dicta

### **3.2.2. Bloc de Docentes**

El Bloc de Docente tiene las siguientes características:

- Ver Plan Analítico de Materia
- Subir Software relacionado con cada materia del Docente
- Subir Deberes
- Ver Documentación extra

### **3.2.3. Bolsa de Trabajo**

La Bolsa de Trabajo va a permitir:

- Que las empresas ingresen sus datos para estar en la lista de candidatos
- Que las empresas puedan publicar ofertas para pasantías
- Que las empresas puedan publicar ofertas de trabajo

#### **3.2.4. Plan de Tesis**

Con este módulo los estudiantes podrán:

- Averiguar el estado en el que se encuentra su plan de tesis
- Ver Profesores Informantes
- Saber si su Plan de Tesis ha sido aprobado
- Asignación de Director y Co – Director de la Tesis

#### **3.2.5. Foros y Chat**

Los servicios de foro y Chat nos van a permitir:

- Comunicación entre docentes, estudiantes y personal administrativo de la FISI
- Comunicación entre estudiantes
- Crear foros de discusión sobre temas y/o proyectos de investigación

#### **3.2.6. Servicio de e – mail**

Se va a configurar un Web Mail de tipo free – software que esté acorde a las necesidades de la Facultad.

#### **3.2.7. Servicio de ftp**

Se va a realizar la configuración al acceso del ftp de la plataforma de implantación del Portal

### **3.3. Requerimientos de funcionamiento**

- El Sistema va a constar de 4 áreas las cuales son: Información de la Facultad de tipo pública, Sección Docentes, Sección Alumnos y Servicios
- Se debe implantar un servidor con un Sistema Operativo LINUX o UNIX (capaz de soportar JSP y PostgreSQL) con 512 MB de RAM y 160 GB en Disco Duro
- Para poder ser accedida por cualquier usuario sea este de la Facultad o externo, se debe tener acceso al internet ya que el Portal es una aplicación orientada a la Web

### 3.4. Restricciones del diseño

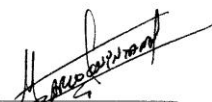
- Evitar los callejones
  - Que los usuarios no lleguen a un página o sección de la cual no puedan regresar a páginas anteriores o al Menú Principal
- Retroalimentación y mensajes del Sistema
- Cumplir con los estándares para el diseño de páginas Web
- Seguridades al acceder a las páginas del Portal
- Seguridad del Portal y de su Base de Datos

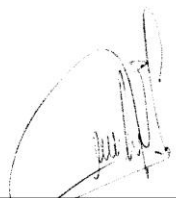
### 3.5. Atributos


- Tener un diseño agradable y atractivo para los usuarios y público en general
- La Arquitectura es una aplicación Cliente / Servidor de 3 capas con orientación Web
- El desarrollo se realizará utilizando herramientas Open Source, JAVA para las páginas dinámicas y la Base de Datos es PostgreSQL

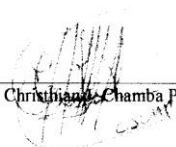
### 3.6. Otros requerimientos

- Las Opciones del Menú debe tener una Categorización por diferentes temas y permitir el regreso a páginas anteriores
- Realizar Diagrama Navegacional del Portal

  
\_\_\_\_\_  
Mayor Ing. Marco Quintana  
Decano de la FISI

  
\_\_\_\_\_  
Ing. Walter Fuertes  
Sub - Decano de la FISI

  
\_\_\_\_\_  
Ing. Cecilia Hinojosa  
Planificadora Académica

  
\_\_\_\_\_  
Christiana Chamba P.

  
\_\_\_\_\_  
Juan Pablo Balda

  
\_\_\_\_\_  
Carlos Anchundia

## **Anexo B: Manual de Instalación**













































## **Anexo C: Manual de Usuario**

























































































## Biografías

### Carlos Eduardo Anchundia Valencia

**Lugar y Fecha de Nacimiento:** Manta, 14 de octubre de 1982

**Estudios Primarios:** Unidad Educativa FAE No. 4 (Manta)

**Estudios Secundarios:** Unidad Educativa FAE No. 4 (Manta)

**Títulos Obtenidos:** Bachiller en Comercio y Administración Especialidad  
Computación

**Honores Obtenidos:** Primer Escolta (Primaria y Secundaria) – 2000  
Mejor Bachiller – 2000

### Juan Pablo Balda Andrade

**Lugar y Fecha de Nacimiento:** Guayaquil, 31 de octubre de 1981

**Estudios Primarios:** Liceo Los Andes (Guayaquil)  
Academia USA (Quito)

**Estudios Secundarios:** Academia USA (Quito)

**Títulos Obtenidos:** Bachiller en Informática

**Honores Obtenidos:** Abanderado (Primaria y Secundaria) – 2000  
Mejor Bachiller – 2000

## HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

---

Carlos Eduardo Anchundia Valencia

---

Juan Pablo Balda Andrade

**COORDINADOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS E  
INFORMÁTICA**

---

Ing. Ramiro Delgado

Sangolquí, diciembre del 2006