

**UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**“DISEÑO DE UNA APLICACIÓN WEB PARA LA GESTIÓN EN LÍNEA DE  
LOS SERVICIOS ACADÉMICOS DE UNA INSTITUCIÓN  
DE EDUCACIÓN SUPERIOR”**

*Realizado por:*

**Oscar Enrique Rondón Guariman**

**Trabajo de Grado presentado como Requisito Parcial para optar al Título de  
INGENIERO DE SISTEMAS**

**BARCELONA, OCTUBRE DE 2009**

**UNIVERSIDAD DE ORIENTE**  
**NÚCLEO DE ANZOÁTEGUI**  
**ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS**  
**DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**“DISEÑO DE UNA APLICACIÓN WEB PARA LA GESTIÓN EN LÍNEA DE  
LOS SERVICIOS ACADÉMICOS DE UNA INSTITUCIÓN  
DE EDUCACIÓN SUPERIOR”**

**Asesor Académico:**

---

M.Sc. Manuel Carrasquero

**BARCELONA, OCTUBRE DE 2009**

**UNIVERSIDAD DE ORIENTE**  
**NÚCLEO DE ANZOÁTEGUI**  
**ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS**  
**DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**DISEÑO DE UNA APLICACIÓN WEB PARA LA GESTIÓN EN LÍNEA DE**  
**LOS SERVICIOS ACADÉMICOS DE UNA INSTITUCIÓN**  
**DE EDUCACIÓN SUPERIOR**

---

**M.Sc. Manuel Carrasquero**  
Asesor Académico

---

**M.Sc. Andrés Martínez**  
Jurado Principal

---

**M.Sc. Felysol Siso**  
Jurado Principal

**BARCELONA, OCTUBRE DE 2009**

## **RESOLUCIÓN**

### **Artículo 44**

Del reglamento de Trabajo de Grado

“Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y solo podrán ser utilizados a otros fines con el conocimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario”.

## ÍNDICE GENERAL

RESOLUCIÓN .....	IV
ÍNDICE GENERAL .....	V
ÍNDICE DE FIGURAS Y TABLAS .....	XI
RESUMEN .....	XVIII
CAPITULO 1 .....	19
PLANTEAMIENTO DEL PROBLEMA .....	19
1.1 Planteamiento del problema .....	19
1.2 Objetivos.....	22
1.2.1 Objetivo general .....	22
1.2.2 Objetivos específicos .....	23
CAPITULO 2 .....	24
MARCO TEORICO .....	24
2.1 Antecedentes.....	24
2.2 Ingeniería de software .....	25
2.3 Lenguaje Unificado de ModeladoUML.....	27
2.3.1 Objetivos del lenguaje unificado de modelado.....	27

2.3.2	Uso del lenguaje unificado de modelado.....	28
2.3.3	Fases del ciclo de desarrollo que soporta UML. ....	29
2.3.4	Diagramas que ofrece el UML. ....	29
2.4	Modelo cliente-servidor.....	40
2.5	Programación orientada a objetos.....	41
2.5.1	Los beneficios del enfoque OO. ....	42
2.6	Servidor Web seguro. ....	43
2.7	Páginas Web. ....	44
2.8	El lenguaje SQL.....	45
2.9	Las bases de datos.....	46
2.9.1	El modelo entidad-relación.....	47
2.9.2	Normalización. ....	50
2.10	Lenguaje de programación PHP. ....	55
2.10.1	Ventajas del lenguaje PHP. ....	56
2.10.2	Desventajas del Lenguaje PHP.....	56
2.11	Common Gateway Interface (CGI). ....	57
2.12	Secure Socket Layer (SSL).....	58
2.12.1	El Protocolo SSL Handshake. ....	59

2.12.2 Protocolo SSL Record. ....	60
2.12.3 Cómo saber si una Conexión se está Realizando Mediante SSL. ....	60
2.12.4 OpenSSL.....	60
2.13 Sistema operativo gnu/linux .....	61
CAPITULO 3 .....	63
ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS.....	63
3.1 Análisis de la situación actual .....	64
3.2 Compresión de los procesos del negocio.....	65
3.2.1 Encontrar actores y Casos de Uso que Participan en el proceso de registro de Notas. ....	66
3.3 Captura de requisitos .....	71
3.3.1 Requisitos funcionales del sistema propuesto. ....	72
3.3.2 Identificación de actores y casos de uso del sistema propuesto. ....	73
3.4 Requisitos no funcionales del sistema propuesto. ....	94
3.5 Análisis de factibilidad para implementar el sistema propuesto .....	98
3.6 Análisis de casos de uso mediante diagramas de colaboración.....	99
3.6.1 Realización del Caso de Uso Iniciar Sesión Estudiante. ....	100
3.6.2 Realización del Caso de Uso Iniciar Sesión Docente.....	101
3.6.3 Realización del Caso de Uso Iniciar Sesión Coordinador Eje UBV. ....	102

3.6.4 Realización del Caso de Uso Iniciar Sesión Administrador.....	103
3.6.5 Realización del Caso de Uso Salir de Sesión Estudiante. ....	104
3.6.6 Realización del Caso de Uso Registrar Notas. ....	107
3.6.7 Consultar Carga Académica. ....	108
3.6.8 Elaborar Carga Académica. ....	110
3.6.9 Modificar Carga Académica.....	111
3.6.10 Registrar Docente. ....	112
3.6.11 Modificar Registro de Docente.....	113
3.6.12 Generar Notas Certificadas.....	114
3.6.13 Reiniciar Cuenta Estudiante. ....	115
3.6.14 Generar Reporte de Inscripción.....	116
3.6.15 Elaborar Oferta Académica. ....	117
3.6.16 Modificar Oferta Académica. ....	118
3.6.17 Modificar inscripción de Unidades Curriculares.....	119
3.6.18 Registrar Notas. ....	120
3.6.19 Modificar Registro de Notas.....	121
3.6.20 Crear Cuenta Estudiante. ....	122
3.6.21 Consultar Registro de Notas. ....	123

3.6.22 Generar Constancias de Estudio. ....	124
3.6.23 Inscribir Unidades Curriculares. ....	125
3.6.24 Generar Constancia de inscripción. ....	126
3.6.25 Generar Backup. ....	127
3.6.26 Registrar Coordinador de Eje UBV. ....	128
3.6.27 Modificar Registro de Coordinadores de Eje UBV. ....	129
3.7 Evaluación de la fase de análisis y definición de requerimientos. ....	130
CAPITULO 4 .....	131
DISEÑO DEL SISTEMA PROPUESTO. ....	131
4.1 Diseño de casos de uso-análisis. ....	114
4.1.1 Identificación de las Clases de Diseño Participantes. ....	114
4.1.2 Diagrama de Clases involucradas en el Inicio de Sesión y Salida de Sesión. ....	117
4.1.3 Diagrama de Clases involucradas en el Proceso de Inscripción de Unidades Curriculares. ....	119
4.1.4 Diagrama de Clases involucradas en el Proceso de Registro de notas. ...	121
4.1.5 Diagrama de Clases involucradas en el Proceso de Administración de Usuarios. ....	122
4.2 Diseño de la base de datos. ....	125
4.2.1 Modelo Relacional del sistema SURUBV. ....	126

4.3 Diseño de la interfaz del sistema. ....	131
4.3.1 Diseño de la Interfaz Principal del Sistema.....	131
4.3.2 Diseño de la Interfaz General de Usuario.....	133
4.3.3 Diseños para Reportes Impresos.....	136
4.4 Plataforma de soporte para el sistema propuesto.....	140
CONCLUSIONES.....	141
RECOMENDACIONES .....	143
BIBLIOGRAFÍA .....	144
METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO: ....	147

## ÍNDICE DE TABLAS

<a href="#"><u>Tabla 2.1 Estereotipos Utilizados en la Notación WAE</u></a>	39
<a href="#"><u>Tabla 3.1 Actores del sistema de la Misión Sucre</u></a>	66
<a href="#"><u>Tabla 3.2 Distribución de la Universidad Bolivariana de Venezuela en Ejes</u></a>	69
<a href="#"><u>Tabla 3.3 Funcionalidades del Sistema Propuesto</u></a>	73
<a href="#"><u>Tabla 3.4 Identificación de Actores para el Sistema Propuesto</u></a>	74

## ÍNDICE DE FIGURAS

<a href="#"><u>Figura 2.1 Modelo de Cascada de Desarrollo de Software</u></a>	8
<a href="#"><u>Figura 2.2 Diagramas del UML que expresan gráficamente un Modelo</u></a>	11
<a href="#"><u>Figura 2.3 Ejemplo de Modelo de Casos de Uso</u></a>	12
<a href="#"><u>Figura 2.4 Ejemplo de un Diagrama de Clases</u></a>	15
<a href="#"><u>Figura 2.5 Ejemplo de un Diagrama de Colaboración</u></a>	16
<a href="#"><u>Figura 2.6 Ejemplo de un Diagrama de Secuencia</u></a>	19
<a href="#"><u>Figura 2.7 Modelo Cliente-Servidor en un entorno Web</u></a>	23
<a href="#"><u>Figura 2.8 Intercambio de Información entre un Navegador Web (cliente) y un Servidor Web Seguro</u></a>	26
Figura 2.9 Tabla en Primera Forma Normal	33
Figura 2.10 Tabla que no está en 2FN	34
Figura 2.11 Tabla Productos	35
Figura 2.12 Tabla Proveedores	35

Figura 2.13 Tabla Atletas	36
Figura 2.14 Tabla Atletas parte 1.	36
Figura 2.15 Tabla Atletas parte 2.	36
Figura 2.16 Transacción usando Cifrado SSL	40
Figura 2.17 Indicación de una Conexión Segura en Navegadores Web	41
Figura 3.1 Modelo del Negocio para el Proceso de Registro de Notas en el Sistema Actual	52
Figura 3.2 Modelo de Casos de Uso del Sistema de SURUBV	78
Figura 3.3 Representación de la Tecnología a usar en el Sistema Propuesto	85
Figura 3.4 Clases de Análisis	85
Figura 3.5 Diagrama de Colaboración de la Realización del Caso de Uso Iniciar Sesión Usuario.	86
Figura 3.6 Diagrama de Colaboración de la Realización del Caso de Uso Salir de Sesión Usuario	87
Figura 3.7 Diagrama de Colaboración de la Realización del Caso de Uso Registrar notas Docente	88
Figura 3.8 Diagrama de Colaboración de la Realización del Caso de Uso Registrar notas Coordinador Eje UBV	89

Figura 3.9 Diagrama de Colaboración de la Realización del Caso de Uso Consultar Carga Académica Docente	90
Figura 3.10 Diagrama de Colaboración de la Realización del Caso de Uso Consultar Carga Académica de Docente por el Coordinador de Eje UBV	91
Figura 3.11 Diagrama de Colaboración de la Realización del Caso de Uso Elaborar Carga Académica	92
Figura 3.12 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Carga Académica	93
Figura 3.13 Diagrama de colaboración de la realización del Caso de Uso Registrar docente	94
Figura 3.14 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Registro de Docente	95
Figura 3.15 Diagrama de Colaboración de la Realización del Caso de Uso Consultar Notas Registradas	96
Figura 3.16 Diagrama de Colaboración de la Realización del Caso de Uso Consultar Notas Registradas Estudiante	97
Figura 3.17 Diagrama de Colaboración de la Realización del Caso de Uso Reiniciar Cuenta Estudiante	98
Figura 3.18 Diagrama de Colaboración de la Realización del Caso de Uso Generar Reporte de Inscripción	99
Figura 3.19 Diagrama de Colaboración de la Realización del Caso de Uso Elaborar Oferta Académica	100
Figura 3.20 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Oferta Académica	101

Figura 3.21 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Inscripción de Unidades Curriculares	102
Figura 3.22 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Registro de Notas	103
Figura 3.23 Diagrama de Colaboración de la Realización del Caso de Uso Crear Cuenta Estudiante	104
Figura 3.24 Diagrama de Colaboración de la Realización del Caso de Uso Generar Constancia de Estudio por el usuario Estudiante	105
Figura 3.25 Diagrama de Colaboración de la Realización del Caso de Uso Generar Constancia de Estudio por el Usuario Coordinador Eje UBV	105
Figura 3.26 Diagrama de Colaboración de la Realización del Caso de Uso Generar Constancia de Inscripción por el Usuario Estudiante	106
Figura 3.27 Diagrama de Colaboración de la Realización del Caso de Uso Generar Constancia de Inscripción por el Usuario Coordinador Eje UBV	107
Figura 3.28 Diagrama de Colaboración de la Realización del Caso de Uso Inscribir Unidades Curriculares por Usuario Estudiante.	108
Figura 3.29 Diagrama de Colaboración de la Realización del Caso de Uso Inscribir Unidades Curriculares por Usuario Coordinador Eje UBV.	108
Figura 3.30 Diagrama de Colaboración de la Realización del Caso de Uso Generar Backup	109
Figura 3.31 Diagrama de Colaboración de la Realización del	110

## Caso de Uso Registrar Coordinador de Eje UBV

Figura 3.32 Diagrama de Colaboración de la Realización del Caso de Uso Modificar Registro de Coordinador de Eje UBV.	111
Figura 4.1a Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis	115
Figura 4.1b Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis	115
Figura 4.1c Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis	116
Figura 4.2 Identificación de Clases de Diseño a partir de Clases de Control del Modelo de Análisis	116
Figura 4.3 Identificación de Clases de Diseño a partir de Clases de Entidad del Modelo de Análisis	117
Figura 4.4 Diagrama de Clases involucradas en los Casos de uso Inicio y Salida de Sesión	118
Figura 4.5 Diagrama de Clases involucradas en los Procesos de Inscripción de Unidades Curriculares	120
Figura 4.6 Diagrama de Clases Involucradas en los Procesos de Registro de Notas	123
Figura 4.7 Diagrama de Clases Involucradas en los Procesos de Administración de Usuarios	124
Figura 4.8 Estructura General de una Tabla en una Base de Datos Relacional	125
Figura 4.9 Modelo Relacional de la Base de Datos para del	130

## Sistema SURUBV

Figura 4.10 Diseño Preliminar de Interfaz de Inicio del Sistema SURUBV	132
Figura 4.11 Interfaz General de Usuario	133
Figura 4.12 Interfaz del Usuario Estudiante	134
Figura 4.13 Interfaz del Usuario Docente	134
Figura 4.14 Interfaz del Usuario Coordinador Eje UBV	135
Figura 4.15 Interfaz del Usuario Administrador	135
Figura 4.16 Diseño de Constancia de Inscripción	136
Figura 4.17 Diseño de Constancia de Estudios	137
Figura 4.18 Diseño de Notas Registradas	138
Figura 4.19 Diseño de Acta de Notas	139
Figura 4.20 Diseño de la Plataforma	140

## **RESUMEN**

El presente Trabajo se refiere al diseño de una aplicación informática utilizando tecnología Web. Este permitirá la gestión en línea de los servicios académicos de la Universidad Bolivariana de Venezuela (UBV), la cual está distribuida en 5 sedes en todo el territorio nacional. La UBV ofrece Programas de Formación de Grado que se imparten no sólo en la sede, sino también en otras instalaciones denominadas “aldeas”. Para la recolección de información acerca de los procesos que dan soporte a los servicios académicos como son: las inscripciones, solicitud de documentos, registro de notas, prosecución del estudiante, entre otros, se emplearon técnicas como la entrevista y observación directa. El modelado de los procesos se realizó mediante el Lenguaje de Modelado Unificado (UML). En la fase de análisis se elaboró un modelo general de casos de uso, lo cuales se desarrollaron a través de diagramas de colaboración empleándose clases de análisis, estas últimas representan las clases que se usan en el modelo de diseño de clases. Estos modelos se construyeron por medio de los estereotipos del UML para aplicaciones Web denominada “Extensión para aplicaciones Web” (WAE). El diseño de la base de datos se elaboró siguiendo el modelo relacional, el cual permitió establecer las relaciones entre las distintas entidades participantes. Las interfaces se diseñaron considerando los diversos tipos de usuarios que van a interactuar con el sistema de acuerdo a su nivel de conocimiento tecnológico.

# CAPITULO 1

## PLANTEAMIENTO DEL PROBLEMA

### 1.1 Planteamiento del problema

La Universidad Bolivariana de Venezuela nace mediante decreto Presidencial n° 2.517, en fecha 18 de julio de 2003 para hacer la educación superior accesible a todos los venezolanos, así mismo *como un* proyecto educativo y social vinculado a las demandas del desarrollo integral de la nación que plantea entre sus condiciones fundamentales la elevación del nivel cultural y educativo del pueblo venezolano. Esta universidad se crea igualmente para ser la cara institucional de la “Misión Sucre”, la cual tiene por objeto la participación comunitaria, para garantizar el acceso a la educación universitaria a todos los bachilleres sin cupo y transformar la condición de excluidos del subsistema de educación superior, la Misión Sucre, fue creada mediante Decreto Presidencial Número 2601, del 8 de septiembre del 2003 y se propone municipalizar la educación superior, orientarla hacia las regiones y las localidades.

En la actualidad 11 universidades y 28 colegios e instituciones participan activamente en satisfacer la demanda educativa, entre los cuales están, UNELLEZ (Universidad Experimental de los Llanos Ezequiel Zamora), UNESR (Universidad Experimental Simón Rodríguez), UEFM (Universidad Experimental Francisco Miranda), UNERG (Universidad Experimental Rómulo Gallegos), UNERMB (Universidad Experimental Rafael María Baralt), UBV (Universidad Bolivariana de Venezuela) y la UNEFA (Universidad Experimental de las Fuerzas Armadas). Una vez creada la Misión Sucre, diseña e implanta una plataforma tecnológica para atender las necesidades de los procesos administrativos y los servicios académicos de las universidades adscritas a ella, utilizando un sistema informático denominado

SINES (Sistema de Información Nacional de Educación Superior), el cual cuenta con una arquitectura cliente-servidor utilizando tecnología Web, permitiéndole de esta manera obtener control sobre los procesos antes mencionados.

Los espacios donde se ofrecen las clases por parte de las distintas universidades se conocen como “aldeas universitarias” y están diseminados por todo el país en escuelas, universidades e institutos, cada uno a cargo de un coordinador de aldea, todo esto bajo la administración de Misión Sucre, la cual es entonces la responsable del registro de notas, pago de nóminas y procedimientos administrativos que van a permitir el buen desempeño de los procesos educativos en cada una de las aldeas universitarias. Por otro lado, Misión Sucre tiene la responsabilidad de realizar los procesos de admisión, prosecución y egreso de cada cohorte de estudiantes en los diferentes Programas de Formación de cada una de las escuelas, universidades e institutos adscritos.

El proceso de admisión e inscripción es realizado periódicamente a través de la plataforma SINES, donde se ofrecen los Programas de Formación de las distintas universidades, el proceso de inscripción es realizado una sola vez, dejando parte de trabajo de prosecución a los coordinadores de aldeas universitarias, ya que el alumno no se inscribe nuevamente, por otro lado el coordinador de aldea debe administrar el vaciado de notas a través de la misma plataforma, lo que implica más trabajo y responsabilidades tomando en cuenta el volumen de estudiantes en una aldea universitaria. Desde sus inicios el sistema ha presentado inconsistencias en la data, por ejemplo existen códigos de aldeas que hacen referencia a una misma aldea universitaria, por otra parte el hecho de que los alumnos se inscriban una sola vez conllevó a que la deserción de estudiantes sea administrada por los mismos coordinadores de aldeas quienes deben realizar esas labores de depuración de data, lo que no garantiza que se realice, manteniéndose así una cantidad de estudiantes que no es real en la actualidad.

En este sentido para la Universidad Bolivariana de Venezuela no es factible depender de un sistema en esas condiciones, dado que en recientes graduaciones en Programas de Formación de la UBV se pudo evidenciar que no se contaban la data actualizada, lo que la coloca en una posición vulnerable en ese sentido, siendo que la mayoría de la población estudiantil de UBV es municipalizada, la universidad cuenta actualmente con 5 sedes distribuidas en el país y unos 13 Programas de Formación, tomando en cuenta lo antes mencionado se plantea diseñar una plataforma con tecnologías actuales que permita administrar los procesos de admisión, prosecución y egreso de los estudiantes de la UBV, un sistema independiente que satisfaga las necesidades de las situaciones que se presenten.

Es por ellos que el propósito de este trabajo será diseñar una aplicación que permita administrar los servicios académicos de la Universidad Bolivariana de Venezuela.

Para elaborar este trabajo se utilizará el modelo de desarrollo en cascada de la Ingeniería de Software, este modelo comprende etapas en el ciclo de vida del software, entre sus fases principales están; el análisis de requerimientos, diseño de la aplicación, codificación, pruebas, implantación y mantenimiento. Para el desarrollo de estas fases se utilizará la notación *UML (Unified Modeling Language)* o Lenguaje Unificado de Modelamiento, ya que es hoy un estándar para el de diseño orientado a objetos y es el resultado de tres opciones existentes en el mercado: BOOCH, RUMBAUGH y COAD-YOURDON, el UML fue adoptada como estándar por el *OMG (Object Management Group)* el 14 de noviembre de 1997.

Desde sus inicios la universidad no contó con una plataforma integrada de servicios académicos que le permitiera controlar y obtener información a nivel nacional de manera rápida, por el contrario cada sede desarrolló mecanismos para automatizar los procesos necesarios y dar servicios como los de inscripción, de

registro de notas, emisión de documentos, etc. y el control de las aldeas se dejó a la Misión Sucre, la cual contaba con una plataforma para administrar los servicios académicos de diferentes universidades y el personal docente. Tomando en cuenta esto la Universidad Bolivariana de Venezuela se ve en la necesidad de unificar todos los esfuerzos en una sola plataforma única de registro y servicios académicos de allí la gran importancia de este trabajo, el cual sentará la base y replanteará las estructuras que dan apoyo a los procesos académicos en las distintas sedes de esta universidad.

Con el decreto 3390, en Venezuela se busca impulsar el uso de software libre en las instituciones públicas del estado, las cuales deben emplear prioritariamente Software Libre desarrollado con estándares abiertos, en sus sistemas, proyectos y servicios informáticos, las tecnologías actuales permiten la integración de diferentes plataformas de desarrollo, tomando en cuenta lo anterior la universidad decidió iniciar el proyecto del desarrollo de la plataforma única de registro y servicios académicos bajo estándares abiertos.

Para llevar a cabo esta tarea se debe crear una aplicación, lo cual involucrará diferentes etapas en la vida del desarrollo del sistema software, éstas abarcarán hasta la fase de diseño, dando inicio a solucionar un problema de gran importancia en las actividades académicas de la Universidad Bolivariana de Venezuela, lo que motiva primordialmente a realizar este proyecto resaltando su carácter de importancia en el diseño.

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

Diseñar una aplicación Web para la gestión en línea de los servicios académicos de una institución de educación superior.

### **1.2.2 Objetivos específicos**

1. Describir el funcionamiento de los procesos en los servicios académicos ofrecidos por la Universidad Bolivariana de Venezuela.
2. Analizar la plataforma actual a través de la cual se automatizan los servicios académicos.
3. Definir los requerimientos que permitan satisfacer las necesidades de los usuarios.
4. Definir la arquitectura de software para el diseño de la aplicación.
5. Diseñar la Base de Datos del sistema propuesto.
6. Diseñar las interfaces que darán soporte a los procesos de los servicios académicos.

## CAPITULO 2

### MARCO TEORICO

#### 2.1 Antecedentes

Hoy día se encuentran disponibles una gama de proyectos cuyo objetivo fundamental es el diseño y/o desarrollo de aplicaciones en ambientes cliente/servidor, entre los que se citan:

En el año 1998, el estudiante José Guevara, elaboró un trabajo de grado titulado “*Desarrollo e Implementación de los Servicios Académicos del Departamento de Computación y Sistemas, usando Tecnología WWW*”, donde se planteaba que la información de los servicios académicos, administrativos y de investigación del departamento de Computación y Sistemas era manejada manualmente lo cual generaba un desaprovechamiento de los recursos humanos de éste departamento, para solucionar este problema desarrolló e implementó una infraestructura red basada en tecnología Internet, que permitió el desarrollo e implementación de una serie de aplicaciones y páginas Web que proporcionaron un medio para la administración e intercambio de información entre dichos servicios [9].

Para el año 1999, los estudiantes Alejandra Galantón y Oswaldo Arocha, desarrollaron como trabajo de grado “*Modernización de los Sistemas Automatizados de Admisión, Inscripción de Estudiantes de Nuevo Ingreso y Validación de la Programación Académica del Núcleo de Anzoátegui de la Universidad de Oriente*”, donde se plantea una migración de plataforma centralizada a una cliente/servidor con el fin de modernizar el sistema existente de inscripción de estudiantes y validación de la programación académica. Para ello se diseñó e implantó un sistema con el concepto cliente/servidor el cual fue desarrollado con la

aplicación de programación Power Builder y se uso el gestor de bases de datos Sybase SQL, logrando de ésta manera un sistema con las prestaciones necesarias para tal fin [8].

Para el año 2001 el estudiante Iván José Puglieser Saroff realizó el trabajo de grado *“Desarrollo del Sistema de Compras Cliente/Servidor para la Universidad de Oriente, Núcleo Anzoátegui”*, En este trabajo se plantea la necesidad de que en la Universidad de Oriente Núcleo Anzoátegui se desarrolle un sistema que permita gestionar las compras de la Universidad de Oriente núcleo Anzoátegui, para lo cual el estudiante Ivan Puglieser diseñó una herramienta para gestionar el procesamiento de las solicitudes de compras, ordenes de compras, hojas de análisis e informe de recepción. El análisis y diseño de esta aplicación fue realizado mediante la notación *UML (Unified Modeling Language)* y fue implantado bajo la tecnología cliente/servidor [20].

Para el año 2001 el estudiante Alfredo Molero desarrolló el trabajo titulado *“Diseño de la Intranet de la Escuela de Medicina de la Universidad de Oriente Núcleo de Anzoátegui”*. Donde se plantea el diseño e implantación de un proyecto de alto nivel tecnológico que solvete los problemas de comunicación y coordinación de índole académico-administrativo de la escuela de medicina. Para ello de diseño una infraestructura de hardware y software que conformó la Intranet de dicha escuela la cual permitió el uso de aplicaciones que se diseñaron para el uso en la Intranet así como herramientas que permitieran ayudar al control de las distintas actividades administrativas [17].

## **2.2 Ingeniería de software**

El proceso de ingeniería de software se define como “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la

obtención de un producto de software de calidad" [7]. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [7].

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición (*véase* figura 2.1).

La concepción define el alcance del proyecto y desarrolla un caso de negocio, la elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura, la construcción crea el producto y la transición transfiere el producto a los usuarios.



**Figura 2.1** Modelo de Cascada de Desarrollo de Software.  
**Fuente:** elaboración propia, año:2009.

Actualmente se encuentra en una etapa de madurez el enfoque *OO* (*Orientado a Objetos*) como paradigma del desarrollo de sistemas de información. El (OMG) Object Management Group es un consorcio a nivel internacional que integra a los

principales representantes en la industria de la tecnología de información OO, éste tiene como objetivo central la promoción, fortalecimiento e impulso de la tecnología OO, propone y adopta por consenso especificaciones entorno a esta. Una de las especificaciones más importantes es la adopción en 1998 del Lenguaje de Modelado Unificado o UML como un estándar, que junto con el *Proceso Unificado* están consolidando la tecnología OO.

### **2.3 Lenguaje Unificado de ModeladoUML.**

El Lenguaje Unificado de Modelado o UML es una técnica para la especificación de sistemas en todas sus fases. Esta ha sido desarrollada por los más importantes autores en materia de análisis y diseño de sistemas, ha sido usada con éxito en sistemas hechos para toda clase de industrias alrededor del mundo: salud, bancos, comunicaciones, aeronáutica, finanzas, etc, [19].

UML no es un lenguaje de programación. Existen herramientas que pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. Este es pues un lenguaje de propósito general para el modelado orientado a objetos, UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

#### **2.3.1 Objetivos del lenguaje unificado de modelado.**

UML es un lenguaje de modelado que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática [11].

UML no pretende ser un método de desarrollo completo, pues no incluye un proceso de desarrollo paso a paso, pero puede manejar todos los conceptos que se consideran necesarios para utilizar un proceso moderno de desarrollo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso, por otro lado busca ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesiten construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software como son la encapsulación y componentes.

### **2.3.2 Uso del lenguaje unificado de modelado.**

UML sirve para hacer modelos que permitan:

- a) Visualizar como es un sistema o como de desea.
- b) Especificar la estructura y/o comportamiento de un sistema.
- c) Hacer una plantilla que guíe la construcción de los sistemas

El modelado sirve no solamente para los grandes sistemas; aún en aplicaciones de pequeño tamaño se obtienen beneficios de modelar, sin embargo, es un hecho que entre más grande y más complejo es el sistema, el modelado juega un papel más importante, esto se debe a una razón simple: se hacen modelos de sistemas complejos porque no se pueden entender en su totalidad [11].

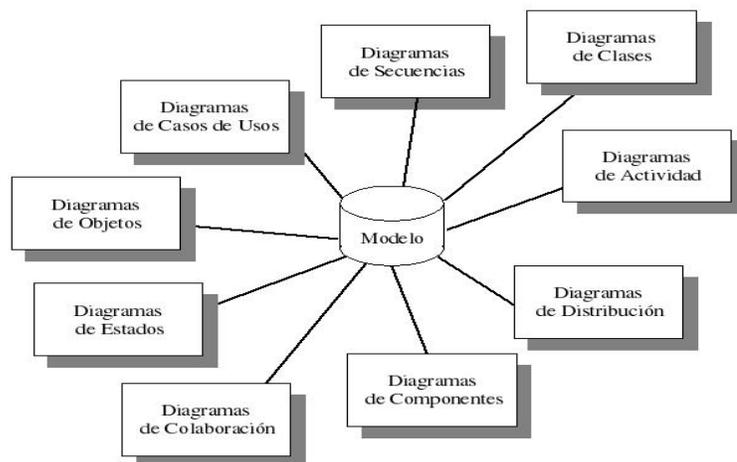
El UML es independiente de metodología, por lo que puede ser usada y lo es en distintas metodología como: *Fusion*, *Objectory*, *Rational Unified Process*, *OMT*, *ECM*, *Catalysys*, etc. La independencia antes mencionada permite que las organizaciones adapten el uso de UML a la metodología que consideren más apropiada.

### 2.3.3 Fases del ciclo de desarrollo que soporta UML.

Cada diagrama puede ser usado con énfasis distinto en las fase de desarrollo: análisis, diseño e implementación, un diagrama cualquiera en una fase de tendrá un estudio lógico, cabe aclarar que aunque UML es orientado a objetos preferentemente, esto es útil en cualquier modelo tecnológico ya que es independiente de lenguajes de programación o tecnología determinada [19].

### 2.3.4 Diagramas que ofrece el UML.

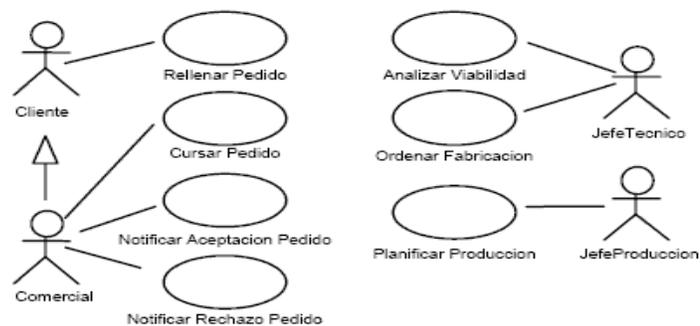
El UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático pasando por el análisis, diseño, implementación y hasta configuración. Estos gráficos son un conjunto de elementos con sus relaciones, por otro lado ofrecen una vista del sistema a modelar. Para poder representar correctamente un sistema UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas, entre estos diagramas se tienen los siguientes [11]:



**Figura 2.2** Diagramas del UML que expresan gráficamente un Modelo.  
**Fuente:** elaboración propia, año:2009.

### 2.3.4.1 Diagrama de Casos de Usos.

El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema *véase* figura 2.3. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar donde se representan los requisitos funcionales. Es decir se está diciendo lo que tiene que hacer un sistema



**Figura 2.3** Ejemplo de Modelo de Casos de Uso.  
**Fuente:** <http://www.cyta.com.ar/ta0604/v6n4a1.htm>, año:2007

### 2.3.4.2 Diagrama de Clase.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases esta compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** Herencia, Composición, Agregación, Asociación y Uso.

## **Clase:**

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

## **Relaciones entre Clases:**

Ahora ya definido el concepto de Clase, es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes). Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- uno o muchos: **1..\*** (**1..n**)
- 0 o muchos: **0..\*** (**0..n**)
- número fijo: **m** (**m denota el número**).

## **Herencia (Especialización/Generalización):**

Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected).

## **Agregación:**

Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere

componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

- **Por Valor:** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada **Composición** (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").
- **Por Referencia:** Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada **Agregación** (el objeto base utiliza al incluido para su funcionamiento).

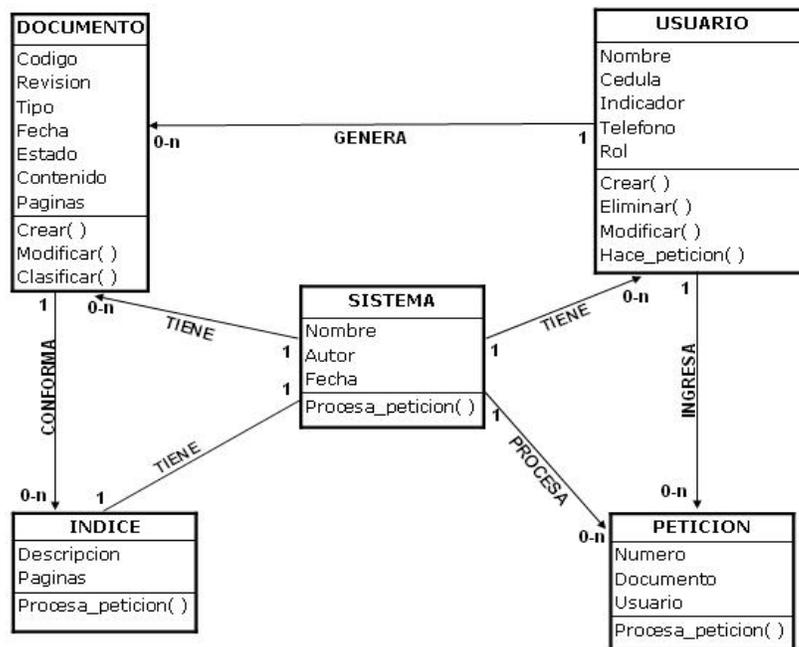
**Asociación:** 

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

**Dependencia o Instanciación (uso):** 

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra, como por ejemplo una aplicación gráfica que instancia una ventana (la creación del Objeto Ventana esta condicionado a la instanciación proveniente desde el objeto Aplicación):



**Figura 2.4** Ejemplo de un Diagrama de Clases.

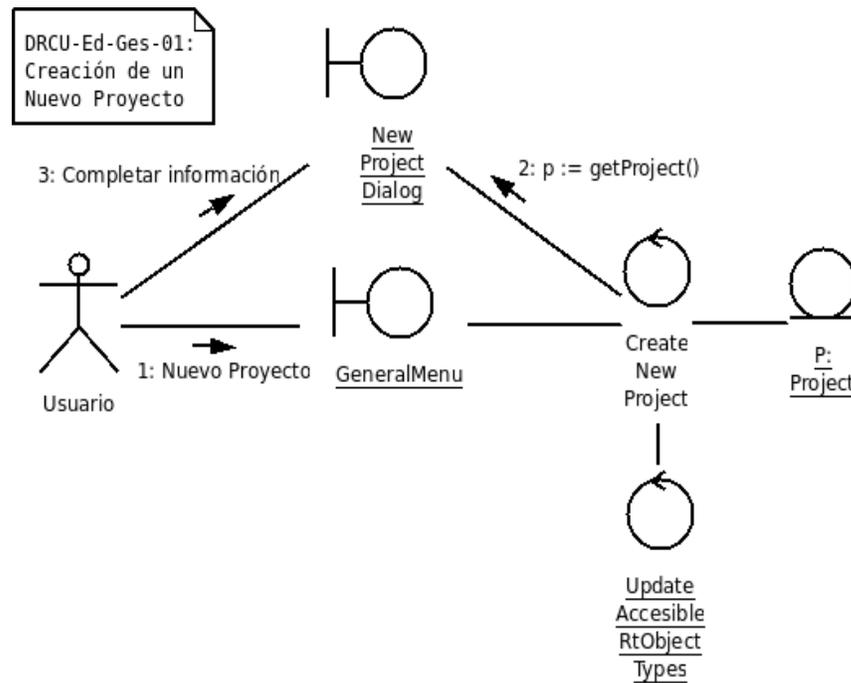
**Fuente:** [http://es.geocities.com/nacarit\\_espaa/fase2/t1.html](http://es.geocities.com/nacarit_espaa/fase2/t1.html), año:2007

### 2.3.4.3 Diagrama de Colaboración.

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia para mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos y los enlaces entre ellos.

Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal - qué pasa primero, qué pasa después -, los clientes entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis. Por tanto los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los

enlaces de unos objetos con otros. Este tipo de diagramas se utilizan frecuentemente en la fase de diseño, véase figura 2.5 donde se muestra un ejemplo.



**Figura 2.5** Ejemplo de un Diagrama de Colaboración.  
**Fuente:** <http://rtlabnet.wikidot.com/doc:diseño:rcu:editor>, año:2007.

#### 2.3.4.4 Diagrama de Secuencia.

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos, véase figura 2.6.

## **Línea de Vida**

Una línea de vida representa un participante individual en un diagrama de secuencia. Una línea de vida usualmente tiene un rectángulo que contiene el nombre del objeto. Si el nombre es self entonces eso indica que la línea de vida representa el clasificador que posee el diagrama de secuencia.

Algunas veces un diagrama de secuencia tendrá una línea de vida con un símbolo del elemento actor en la parte superior. Este usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso. Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida.

## **Mensajes**

Los mensajes se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales.

## **Ocurrencia de ejecución**

Un rectángulo fino a lo largo de la línea de vida denota la ocurrencia de ejecución o activación de un foco de control.

## **Mensaje Self**

Un mensaje self puede representar una llamada recursiva de una operación, o un método llamando a otro método perteneciente al mismo objeto. Este se muestra como cuando crea un foco de control anidado en la ocurrencia de ejecución de la línea de vida.

## **Mensajes perdidos y encontrados**

Los mensajes perdidos son aquellos que han sido enviados pero que no han llegado al destino esperado, o que han llegado a un destino que no se muestra en el diagrama actual. Los mensajes encontrados son aquellos que llegan de un remitente no conocido, o de un remitente no conocido en el diagrama actual. Ellos se denotan yendo o llegando desde un elemento de punto final.

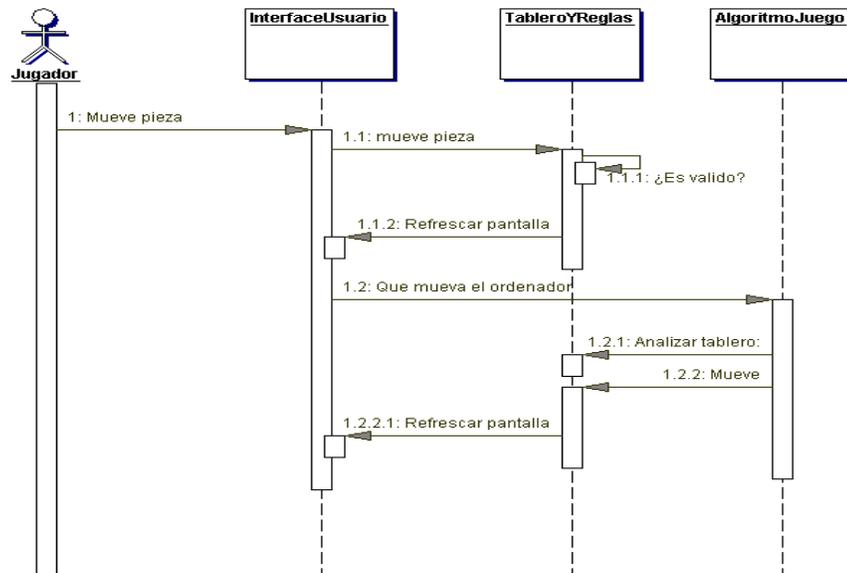
## **Inicio y final de línea de vida**

Una línea de vida se puede crear o destruir durante la escala de tiempo representada por un diagrama de secuencia. En el último caso, la línea de vida se termina por un símbolo de detención, representado como una cruz. En el primer caso, el símbolo al inicio de la línea de vida se muestra en un nivel más bajo de la página que el símbolo del objeto que causó la creación.

## **Restricciones de tiempo y duración**

En forma predeterminada, un mensaje se muestra como una línea horizontal. Ya que la línea de vida representa el pasaje de tiempo hacia abajo, cuando se modela un sistema en tiempo real, o incluso un proceso de negocios en tiempo límite, puede ser importante considerar el tiempo que toma realizar las acciones.

Al configurar una restricción de duración para un mensaje, el mensaje se mostrará como una línea inclinada.



**Figura 2.6** Ejemplo de un Diagrama de Secuencia.

**Fuente:** [http://www.chuidiang.com/ood/metodologia/diagrama\\_secuencia.php](http://www.chuidiang.com/ood/metodologia/diagrama_secuencia.php), año:2007.

### 2.3.4.5 Extensión WAE del UML.

Una de las características más relevantes de la notación UML es su capacidad para absorber nueva semántica sin romper su lógica interna. La necesidad de implementar complejas arquitecturas con múltiples capas y una gran dispersión geográfica de nodos, ha supuesto todo un reto al abordar su modelado y especificación. Jim Conallen ha desarrollado desde 1998 una extensión de la notación UML denominada WAE “Web Application Extensión” que permite rentabilizar toda la gramática interna de UML para modelar aplicaciones con elementos específicos de la arquitectura de un entorno WEB. [15].

La tabla 2.1, muestra los estereotipos que se utilizan en WAE. Una página cliente es una unidad de código HTML que es distribuida por el servidor Web a sus clientes, que son los navegadores Web, los navegadores interpretan el código y

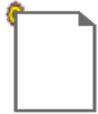
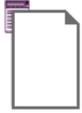
presentan la información que contiene al usuario. Para obtener una página cliente, el navegador envía al servidor Web la dirección *URL (Uniform Resource Locator)* de la página.

Una página servidora, por su parte, es una secuencia de comandos en algún lenguaje de programación como ser PHP, ASP, JSP, PERL, etc que son procesados por el mismo servidor.

Al igual que las páginas cliente la página servidora tienen una URL que es enviada por el navegador al servidor Web, pero éste, en lugar de distribuir la página, ejecuta las instrucciones que contiene. Estas instrucciones pueden ser, por ejemplo, para consultar una base de datos y extraer de ella información que interesa al usuario del navegador, y terminan generalmente con la construcción de una página cliente que contiene la información obtenida, y que es enviada entonces por el servidor Web al navegador para que se la presente al usuario.

Desde el punto de vista de éste, el servidor Web le envía la página cliente construida en forma dinámica, en respuesta a la URL de la página servidora enviada previamente.

**Tabla 2.1** Estereotipos Utilizados en la Notación WAE.

Estereotipos para las Clases	
Estereotipo	Descripción
 Server Page	Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor
 Client Page	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, scripts del lado del cliente, islas de datos, etc.
 Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page). Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.
 ClientScript Object	Es una colección de scripts del lado del cliente que existe como un archivo separado y que son incluidos mediante una petición independiente por parte del navegador.
Estereotipos para las Relaciones entre las Clases	
Link	Representa un apuntador desde una “client page” hacia una “client page” o “server page”. Corresponde directamente con una etiqueta <a> (ancla) de HTML
Submit	Esta relación siempre se da entre una “form” y una “server page”, por supuesto, la “server page” procesa los datos que la “form” le envía (submits)
Build	Sirve para identificar cuales “server page” son responsables de de la creación de una “client page”. Una “server page” puede crear varias “client page”, pero una “client page” sólo puede ser creada por una sola “server page”. Esta relación siempre es unidireccional
Redirect	Esta es también una relación unidireccional que indica que una página Web redirige hacia otra. En caso de que la página origen sea una “client page” esta asociación corresponderá con la “META” etiqueta y valor HTTP-EQUIV de “Refresh”*.

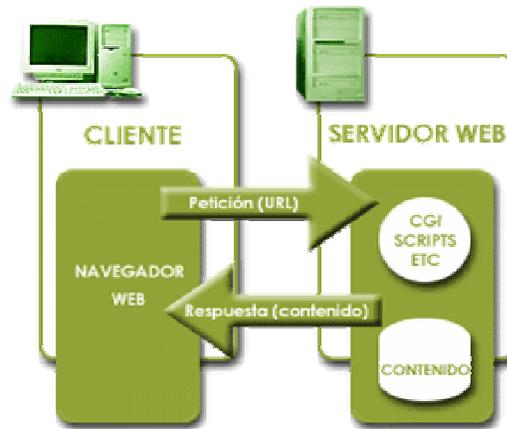
## 2.4 Modelo cliente-servidor.

La tecnología denominada Cliente/Servidor es utilizada en todas las aplicaciones de Internet/intranet, un servidor es un ordenador remoto en algún lugar de la red que proporciona información según la petición *véase* figura 2.7. Un cliente funciona en su computador local que se comunica con el servidor remoto y pide a éste información. Un servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente [23].

Los sistemas Cliente/Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes, entre otros, existen los siguientes:

- **Servidor de Impresión:** mediante el cual los usuarios imprimen remotamente.
- **Servidor de Archivos:** con el cual los clientes comparten y/o almacenan archivos, a este servicio se le conoce como Servidor FTP.
- **Servidor de Bases de Datos:** donde existe uno o varios sistemas de Base de Datos.
- **Servidor de Nombres:** el cual convierte las direcciones IP (Protocolo Internet) en nombres y viceversa también se conoce como Servidor DNS.
- **Servidor Web:** el cual sirve páginas Web.

- **Servidor de Correo:** el cual permite enviar y/o recibir correo electrónicos mediante un cliente de correo electrónico.



**Figura 2.7** Modelo Cliente-Servidor en un entorno Web.

**Fuente:**<http://www.ingeniosarrieta.com/category/java-sistempres>, año:2008

## 2.5 Programación orientada a objetos.

El paradigma OO se basa en el concepto de objeto, un objeto es aquello que tiene estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos). La estructura y comportamiento de objetos similares están definidos en una clase común, los términos instancia y objeto son intercambiables.

Una clase es un conjunto de objetos que comparten una estructura y comportamiento común, la diferencia entre un objeto y una clase es que un objeto es una entidad concreta que existe en tiempo y espacio, mientras que una clase representa una abstracción, la "esencia" de un objeto, tal como son, de aquí que un objeto no es una clase, sin embargo, una clase puede ser un objeto [21].

En el enfoque OO las propiedades del objeto son claves, los principios del modelo OO son:

- **Abstracción:** Es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros.
- **Encapsulación:** Es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.
- **Modularidad:** Es la propiedad de un sistema que ha sido descompuesto en un conjunto de módulos coherentes e independientes.
- **Jerarquía o herencia:** Es el orden de las abstracciones organizado por niveles.
- **Tipificación:** Es la definición precisa de un objeto de tal forma que objetos de diferentes tipos no puedan ser intercambiados o cuando mucho, puedan intercambiarse de manera muy restringida.
- **Concurrencia:** Es la propiedad que distingue un objeto que está activo de uno que no lo está.
- **Persistencia:** Es la propiedad de un objeto a través de la cual su existencia trasciende el tiempo (es decir, el objeto continua existiendo después de que su creador ha dejado de existir) y/o el espacio es decir, la localización del objeto se mueve del espacio de dirección en que fue creado. Si un modelo que se dice OO no contiene alguno de los primeros cuatro elementos, entonces no es OO [9] .

### 2.5.1 Los beneficios del enfoque OO.

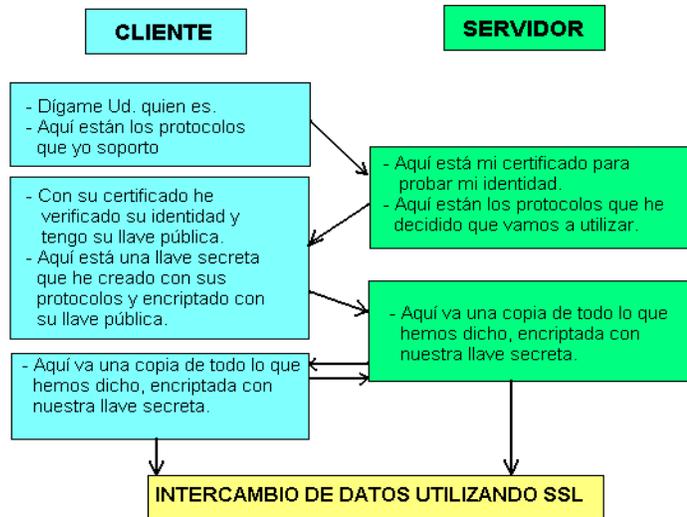
- El uso del modelo OO ayuda a explotar el poder expresivo de todos los lenguajes de programación basados en objetos y los orientados a objetos, como Smalltalk, Object Pascal, C++, CLOS, Ada, y Java .

- El uso del modelo OO alienta el rechazo no solo del software, sino de diseños completos.
- Produce sistemas que están contruidos en formas intermedias estables y por ello son más resistentes al cambio en especificaciones y tecnología [3].

## **2.6 Servidor Web seguro.**

Existen ocasiones en las que se hace necesario recibir/enviar información sensible a un Servidor Web, es por ello que se hace imprescindible el contar con un mecanismo que dé fe de si, un servidor seguro es en quien se cree y se puede confiar en él a la hora de transmitirle la información. La forma como se hace es mediante las Autoridades de Certificación (AC), conocidas informalmente como notarios electrónicos, encargadas de autenticar a los participantes en transacciones y comunicaciones a través de la red. Su función es emitir certificados a los usuarios, de manera que se pueda estar seguro de que el interlocutor (cliente o servidor) es quien pretende ser, garantizando así la seguridad de las transacciones, *véase* figura 2.8.

El certificado de seguridad se concede a una entidad después de comprobar una serie de referencias, para asegurar la identidad del receptor de los datos cifrados. Se construye a partir de la clave pública del servidor solicitante, junto con algunos datos básicos del mismo y es firmado por la autoridad de certificación correspondiente con su clave privada [1]. En la práctica, se sabe que el servidor es seguro porque en el navegador de Internet se ve una llave o un candado cerrado en la barra de estado de éste.



**Figura 2.8** Intercambio de Información entre un Navegador Web (cliente) y un Servidor Web Seguro.

**Fuente:** <http://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/ssl.html>, año 2007.

## 2.7 Páginas Web.

Una página Web estática incluye un contenido que se muestra del mismo modo cada vez que se solicita desde un navegador. Un ejemplo de una página Web estática es una página de servicio al cliente que contiene información de contacto como por ejemplo: los números de teléfono, los números de fax y las direcciones de correo electrónico que no suelen cambiar con frecuencia [4].

Una página Web estática se crea utilizando sólo *HTML* lenguaje que interpretan los navegadores Web. Una página Web estática contiene además del código HTML, texto, así como otros elementos apropiados para la página como imágenes y animación, pero no utilizan la información almacenada en Base de Datos.

El contenido de una página Web dinámica en cambio se genera cuando el usuario solicita la página. Generalmente el contenido se extrae de una Base de Datos, lo que permite presentar la información más actual sin volver a codificar la página Web [13]. Una página Web dinámica actúa como una plantilla: contiene código para recuperar la información solicitada y dar formato a la salida.

## **2.8 El lenguaje SQL.**

El SQL (Structured Query Language) o Lenguaje de Consultas Estructurado, es el lenguaje que permite la comunicación con el Sistema Gestor de Bases de Datos. Es un lenguaje unificado, y lo utilizan todo tipo de usuarios, desde el administrador de la Base de Datos, hasta el usuario final. El SQL es un lenguaje no procedimental esto quiere decir que el usuario especifica *Qué* quiere, no *Cómo* ni *Dónde* conseguirlo. El SQL es relacionalmente completo esto permite la realización de cualquier consulta de datos. [6]

Las sentencias SQL pertenecen a dos categorías principales: Lenguaje de Definición de Datos (DDL) y Lenguaje de Manipulación de Datos (DML). Estos dos lenguajes no son lenguajes en sí mismos, sino que es una forma de clasificar las sentencias de lenguaje SQL en función de su cometido. La diferencia principal reside en que el DDL crea objetos en la base de datos y sus efectos se pueden ver en el diccionario de la base de datos, mientras que el DML es el que permite consultar, insertar, modificar y eliminar la información almacenada en los objetos de la base de datos.

El lenguaje SQL está basado en el cálculo relacional de tuplas. Como resultado, toda consulta formulada utilizando el cálculo relacional de tuplas (o su equivalente, el álgebra relacional) se puede formular también utilizando SQL. Hay, sin embargo, capacidades que van más allá del cálculo o del álgebra relacional. A continuación se

muestra una lista de algunas características proporcionadas por SQL que no forman parte del álgebra y del cálculo relacionales [6]:

- Comandos para inserción, borrado o modificación de datos.
- Capacidades aritméticas: En SQL es posible incluir operaciones aritméticas así como comparaciones, por ejemplo  $A < B + 3$ . Nótese que ni + ni otros operadores aritméticos aparecían en el álgebra relacional ni en cálculo relacional.
- Asignación y comandos de impresión: es posible imprimir una relación construida por una consulta y asignar una relación calculada a un nombre de relación.
- Funciones agregadas: Operaciones tales como *promedio (average)*, *suma (sum)*, *máximo (max)*, etc. se pueden aplicar a las columnas de una relación para obtener una cantidad única.

## 2.9 Las bases de datos.

Una base de datos es un conjunto de datos estructurados, almacenados en algún soporte de almacenamiento de datos y se puede acceder a ella desde uno o varios programas. Antes de diseñar una base de datos se debe establecer un proceso partiendo del mundo real, de manera que sea posible plasmar éste mediante una serie de datos. La imagen que se obtiene del mundo real se denomina modelo conceptual y consiste en una serie de elementos que definen perfectamente lo que se quiere plasmar del mundo real en la base de datos [12].

El Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas, procedimientos y lenguajes que proporcionan a los usuarios las herramientas necesarias para operar con una base de datos. Por tanto, el SGBD actúa como un intermediario entre los usuarios y los datos. Debe cumplir una serie de funciones

como descripción de los datos, de manera que debe permitir definir los registros, sus campos, sus relaciones de autorización, etc. Debe manipular los datos permitiendo a los usuarios insertar, suprimir, modificar y consultar datos de la base de datos y por último, debe permitir usar la base de datos, dando un interfaz adecuado a cada tipo de usuario.

### **2.9.1 El modelo entidad-relación.**

Es una técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del mundo real sobre un soporte informático. Sus características fundamentales son [6]:

2. Reflejan tan sólo la existencia de los datos sin expresar lo que se hace con ellos.
3. Es independiente de las bases de datos y de los sistemas operativos.
4. Incluye todos los datos que se estudian sin tener en cuenta las aplicaciones que se van a tratar.

Las entidades se representan como rectángulos, los atributos como elipses y las relaciones como rombos.

#### **Conceptos fundamentales.**

**Entidad:** Una entidad es un objeto concreto o abstracto que presenta interés para el sistema y sobre el que se recoge información la cual va a ser representada en un sistema de base de datos. La mayoría de las entidades modelan objetos o eventos del mundo real, por ejemplo, clientes, productos o llamadas de pedidos.

**Atributo:** Es una unidad básica e indivisible de información acerca de una entidad o una relación y sirve para identificar y describir a las mismas. Por ejemplo, si se va a modelar un evento como una llamada al servicio de asistencia, probablemente se querrá saber quién era el cliente, quién hizo la llamada y cuándo, así como si se resolvió o no el problema. La determinación de los atributos que hay que incluir en el modelo es un problema semántico (de significado). Se deben tomar decisiones basadas en el significado de los datos y en cómo se utilizarán.

**Dominio:** Un dominio es el conjunto de valores que puede tomar cada uno de los atributos.

**Tabla:** Organización de los datos en forma de filas y columnas. Cada fila se llama tupla, y cada columna dentro de una tupla corresponde al valor de un atributo para esa tupla.

**Relación:** Asociación entre entidades. Por ejemplo, un "alumno" "tiene" una "asignatura".

**Tabla relacional:** Es una tabla que debe cumplir las siguientes características:

- Cada fila debe ser única.
- Cada columna debe ser única.
- Los valores de las columnas deben pertenecer al dominio de cada atributo.
- Debe tener un solo tipo de fila, cuyo formato está definido por el esquema de la tabla o relación .
- El valor de la columna para cada fila debe ser único.

**Clave candidata:** Atributo o atributos que pueden distinguir de forma unívoca una tupla dentro de una tabla. Puede haber varias claves candidatas para distinguir

una misma entidad. Se elegirá como clave candidata aquel atributo que posea un dominio en el que se tenga valores únicos. Si esto no es posible, entonces se usa como clave candidata la combinación de varios atributos, de manera que esta combinación sí sea única.

**Clave principal:** Aquella de las claves candidatas que es designada para distinguir de forma unívoca una tupla dentro de una tabla.

**Clave ajena:** Se trata de un atributo que es clave principal en otra tabla.

**Vista:** Una vista es una tabla ficticia cuya definición y tuplas se obtiene a partir de una o más tablas base. Sus características son:

2. Sus columnas se obtienen a partir de varias tablas base.
3. Pueden estar definidas a partir de otras vistas.
4. Sus datos se obtienen como resultado de una consulta a la base de datos.
5. Se puede almacenar su estructura.

Se trata de una tabla virtual que no existe como tabla en el disco.

**Inconsistencia:** Se da cuando se encuentra un valor en una clave ajena no existente en la entidad donde ésta sea clave principal.

Asociaciones entre entidades

**Asociaciones uno-a-uno:** Si es cierto que cualquier ejemplar de la entidad X se puede asociar con tan sólo un ejemplar de la entidad Y, entonces se dice que la asociación es uno-a-uno. Cuando se elige una asociación uno-a-uno se debe asegurar de que se mantenga la asociación en todo momento.

**Asociaciones uno-a-muchos:** Es el tipo de asociación más común, donde un

solo ejemplar de una entidad se puede asociar con cero, uno o muchos ejemplares de otra entidad. Por ejemplo, una persona puede tener varios números de teléfono.

**Asociaciones muchos-a-muchos:** Los clientes compran en muchas tiendas, una tienda tiene muchos clientes. Como este tipo de relaciones no se puede modelar directamente en una base de datos relacional, se modela usando una tabla intermedia que tenga una asociación uno-a-muchos con cada uno de los participantes originales. Por ejemplo, un pedido puede tener muchos tipos de confección, y un tipo de confección puede aparecer en varios pedidos.

## **2.9.2 Normalización.**

La normalización se encarga de obtener los datos agrupados en distintas tablas siguiendo una serie de pasos, de tal manera que los datos obtenidos tienen una estructura óptima para su implementación, gestión y explotación desde distintas aplicaciones futuras. Una de las ventajas principales que se obtiene al realizar la normalización es que la información no estará duplicada innecesariamente dentro de las estructuras: habrá mínima redundancia [6].

### **2.9.2.1 Dependencia funcional.**

Se dice que un atributo B depende funcionalmente de A ( $A \rightarrow B$ ) si cada valor de A se corresponde con un único valor de B o, visto de otra manera, si dado A se puede obtener B. Un caso típico podría ser DNI  $\rightarrow$  Nombre, pues dado un DNI se puede obtener el nombre de la persona con ese DNI. Existen reglas que se pueden realizar entre atributos para poder obtener dependencias funcionales adicionalmente. Supóngase que T es una tabla relacional y X, Y, Z son subconjuntos de atributos de la tabla T.

**Reflexividad:** Si los valores de un subconjunto de atributos Y están incluidos dentro de un subconjunto de atributos X, se dice que X depende funcionalmente de Y ( $Y \rightarrow X$ ).

**Aumentación:** Si un subconjunto X depende funcionalmente de otro Y, dicha dependencia se mantendrá aunque se añada otro atributo a los dos subconjuntos ( $X \rightarrow Y$  entonces  $X.a \rightarrow Y.a$ ).

**Transitividad:** Si Y depende funcionalmente de X y Z depende funcionalmente de Y, entonces Z depende funcionalmente de X ( $X \rightarrow Y$  e  $Y \rightarrow Z$  entonces  $X \rightarrow Z$ ). Por ejemplo,  $DNI \rightarrow NOMBRE$  y  $NOMBRE \rightarrow DIRECCIÓN$ , luego  $DNI \rightarrow DIRECCIÓN$ .

**Dependencia funcional total:** Un atributo Y tiene una dependencia funcional total con otro atributo X si tiene una dependencia funcional con X y no depende funcionalmente de ningún subconjunto de X. Por ejemplo, supóngase que una empresa tiene empleados y que una persona puede ser empleado de varias empresas. Según esto, se podría decir que  $DNI.EMPRESA \rightarrow NOMBRE$ , pero esta dependencia no es total porque también es cierto que  $DNI \rightarrow NOMBRE$ . Sin embargo, no se puede identificar el sueldo de un empleado sin saber a qué empresa pertenece, por lo tanto,  $DNI.EMPRESA \rightarrow SUELDO$  sí es una dependencia funcional total.

### 2.9.2.2 Primera Forma Normal (1FN).

Se dice que una tabla está en primera forma normal si todos los valores que componen a sus tuplas son atómicos: un atributo no puede tener más de un valor. Para normalizar una tabla que no esté en 1FN han de seguirse los siguientes pasos:

- Se localizan los atributos correspondientes a la clave principal

- Se realiza una proyección sobre la tabla y así se descompone en varias, de manera que se hace la proyección de la clave con los atributos que tengan los valores únicos.

Por ejemplo, la figura 2.9 muestra una tabla que no está en 1FN:

IdPedido	IdCliente	IdVendedor	Fecha_Pedido	Fecha_Entrega	Productos
10548	Wilman	5	25/02/02	26/02/02	3 chorizos, 2 ajos, 1 cebolla
10549	Toldar	6	01/04/02	05/04/02	4 botes, 2 fresas
10550	Canair	3	05/06/02	06/06/02	1 revista, 2 zumos, 1 pan
10551	Suprem	5	07/07/02	07/07/02	2 habas, 1 pan, 2 quesos
10552	Wilman	1	08/07/02	09/07/02	1 bote, 2 panes

**Figura 2.9** Tabla en Primera Forma Normal.

**Fuente:** <http://www.formauri.es/arrobamas/Cursos/index.php?apdo=05&curso=51&cap=4>, año:2009.

### 2.9.2.3 Segunda Forma Normal (2FN).

Esta forma normal se considerará únicamente cuando la clave principal sea compuesta, si no (la clave principal está formada por un único atributo) la tabla estaría en segunda forma normal.

Se dice que una tabla está en segunda forma normal si se cumplen las siguientes condiciones:

- Está en 1FN.
- Todo atributo secundario (los que no pertenecen a la clave principal) tiene una dependencia funcional total de la clave completa y no de una parte de ella.

Para convertir una tabla que no esté en 2FN a 2FN se creará una tabla con la clave y todas sus dependencias funcionales totales y otra tabla con la parte de la clave

que tiene dependencias con los atributos secundarios. Por ejemplo, la figura 2.10 muestra un tabla que no está en 2FN:

NombreProducto	NombreProveedor	Categoria	TeléfonoProveedor
Diarios	Exotic Kiosk	Prensa	968582222
Revistas	Exotic Kiosk	Prensa	968582222
Habas	Tnj export	Alimentacion	975869999
Botes	Tnj export	Bebidas	975869999

**Figura 2.10** Tabla que no esta en 2FN.

**Fuente:** <http://www.formauri.es/arrobamasmamas/Cursos/index.php?apdo=05&curso=51&cap=4, año:2009>.

Ya que el campo "TeléfonoProveedor" no es dependiente de la clave candidata {"NombreProducto, "NombreProveedor"} sino únicamente de "NombreProveedor". Se trata de no representar dos entidades distintas en una sola tabla.

En este ejemplo, se reorganizan los datos de la siguiente manera:

Tabla Productos:

IdProducto	NombreProduct	Categoria	IdProveedor
1	Diarios	Prensa	1
2	Revistas	Prensa	1
3	Habas	Alimentación	2
4	Botes	Alimentación	2

**Figura 2.11** Tabla Productos

**Fuente:** <http://www.formauri.es/arrobamasmamas/Cursos/index.php?apdo=05&curso=51&cap=4, año:2009>.

Tabla Proveedores:

IdProveedor	NombreProveedor	TeléfonoProveedor
1	Exotic Kiosk	968582222
2	Tnj Export	975869999

**Figura 2.12** Tabla Proveedores.

**Fuente:** <http://www.formauri.es/arrobamas/Cursos/index.php?apdo=05&curso=51&cap=4>, año:2009.

### 2.9.2.4 Tercera Forma Normal (3FN).

Una tabla está en 3FN si:

- Está en 2FN
- No existen atributos no primarios (no pertenecen a la clave) que son transitivamente dependientes de cada posible clave de la tabla, o lo que es lo mismo, un atributo secundario sólo puede ser conocido a través de la clave principal o claves secundarias de la tabla y no por medio de otro atributo no primario.

Para convertir una tabla a 3FN se realizará una proyección de la clave a los elementos que no tengan dependencia funcional transitiva y otra tabla con una nueva clave a los elementos que anteriormente tenían esta dependencia.

Por ejemplo, la siguiente tabla no está en 3FN:

Tabla Atletas:

Licencia	Nombre	Club	Edad	Categoría
189585	Pepe	Alcoy	12	Junior
205888	Tomas	Jaca	10	Cadete
748523	Andres	Almansa	9	Alevín

**Figura 2.13** Tabla Atletas.

**Fuente:** <http://www.formauri.es/arrobamas/Cursos/index.php?apdo=05&curso=51&cap=4>, año:2009.

Ya que, dado un número de licencia, se puede obtener la edad del inscrito, y dada la edad del inscrito, se puede averiguar la categoría a la que pertenece: teniendo entonces una dependencia funcional transitiva. Evidentemente, dado el número de licencia se puede averiguar la categoría pero lo importante aquí es que la categoría depende de un atributo que no forma parte de la clave. Para normalizar, se descompone la tabla en las siguientes:

Licencia	Nombre	Club	Edad
189585	Pepe	Alcoy	12
205888	Tomas	Jaca	10
748523	Andres	Almansa	9

**Figura 2.14** Tabla Atletas parte 1.

**Fuente:** <http://www.formauri.es/arrobamas/Cursos/index.php?apdo=05&curso=51&cap=4>, año:2009.

Edad	Categoría
9	Alevín
10	Cadete
12	Junior

**Figura 2.15** Tabla Atletas parte2.

**Fuente:** <http://www.formauri.es/arrobamas/Cursos/index.php?apdo=05&curso=51&cap=4>, año:2009.

## 2.10 Lenguaje de programación PHP.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+ [5].

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

### **2.10.1 Ventajas del lenguaje PHP.**

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial ([2]), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde php5).

### **2.10.2 Desventajas del Lenguaje PHP.**

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aún estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de

programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.

Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes (ver más abajo Frameworks en PHP).

### **2.11 Common Gateway Interface (CGI).**

El CGI (Por sus siglas en inglés "Common Gateway Interface") cambió la forma de manipular información en el Web. En sí, es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos Web que por medio del HTML se presentan de forma estática [5].

El CGI es utilizado comúnmente para contadores, bases de datos, motores de búsqueda, formularios, generadores de email automático, foros de discusión, chats, comercio electrónico, rotadores y mapas de imágenes, juegos en línea y otros. Esta tecnología tiene la ventaja de correr en el servidor cuando el usuario lo solicita por lo que es dependiente del servidor y no de la computadora del usuario.

De acuerdo a la traducción de la NCSA: "Un documento HTML es estático, lo que significa que existe en un estado constante; es un archivo de texto que no cambia. Un script CGI por otro lado, es ejecutado en tiempo real, lo que permite que regrese información dinámica". Por ejemplo, si se desea conectar las bases de datos de Unix al World Wide Web para permitir que las personas de todo el mundo la manipulen básicamente, lo que se debe hacer es crear un script CGI que será ejecutado por el servidor para transmitir información al motor de la base de datos, recibir los

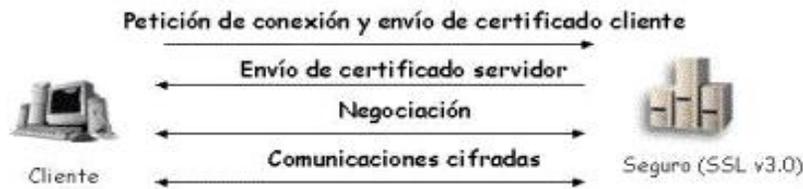
resultados y mostrárselos al cliente. Los programas que maneja el CGI pueden estar compilados en diferentes lenguajes de programación.

El más popular para el desarrollo de contenidos Web es el lenguaje Perl de distribución gratuita, aunque también podemos mencionar: C, C++ y Java. El funcionamiento de esta tecnología es muy sencillo. Los scripts residen en el servidor, donde son llamados, ejecutados y regresan información de vuelta al usuario.

### **2.12 Secure Socket Layer (SSL).**

El protocolo SSL es un sistema diseñado y propuesto por *Netscape Communications Corporation*. Este se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones (véase figura 2.16). El MD5 se usa como algoritmo de hash. [1].

El SSL proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y opcionalmente autenticación de cliente para conexiones TCP/IP. Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.



**Figura 2.16** Transacción usando Cifrado SSL.

**Fuente:** <http://www.enelnombredetux.com/project.php?project=pgp>, año:2008.

### 2.12.1 El Protocolo SSL Handshake.

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases:

- La fase “hola”, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.
- La fase de intercambio de claves, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.
- La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.
- La fase de verificación del servidor, presente sólo cuando se usa RSA como algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.
- La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).
- Por último, la fase de fin, que indica que ya se puede comenzar la sesión

segura.

### 2.12.2 Protocolo SSL Record.

El Protocolo SSL Record especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

- MAC-DATA, el código de autenticación del mensaje.
- ACTUAL-DATA, los datos de aplicación a transmitir.
- PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

### 2.12.3 Cómo saber si una Conexión se está Realizando Mediante SSL.

Los navegadores Web disponen de un icono que lo indica, generalmente un candado en la parte inferior de la ventana, véase figura 2.17. Si el candado está abierto se trata de una conexión normal, y si está cerrado de una conexión segura. Si hace “doble click” sobre el candado cerrado aparecerá el Certificado Digital del Servidor Web Seguro.



**Figura 2.17** Indicación de una Conexión Segura en Navegadores Web.

**Fuente:**<http://www.microsoft.com/spain/protect/yourself/phishing/spoof.msp>, año:2009

### 2.12.4 OpenSSL.

El software *OpenSSL* es un proyecto de software desarrollado por lo miembros de la comunidad Open Source (código abierto). Es un robusto juego de herramientas

que le ayudan a su sistema a implementar el *Secure Sockets Layer* (SSL), así como otros protocolos relacionados con la seguridad, tales como el *Transport Layer Security* (TLS). También incluye una librería de criptografía [1].

### **2.13 Sistema operativo gnu/linux**

GNU/Linux es un Sistema Operativo, es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium, Pentium Pro y Pentium II, así como los clones AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0 [13].

Como sistema operativo, GNU/Linux es muy eficiente y tiene un excelente diseño. 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.

GNU/Linux es una muy buena alternativa frente a los demás sistemas operativos. Más allá de las ventajas evidentes de costo, ofrece algunas características muy notables.

En comparación con las otras versiones de Unix para PC, la velocidad y confiabilidad de GNU/Linux son muy superiores. También está en ventaja sobre la disponibilidad de aplicaciones, ya que no hay mucha difusión de estos otros Unixes (como Solaris, XENIX o SCO) entre los usuarios de PC por sus altos costos.

Comparado con sistemas operativos como Microsoft Windows, GNU/Linux también sale ganando. Los bajos requisitos de hardware permiten hacer un sistema potente y útil de aquel 486 que algunos guardan en un armario. Esta misma característica permite aprovechar al máximo las capacidades de las computadoras más modernas. Es poco práctico tener una PC con 16 Mb de RAM y ponerle un sistema operativo que ocupa 13 (que es lo que reporta sobre Windows 95 el System Information de Symantec).

No solo es superior respecto a el sistema de multitarea y de administración de memoria, sino también en la capacidades de networking (conectividad a redes) y de multiusuario (aún comparando con sistemas multiusuario como NT). La única desventaja de GNU/Linux frente a estos sistemas, es la menor disponibilidad de software, pero este problema disminuye con cada nuevo programa que se escribe para el proyecto GNU, y con algunas empresas que están desarrollando software comercial para GNU/Linux.

## **CAPITULO 3**

### **ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS**

El propósito fundamental de esta fase es establecer un modelo de análisis y definición de los requisitos, lo cual permitirá guiar el desarrollo del sistema a diseñar durante el ciclo de vida, para lo cual se seguirá el enfoque en “cascada” de la Ingeniería de Software que establece los siguientes pasos; Análisis, Diseño, Desarrollo, Prueba e Implantación. La ingeniería de Software, como cualquier ingeniería, requiere modelar tanto el problema como sus posibles soluciones.

Los modelos ayudan a determinar qué información es requerida y qué datos son necesarios recoger para obtener esa información, adicionalmente la abstracción y el rigor de los modelos, permite usar los modelos como medios para especificar, analizar y evaluar propiedades del producto.

Para capturar los requisitos de manera eficaz y construir el sistema correctamente se requiere un sólido conocimiento del contexto donde se ubique la problemática, para lo cual es necesario elaborar un modelo de la situación actual que permita conocer el negocio y los objetos del dominio.

Luego plantear como requerimientos los cambios o mejoras que se desean y establecer así las especificaciones de esos cambios, igualmente estudiar la factibilidad de llevar a cabo el proyecto.

### **3.1 Análisis de la situación actual**

Actualmente la organización Misión Sucre es quien administra la data de ingreso, prosecución y egreso a nivel de municipalización de los diferentes programas en aldeas universitarias (espacio donde se procesa y se satisfacen las necesidades de la comunidad por distintas universidades) llega la Universidad Bolivariana de Venezuela, la universidad debe solicitar información a la organización cuando la necesita, por otro lado la data suministrada siempre ha presentado incongruencias, siendo esto un elemento que pone en desventaja a la universidad al momento de tomar decisiones, como son en los procesos de graduación de programas de formación municipalizados y en las estadísticas de la población estudiantil a nivel nacional.

Esta situación ocasionó que en las distintas sedes de la universidad, las unidades de control de estudio manejasen una data paralela, lo que contribuye a una mayor incertidumbre en cuanto al número real de estudiantes en cada programa a nivel de municipalización.

En los actuales momentos la organización Misión Sucre no cuenta con una plataforma que permita dar respuesta a la dinámica de la universidad en su extensión hacia la municipalización, esta situación ya era conocida por parte de las autoridades desde hace algún tiempo, motivado a esto la universidad adquirió un software propietario con un considerable costo y nunca llego a cumplir las expectativas planteadas por la universidad.

En tal sentido se propone el diseño de una plataforma que ofrezca servicios estudiantiles, permita manejar estadísticas, evaluar la prosecución, que tome en cuenta la realidad de la universidad.

### **3.2 Compresión de los procesos del negocio.**

Esta técnica busca presentar el negocio desde su perspectiva de uso mediante un modelo de Casos de Uso, igualmente capturar los tipos de objetos más importantes en el contexto, estos objetos del dominio son los que representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

La plataforma que ofrece la Misión Sucre esta basada en una arquitectura cliente-servidor utilizando tecnología Web, a través de este sistema acceden los estudiantes para revisar sus calificaciones, así mismo los coordinadores de las diferentes aldeas para registrar las calificaciones, la carga académica de los docentes y generar diferentes reportes.

El proceso de ingreso a la Universidad Bolivariana de Venezuela en el área de la municipalización se realiza a través de este sistema, una vez registrado el aspirante se dirige a la aldea correspondiente a su inscripción a formalizar ésta, consignando los recaudos solicitados.

Luego de un periodo donde cursa un trayecto inicial (etapa de nivelación donde se cursan 6 unidades curriculares), el estudiante se inicia en uno de los programas de formación que ofrezca la aldea a través de la Universidad Bolivariana, inicialmente en el sistema de la Misión Sucre se registra al estudiante como alumno del programas de formación que haya seleccionado. Pasado el período académico, el sistema realiza la inscripción de las Unidades Curriculares del semestre siguiente sin que el estudiante participe de ello, esto mismo se hace para los siguientes semestres.

### 3.2.1 Encontrar actores y Casos de Uso que Participan en el proceso de registro de Notas.

Cada trabajador es responsable de un conjunto completo de actividades, por lo tanto, cada vez que un usuario (humano o sistema) interactúa con el sistema en estudio, la instancia correspondiente al actor es quien está desarrollando ese papel. Los actores suelen corresponderse con trabajadores, del mismo modo se pueden representar mediante actores cada sistema externo con el que interactúa el sistema [9].

La búsqueda de actores y Casos de Uso permitirá delimitar el sistema existente de su entorno y de este modo identificar qué actores participan al realizar las actividades involucradas en el proceso de registro de notas.

Para llevar a cabo esta actividad se debe señalar un actor por cada rol de un trabajador como se señaló anteriormente. Posteriormente se establecerá una conexión entre estos y los Caso de Uso que luego se identificaran, con lo cual se tendrá constituido un modelo de Casos de Uso que represente las actividades, pues son éstos los procesos que se desean analizar. A continuación en la tabla 3.1, se muestran los actores que representan a los diferentes tipos usuarios identificados según su rol en el sistema existente, así como las funciones que realizan.

**Tabla 3.1.** Actores del sistema de la Misión Sucre.

<b>ACTOR</b>	<b>FUNCIONES</b>
<b>Administrador del sistema</b>	Es el responsable de controlar y supervisar las acciones y recursos del sistema. Es quien autoriza el acceso a los coordinadores de aldeas.
<b>Coordinador de Aldea</b>	Es el encargado del registro de la carga académica de los Docentes, así como velar por el vaciado de las notas en el sistema de todas las universidades que participan en la Aldea.
<b>Docente o Colaborador</b>	Es quien registra las notas de los estudiantes una vez culminado el periodo académico.

<b>Estudiante</b>	Es quien accede al sistema para inscribir unidades curriculares y consultar notas registradas.
-------------------	--

Una vez identificados los actores, se continuará con la actividad de encontrar los Casos de Uso. Un Caso de Uso especifica una funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores. La técnica que permitirá encontrar Casos de Uso será elegir “nombre” de forma que haga pensar en una secuencia de acciones concretas que añaden valor a un actor, este nombre debe reflejar cuál es el objetivo de la interacción entre el actor y el sistema, siendo esta una manera directa de identificar un conjunto tentativo de Casos de Uso.[9]

Seguidamente se describen los Casos de Uso correspondientes a los procesos observados y existentes dentro del proceso del registro de notas de la Misión Sucre:

- **Caso de uso:** Iniciar sesión.

**Actores participantes:** Administrador, Coordinador de Aldea, y Docente, SGBD.

**Resumen:** Este caso de uso permite a los usuarios Administrador y Coordinador de aldea y Docente iniciar una sesión con la cual el sistema podrá identificarlo, y permitirle usar las diferentes operaciones que le son pertinentes para realizar sus tareas. El SGBD es quien lleva a cabo la consulta del usuario.

- **Caso de uso:** Registro de Docente.

**Actores participantes:** Coordinador de aldea, SGBD.

**Resumen:** Este caso de uso le permite al usuario Coordinador de aldea registrar a un usuario Docente para un período académico. El SGBD realiza las transacciones.

- **Caso de uso:** Asignar cargar académica.

**Actores participantes:** Coordinador de aldea, SGBD.

**Resumen:** Este caso de uso le permite al usuario Coordinador de aldea asignar una carga académica a los Docentes que van a participar en el período académico. El SGBD es quien realiza la transacción.

- **Caso de uso:** Consultar carga académica.

**Actores participantes:** Coordinador de aldea, Docente, SGBD.

**Resumen:** Este caso de uso permite que el Docente o el Coordinador de aldea puedan consultar la carga académica asignada a un Docente. El SGBD es quien registra las transacciones.

- **Caso de uso:** Actualizar carga académica.

**Actores participante:** Coordinador de aldea, SGBD.

**Resumen:** Este Caso de Uso permite que el Coordinador de aldea pueda modificar la carga académica asignada a un Docente. El SGBD es quien realiza las diferentes transacciones.

- **Caso de uso:** Consultar notas.

**Actores participante:** Estudiante, SGBD.

**Resumen:** El caso de uso Consultar notas, permite al usuario Estudiante una vez finalizado el periodo académico consultar las notas registradas por el Docente. El SGBD es quien realiza la transacción.

Después de haber realizado la identificación de los actores basados en el rol de cada “trabajador” y haber realizado una breve descripción de los Casos de Uso que dan soporte al proceso de registro de notas en el sistema existente, es necesario ver las relaciones entre éstos.

La Figura 3.1 representa un modelo del negocio a través de un diagrama de Casos de Uso, el cual muestra las interacciones entre los actores y Casos de Uso. Este modelo representa solo una parte del sistema actual donde se desea analizar los

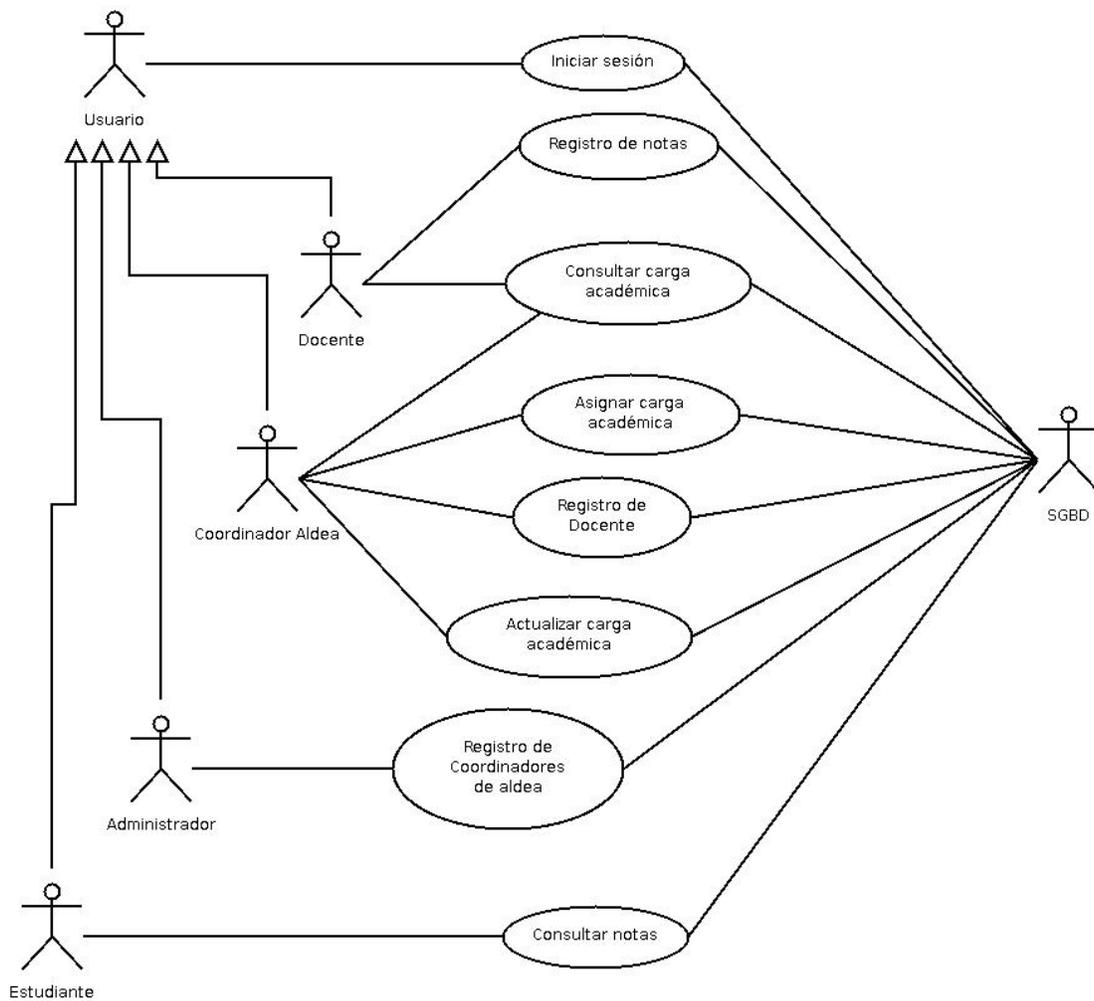
procesos involucrados en la problemática a fin de comprender el funcionamiento del negocio que lo rodea. En lo sucesivo el modelo del negocio representado en la figura 3.1, servirá como punto de partida para establecer las actividades necesarias que transformen los requisitos del cliente en un conjunto de requerimientos de software, lo que a su vez permitirá el desarrollo de la arquitectura del software a diseñar.

### Contexto actual del sistema

La Universidad Bolivariana de Venezuela está dividida estratégicamente en “Ejes”, éstos a su vez están bajo la responsabilidad en cuanto a control de estudios de un coordinador, estos ejes pueden contener uno o varios estados de Venezuela, en la tabla 3.2 se muestran los ejes y sus estados asociados.

**Tabla 3.2.** Distribución de la Universidad Bolivariana de Venezuela por ejes.

EJE UBV	ESTADO DE VENEZUELA
ARAGUA	ARAGUA, CARABOBO, COJEDES y GUÁRICO.
BARINAS	BARINAS, PORTUGUESA
BOLÍVAR	BOLÍVAR
CARACAS	AMAZONAS, APURE, DISTRITO CAPITAL, MIRANDA y VARGAS.
FALCÓN	FALCÓN, LARA y YARACUY
MATURÍN	ANZOÁTEGUI, DELTA AMACURO, MONAGAS NUEVA ESPARTA y SUCRE
TACHIRA	TACHIRA
ZULIA	MÉRIDA, TRUJILLO y ZULIA



**Figura 3.1.** Modelo del Negocio del el proceso de registro de notas en el sistema actual.  
**Fuente:** elaboración propia, año:2009.

Una vez elaborado un Modelo del Negocio mostrando así los procesos del dominio, ahora se realizará una descripción de cómo se relacionan los Casos de Uso entre sí y con los Actores.

El actor *Usuario* es generalizado en los actores *Administrador*, *Coordinador de aldea* y *Docente*, este actor utiliza el Caso de Uso *Iniciar Sesión* para identificarse ante el sistema y de este modo llevar a cabo las tareas que le competen. El actor *Estudiante* utiliza el caso de uso *Consultar notas* para consultar los registros de notas realizados por el actor *Docente* al finalizar un periodo académico.

Cada vez que los usuarios *Administrador*, *Coordinador de aldea* y *Docente* interactúan con los casos de uso ya mencionados, el Sistema Gestor de Base de Datos (SGBD) es quien se ocupa de ejecutar las transacciones (lectura, escritura) de los datos.

El sistema responde de manera satisfactoria si las acciones se llevaron a cabo exitosamente o advierte en caso de que ocurra algún problema al momento de ejecutar la acción.

Una vez analizado el contexto del proceso de registros de notas en el sistema actual, ahora el objetivo será elaborar el modelado de una aplicación que satisfaga las necesidades de los usuarios.

### **3.3 Captura de requisitos**

La forma más adecuada para la captura de requisitos es la utilización de los Casos de Uso, por ser un medio intuitivo y sistemático que proporciona la entrada fundamental para el análisis y el diseño.[9]

El propósito de esta fase es desarrollar un modelo del sistema que se va a satisfacer las necesidades de la universidad, pues ésta desea controlar a nivel nacional

el proceso de ingreso, prosecución y egreso. El objetivo es obtener información en tiempo real así mismo obtener estadísticas que permitan tomar decisiones.

### **3.3.1 Requisitos funcionales del sistema propuesto.**

El sistema a diseñar “Sistema Único de Registro UBV” cuyo acrónimo será *SURUBV*, permitirá a la universidad ofrecer servicios académicos, como consultas de notas, realizar inscripciones en línea, entre otros.

Por otro lado la universidad contaría con una plataforma para dar respuestas a las necesidades, colocándose a la vanguardia en ofrecer servicios académicos a través de una plataforma informática.

La aplicación estará basada en una arquitectura Cliente-Servidor, en la cual se usará un Browser o navegador Web como interfaz gráfica de usuario (GUI) cliente, lo cual le permitirá a los usuarios acceder desde cualquier lugar donde exista una conexión a Internet. Otra de las ventajas de utilizar una interfaz Web, es que no es necesario desarrollar un programa “cliente” para interactuar con el sistema principal, debido a que los navegadores Web están muy difundidos en la mayoría de los sistemas operativos, por lo tanto esta interfaz es la idónea para la aplicación a diseñar. Seguidamente en la tabla 3.3, se muestran las funcionalidades que debe poseer el sistema, producto de conversaciones con los diferentes actores.

**Tabla 3.3** Funcionalidades del sistema propuesto.

FUNCIONALIDADES	DESCRIPCIÓN
2. Autenticación de Usuarios.	Esta función le permite al sistema identificar al usuario y dependiendo del tipo de usuario ofrecer las diferentes funcionalidades.
3. Inscripciones de los estudiantes.	El sistema permitirá realizar las inscripciones de los estudiantes en los períodos académicos, haciendo uso de la oferta académica de los programas de formación de grado tanto en sedes como municipalizados.
<ul style="list-style-type: none"> <li>• Registrar nuevos estudiantes, actualizar datos de estudiantes y registro de notas de los estudiantes.</li> </ul>	El sistema haciendo uso de la arquitectura cliente-servidor permitirá mediante un servidor Web, ofrecer una página Web a los coordinadores de de control de estudios de los Ejes UBV, para coordinar las inscripciones y registro de notas.
<ul style="list-style-type: none"> <li>• Generar reportes de inscripciones.</li> </ul> <p>Generar constancia de inscripción.</p> <p>Generar constancia de estudio.</p> <p>Generar récord académico.</p> <p>Generar notas certificadas.</p>	El sistema tendrá la capacidad de ofrecer servicios académicos a los estudiantes a través del sitio Web con la colaboración de los coordinadores de aldeas.

### 3.3.2 Identificación de actores y casos de uso del sistema propuesto.

La identificación de actores se llevó a cabo asociando un actor por cada rol de usuario en el sistema a diseñar *véase* la sección 3.1.1, donde se detalla este método, de tal manera que los actores encontrados vienen a representan a los diferentes tipos de usuarios que interactúan con el sistema, *véase* tabla 3.4., donde se establecen y se recogen las necesidades de éstos, las cuales se obtuvieron a través de conversaciones directas con los usuarios y observaciones directas de los procesos.

**Tabla 3.4.** Identificación de actores para el sistema propuesto.

<b>ACTOR</b>	<b>¿QUE NECESITA DEL SISTEMA?</b>
<b>Coordinador de Eje UBV</b>	4. Elaborar oferta académica para los diferentes programas de formación en los estados asociados a su Eje UBV. 5. Abrir inscripciones. 6. Administrar el registro de las notas. 7. Actualizar datos de los alumnos. 8. Generar constancia de estudio, notas certificadas, récord de notas, comprobantes de inscripción.
<b>Estudiante</b>	9. Inscribir unidades curriculares. 10. Consultar notas. 11. Obtener constancia de inscripción, constancia de estudios, récord de notas, comprobante de inscripción.
<b>Docente o Colaborador</b>	12. Registrar notas de los estudiantes una vez culminado el periodo académico
<b>Administrador del sistema</b>	5. Realizar respaldos de la data 6. Administración de usuarios

Una vez encontrados los actores y haber obtenido información acerca de sus necesidades, el paso siguiente será encontrar casos de uso y nuevos actores que darán soporte a las necesidades de los usuarios.

El modelo del negocio elaborado en la sección 3.1.1 representado mediante un Diagrama de casos de uso, ver figura 3.1, servirá como punto de inicio para la búsqueda.

Para encontrar actores y casos de uso de debe proceder como se describió en la sección 3.1.1, pensando en una secuencia de interacciones usuario-sistema cuyo nombre refleje cuál es el objetivo de la interacción entre el actor y el sistema, dicho de otra manera se debe especificar el comportamiento de “cosas” dinámicas, en este caso de instancias de casos de uso, pero también se debe considerar que no todas las tareas pueden automatizarse mediante un sistema informático. De este modo se describen los siguientes casos de usos tomando en cuenta las necesidades de la tabla 3.3.

### 13. Caso de Uso: Iniciar Sesión

**Actores participantes:** Coordinador de Eje UBV, Docente, Estudiante y Administrador del sistema, SGBD.

**Descripción:** Este Caso de Uso es activado por los actores Coordinador de aldea, Docente y Estudiante para iniciar la interacción con el sistema desde un navegador Web, creándose así una “sesión” que permitirá utilizar funcionalidades para llevar a cabo su tarea. Esta sesión se mantendrá hasta que el usuario salga voluntariamente activando el caso de uso “*Salir de sesión*” el cual eliminará la sesión iniciada por el actor participante.

**Precondiciones:** El usuario está registrado en el sistema.

**Escenario principal:**

- El usuario llega a un dispositivo conectado a Internet que tiene un navegador e ingresa la dirección del sitio Web de la aplicación.
- El usuario introduce su login y password.
- El sistema realiza la validación del usuario.
- Finaliza el caso de uso.

**Flujos alternos:**

*A1: login o password de usuario inválidos.*

La secuencia A1 comienza en el punto 3 del escenario principal.

3 El sistema comunica que el login o password son inválidos.  
El escenario vuelve al punto 2.

#### **14. Caso de Uso: Salir de Sesión.**

**Actores participantes:** Coordinador de Eje UBV, Docente, Estudiante y Administrador del sistema.

**Descripción:** Este Caso de Uso le permitirá al actor eliminar la sesión generada por el sistema, lo cual le permite ingresar como otro tipo de usuario.

**Precondiciones:** El usuario ya ha iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción de Salir de sesión en el sistema.
- El sistema responde que confirme si esta seguro que desea salir de la sesión.
- Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 2 del escenario principal.

*A1: El usuario decide no salir de la sesión.*

3. El sistema permanece en la sesión iniciada.

#### **15. Caso de Uso: Registrar notas.**

**Actores participantes:** Docente y SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Docente registrar las notas finales de la unidad curricular que él dicta.

**Precondiciones:** La unidad curricular esta registrada en el sistema y asignada al Docente. El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción para Registrar notas.
- El sistema responde desplegando la información sobre las unidades curriculares y secciones asignadas al Docente.

- El usuario selecciona la sección y la unidad curricular deseada.
- El sistema responde desplegando un formulario para ingresar las notas.
- El usuario guarda las calificaciones.
- El sistema responde que confirme si esta de acuerdo
- Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 10 del escenario principal.

*AI: El usuario decide no guardar el registro.*

10. El sistema permanece en la sesión iniciada.

- **Caso de Uso:** Consultar registro de notas

Actores participantes: Docente, Coordinador de Eje UBV, Estudiante y SGBD.

**Descripción:** Este Caso de Uso es activado por el actor Docente, Coordinador de aldea y Estudiante consultar las notas registradas de las unidades curriculares en las diferentes secciones, el SGDB es quien realiza las transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción de Consultar registro de notas.
- El sistema responde solicitando el número cédula a través de un formulario.
- El usuario ingresa el valor de la cédula.
- El sistema responde mostrando las notas registradas de la cédula ingresada.
- Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*AI: No se consiguieron notas registradas.*

4. El sistema comunica que no hay registros.

El escenario vuelve al punto 3.

- **Caso de Uso:** Elaborar carga académica.

**Actor participante:** Coordinador Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea asignar las unidades curriculares y secciones a los docentes en las diferentes aldeas. El SGBD es quien registra las transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

3.5.1 El usuario selecciona la opción Asignar carga académica.

4.5.1 El sistema responde mostrando un formulario que le permitirá buscar el docente a través de la cédula.

5.5.1 El usuario ingresa el valor de la cédula.

6.5.1 El sistema responde mostrando un formulario para asignar la carga académica.

7.5.1 El usuario guarda el registro.

8.5.1 Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: La cédula no esta registrada.*

4. El sistema informa que la cédula no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 5 del escenario principal.

*A2: El usuario decide no guardar el registro.*

6. El sistema permanece en la sesión iniciada

El escenario vuelve al punto 3.

- **Caso de Uso:** Modificar carga académica.

**Actor participante:** Coordinador de Eje UBV y SGBD

**Descripción:** Este Caso de Uso le permitirá al Coordinador de aldea modificar la asignaciones de cargas académica que se han asociado a los Docentes. El SGBD SACI es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

3.5.3 El usuario selecciona la opción Modificar carga académica.

4.5.3 El sistema responde mostrando un formulario que le permitirá buscar el docente a través de la cédula.

5.5.3 El usuario ingresa el valor de la cédula.

6.5.3 El sistema responde mostrando un formulario para asignar la Modificar la carga académica.

7.5.3 El usuario guarda el registro.

8.5.3 Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: La cédula no esta registrada.*

5. El sistema informa que la cédula no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 5 del escenario principal.

*A2: El usuario decide no guardar el registro.*

6. El sistema permanece en la sesión iniciada

El escenario vuelve al punto 3.

• **Caso de Uso:** Registrar docentes.

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea Registrar los datos de los Docentes como cédula, nombres, apellidos, cargo, escalafón, etc. El SGBD es quien registra las diferentes transacciones.

Precondiciones: El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- 3.5.4 El usuario selecciona la opción Registrar Docentes.
- 4.5.4 El sistema responde mostrando un formulario que le permitirá registrar los datos de los docentes.
- 5.5.4 El usuario ingresa los datos.
- 6.5.4 El usuario guarda el registro.
- 7.5.4 Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: El usuario decide no guardar el registro.*

- 5. El sistema permanece en la sesión iniciada
  - El escenario vuelve al punto 3.

- **Caso de Uso:** Modificar registro de docentes.

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea modificar los datos registrados de los Docentes. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- 1. El usuario selecciona la opción Modifica registro de docentes.
- 2. El sistema responde mostrando un formulario que le permitirá buscar el docente a través de la cédula.
- 3. El usuario ingresa el valor de la cédula.
- 4. El sistema responde mostrando un formulario que le permitirá modificar los datos de los docentes.
- 5. El usuario actualiza los datos.

6. El usuario guarda el registro.

7. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: La cédula no esta registrada.*

6. El sistema informa que la cédula no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 6 del escenario principal.

*A1: El usuario decide no guardar el registro.*

7. El sistema permanece en la sesión iniciada

El escenario vuelve al punto 3.

- **Caso de Uso:** Generar notas certificadas

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea generar notas certificadas de los registros de notas de cualquier estudiante. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

3.5.5 El usuario selecciona la opción Generar notas certificadas.

4.5.5 El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.

5.5.5 El usuario ingresa el valor de la cédula.

6.5.5 El sistema responde mostrando un formulario que le permitirá imprimir las notas certificadas.

7.5.5 Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: El estudiante no esta registrado.*

7. El sistema informa que la cédula ingresada no esta registrada.

El escenario vuelve al punto 3.

- **Caso de Uso:** Reiniciar cuenta estudiante

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea reiniciar la cuenta de estudiante, lo cual colocará como password el número de cédula del estudiante. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción Reiniciar cuenta estudiante.
- El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.
- El usuario ingresa el valor de la cédula.
- El sistema responde mostrando un formulario que le permitirá reiniciar la cuenta.
- El usuario reinicia la cuenta.
- Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: El estudiante no esta registrado.*

- El sistema informa que la cédula ingresada no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 5 del escenario principal.

*A2: El usuario decide no reiniciar la cuenta.*

- El sistema permanece en la sesión iniciada. El escenario vuelve al punto 3.

.

- **Caso de Uso:** Generar reporte de inscripción

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso es activado por el usuario Coordinador de aldea para generar un reporte de las listas de estudiantes inscritos por sección, unidad curricular, aldea y programa de formación de grado. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Generar reporte de inscripción.
2. El sistema responde mostrando un formulario que le permitirá imprimir el reporte tomando en cuenta el período académico.
3. El usuario imprime el reporte.
4. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 3 del escenario principal.

*A1: El usuario decide no imprimir el reporte.*

4. El sistema permanece en la sesión iniciada.

- **Caso de Uso:** Elaborar oferta académica.

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea elaborar la oferta de secciones y unidades curriculares que se dictarán en los diferentes estados de Venezuela donde la universidad ofrece programas de formación a través de aldeas. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Elaborar oferta académica.

2. El sistema responde mostrando un formulario que le permitirá crear secciones para las unidades curriculares que se seleccionen.
3. El usuario guarda el registro.
4. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 3 del escenario principal.

*AI: El usuario decide no guardar el registro.*

2. El sistema permanece en la sesión iniciada

• **Caso de Uso:** Modificar oferta académica.

**Actor participante:** Coordinador de Eje UBV y SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Coordinador de aldea modificar las secciones y unidades curriculares que se asignaron a un programa de formación en un estado determinado de Venezuela. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Modificar oferta académica.
2. El sistema responde mostrando un formulario que le permitirá realizar la modificación de la secciones de unidades curriculares que se seleccionaron.
3. El usuario actualiza el registro.
4. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 3 del escenario principal.

*AI: El usuario decide no actualizar el registro.*

4. El sistema permanece en la sesión iniciada.

- **Caso de Uso:** Modificar inscripción de unidades curriculares.

**Actor participante:** Coordinador de Eje UBV, SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Administrador modificar las inscripciones de unidades curriculares que hayan realizado los estudiantes durante un proceso de inscripciones. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Modificar inscripción de unidades curriculares.
2. El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.
3. El usuario ingresa el valor de la cédula.
4. El sistema responde mostrando un formulario que le permitirá realizar la modificación de la secciones de unidades curriculares seleccionadas por el estudiante.
5. El usuario actualiza el registro.
6. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: El estudiante no esta registrado.*

- El sistema informa que la cédula ingresada no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 5 del escenario principal.

*A2: El usuario decide no actualizar el registro.*

1. El sistema permanece en la sesión iniciada.

- **Caso de Uso:** Modificar Registro de Notas.

**Actor participante:** Coordinador de Eje UBV, SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Coordinador de Eje UBV Actualizar el registro las notas de los estudiantes. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Modificar el registrar notas de unidades curriculares.
2. El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.
3. El usuario ingresa el valor de la cédula.
4. El sistema responde mostrando un formulario que le permitirá realizar el registro de las notas de unidades curriculares seleccionadas al estudiante.
5. El usuario realiza el registro.
6. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 4 del escenario principal.

*A1: El estudiante no esta registrado.*

1. El sistema informa que la cédula ingresada no esta registrada.

El escenario vuelve al punto 3.

La secuencia A2 comienza en el punto 5 del escenario principal.

*A2: El usuario decide no actualizar el registro.*

6. El sistema permanece en la sesión iniciada.

• **Caso de Uso:** Crear cuenta estudiante.

**Actor participante:** Estudiante, SGBD.

**Descripción:** Este Caso de Uso le permitirá al usuario Estudiante crear una cuenta de acceso al sistema, proporcionándole la opción de agregar una clave. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe estar registrado como estudiante de la universidad.

**Escenario principal:**

1. El usuario llega a un dispositivo conectado a Internet que tiene un navegador e ingresa la dirección del sitio Web de la aplicación.
2. El usuario introduce su cédula.
3. El sistema realiza la validación del usuario y muestra un formulario para crear la cuenta del usuario.
4. El usuario crea la cuenta
5. Finaliza el caso de uso.

**Flujos alternos:**

*A1: la cédula no esta registrada.*

La secuencia A1 comienza en el punto 3 del escenario principal.

3. El sistema comunica que la cédula no esta registrada como estudiante.

El escenario vuelve al punto 2.

*A2: El usuario decide no crear la cuenta.*

La secuencia A1 comienza en el punto 4 del escenario principal.

5. El sistema regresa a la página principal de la aplicación.

• **Caso de Uso:** Generar constancia de estudios.

**Actor participante:** Coordinador de Eje UBV, Estudiante y SGBD.

**Descripción:** Este Caso de Uso es activado por actor Coordinador de aldea y el actor Estudiante para generar una constancia de estudios, en el caso del estudiante éste debe llevar el documento para ser firmado y debidamente sellado en el control de estudios de la universidad de su estado o en la aldea donde cursa estudios, cuando el usuario el actor es un estudiante el sistema toma el valor cédula de la sesión. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Generar constancia de estudio.
2. El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.
3. El usuario ingresa el valor de la cédula.
4. El sistema responde generando el documento el cual podrá imprimir.
5. Finaliza el caso de uso.

**Flujos alternos:**

*A1: la cédula no esta registrada.*

La secuencia A1 comienza en el punto 4 del escenario principal.

2. El sistema comunica que la cédula no esta registrada como estudiante.  
El escenario vuelve al punto 3.

- **Caso de Uso:** Inscribir unidades curriculares

**Actor participante:** Coordinador de Eje UBV, Estudiante y SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Coordinador de aldea y al actor estudiante registrar unidades curriculares que va a cursar un estudiante en un determinado período académico. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Inscribir unidades curriculares.
2. El sistema responde mostrando un formulario donde se seleccionarán unidades curriculares de los programas de formación según la oferta académica elaborada para ese período.
3. El usuario guarda el registro.
4. Finaliza el caso de uso.

**Flujos alternos:**

La secuencia A1 comienza en el punto 3 del escenario principal.

*AI: El usuario decide no guardar el registro.*

- El sistema permanece en la sesión iniciada

- **Caso de Uso:** Generar constancia de inscripción.

**Actor participante:** Coordinador de Eje UBV, Estudiante, SGBD.

**Descripción:** Este Caso de Uso es activado por actor Coordinador de aldea y el actor Estudiante para generar una constancia de inscripción, en el caso del estudiante éste debe llevar el documento para ser firmado y debidamente sellado en el control de estudios de la universidad de su estado o en la aldea donde cursa estudios, cuando el usuario el actor es un estudiante el sistema toma el valor cédula de la sesión. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Generar constancia de inscripción.
2. El sistema responde mostrando un formulario que le permitirá buscar al estudiante a través de la cédula.
3. El usuario ingresa el valor de la cédula.
4. El sistema responde generando el documento el cual podrá imprimir.
5. Finaliza el caso de uso.

**Flujos alternos:**

*AI: la cédula no esta registrada o el estudiante no ha inscrito unidades curriculares.*

La secuencia A1 comienza en el punto 4 del escenario principal.

3. El sistema comunica que la cédula no esta registrada como estudiante o que el estudiante no ha inscrito unidades curriculares.

El escenario vuelve al punto 3.

- **Caso de Uso:** Generar backup.

**Actor participante:** Administrador y SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Administrador generar un respaldo “backup” del sistema. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción Generar backup.
- El sistema responde mostrando un formulario que le permitirá generar un respaldo del sistema.
- El usuario genera el respaldo.
- Finaliza el caso de uso.

**Flujos alternos:**

*A1: El usuario decide no generar el respaldo.*

La secuencia A1 comienza en el punto 3 del escenario principal.

2. El sistema permanece en la sesión iniciada.

- **Caso de Uso:** Registrar coordinadores de eje.

**Actor participante:** Administrador y SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Administrador crear las cuentas de los usuarios coordinadores de aldeas que pertenecen a los ejes UBV. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

- El usuario selecciona la opción Registrar coordinadores de aldeas.
- El sistema responde mostrando un formulario que le permitirá crear las cuentas del usuario coordinador de aldea.
- El usuario crea la cuenta.

- Finaliza el caso de uso.

**Flujos alternos:**

*A1: El usuario ya existe.*

La secuencia A1 comienza en el punto 3 del escenario principal.

3. El sistema comunica que el usuario ya está registrado en el sistema.

El escenario vuelve al punto 1.

*A2: El usuario decide no crear la cuenta.*

La secuencia A1 comienza en el punto 3 del escenario principal.

- El sistema regresa a la página principal de la aplicación.

- **Caso de Uso:** Modificar registro de coordinadores de eje.

**Actor participante:** Administrador y SGBD.

**Descripción:** Este Caso de Uso le permitirá al actor Administrador modificar las cuentas de los usuarios coordinadores de aldeas que pertenecen a los ejes UBV. El SGBD es quien registra las diferentes transacciones.

**Precondiciones:** El usuario debe haber iniciado sesión en el sistema.

**Escenario principal:**

1. El usuario selecciona la opción Modificar cuentas de coordinadores de aldeas.
2. El sistema responde mostrando un formulario que le permitirá buscar al coordinador de aldea a través de la cédula.
3. El usuario ingresa el valor de la cédula.
4. El usuario actualiza la cuenta.
5. Finaliza el caso de uso.

**Flujos alternos:**

*A2: El usuario decide no actualizar la cuenta.*

La secuencia A1 comienza en el punto 4 del escenario principal.

4. El sistema permanece en la sesión iniciada.

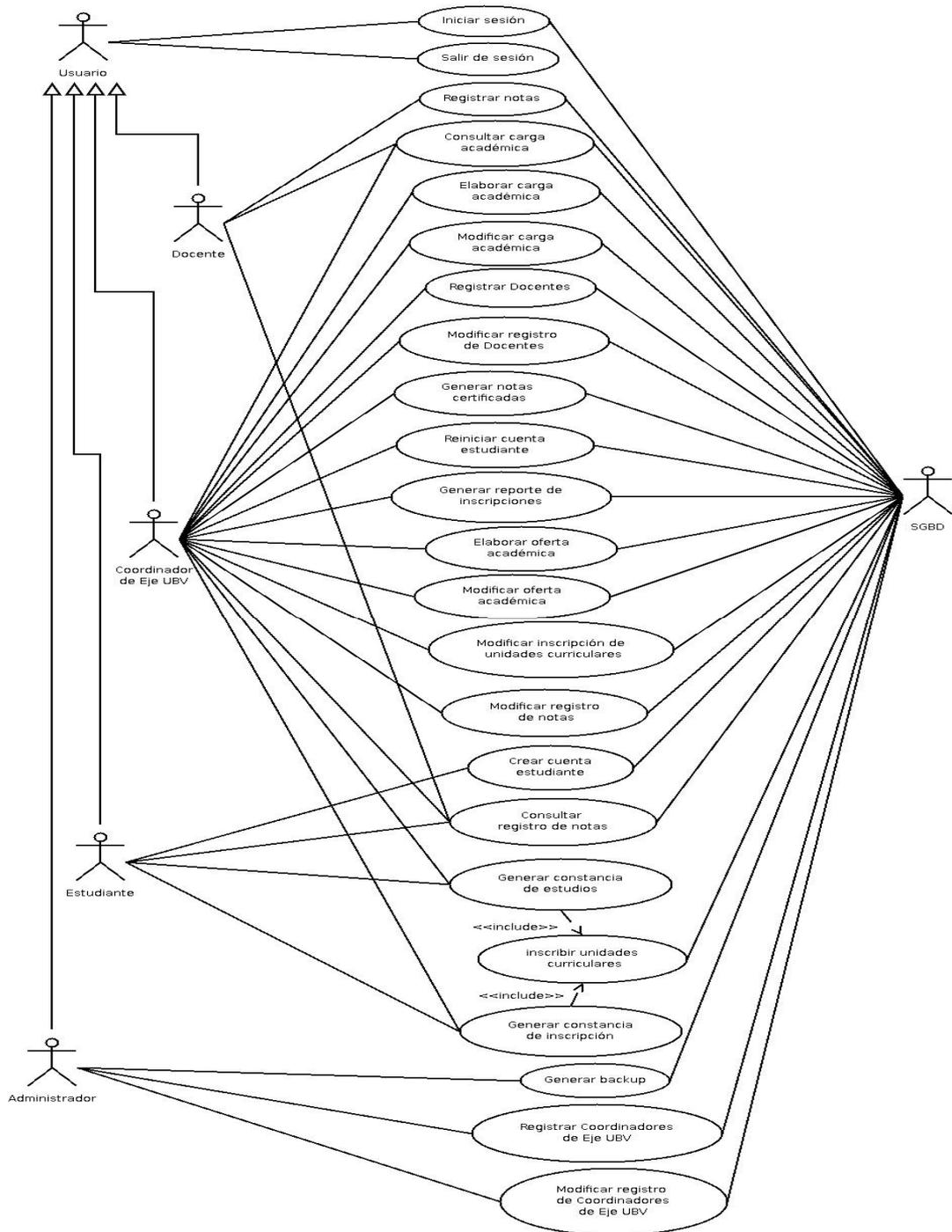
Una vez identificados los actores tomando en cuenta el rol de cada trabajador que participa en el Proceso de inscripciones de la Universidad se elabora un modelo de Casos de Uso *véase* figura 3.2 donde se muestran combinados, lo cual muestra el sistema desde la perspectiva de uso de los usuario.

Este modelo viene a representar el sistema propuesto desde la perspectiva en que los actores lo usan, siendo los Casos de Uso “fragmentos” que el sistema ofrece para aportar un resultado de valor para sus actores.

Las instancias de los Casos de Uso se consideran atómicas, es decir cada una de ellas se ejecuta por completo o no se ejecuta nada, por lo tanto el comportamiento de un Caso de Uso puede interpretarse independientemente de los otros Casos de Uso.

Una instancia de Caso de Uso es la realización (ejecución) de un Caso de Uso, es decir que una instancia de un Caso de Uso es lo que el sistema lleva a cabo cuando “obedece” a un Caso de Uso [9]. Igualmente ocurre con los Actores y los usuarios como se menciono en la sección 3.4.1.1.

Se debe considerar también que cada ejecución satisfactoria de un Caso de Uso debe proporcionar algún valor al usuario para alcanzar su objetivo, esto se debe considerar como norma para la identificar un buen Caso de Uso, así mismo ayuda a determinar el ámbito apropiado para ellos.



**Figura 3.2.** Modelo de Casos de Uso del Sistema de SURUBV.  
**Fuente:** elaboración propia, año:2009.

Ahora, se explicará como se relacionan los Casos de Uso y los Actores que dan soporte a los procesos de servicios académicos a través del sistema propuesto, los cuales en conjunto constituyen el Modelo de Casos de Uso del Sistema Único de Registro Académico UBV.

Los actores Estudiante, Coordinador de Eje UBV, Docente y Administrador son generalizados en el Actor “Usuario” a fin de comprender la funcionalidad de inicio de sesión lo cual permitirá que el sistema ofrezca una interfaz de trabajo diferente dependiendo del tipo de usuario donde se activen los casos de uso, el sistema gestor de base de datos (SGBD) es el encargado de realizar las operaciones de lectura-escritura en los diferentes procesos. Para activar los casos de uso “Generar constancia de estudios” y “Generar constancia de inscripción” el estudiante debe haber inscrito unidades curriculares en el período lectivo, de lo contrario el sistema no dará acceso a los caso de uso antes mencionados, los actores Estudiante, Coordinador de Eje UBV, Docente y Administrador pueden salir del sistema activando el caso de uso “Salir de sesión”.

### **3.4 Requisitos no funcionales del sistema propuesto.**

Una vez planteada la necesidad del acceso a una plataforma, el sistema de acceso que se adapta mejor a esta situación es la implementación de un modelo Cliente-Servidor utilizando la red pública Internet, ésto por ser una tecnología muy utilizada hoy en día y de amplia trayectoria.

De esta manera se brinda posibilidad de acceder desde cualquier lugar donde exista una conexión a la Internet permitiendo así acceder a la información en tiempo real, logrando que se puedan obtener información para tomar para la toma de decisiones. En líneas generales el sistema SURUBV permitirá administrar y

mantener la información sobre los procesos de Ingreso, prosecución y egreso. Todo esto debe ofrecer un alto nivel de seguridad, ya que existirá una conexión directa entre el sistema de la universidad y el exterior, en este caso la red pública Internet, por ellos es necesario establecer una barrera de protección para el buen funcionamiento del sistema a diseñar.

El llevar a cabo el desarrollo de un sistema basado en un modelo Cliente-Servidor implica establecer requisitos inherentes a este tipo de sistema, especificando requerimientos tanto de hardware como de software que permitirían crear su diseño.

Actualmente el desarrollo de aplicaciones demanda la necesidad de compatibilidad con otras plataformas de desarrollo, ello es debido a la gran variedad de sistemas, es por ello que desarrollar aplicaciones basadas en estándares abiertos es bien visto por su bajo costo. Los sistemas de Bases de Datos no escapan a esto, independiente de la casa de desarrollo, estos cumplen con el lenguaje estándar SQL, así también con protocolos de conexiones que cumplen con estándares como el TCP/IP, lo que facilitó que el desarrollo de aplicaciones pueda comunicarse y acceder a Bases de Datos independientemente de la plataforma de desarrollo.

Este es el caso del lenguaje de programación PHP, el cual posee librerías para establecer conexión con muchos sistemas de Bases de datos, siendo un sistema robusto y muy probado es el más adecuado a usar en el sistema propuesto. Por otra parte el lenguaje de programación PHP es casi un estándar en la elaboración de aplicaciones Web.

El lenguaje de programación PHP es muy flexible y versátil, disponible para varias plataformas de desarrollo permitiendo incorporar de manera transparente diversas tecnologías.

El modelo Cliente-Servidor demanda el uso de un Servidor Web, teniendo en cuenta que la organización requiere un sistema con alta prestaciones de seguridad y estabilidad, el servidor Web más apropiado sería el Apache Web Server, por ser el más usado en el mundo a nivel de servidores y poseer soporte para SSL (Secure Sockets Layer) del más alto nivel.

Tomando en cuenta que el sistema de conexión a Bases de Datos debe ser de los más estables y escalables, el más apropiado sería Postgre-SQL el cual es comparable con la versión propietaria de ORACLE, el sistema gestor de base de datos Postgre-SQL es de código abierto y soporta inclusive programación orientada a objetos.

El Servidor Web y el SGBD son dos herramientas fundamentales para el funcionamiento del sistema propuesto, es necesario contar con un sistema base o sistema operativo a la altura de éstos, es por ello que el sistema operativo GNU/Linux puede ofrecer ese nivel.

Este sistema operativo GNU/Linux cuenta con el tipo de tecnología que permitirían implementar las herramientas de software antes mencionadas. Esto no quiere decir que sea el único sistema operativo presto para usar con estas tecnologías, pues éstas son aplicaciones en la actualidad multiplataforma, lo que quiere decir que se puede implementar en otros sistemas operativos como UNIX, Windows, FreeBSD, etc.. Otra de las ventajas que ofrece el sistema operativo GNU/Linux al igual que los sistemas Postgre-SQL y Apache Web Server, es el bajo costo de adquisición ya que son desarrollados bajo licencia de código abierto, ofreciendo una alta calidad y escalabilidad.

Otro punto importante en el uso del sistema operativo GNU/Linux es que cuenta con tecnologías que permiten implementar un nivel de seguridad acorde a las

demandas, permitiendo establecer políticas de seguridad mediante el sistema Firewall, estas herramientas de software sustituyen equipos muy costosos para la implantación de medidas de seguridad en redes.

Una vez mencionados los elementos estructurales a partir de los cuales se compone el sistema propuesto se establecen los requisitos adicionales como sigue.

- **Requisitos de Plataforma Hardware**

- Servidores (características generales)*

1. Computador Intel(R) con 4 CPU Xeon(R) 1.86GHz
2. Disco duro con tecnología scsi de 100 GB.
3. Unidad de respaldo de energía.
4. Adaptador de Red de 1000Mbps.

- Clientes*

- Computador Pentium o compatible.

- **Requisitos de Plataforma software**

- Software para uso desde Internet*

- Navegador Web con soporte para HTML 4.0, CSS y SSL.

- Software para el servidor*

1. Sistemas operativo GNU/Linux (kernel 2.6 ó superior)
2. Apache Web Server (versión 2.0)
3. libapache2-mod-php5
4. Iptables (versión 1.2.6)
5. Openssl (versión 0.9.6)
6. PHP (versión 5)
7. Gestor de Bases de Datos Postgre-SQL server (versión 8.2)
8. php5-pgsql

### **3.5 Análisis de factibilidad para implementar el sistema propuesto**

Un análisis de factibilidad permitirá responder a las preguntas ¿El producto final podrá ser realizado?, ¿El producto final beneficiará a los usuarios interesados? o ¿El producto final se justifica? Dentro de un análisis de este tipo existen cuatro aspectos importantes relacionados con el estudio de la factibilidad:

#### **Factibilidad Técnica:**

Esta estudia si el trabajo para el proyecto, puede desarrollarse con el software y el personal existente, y si en caso de necesitar nueva tecnología, cuales son las posibilidades de desarrollarla.

#### **Factibilidad Económica:**

Esta investiga si los costos se justifican con los beneficios que se obtienen, y si se ha invertido demasiado, como para no crear el sistema si se cree necesario.

#### **Factibilidad Operacional:**

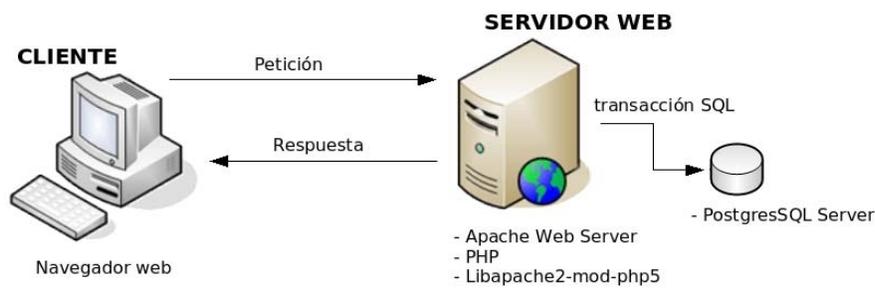
Esta investiga si será utilizado el sistema, si los usuarios usaran el sistema, como para obtener beneficios.

Considerando los puntos anteriores se asume que el sistema funcionará cuando este terminado, ya que no existen barreras que impidan su diseño, pues se dispone del apoyo por parte de la organización y de los usuarios, por lo tanto la factibilidad operacional está asegurada.

Se considera también que el proyecto es factible técnicamente, pues la tecnología que se empleará existe actualmente en el mercado, básicamente consistirá en la implantación de una arquitectura Cliente-Servidor que se apoye en un SGBD al cual se accederá a través de una aplicación informática desarrollada en el lenguaje de

programación PHP mediante tecnología Web, véase la Figura 3.3 donde se representa ésta tecnología que permite acceder a bases de datos con el uso de los programas CGI.

Por otra parte la universidad ha incluido este proyecto dentro de sus presupuestos, por tanto se dispone del dinero suficiente para su realización, con lo cual la factibilidad económica está asegurada.



**Figura 3.3** Representación de la tecnología usada en el sistema propuesto.  
**Fuente:** elaboración propia, año:2009.

### 3.6 Análisis de casos de uso mediante diagramas de colaboración.

En este paso se elaboran la realización de los Casos de Uso establecidos en la sección 3.2.2 haciendo uso de las clases del análisis, en las cuales se utilizan los estereotipos de control, interfaz y entidad, véase figura 3.4.

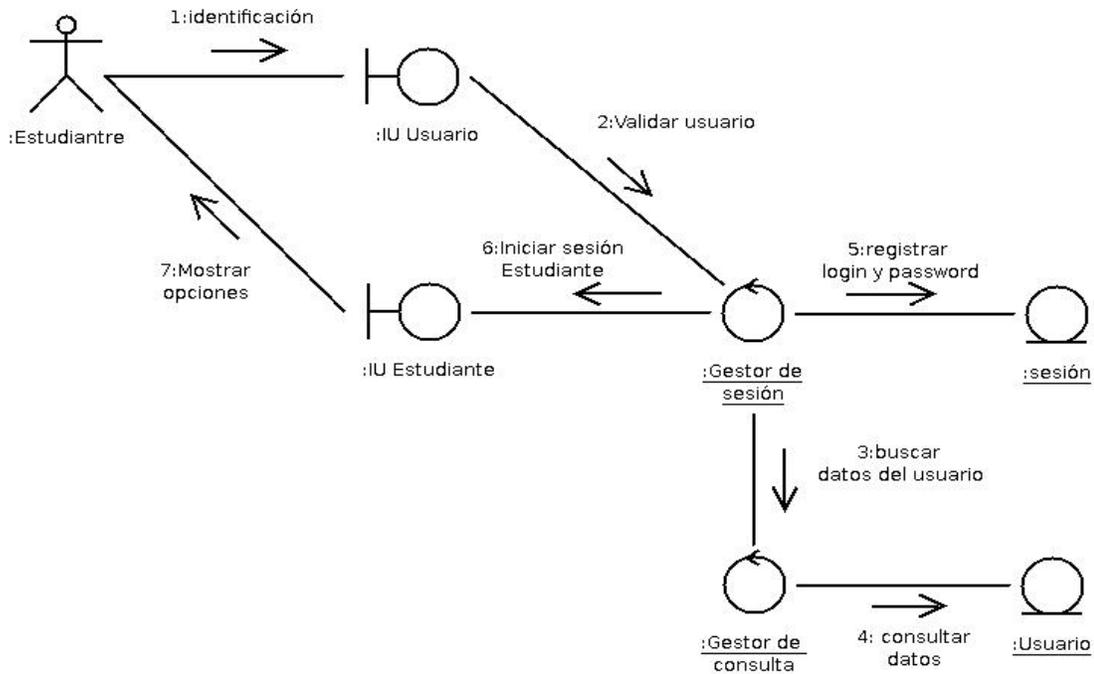


**Figura 3.4** Clases de análisis.  
**Fuente:** elaboración propia, año:2009

La realización de los Casos de Uso usando estos tres estereotipos conforman diagramas de colaboración. Esta realización se conocen como Caso de Uso-Análisis, esto permite mostrar cómo se llevan a cabo éstos y se ejecuta un Caso de Uso determinado en términos de las clases del análisis y de sus objetos del análisis en interacción, lo que permite un mecanismo que describe características estructurales y de comportamiento incluyendo interfaces, clases, tipos de datos componentes y nodos.

En este análisis con diagramas de colaboración el objetivo fundamental es identificar requisitos y responsabilidades sobre los objetos mostrando interacciones entre ellos y añadiendo mensajes a esos enlaces, los cuales denotan el propósito del objeto invocante en la interacción con el objeto invocado.

### 3.6.1 Realización del Caso de Uso Iniciar Sesión Estudiante.

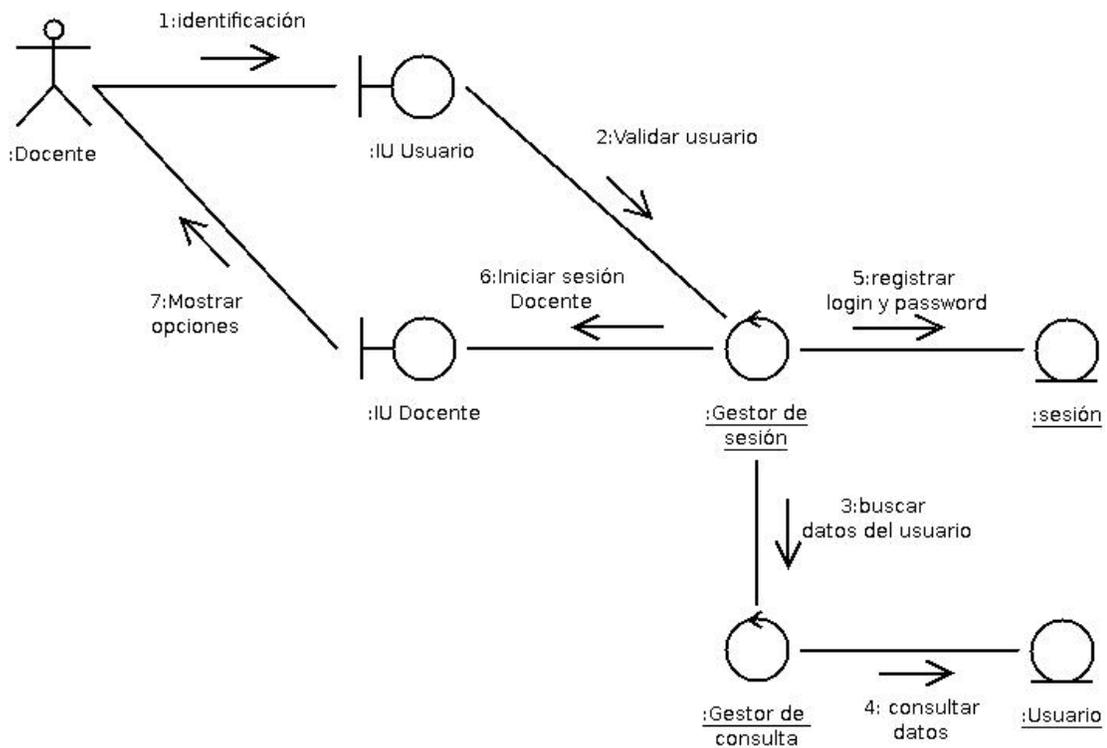


**Figura 3.5** Diagrama de colaboración de la realización del Caso de Uso Iniciar Sesión Estudiante.  
**Fuente:** elaboración propia, año:2009.

El usuario Estudiante iniciar una sesión para comenzar la interacción con el sistema. Una vez que el usuario decide iniciar una sesión debe activar el objeto interfaz *Usuario*, véase Figura 3.5, éste se identifica introduciendo su login y password (1). El objeto interfaz *Usuario* solicita al objeto *Gestor de sesión* que verifique la identidad del usuario (2), para lo cual solicita al objeto *Gestor de consulta* (3,4).

Una vez realizada la validación del usuario, el objeto *Gestor de Sesión* guarda los datos del usuario en el objeto entidad *sesión* y solicita al objeto interfaz *Estudiante* que muestre las opciones para el tipo de usuario Estudiante (5, 6, 7).

### 3.6.2 Realización del Caso de Uso Iniciar Sesión Docente.



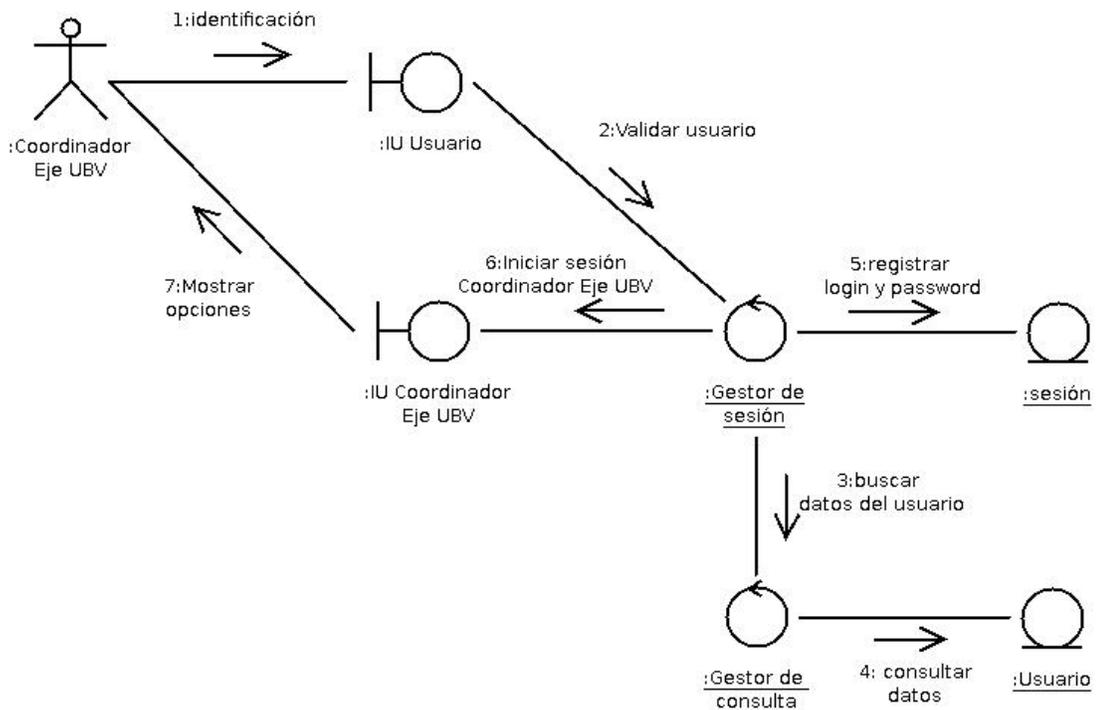
**Figura 3.6** Diagrama de colaboración de la realización del Caso de Uso Iniciar Sesión Docente.

**Fuente:** elaboración propia, año:2009.

El usuario Docente iniciar una sesión para comenzar la interacción con el sistema. Una vez que el usuario decide iniciar una sesión debe activar el objeto interfaz *Usuario*, véase Figura 3.6, éste se identifica introduciendo su login y password (1). El objeto interfaz *Usuario* solicita al objeto *Gestor de sesión* que verifique la identidad del usuario (2), para lo cual solicita al objeto *Gestor de consulta* (3,4).

Una vez realizada la validación del usuario, el objeto *Gestor de Sesión* guarda los datos del usuario en el objeto entidad *sesión* y solicita al objeto interfaz *Docente* que muestre las opciones para el tipo de usuario Docente (5, 6, 7).

### 3.6.3 Realización del Caso de Uso Iniciar Sesión Coordinador Eje UBV.



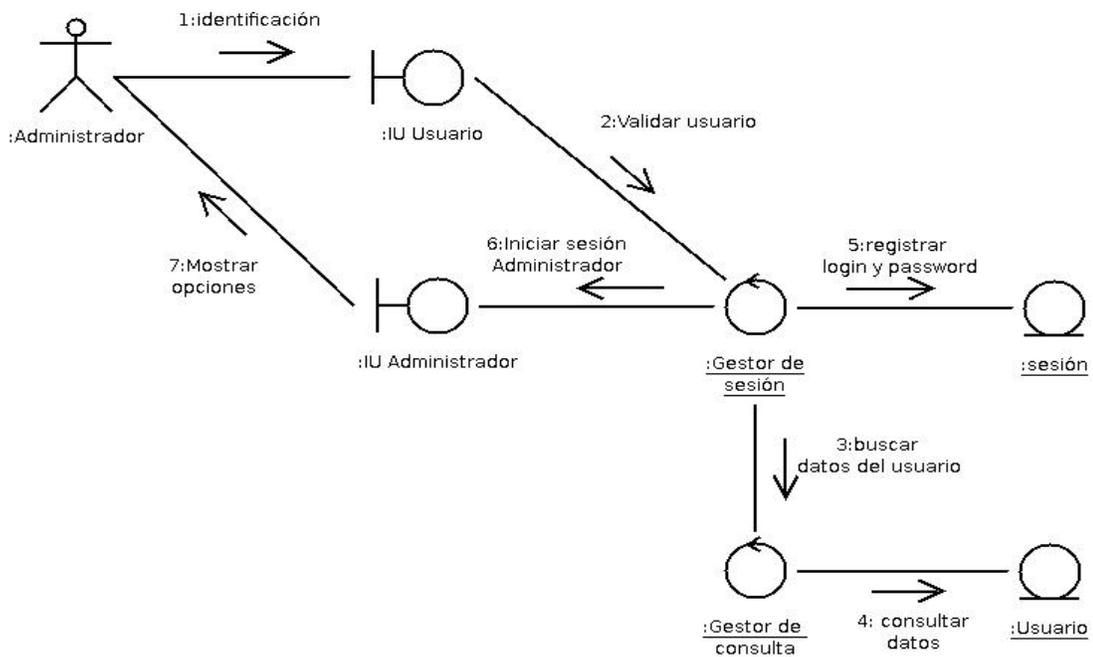
**Figura 3.7** Diagrama de colaboración de la realización del Caso de Uso Iniciar Sesión Coordinador Eje UBV.

**Fuente:** elaboración propia, año:2009.

El usuario Coordinador Eje UBV iniciar una sesión para comenzar la interacción con el sistema. Una vez que el usuario decide iniciar una sesión debe activar el objeto interfaz *Usuario*, véase Figura 3.7, éste se identifica introduciendo su login y password (1). El objeto interfaz *Usuario* solicita al objeto *Gestor de sesión* que verifique la identidad del usuario (2), para lo cual solicita al objeto *Gestor de consulta* (3,4).

Una vez realizada la validación del usuario, el objeto *Gestor de Sesión* guarda los datos del usuario en el objeto entidad *sesión* y solicita al objeto interfaz Coordinador Eje UBV que muestre las opciones para el tipo de usuario Coordinador Eje UBV (5,6,7).

### 3.6.4 Realización del Caso de Uso Iniciar Sesión Administrador.



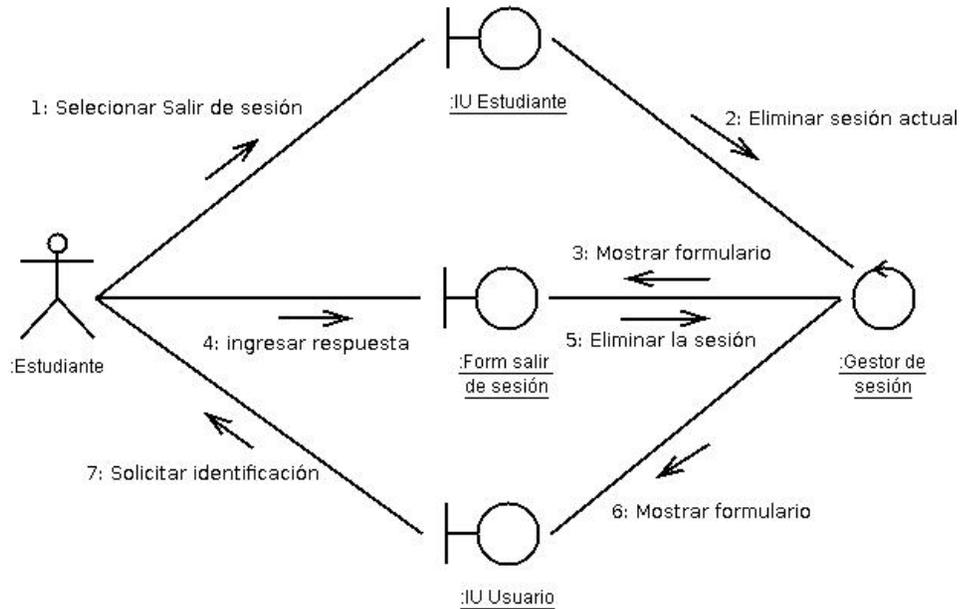
**Figura 3.8** Diagrama de colaboración de la realización del Caso de Uso Iniciar Sesión Administrador.

**Fuente:** elaboración propia, año:2009.

El usuario Administrador iniciar una sesión para comenzar la interacción con el sistema. Una vez que el usuario decide iniciar una sesión debe activar el objeto interfaz *Usuario*, véase Figura 3.8, éste se identifica introduciendo su login y password (1). El objeto interfaz *Usuario* solicita al objeto *Gestor de sesión* que verifique la identidad del usuario (2), para lo cual solicita al objeto *Gestor de consulta* (3,4).

Una vez realizada la validación del usuario, el objeto *Gestor de Sesión* guarda los datos del usuario en el objeto entidad *sesión* y solicita al objeto interfaz Administrador que muestre las opciones para el tipo de usuario Administrador (5,6,7).

### 3.6.5 Realización del Caso de Uso Salir de Sesión Estudiante.



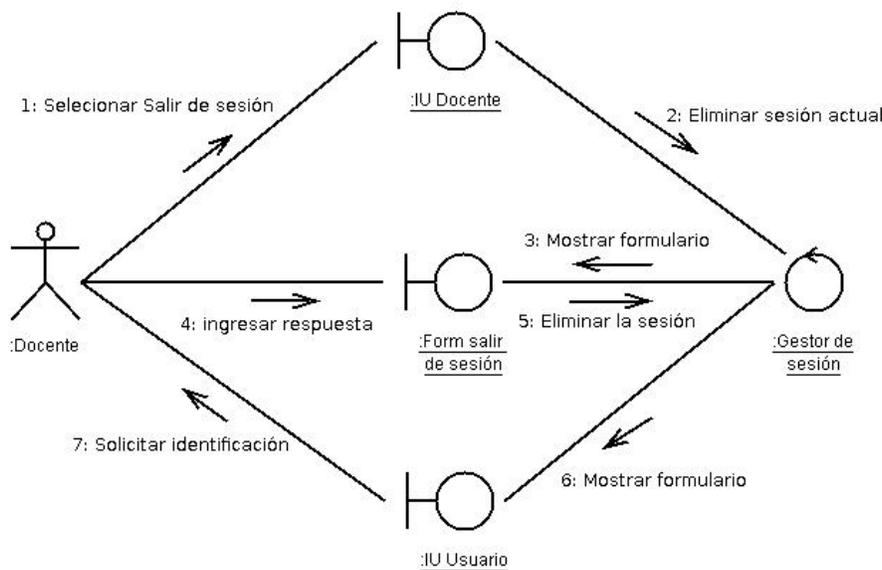
**Figura 3.9** Diagrama de colaboración de la realización del Caso de Uso Salir de Sesión Estudiante.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Salir de sesión” para abandonar la sesión actual (1) véase figura 3.9, ésto permitirá el ingreso de otro tipo de usuario en el mismo navegador o Browser.

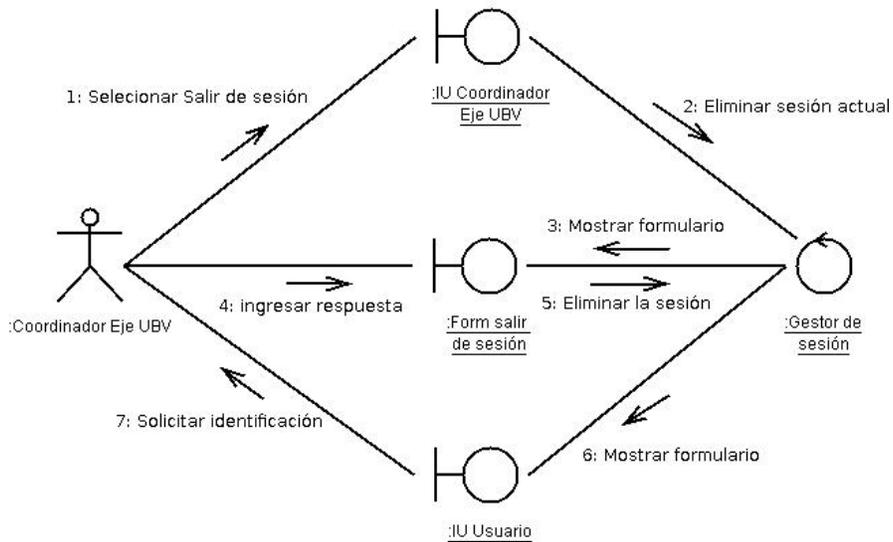
El objeto *IU Estudiante* solicita al objeto *Gestor de sesión* que elimine la sesión iniciada por el usuario (2), el *Gestor de sesión* solicita al objeto *Form salir de sesión* que confirme la solicitud del usuario (3), una vez que el usuario confirma la solicitud (4), *Form salir de sesión* envía una confirmación o negación de la solicitud al objeto *Gestor de Sesión* en el caso de confirmación el objeto *Gestor de sesión* muestra la interfaz de inicio de la aplicación para que el usuario vuelva a identificarse (5,6,7).

Para los usuarios Docente, Coordinador de Eje UBV y Administrador la realización del caso de uso mediante diagramas de colaboración es análogo al mostrado en la figura 3.9, véase figura 3.10 ,3.11 y 3.12.

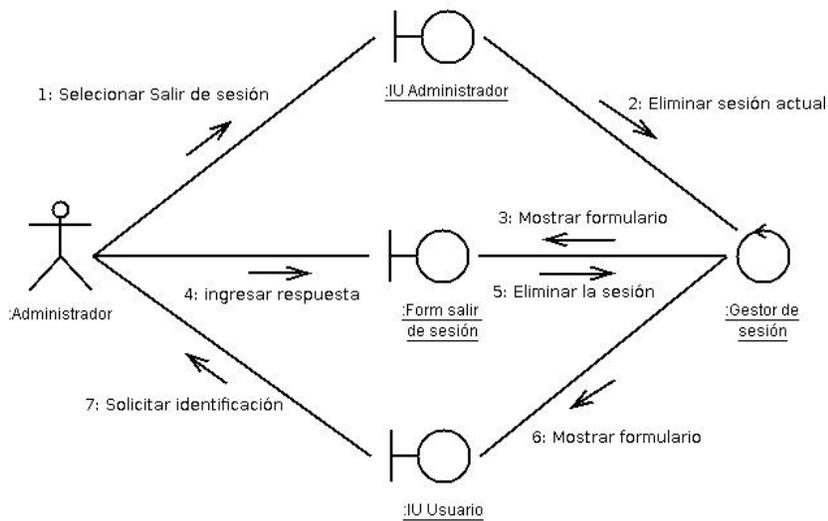


**Figura 3.10** Diagrama de colaboración de la realización del Caso de Uso Salir de Sesión Docente.

**Fuente:** elaboración propia, año:2009.

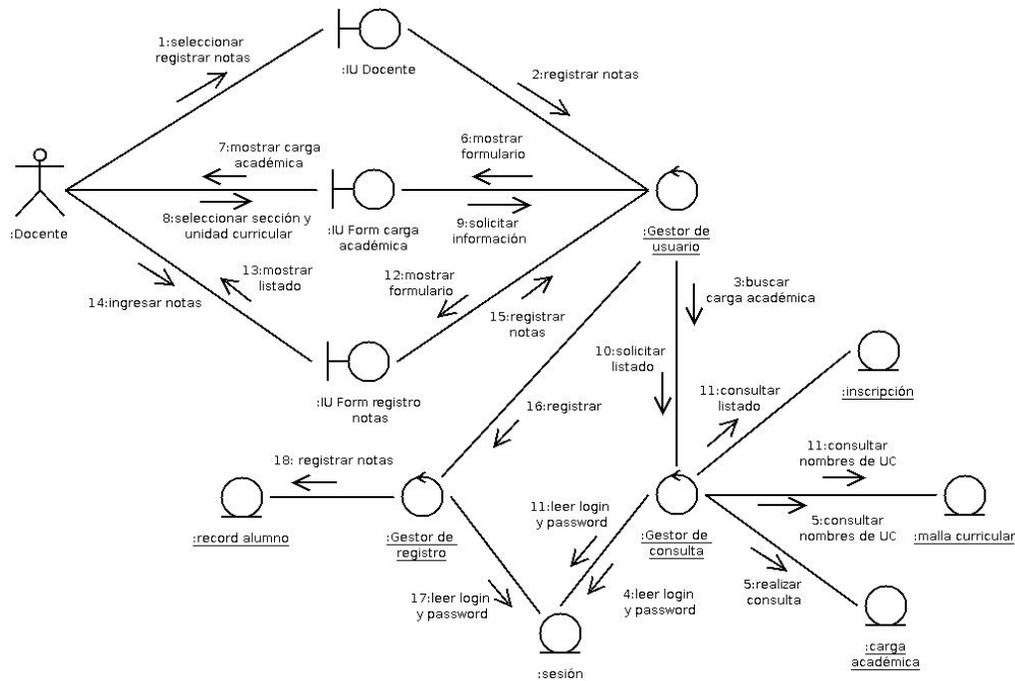


**Figura 3.11** Diagrama de colaboración de la realización del Caso de Uso Salir de Sesión Coordinador Eje UBV.  
**Fuente:** elaboración propia, año:2009.



**Figura 3.12** Diagrama de colaboración de la realización del Caso de Uso Salir de Sesión Administrador.  
**Fuente:** elaboración propia, año:2009.

### 3.6.6 Realización del Caso de Uso Registrar Notas.



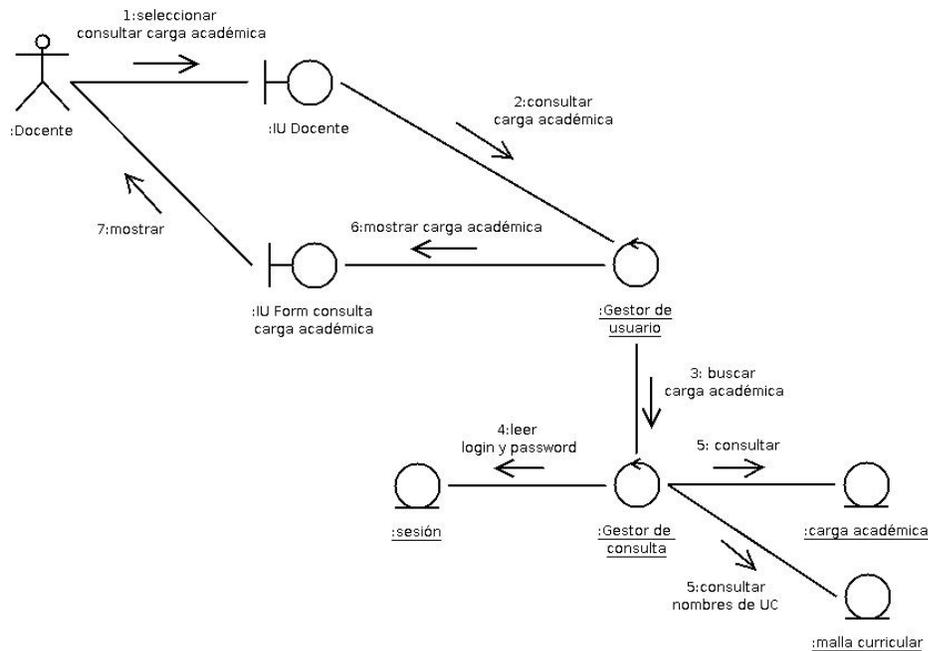
**Figura 3.13** Diagrama de colaboración de la realización del Caso de Uso Registrar notas.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Registrar notas” para llevar a cabo el registro de calificaciones (1) véase figura 3.13, el objeto *IU Docente* solicita al objeto *Gestor de Usuario* que muestre el formulario donde se registrarán las notas (2), el objeto *Gestor de Usuario* muestra la carga académica del Docente para lo cual hace uso del a información registrada en la sesión (3,4,5,6,7), una vez que el usuario ingresa los valores (8) el objeto *Form carga académica* solicita al *Gestor de usuario* que muestre el registro con la información suministrada (9,10,11), una vez obtenida la información el *Gestor de usuario* solicita al objeto *Form registro notas* que muestre al usuario un formulario para al vaciado de las notas (12,13), una vez ingresada las notas (14), el objeto *Form registro notas* solicita al objeto *Gestor de usuario* que registre las notas ingresadas lo cual solicita al objeto *Gestor de registro* (15,16,17,18).

Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.7 Consultar Carga Académica.

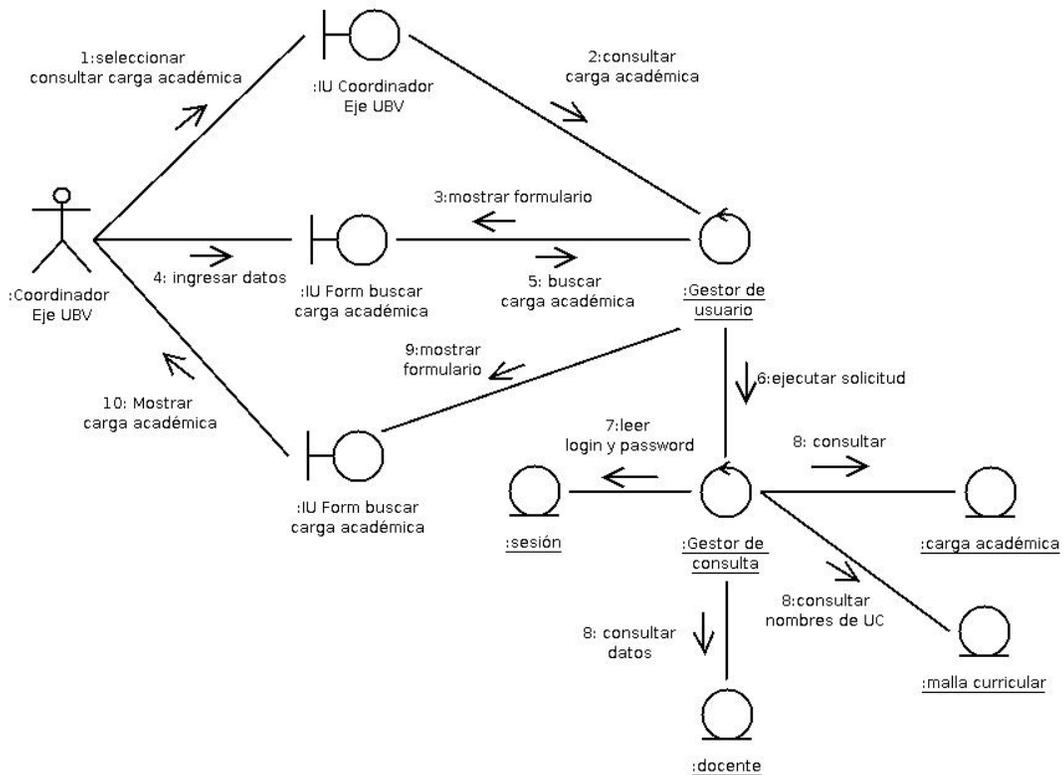


**Figura 3.14** Diagrama de colaboración de la realización del Caso de Uso Consultar carga académica Docente.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Consultar carga académica” para consultar las unidades curriculares asignadas a su persona en un periodo lectivo (1) véase figura 3.14, el objeto *IU Docente* solicita al objeto *Gestor de usuario* (2) que muestre la carga académica de usuario docente para lo cual el objeto *Gestor de usuario* solicita la información al objeto *Gestor de consulta* (3), el objeto *Gestor de consulta* obtiene información de la sesión para llevar a cabo la solicitud (4,5) luego el objeto *Gestor de usuario* solicita al objeto *Form consulta carga académica* que muestre la carga académica (6,7), Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

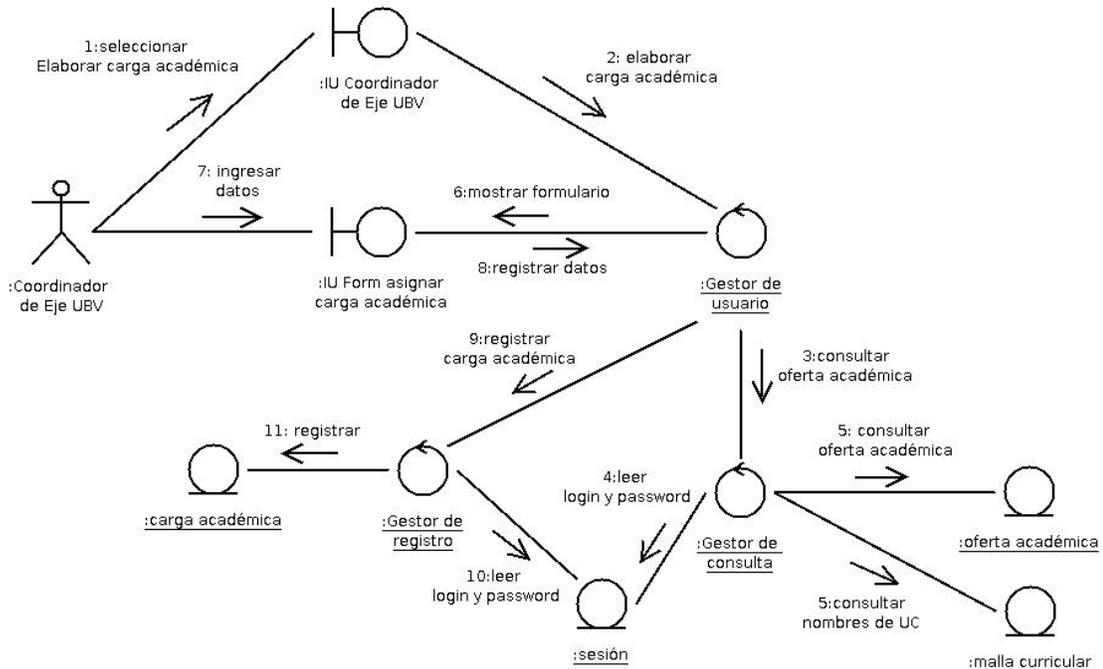
Este caso de uso también es activado por el actor Coordinador de Eje UBV, la realización es análoga, excepto que primero se debe ingresar la información para buscar la carga a académica, véase figura 3.15.



**Figura 3.15** Diagrama de colaboración de la realización del Caso de Uso Consultar carga académica Coordinador de Eje UBV.

**Fuente:** elaboración propia, año:2009.

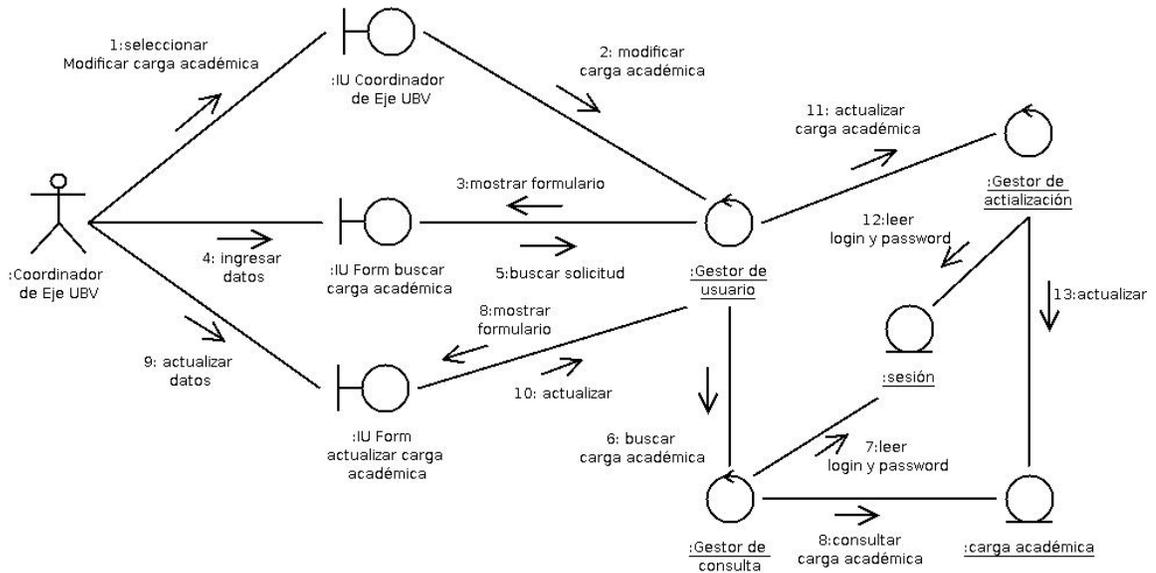
### 3.6.8 Elaborar Carga Académica.



**Figura 3.16** Diagrama de colaboración de la realización del Caso de Uso Elaborar carga académica.  
**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Elaborar carga académica” para asignar las unidades curriculares a un docente en un período lectivo (1) véase figura 3.16, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para asignar carga académica a un usuario docente para lo cual el *Gestor de usuario* solicita al objeto *Gestor de consulta* que busque la información requerida (2,3,4,5), una vez ingresada la información (6,7) el objeto *Form asignar carga académica* solicita al objeto *Gestor de usuario* que registre los datos ingresados (8) para lo cual éste solicita al *Gestor de registro* que lleve a cabo la acción (9,10,11).

### 3.6.9 Modificar Carga Académica.

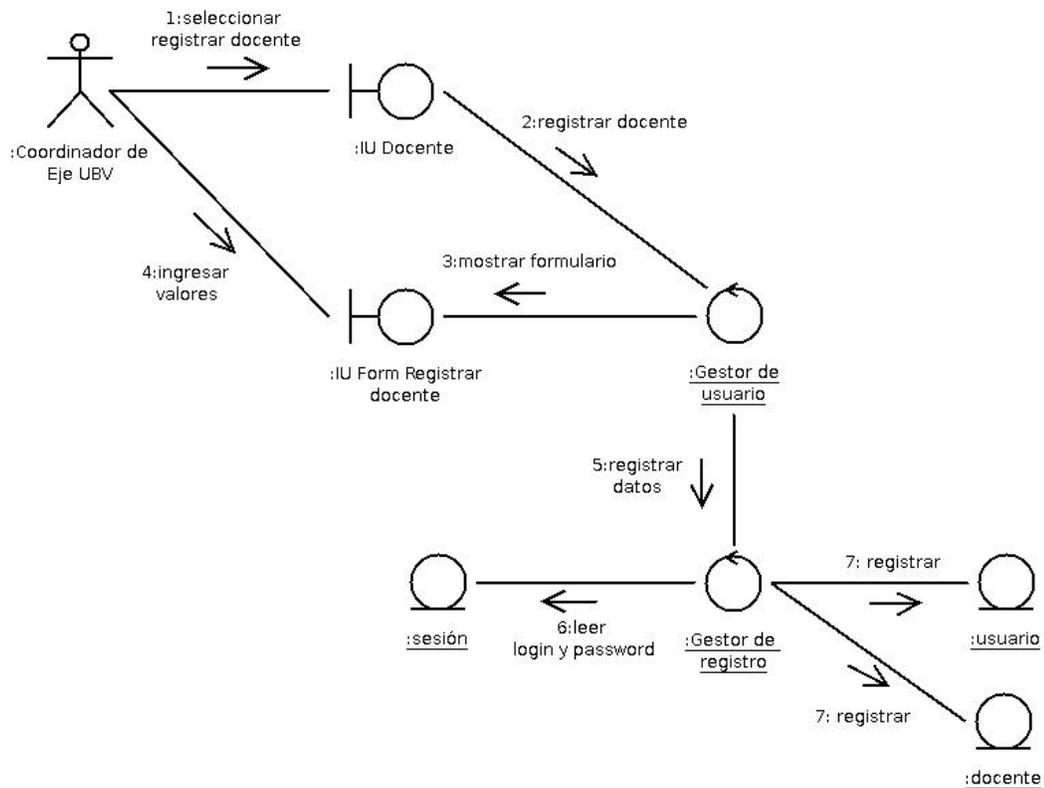


**Figura 3.17** Diagrama de colaboración de la realización del Caso de Uso Modificar carga académica.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar carga académica” para modificar las unidades curriculares a un docente en un período lectivo (1) véase figura 3.17, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar la carga académica de un usuario docente (2,3), una vez ingresada la información (4) el objeto *Form buscar carga académica* del objeto *Gestor de usuario* que busque la información, para lo cual solicita al objeto *Gestor de consulta* (5,6) que lleve a cabo la solicitud (7,8), una vez obtenida la información el objeto *Gestor de usuario* muestra la carga académica a través del objeto *Form actualizar carga académica* (8), una vez actualizado el registro el objeto *Form actualizar carga académica* solicita al objeto *Gestor de usuario* que actualice el registro lo cual solicita al objeto *Gestor de actualización* que lleve a cabo la solicitud (8,9,10,11,12,13). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.10 Registrar Docente.

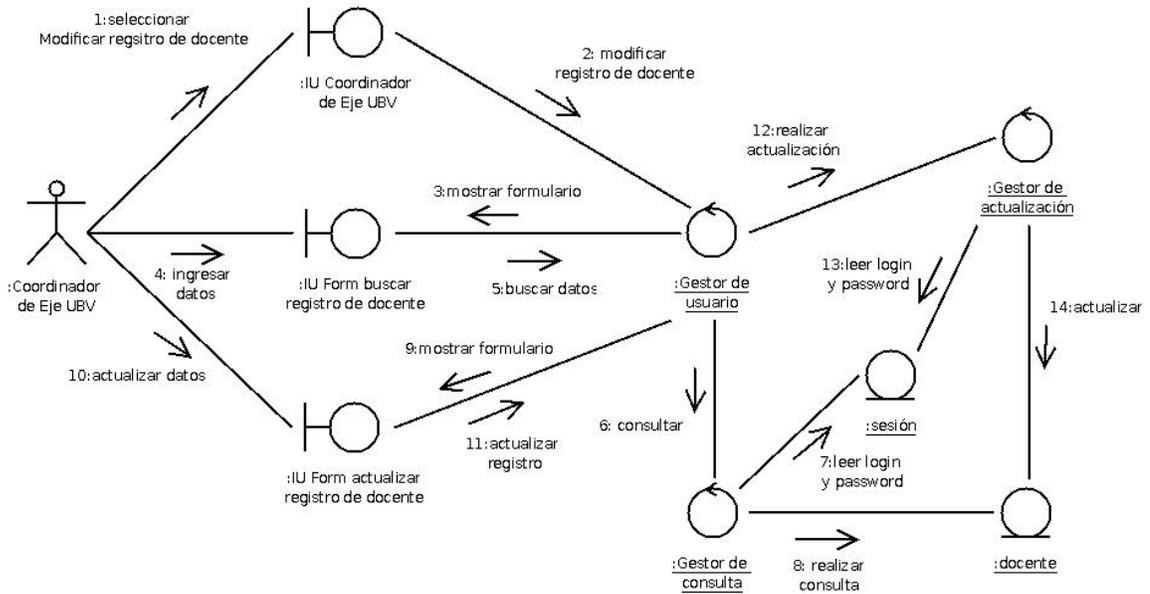


**Figura 3.18** Diagrama de colaboración de la realización del Caso de Uso Registrar docente.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Registrar docente” para llevar a cabo el registro de un docente (1) véase figura 3.18, el objeto *IU Docente* solicita al objeto *Gestor de Usuario* que muestre el formulario donde se registra el docente (2), el objeto *Gestor de Usuario* muestra el formulario (3), una vez que el usuario ingresa los valores (4) el *Gestor de usuario* obtiene datos necesarios del objeto entidad *sesión* para llevar a cabo el registro, una vez obtenida la información el *Gestor de sesión* solicita al *Gestor de registro* que lleve a cabo la acción (5,6,7). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.11 Modificar Registro de Docente.

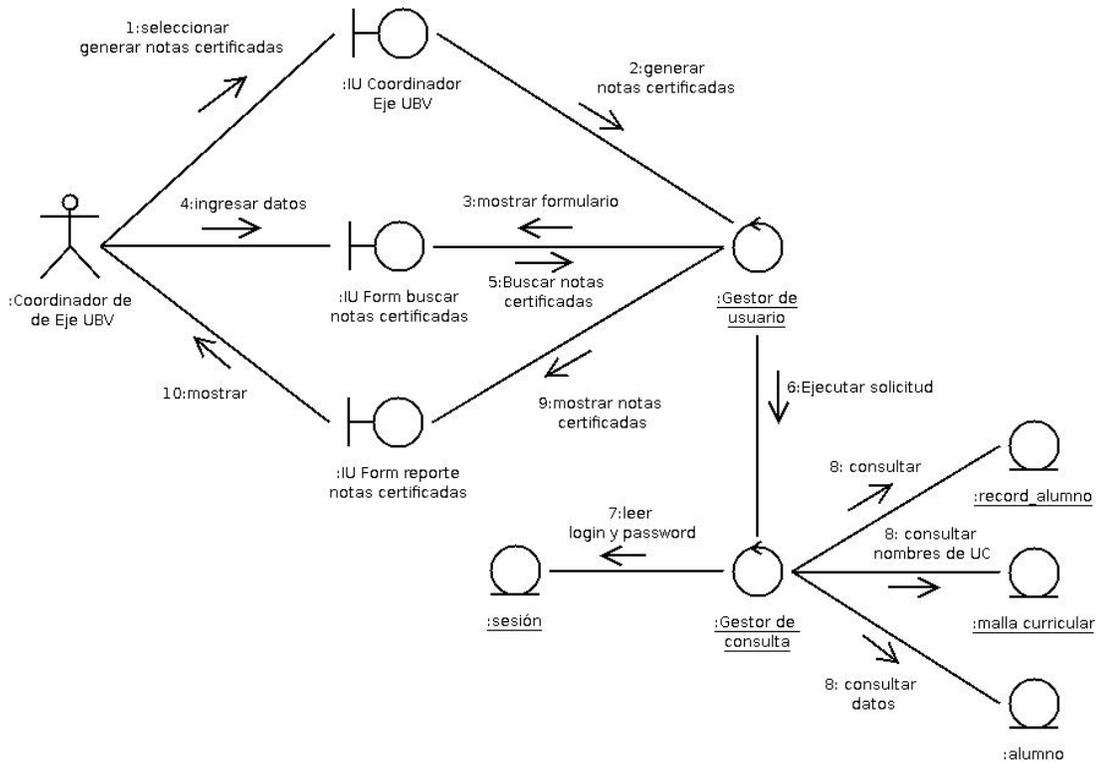


**Figura 3.19** Diagrama de colaboración de la realización del Caso de Uso Modificar registro de docente.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar registro de docente” para modificar los datos de un docente (1) véase figura 3.19, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar el registro del docente (2,3) una vez ingresada la información (4) el objeto *Form buscar registro docente* solicita al objeto *Gestor de usuario* la información solicitada, con lo cual éste solicita al objeto *Gestor de consulta* que lleve a cabo la búsqueda (5,6,7,8), una vez obtenida la información el objeto *Gestor de usuario* muestra el formulario para actualizar el registro a través del objeto *Form actualizar registro docente*, una vez actualizada la información el objeto *Form actualizar registro docente* solicita al objeto *Gestor de usuario* que lleve a cabo la actualización, lo cual solicita al objeto *Gestor de actualización* (9,10,11,12,13,14). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

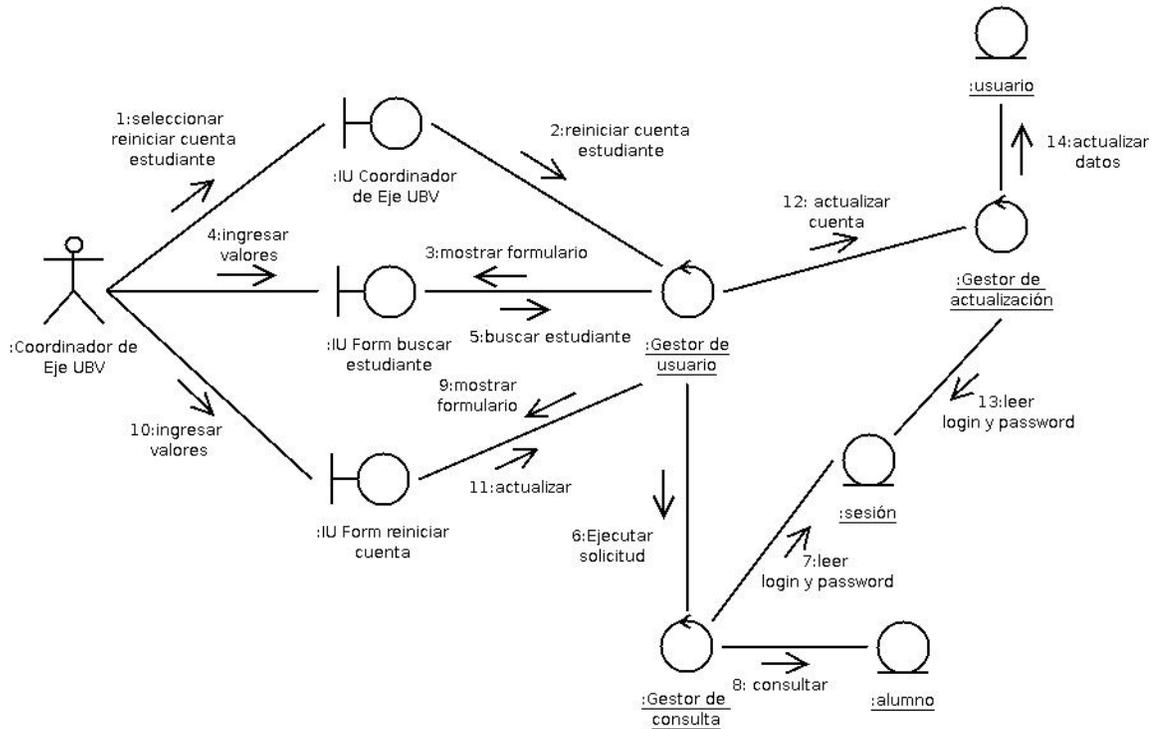
### 3.6.12 Generar Notas Certificadas.



**Figura 3.20** Diagrama de colaboración de la realización del Caso de Uso Generar notas certificadas.  
**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Generar notas certificadas” para obtener una consulta de las notas certificadas del un alumno (1) véase figura 3.20, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar los datos (2,3), una vez ingresada la información (4) el objeto *Form buscar notas certificadas* solicita al objeto *Gestor de usuario* que lleve a cabo la solicitud (5) lo cual solicita al objeto *Gestor de consulta* (6,7,8), una vez realizada la consulta el objeto *Gestor de usuario* solicita al objeto *IU Form reporte notas certificadas* que muestre la información a usuario (9,10). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.13 Reiniciar Cuenta Estudiante.



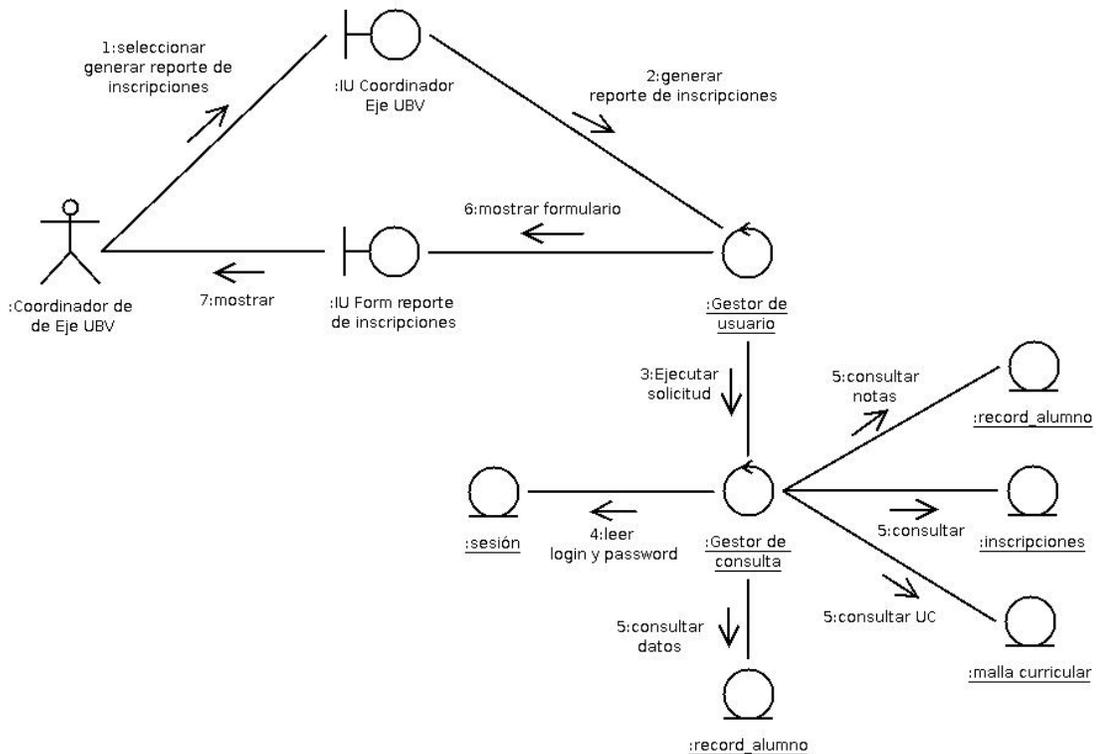
**Figura 3.21** Diagrama de colaboración de la realización del Caso de Uso Reiniciar cuenta estudiante.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Reiniciar cuenta estudiante” para restablecer la contraseña de un usuario estudiante (1) véase figura 3.21, y cambiarla al momento por su número de cédula, véase figura 3.21, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar al estudiante (2,3), una vez ingresada la información (4) el objeto *Form buscar estudiante* solicita al objeto *Gestor de usuario* que busque la información solicitada, lo cual pide al objeto *Gestor de consulta* (5,6,7,8), una vez obtenida la información el objeto *Gestor de usuario* muestra el formulario para que se reinicie la cuenta (9), luego el objeto *Form reiniciar cuenta* solicita al objeto *Gestor de usuario* que actualiza la cuenta, lo cual solicita al objeto *Gestor de actualización* que lleva a cabo

la solicitud (10,11,12,13,14). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.14 Generar Reporte de Inscripción.



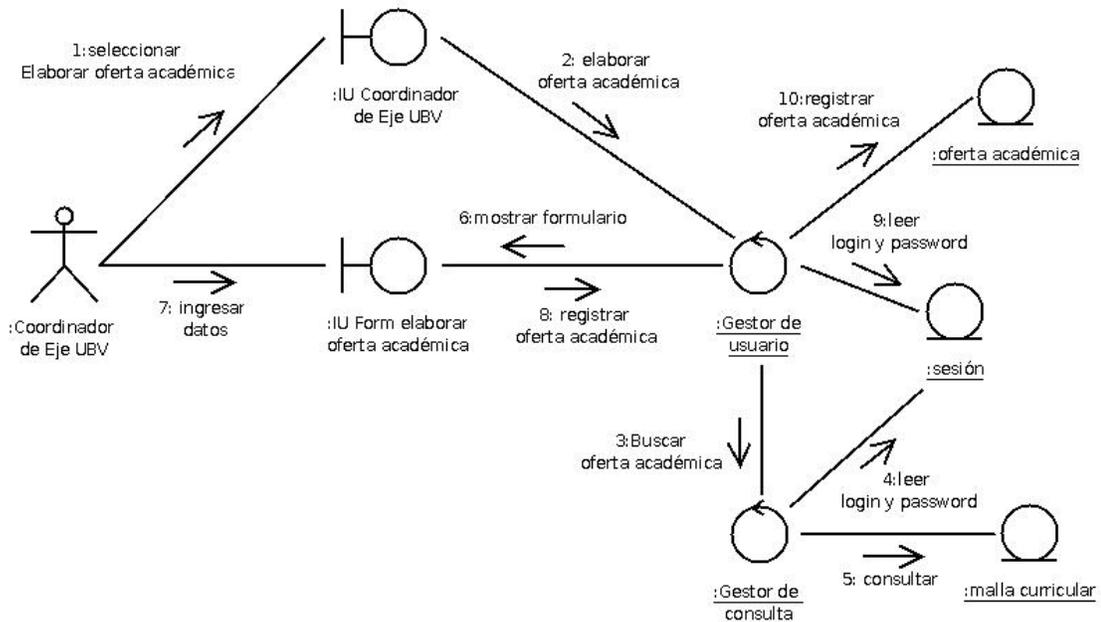
**Figura 3.22** Diagrama de colaboración de la realización del Caso de Uso Generar reporte de inscripción.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Generar reporte de inscripción” para obtener un reporte de las inscripciones en un período lectivo (1), véase figura 3.22, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que genere el reporte de las inscripciones (2), para los cual éste solicita al objeto *Gestor de consulta* que lleve a cabo la solicitud (3,4,5), una vez obtenida la información el objeto *Gestor de usuario* solicita al objeto *IU Form reporte de inscripciones* que muestre la

información a usuario (6,7). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.15 Elaborar Oferta Académica.



**Figura 3.23** Diagrama de colaboración de la realización del Caso de Uso Elaborar oferta académica.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Elaborar oferta académica” para ofertar las unidades curriculares de diferentes programas de formación el los estados que corresponda al Eje del coordinador (1) véase figura 3.23, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para carga la oferta académica en un período lectivo (2,3,4,5), una vez obtenida la información e ingresado los datos (6,7) objeto *Form elaborar oferta académica* solicita al objeto *Gestor de usuario* que lleve a cabo el registro de los datos (8,9,10). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.16 Modificar Oferta Académica.

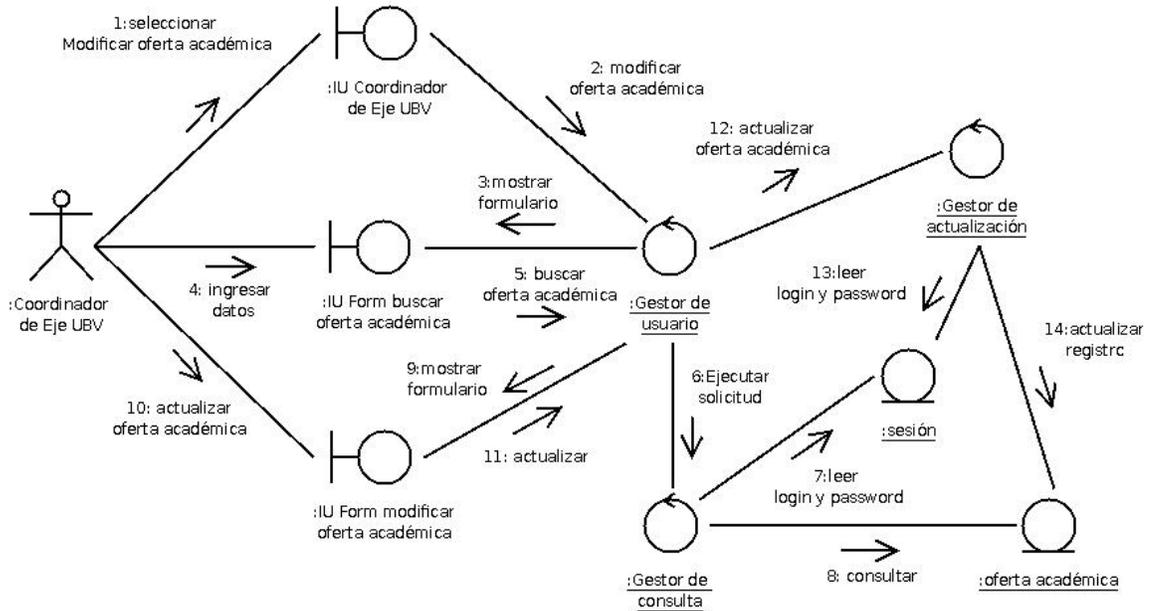
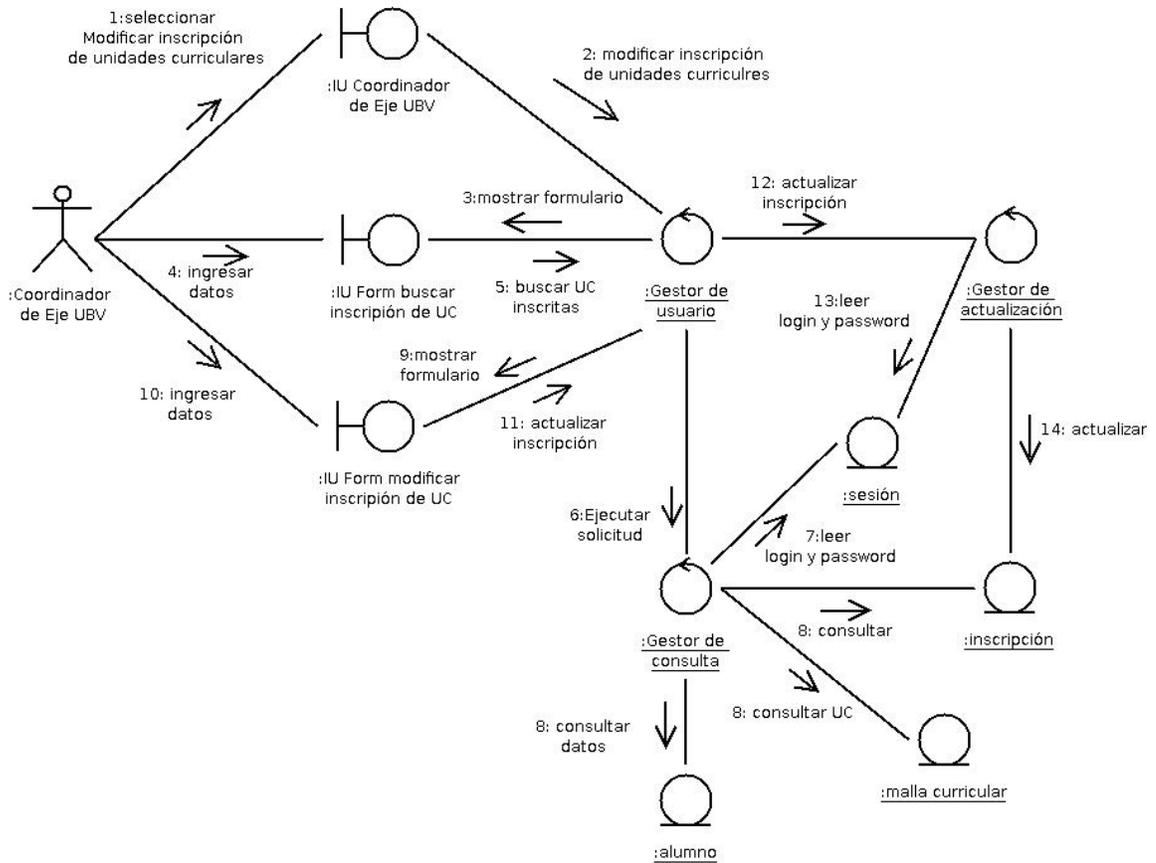


Figura 3.24 Diagrama de colaboración de la realización del Caso de Uso Modificar oferta académica.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar oferta académica” para modificar las unidades curriculares de los diferentes programas de formación de los estados asociados al eje del coordinador de Eje UBV (1) véase figura 3.24, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar la oferta académica (2,3), una vez ingresada la información (4,5) el objeto *Gestor de usuario* solicita la información al objeto *Gestor de consulta* que busque la información solicitada (6,7,8), una vez obtenida la información el objeto *Gestor de usuario* muestra el formulario para modificar la oferta académica a través del objeto *Form modificar oferta académica* (9), luego de actualizar la información el objeto *Form modificar oferta académica* solicita al objeto *Gestor de usuario* que actualice la información, lo cual realiza a través de objeto *Gestor de actualización* (10,11,12,13,14). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.17 Modificar inscripción de Unidades Curriculares.

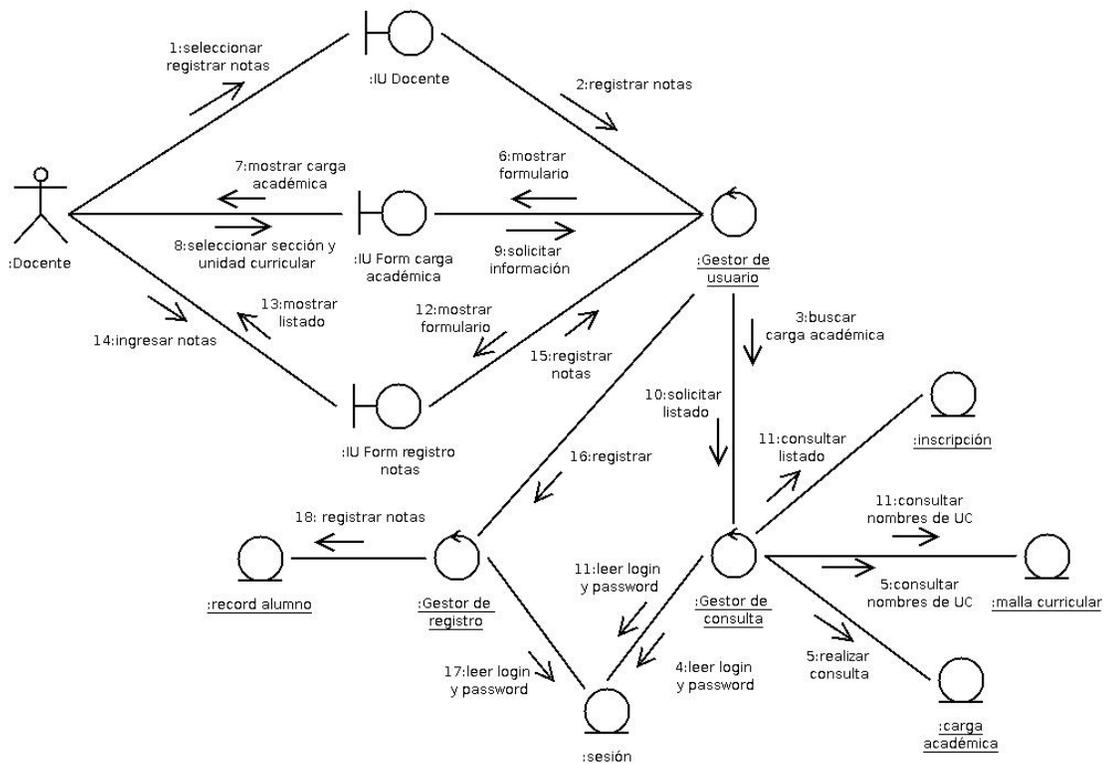


**Figura 3.25** Diagrama de colaboración de la realización del Caso de Uso Modificar oferta académica.  
Fuente: elaboración propia, año:2009.

El usuario selecciona la opción “Modificar inscripción de unidades curriculares” para modificar las unidades curriculares que inscribió un alumno en uno de los programas de formación de los estados asociados al eje del coordinador de Eje UBV (1) véase figura 3.25, el objeto *IU Coordinador de Eje UBV* solicita al objeto *Gestor de usuario* que muestre el formulario para modificar buscar la inscripción realizada (2,3), una vez ingresada la información (4) el objeto *Form buscar inscripción de UC* solicita la objeto *Gestor de usuario* que lleve a cabo la búsqueda lo cual solicita al objeto *Gestor de consulta* (5,6,7,8) una vez obtenida la información el objeto *Gestor de usuario* muestra un formulario para modificar el

registro de inscripción (9), una vez actualizada la información (10) el objeto *Form modificar inscripción de UC* solicita al objeto *Gestor de usuario* que actualice el registro, lo cual solicita al objeto *Gestor de actualización* que lleve a cabo la solicitud (11,12,13,14). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.18 Registrar Notas.



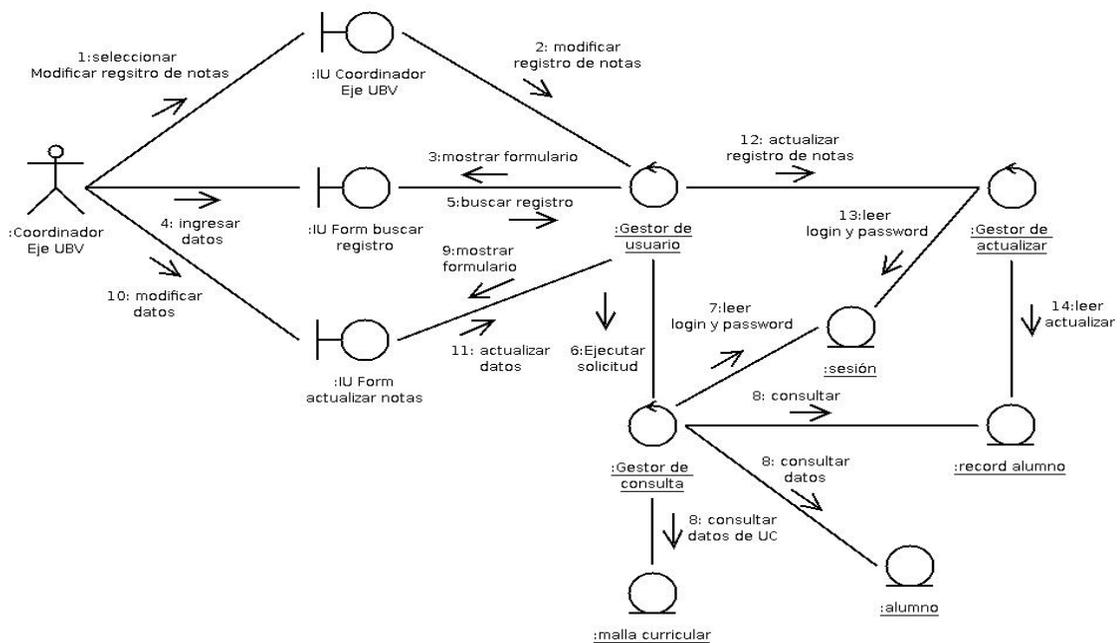
**Figura 3.26** Diagrama de colaboración de la realización del Caso de Uso Registrar notas.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar registrar notas” para registrar las calificaciones en las unidades curriculares asignadas (1) véase figura 3.26, el objeto *IU Docente* solicita al objeto *Gestor de usuario* que muestre el formulario para consultar la carga académica del docente (2,3,4,5), una vez obtenida la información el

objeto *Gestor de usuario* muestra el formulario con la carga académica asignada (6,7), una vez escogida la opción el objeto *Form carga académica* solicita al objeto *Gestor de usuario* que muestre el formulario para cargar las notas lo cual solicita al objeto *Gestor de consulta* que busque la información (8,9,10,11), una vez obtenida la información el objeto *Gestor de usuario* muestra el formulario para registrar las calificaciones (12,13), una vez ingresada las calificaciones el objeto *Form registro notas* solicita al objeto *Gestor de usuario* que registre la información (14,15,16,17,18). Cada vez el *Gestor de consulta* o *Gestor de registro* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.19 Modificar Registro de Notas.



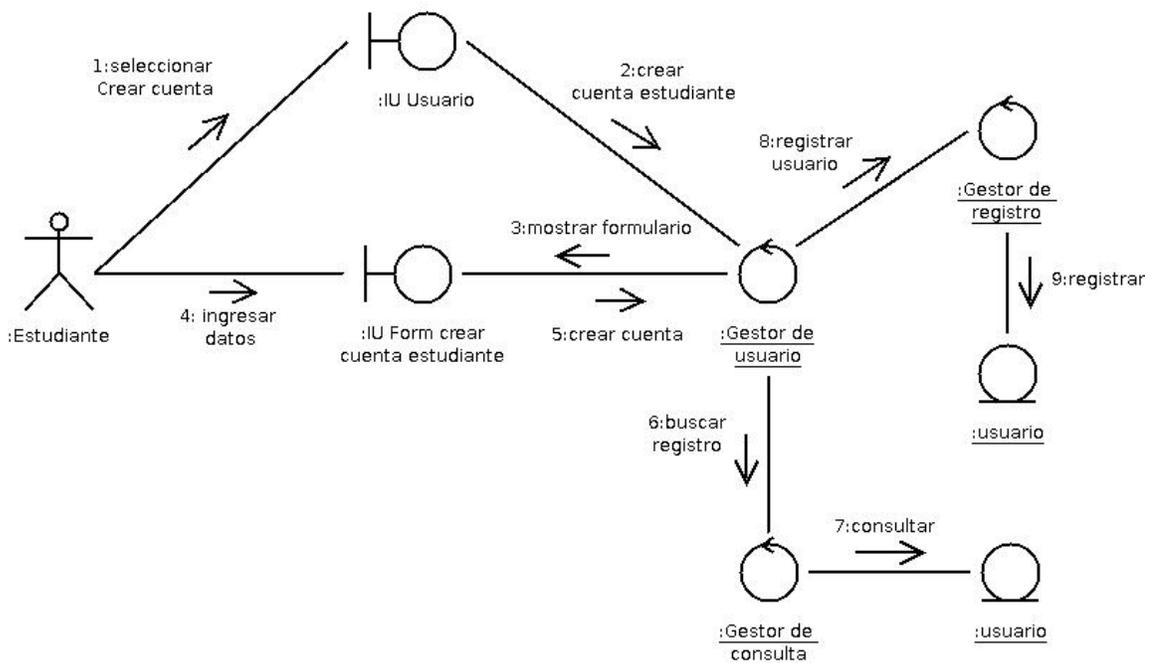
**Figura 3.27** Diagrama de colaboración de la realización del Caso de Uso Modificar registro de notas.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar registro de notas” para modificar los valores de las notas registradas a un estudiante (1) véase figura 3.27, el objeto *IU Docente* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar el

registro (2,3), una vez ingresada la información (4) el objeto *Form buscar registro* solicita al objeto *Gestor de usuario* que busque la información solicitada (5,6,7,8) una vez obtenida la información el objeto *Gestor de usuario* muestra el formulario para actualizar el registro (9), una vez modificado el registro el objeto *Form actualizar notas* solicita al objeto *Gestor de usuario* que actualice el registro, lo cual solicita al objeto *Gestor de actualización* que realiza la solicitud (10,11,12,13,14). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.20 Crear Cuenta Estudiante.



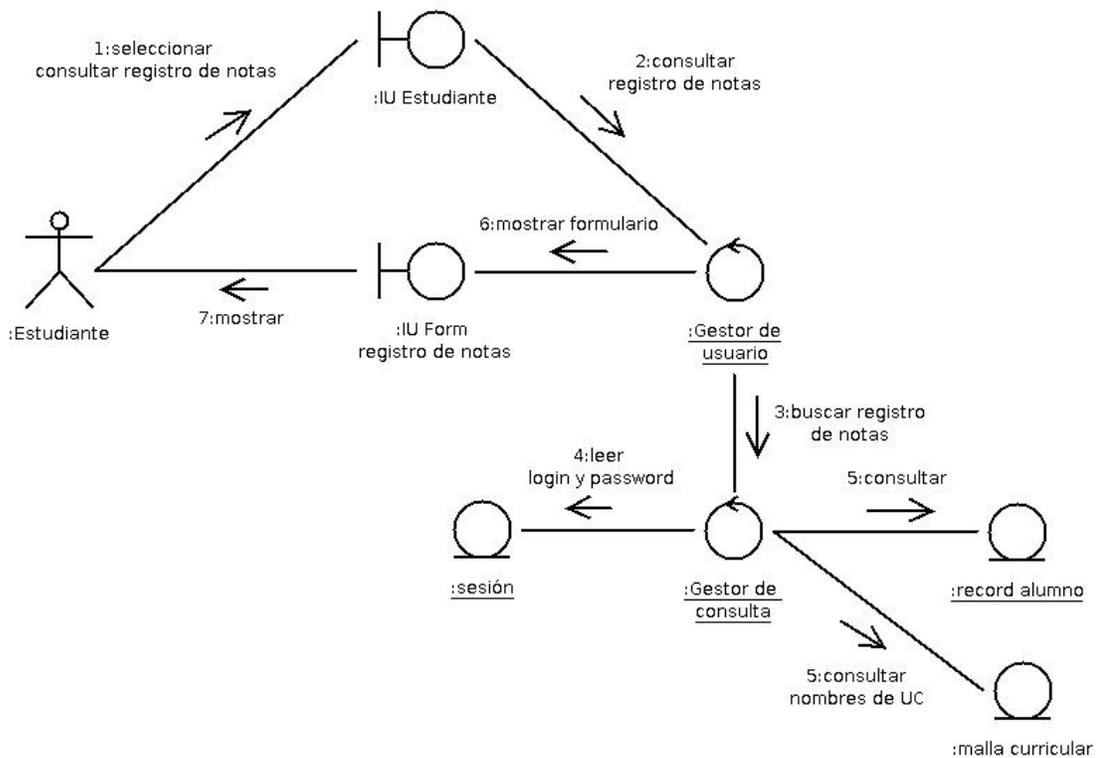
**Figura 3.28** Diagrama de colaboración de la realización del Caso de Uso Crear cuenta estudiante.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Crear cuenta Estudiante” en la pagina principal de la aplicación (1) véase figura 3.28, el objeto *IU Usuario* solicita al objeto *Gestor de usuario* que muestre el formulario para crear cuentas estudiante (2,3), una

vez ingresada la información (4) el objeto *Form crear cuenta estudiante* solicita al objeto *Gestor de usuario* que cree la cuenta del usuario, para lo cual el objeto *Gestor de usuario* solicita al objeto *Gestor de consulta* que consulte si el usuario esta registrado (5,6,7), una vez obtenida la información el objeto *Gestor de usuario* solicita al objeto *Gestor de registro* que lleve a cabo la transacción (8,9).

### 3.6.21 Consultar Registro de Notas.



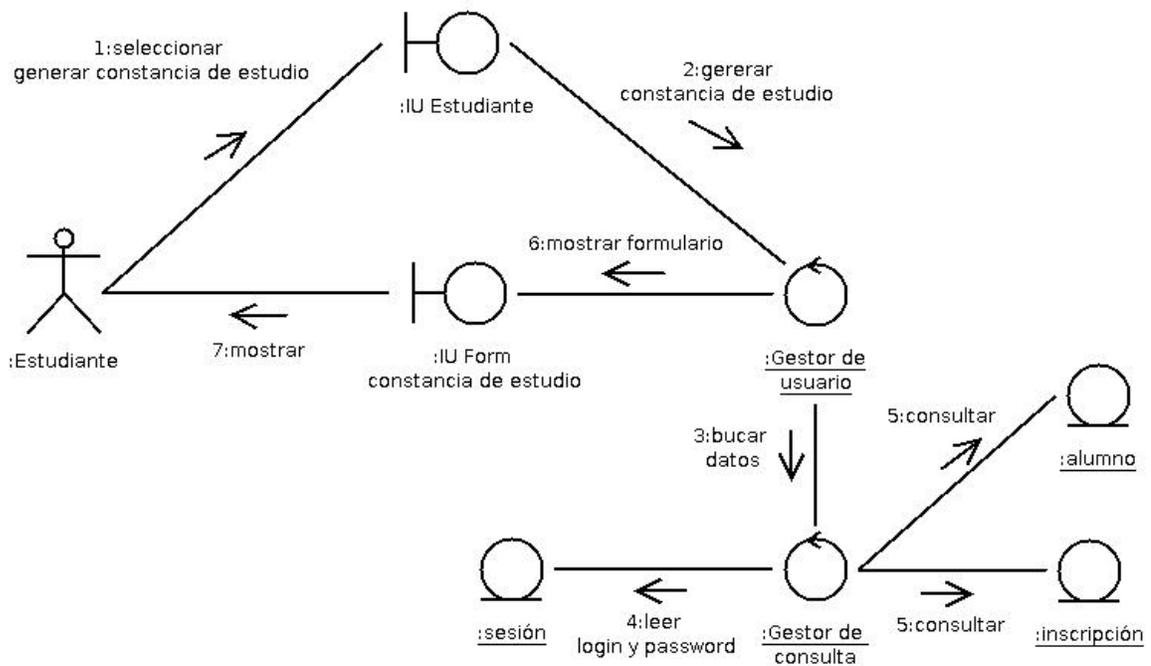
**Figura 3.29** Diagrama de colaboración de la realización del Caso de Uso Consultar registro de notas.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Consultar registro de notas” (1) véase figura 3.29, el objeto *IU Estudiante* solicita al objeto *Gestor de usuario* que muestre las notas registradas para del usuario (2), una vez que el objeto *Gestor de usuario* obtiene

la información (3,4,5) el objeto *Form registro de notas* muestra la información al usuario (6,7). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.22 Generar Constancias de Estudio.

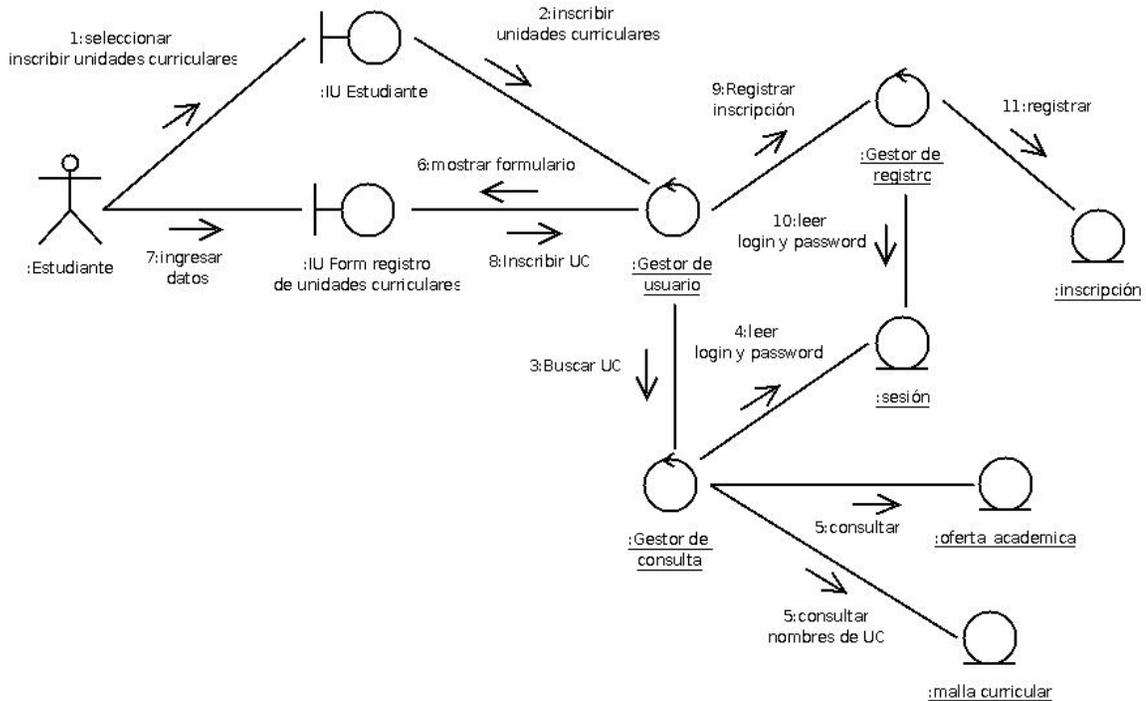


**Figura 3.30** Diagrama de colaboración de la realización del Caso de Uso Generar constancia de estudio.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Generar constancia de estudios” (1) véase figura 3.30, el objeto *IU Estudiante* solicita al objeto *Gestor de usuario* que genere el documento “constancia de estudios” (2), una vez que el objeto *Gestor de usuario* obtiene información (3,4,5) solicita al objeto *Form constancia de estudio* que muestre la información (6,7). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.23 Inscribir Unidades Curriculares.

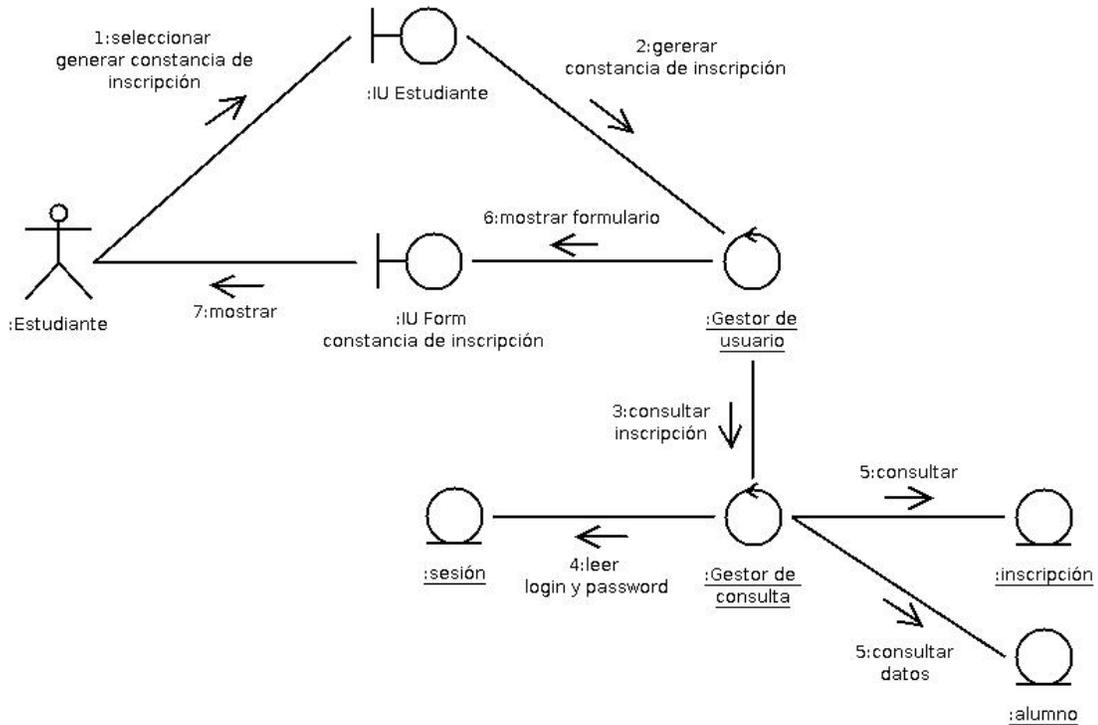


**Figura 3.31** Diagrama de colaboración de la realización del Caso de Uso Inscribir unidades curriculares.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Inscribir unidades curriculares” (1) lo cual le permitirá registrar las unidades curriculares que cursará en el período lectivo, véase figura 3.31, el objeto *IU Estudiante* solicita al objeto *Gestor de usuario* que muestre el formulario para el registro de unidades curriculares a inscribir (2,3,4,5), una vez obtenida la información el objeto *Gestor de usuario* muestra la oferta académica al usuario estudiante (6), una vez ingresada la información objeto *Form registro de unidades curriculares* solicita al objeto *Gestor de usuario* que registre la inscripción, lo cual solicita al objeto *Gestor de registro* que lleve a cabo la solicitud (7,8,9,10,11). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.24 Generar Constancia de inscripción.

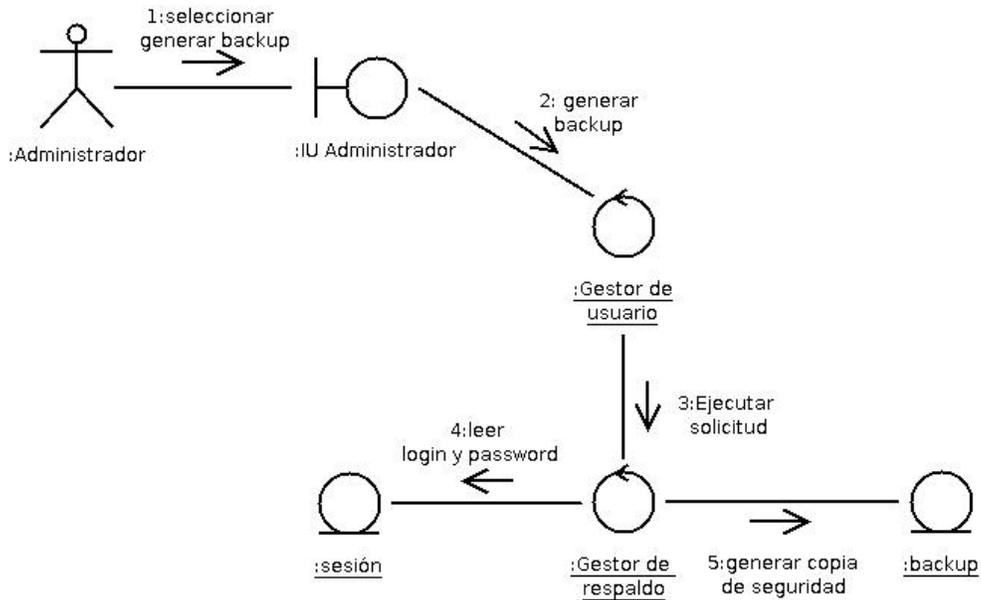


**Figura 3.32** Diagrama de colaboración de la realización del Caso de Uso Generar constancia de inscripción.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Generar constancia de estudios” (1) véase figura 3.32, el objeto *IU Estudiante* solicita al objeto *Gestor de usuario* que muestre el documento “constancia de inscripción” (2), el objeto *Gestor de usuario* solicita al objeto *Gestor de consulta* que consulte la inscripción (3,4,5), una vez verificada la información el objeto *Gestor de usuario* muestra la información a través del objeto *Form constancia de inscripción* (6,7). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.25 Generar Backup.

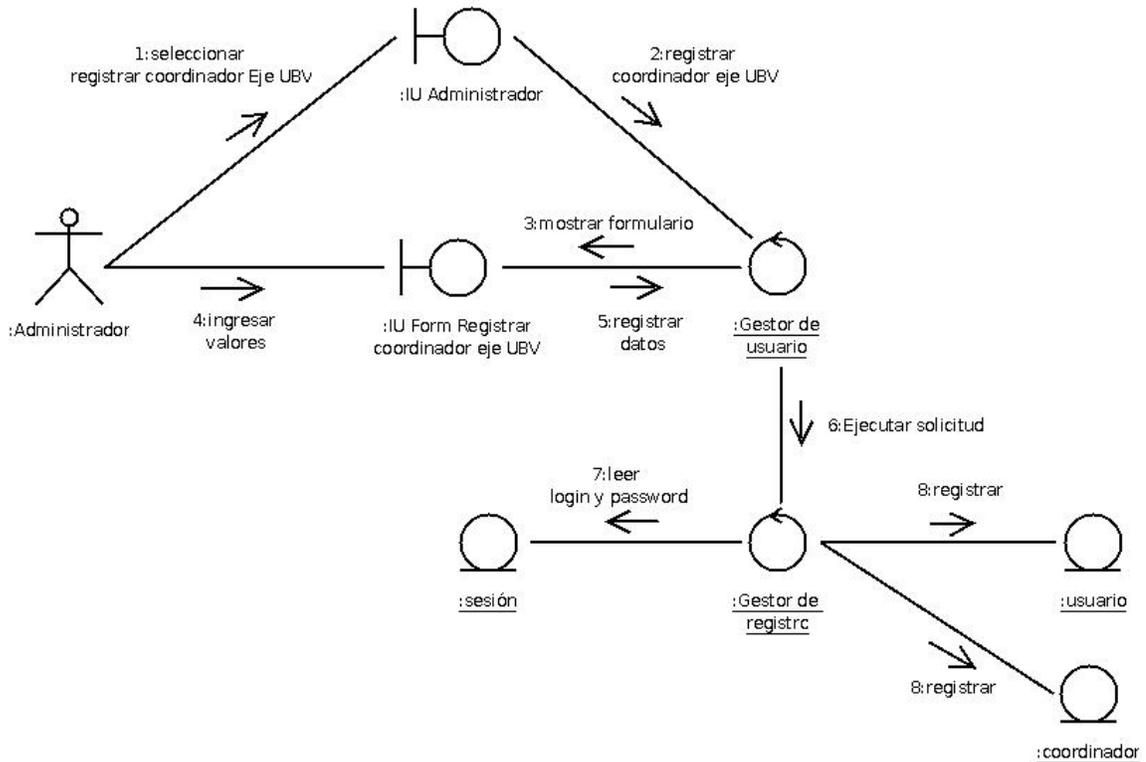


**Figura 3.33** Diagrama de colaboración de la realización del Caso de Uso Generar backup.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Generar backup” (1) véase figura 3.33, el objeto *IU Administrador* solicita al objeto *Gestor de usuario* que genera una copia de seguridad del sistema (2), el objeto *Gestor de usuario* solicita al objeto *Gestor de respaldo* que lleve a cabo la acción (3,4,5). Cada vez el *Gestor de consulta* o *Gestor de respaldo* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.26 Registrar Coordinador de Eje UBV.

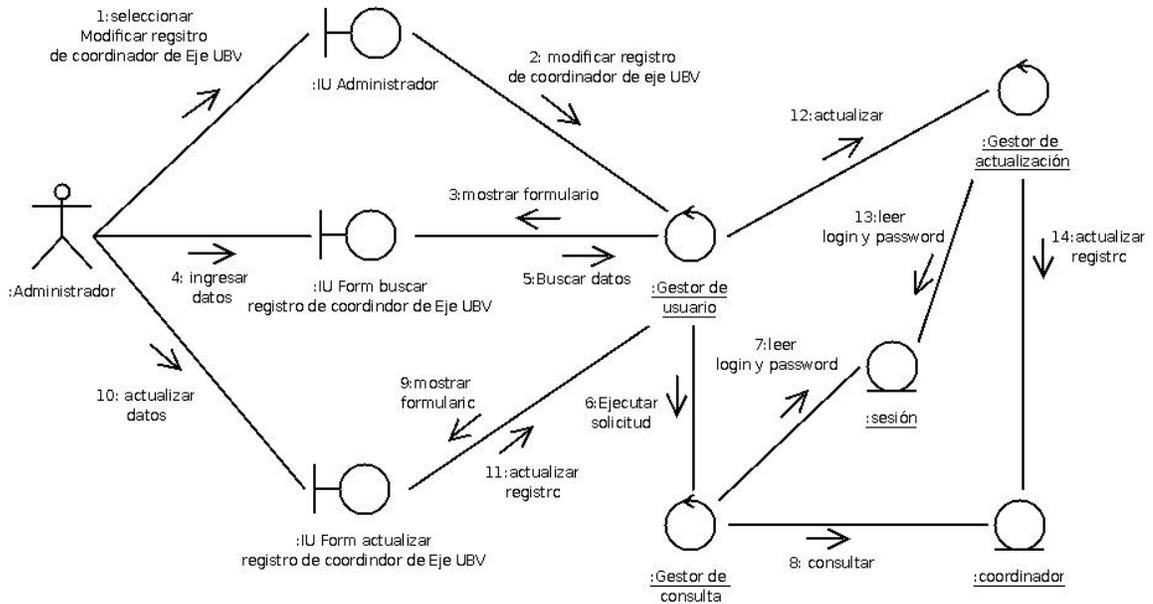


**Figura 3.34** Diagrama de colaboración de la realización del Caso de Uso Registrar coordinador de Eje UBV.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Registrar coordinador eje UBV” para llevar a cabo el registro de un coordinador (1) véase figura 3.34, el objeto *IU Administrador* solicita al objeto *Gestor de Usuario* que muestre el formulario donde se registra el coordinador (2,3), una vez que el usuario ingresa los valores (4) el objeto *Form registrar coordinador* solicita al objeto *Gestor de sesión* que registre los datos lo cual solicita al *Gestor de registro* (5,6,7,8). Cada vez el *Gestor de consulta* o *Gestor de respaldo* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### 3.6.27 Modificar Registro de Coordinadores de Eje UBV.



**Figura 3.35** Diagrama de colaboración de la realización del Caso de Uso Modificar registro de coordinador de Eje UBV.

**Fuente:** elaboración propia, año:2009.

El usuario selecciona la opción “Modificar registro de coordinador de Eje UBV” para modificar los datos de un coordinador de Eje UBV (1) véase figura 3.35, el objeto *IU Administrador* solicita al objeto *Gestor de usuario* que muestre el formulario para buscar los datos de un coordinador de Eje UBV (2,3), una vez ingresada la información (4) el objeto *Form buscar registro coordinador eje UBV* solicita al objeto *Gestor de usuario* que busque los datos solicitados lo cual solicita al objeto *Gestor de consulta* (5,6,7,8), una vez obtenida la información el objeto *Gestor de usuario* muestra un formulario para actualizar el registro del coordinador eje UBV (9), una vez actualizado el registro, el objeto *Form actualizar registro coordinador eje UBV* solicita al objeto *Gestor de usuario* que actualiza el registro (10,11), para lo cual solicita al objeto *Gestor de actualización* que lleve a cabo la solicitud (12,13,14). Cada vez el *Gestor de consulta* o *Gestor de actualización* lleven a cabo alguna operación deben obtener información del objeto entidad *sesión*.

### **3.7 Evaluación de la fase de análisis y definición de requerimientos.**

En esta fase se estableció el análisis inicial del negocio, el cual consistió en un estudio del sistema informático existente en la organización Misión Sucre, para dicho análisis, se realizó un Modelo del Negocio representado por un diagrama de Casos de Uso, lo que permitió conocer los procesos involucrados en la problemática, así como conocer los objetos del dominio.

Como punto de partida para la captura de requerimientos, se elaboró un Modelo de Casos de Uso como producto de conversaciones con la directiva de la organización y los usuarios de sistema existente. Esto permitió crear un marco de referencia para hallar soluciones al problema planteado.

Una vez establecidas las condiciones necesarias para lograr los objetivos, se realizó un análisis sobre la viabilidad de llevar a cabo la realización el proyecto del sistema.

Posteriormente se elaboró la realización de los Casos de Uso-Análisis con diagramas de colaboración y sus requisitos especiales para el sistema propuesto mediante clases de análisis y sus responsabilidades, lo que permitió representar como se lleva a cabo y se ejecuta un Caso de Uso determinado en término de éstas.

Todo lo anterior viene a conformar un modelo de análisis que será la entrada fundamental para las actividades del diseño que se desarrollarán en el capítulo siguiente.

## CAPITULO 4

### DISEÑO DEL SISTEMA PROPUESTO

En esta fase de diseño del ciclo de vida del software, se busca adquirir una mejor comprensión de los aspectos relacionados con los requisitos funcionales, así como capturar las interfaces entre los subsistemas, gestión de transacciones, etc. El análisis y definición de los requerimientos vistos en el capítulo anterior proporcionó una comprensión detallada de éstos y permitió establecer una estructura para el diseño del sistema.

El objetivo a lograr, es crear las clases del diseño, éstas realizarán los casos de uso y sus objetos, incluyendo aspectos como: atributos, operaciones, asociaciones, métodos, etc, las clases de diseño se corresponden a las clases creadas en el lenguaje de programación, igualmente se realizarán diagramas de secuencia que muestren los objetos el escenario ordenados en secuencia temporal así como los mensajes intercambiados entre ellos.

Las clases de interfaz y las clases de control serán diseñadas con el estereotipo que proporciona la extensión del UML denominada *WAE (Web Application Extension)*, la cual permite modelar aplicaciones con elementos específicos de la arquitectura de un entorno Web. Estas clases representan páginas Web escritas en lenguaje HTML procesos realizados del lado del servidor o script escritos en lenguaje JavaScript.

Para el diseño de la Base de datos se utilizará del modelo relacional, el diseño de la interfaz de usuario se realizará tomando en cuenta un diseño centrado en el usuario, lo que implica involucrar desde el comienzo a los usuarios para comprender

claramente sus necesidades y lograr que su interacción con el sistema se realice en forma amigable, de esta manera se optimiza la interfaz tomando en cuenta cómo el usuario

#### **4.1 Diseño de casos de uso-análisis.**

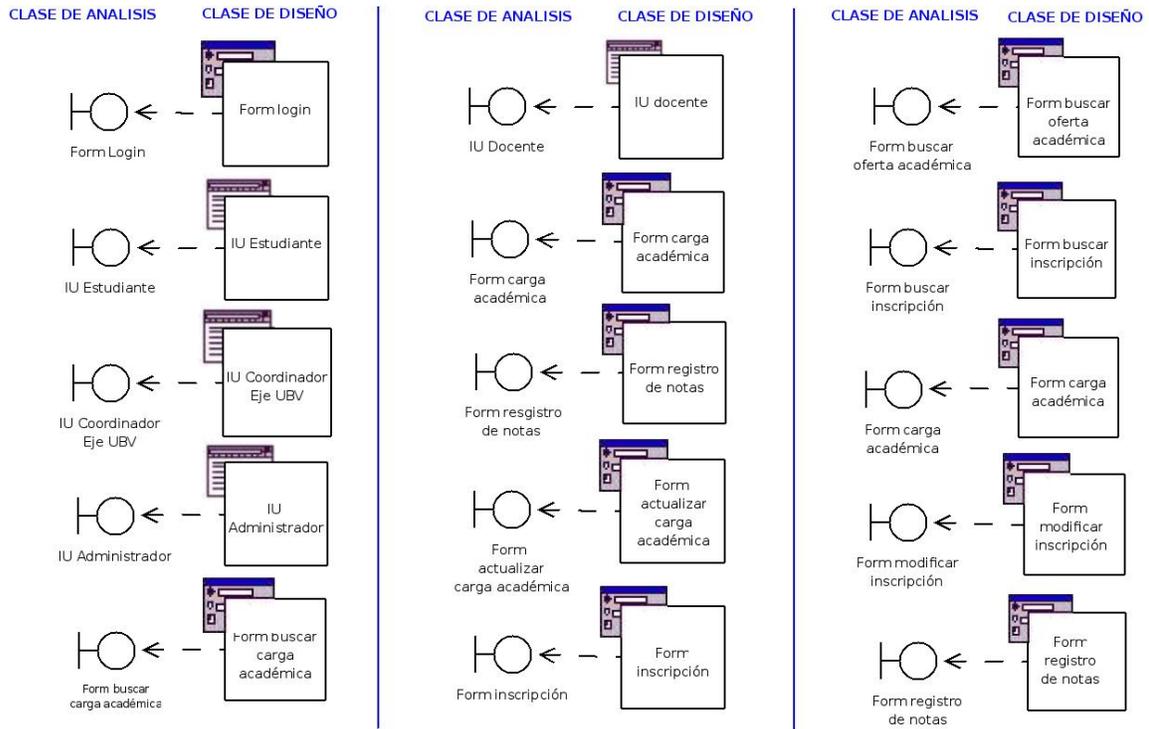
En este paso se elaboraran las clases cuyas instancias son necesarias para la realización de los Caso de Uso, para ello se hará referencia al las clases del análisis que participaron en la correspondiente realización de los Casos de Uso, (*véase* sección 3.5). Con lo cual se describirá como se realiza un caso de uso en términos de interacciones entre objetos con sus respectivos atributos, operaciones y asociaciones.

##### **4.1.1 Identificación de las Clases de Diseño Participantes.**

Para la identificación de las clases de diseño se tomará como referencia las clases de análisis obtenidas en los diagramas de colaboración del capítulo anterior (*véase*, sección 3.5).

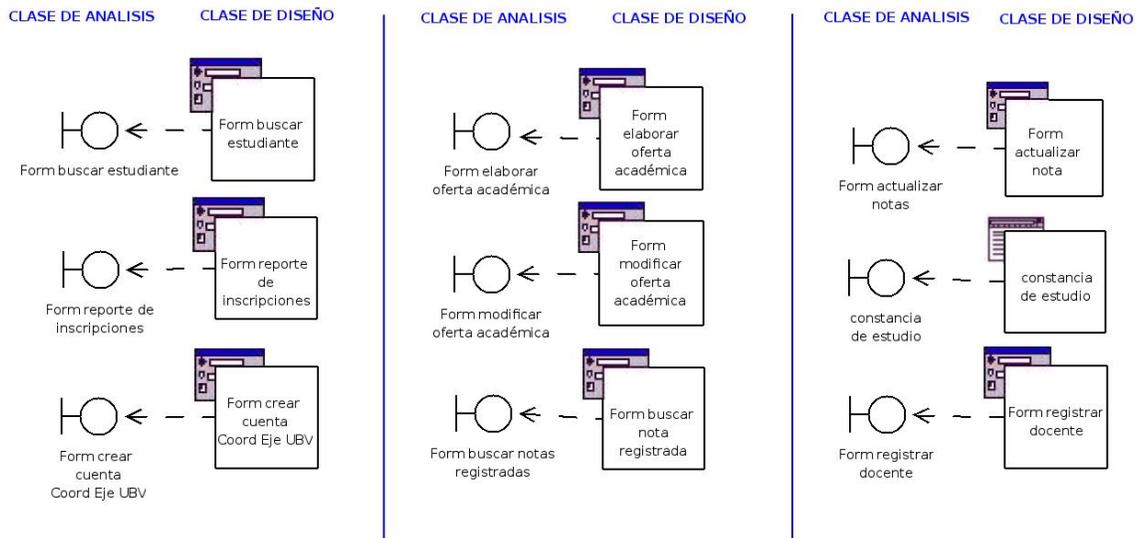
Las clases de interfaz se diseñaran teniendo en cuenta la tecnología empleada para el modelado de una aplicación Web, estas clases por lo general se convertirán en páginas Web del cliente, igualmente para las clases de control se empleará un diseño para el modelado de aplicaciones Web, éstas representarán procesos que lleva a cabo el servidor para realizar las diferentes transacciones que solicite la interfaz de usuario, o realizar procesos asociados a las tareas.

Las clases de entidad representan la información que maneja el sistema así como la información persistente lo que implica el uso de tecnología de base de datos. En las figuras 4.1a, 4.1b, 4.1c, 4.2, 4.3 y 4.4 de muestra la identificación de cada una de las clases utilizadas en el modelo de análisis y su respectiva clase para el modelo de diseño utilizando el estereotipo correspondientes por la notación WAE.



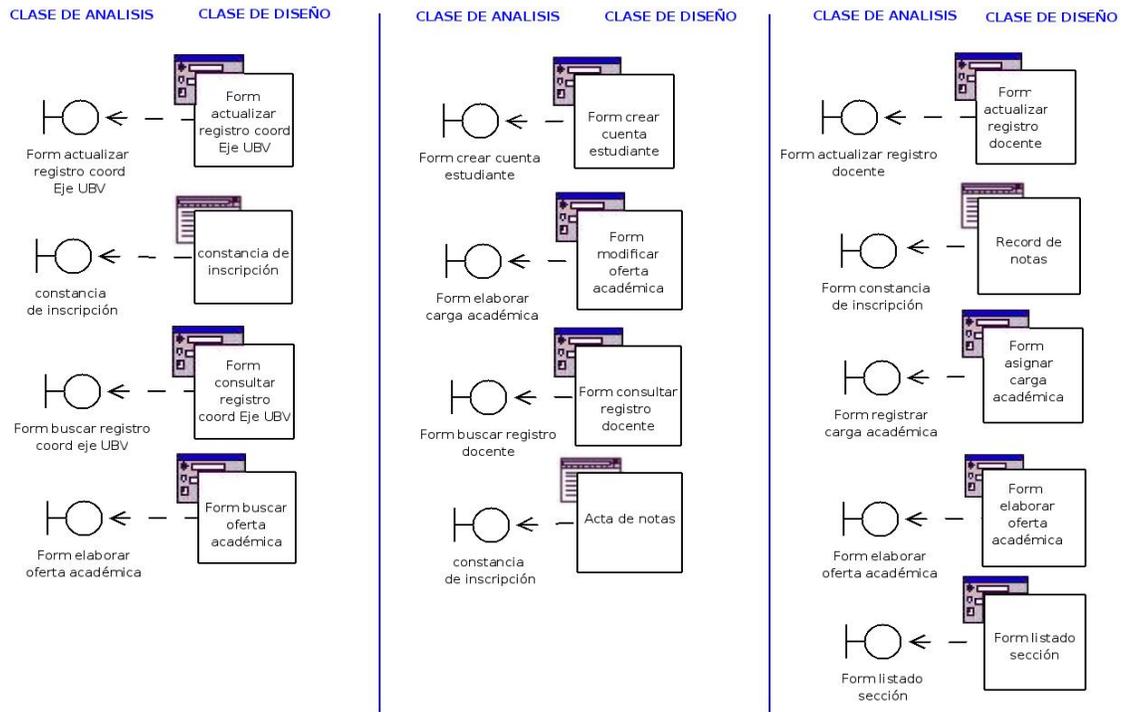
**Figura 4.1a** Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis.

Fuente: elaboración propia, año:2009.



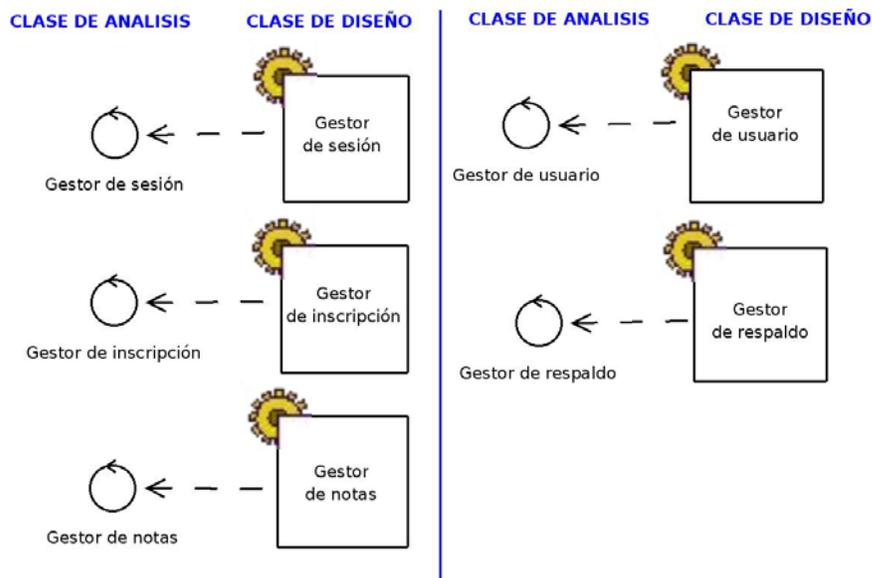
**Figura 4.1b** Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis.

Fuente: elaboración propia, año:2009.



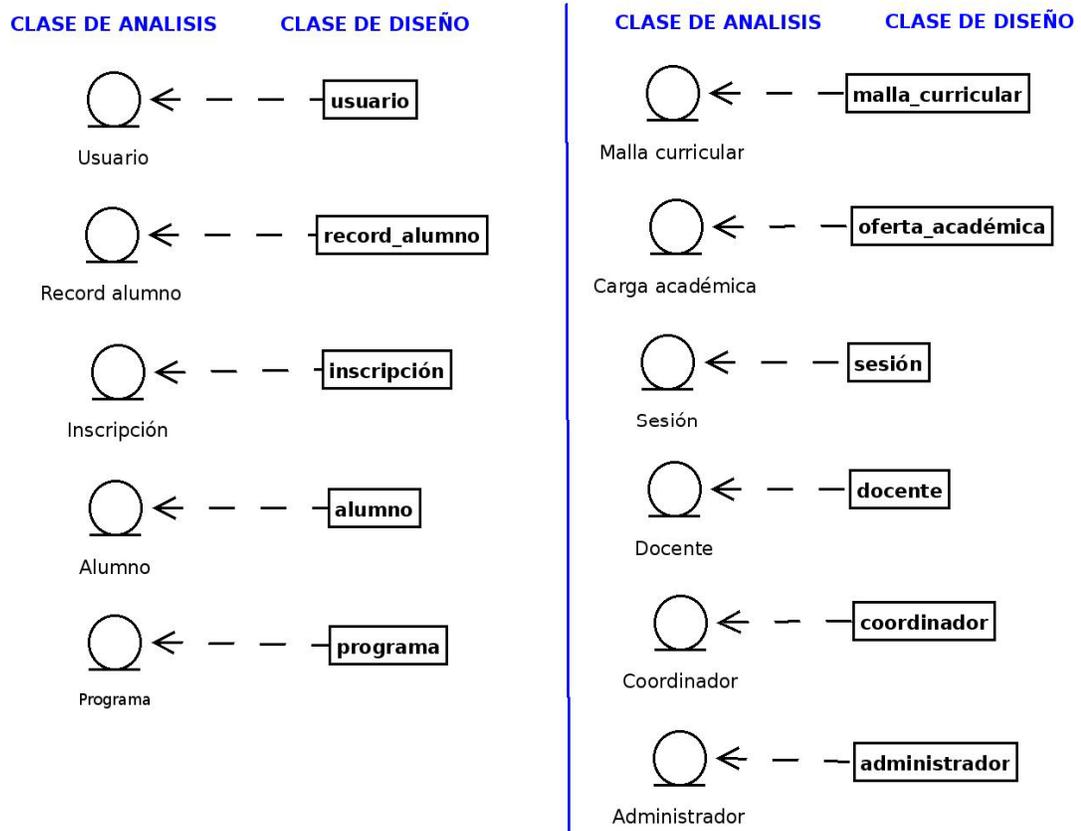
**Figura 4.1c** Identificación de Clases de Diseño a partir de Clases de Interfaz del Modelo de Análisis.

**Fuente:** elaboración propia, año:2009.



**Figura 4.2** Identificación de Clases de Diseño a partir de Clases de Control del Modelo de Análisis.

**Fuente:** elaboración propia, año:2009.

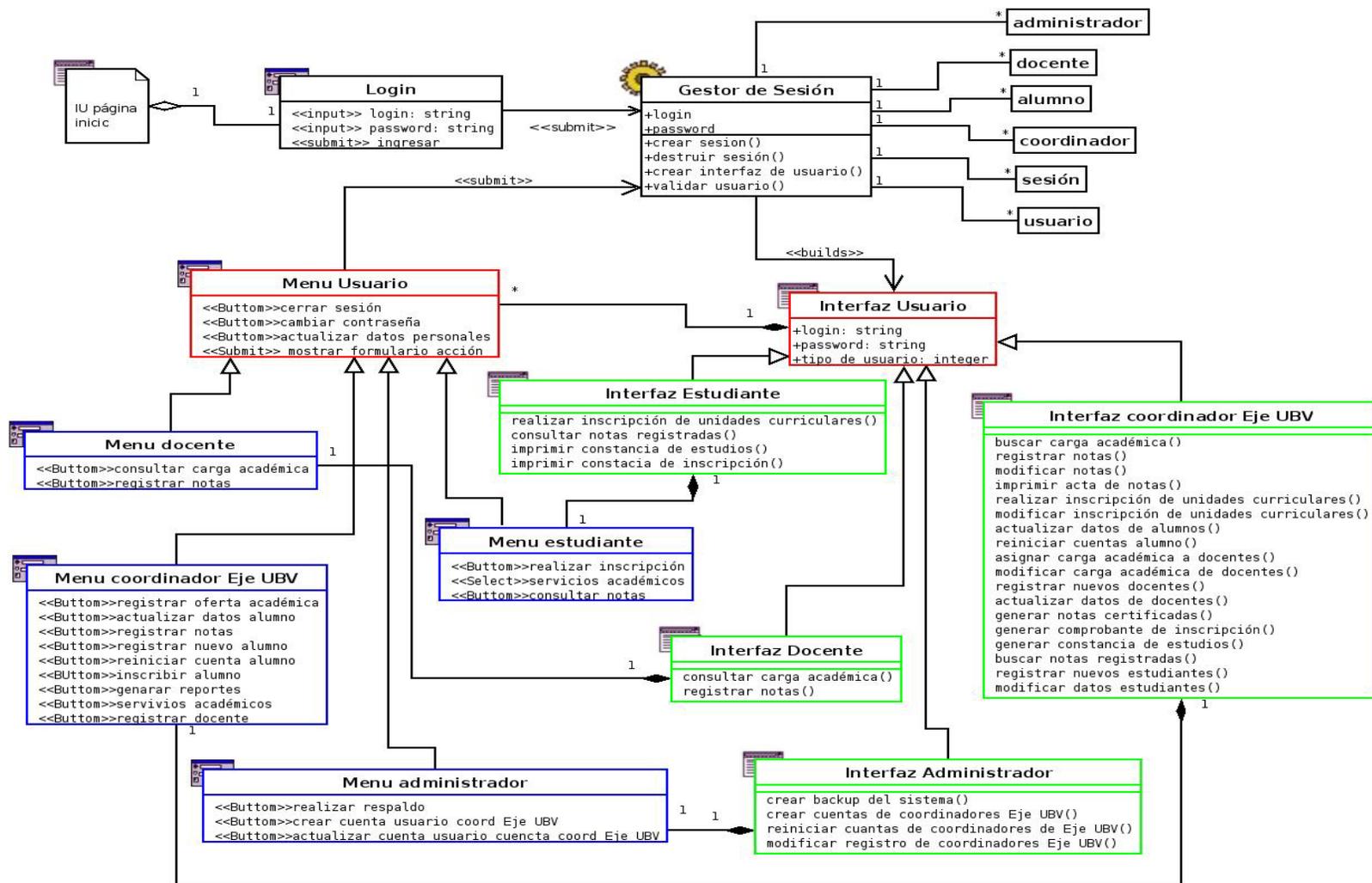


**Figura 4.3** Identificación de Clases de Diseño a partir de Clases de Entidad del Modelo de Análisis.

**Fuente:** elaboración propia, año:2009.

#### 4.1.2 Diagrama de Clases involucradas en el Inicio de Sesión y Salida de Sesión.

La clase interfaz “página inicio” posee una relación por agregación con la clase formulario “login”, véase figura 4.4, esta clase formulario es utilizada para autenticar los diferentes usuarios del sistema, los datos que captura son procesados por la clase “Gestor de sesión”, luego éste último puede construir la interfaz de usuario dependiendo del tipo de usuario, la clase interfaz “Interfaz de usuario” es una generalización de las clases de interfaz “Interfaz docente”, “Interfaz estudiante”, “Interfaz coordinador Eje UBV” e “Interfaz administrador”.



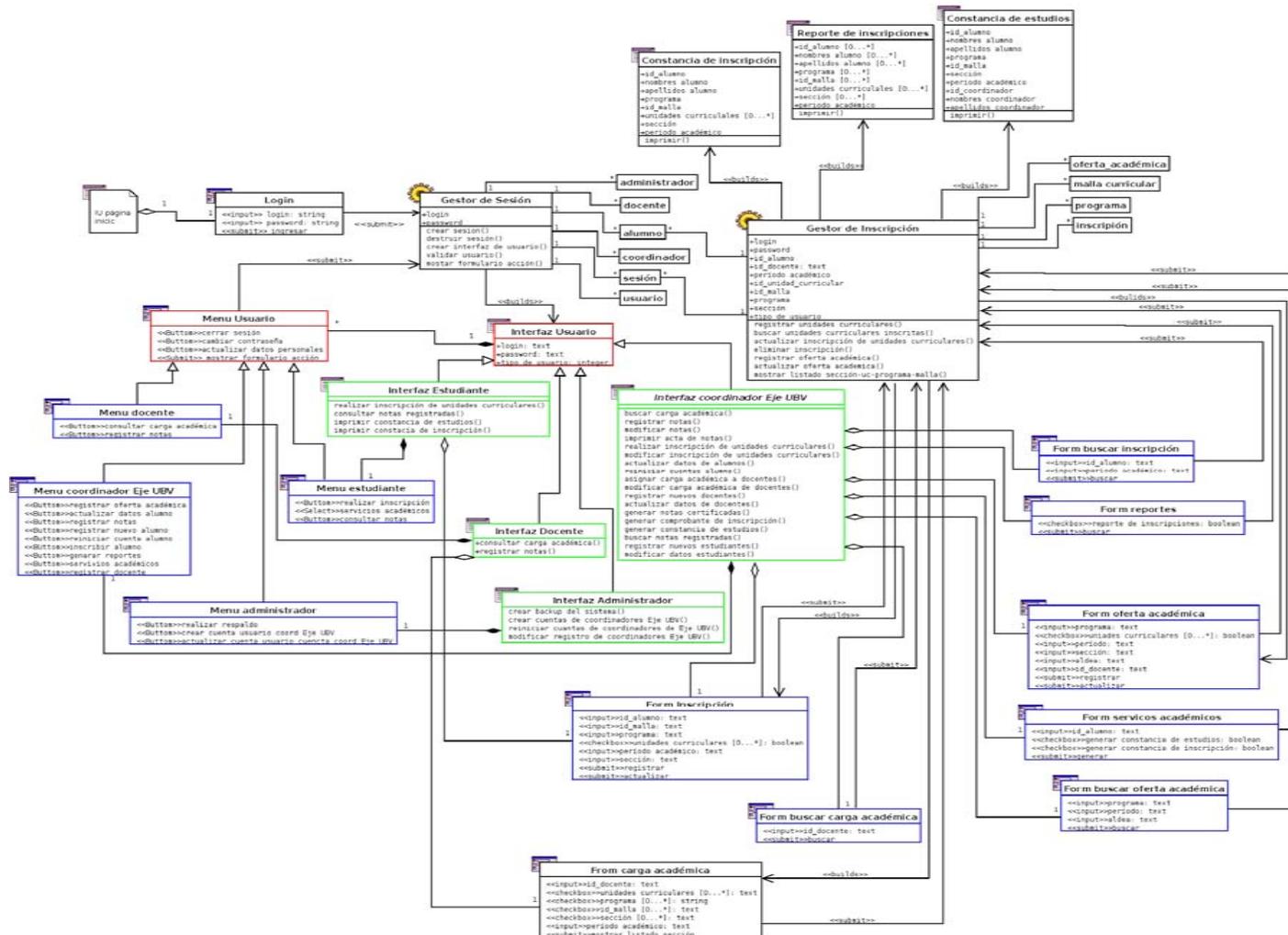
**Figura 4.4** Diagrama de Clases involucradas en los Casos de uso Inicio y Salida de Sesión.  
**Fuente:** elaboración propia, año:2009.

La clase “menú usuario” es una generalización de las clases “menú docente”, “menú estudiante”, “menú coordinador Eje UBV” y “menú administrador”, esta clase “menú usuario” posee una relación por composición con la clase “interfaz de usuario”, ésta envía información a la clase de control “Gestor de sesión” para activar las diferentes opciones de los menús de las interfaces. Cada clases de interfaz posee una relación por composición con la clase menú que le corresponde al tipo de usuario. Una vez obtenido el tipo de usuario, se puede acceder a las clases de entidad relacionadas con ese tipo de usuario a través de la clase Gestor de sesión.

#### **4.1.3 Diagrama de Clases involucradas en el Proceso de Inscripción de Unidades Curriculares.**

En la figura 4.5 se observan la clases que dan soporte a los procesos de inscripción de las unidades curriculares, en este proceso se involucran directamente la clase interfaz “Interfaz estudiante” y la clase interfaz “Interfaz coordinador Eje UBV”.

La clase formulario “Form inscripción” posee una relación por agregación con las clases antes mencionadas. Las clases formulario “Form modificar inscripción”, “Form buscar inscripción”, “Form elaborar oferta académica”, “Form asignar carga académica” y “Form buscar oferta académica” poseen una relación por agregación con la clase interfaz “Interfaz coordinador Eje UBV”, las cuales dan soporte para los diferentes procesos relacionados con la actividad, estas clases envían información a la clase de control “Gestor de inscripción”, el cual puede acceder a las diferentes clase de entidad, para generar los documentos como las constancias de estudio y constancias de inscripción a través de las clases interfaz “Constancia de inscripción” ,”Reporte de inscripciones” y “Constancia de estudios”. Para que la clase de control “Gestor de inscripción” lleve a cabo las diferentes tareas, éste debe acceder a la sesión iniciada por el usuario, con el fin de validar que las transacciones solicitadas provienen de un usuario registrado del sistema.



**Figura 4.5** Diagrama de Clases involucradas en los Procesos de Inscripción de Unidades Curriculares.  
**Fuente:** elaboración propia, año:2009.

La clase interfaz “Form inscripción” posee una relación por agregación con la clase interfaz “Interfaz Estudiante” y la interfaz “Interfaz coordinador Eje UBV”, esta clase es utilizada por los dos usuario para realizar inscripciones de unidades curriculares.

Las clases de interfaz “Form inscripción”, “Form carga académica”, “Form asignar carga académica”, “Form modificar inscripción” y “Form Elaborar oferta académica” son construidas por la clase de control “Gestor de inscripción”, al momento de ser solicitadas por la interfaz respectiva, estas clases antes de iniciar la iteración con el usuario deben contener información que es obtenida de las clases de entidad.

#### **4.1.4 Diagrama de Clases involucradas en el Proceso de Registro de notas.**

El la figura 4.6 se muestran las clases involucradas el procesos de registro de notas, las clases formulario “Form registro de nota”, “Form actualizar nota”, “Form eliminar nota” poseen una relación por agregación con la clase Interfaz “Coordinador Eje UBV”, estas clases formularios envían información la la clase de control “Gestor de notas” para llevar a cabo las diferentes operaciones relacionadas con el registro de notas.

Por otro lado la clase interfaz “Interfaz docente” posee una relación por agregación con la clase formulario “Form carga académica”, esta clase formulario envía datos a la clase de control “Gestor de carga académica”, el cual a su vez construye una interfaz a través de la clase interfaz “From listado sección”, esta última posee una relación por agregación con la clase interfaz “Interfaz Docente”, la cual envía la información a registrar a la clase de control “Gestor de notas”.

Los procesos de las diferentes transacciones para el registro de notas son realizados por la clase de control “Gestor de notas”, esta clase de control genera los documentos que son solicitados por las clases de interfaz que lo requieran a través de las clases de interfaz “Record de notas” y “Acta de notas”.

La clase de control “Gestor de notas” y “Gestor de carga académica” necesitan acceder a la información registrada en la sesión, véase que en la figura (4.7) existe una relación de uno a muchos, entre la clase entidad “sesión” y la clase de control “Gestor de carga académica”, a través de esta relación se puede obtener la información del usuario que esta interactuando con el sistema, igualmente le permitirá acceder a las clases de entidad necesarias para realizar transacciones en los procesos.

La clase formulario “Form buscar nota registrada” posee una relación por agregación con las clases interfaz “Interfaz coordinador Eje UBV”.

#### **4.1.5 Diagrama de Clases involucradas en el Proceso de Administración de Usuarios.**

Las relaciones en el diagrama de clases de la figura 4.7 permiten la administración de los diferentes tipos de usuarios del sistema, la clase interfaz “Interfaz coordinador Eje UBV” se encarga de administrar los usuarios docente, mientras que la clase interfaz “Interfaz administrador” los usuarios coordinadores Eje UBV. Las diferentes clases formulario que dan soporte a la interfaz “Interfaz coordinador Eje UBV” poseen una relación por agregación con ésta, igualmente las clases formulario que dan soporte a la clase interfaz “Interfaz administrador” poseen una relación por agregación con en ésta.

Las clases formulario que dan soporte a la administración de los usuarios envían información a la clase de control “Gestor de usuarios”, ésta al igual que otras clases de control vistas en anteriores diagramas de clases necesita acceder a la sesión para obtener información del usuario que está interactuando en ese momento, lo cual permite acceder a las clases de entidad necesarias para llevar a cabo los procesos.

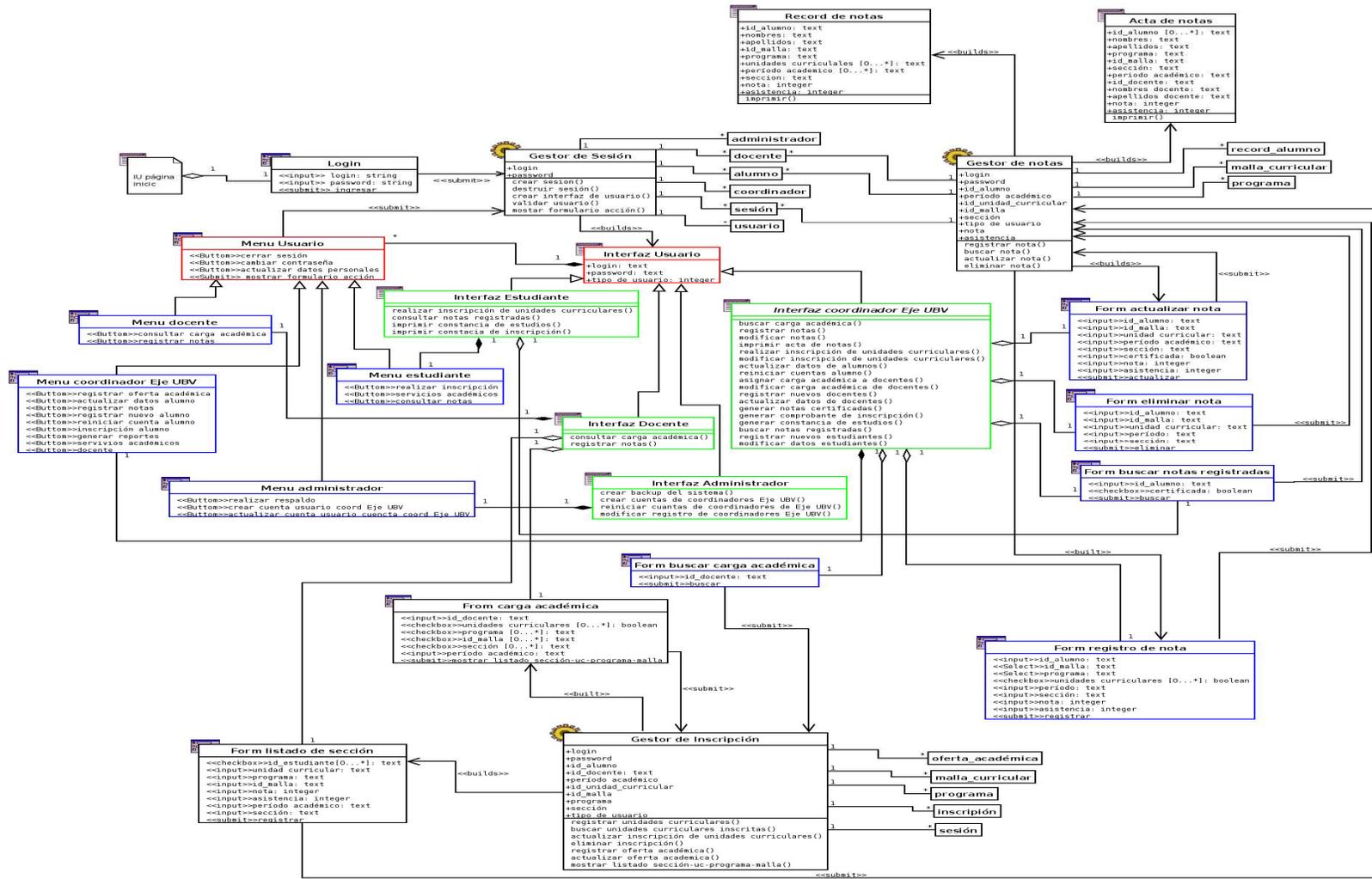
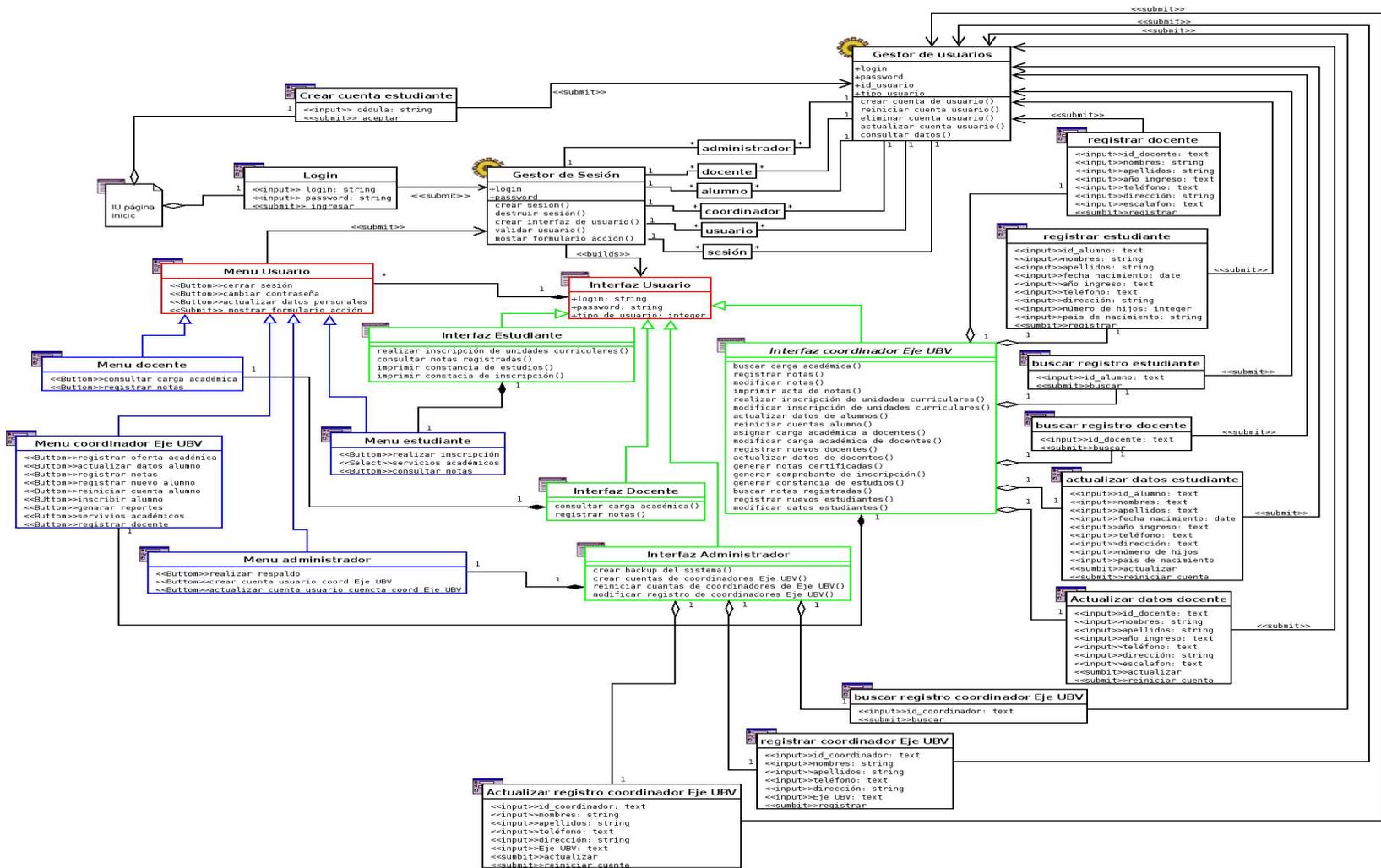


Figura 4.6 Diagrama de Clases Involucradas en los Procesos de Registro de Notas.  
Fuente: elaboración propia, año:2009.



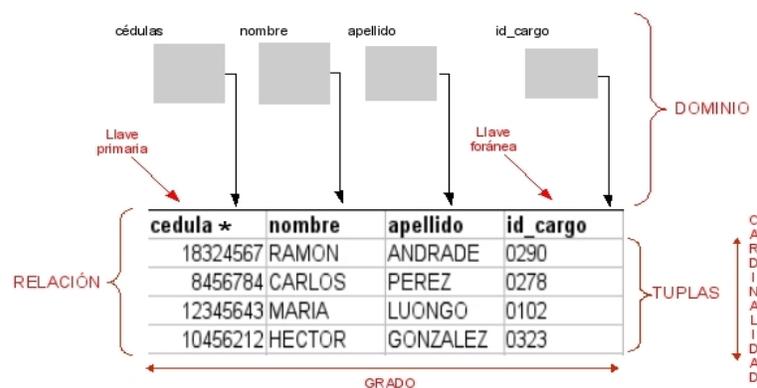
**Figura 4.7** Diagrama de Clases Involucradas en los Procesos de Administración de Usuarios.

Fuente: elaboración propia, año:2009.

## 4.2 Diseño de la base de datos.

El modelo relacional fue propuesto originariamente en la década de los 70 por Edgar Frank Codd (1970), en este modelo la base de datos es percibida por el usuario como un conjunto de tablas, esta percepción es a nivel lógico ya que a nivel físico puede estar implementada en distintas estructuras de almacenamiento, el modelo se caracteriza porque permite unir la información que aparece en varias tablas. La estructura fundamental del modelo es la “relación”, la cual esté representada por una tabla bidimensional construida por líneas (tuplas) y columnas (atributos).

Para distinguir una tupla de otra, se recurre al concepto de "llave primaria", el cual es un conjunto de atributos que permiten identificar unívocamente una tupla en una relación, una “llave foráneas” es el atributo o conjunto de atributos que son llaves primarias de otra relación y son el mecanismo fundamental para relacionar unas tablas con otras. Los atributos de una llave primaria no pueden asumir el valor nulo debido a que no permitirían identificar una tupla concreta en una relación. Cada atributo de una relación se caracteriza por un nombre y por un dominio, el dominio indica qué valores pueden ser asumidos por una columna de la relación véase figura 4.8, otro aspecto importante es la cardinalidad, ésta en una interrelación mide el grado de participación de dicha entidad y pueden ser “uno a uno” “uno a varios” o “varios a varios”.



**Figura 4.8** Estructura General de una Tabla en una Base de Datos Relacional.

**Fuente:** elaboración propia, año:2009.

El objetivo en esta fase es elaborar un modelo lógico que represente las entidades, relaciones y atributos, para ello se contemplarán los siguientes aspectos para elaborar el Modelo Relacional:

- Identificación de las entidades involucradas en el sistema SURUBV
- Determinar las claves de las entidades
- Representación gráfica del modelo
- Descripción de los atributos de cada entidad

#### **4.2.1 Modelo Relacional del sistema SURUBV.**

En la figura 4.3 se establecieron entidades involucradas en el sistema para el desarrollo del modelo de clases, estas entidades derivan del modelo de clases de análisis visto en el capítulo 3, sección 3.5, y serán la base para elaborar el modelo relacional.

En la figura 4.9 se muestra el modelo relacional desarrollado para el sistema SURUBV, en el modelo se puede destacar que la entidad “usuario” es una generalización de las entidades “alumno”, “coordinador”, “docente” y “administrador”, en este caso “usuario” hereda los atributos de otra entidad dependiendo del tipo de usuario, para llevar a cabo este proceso es necesario contar con un SGBD que soporte dicha propiedad.

El modelo muestra los diferentes atributos y su longitud en cada tabla, así como la clave principal y clave foránea, las cuales se destacan con color rojo y la clave principal con una llave de color amarillo.

Dentro de las reglas del negocio se destacan que un estudiante puede estar cursando varios programas, los programas tienen una o dos mallas curriculares, las mallas curriculares representan los pensum de estudio de los programas con la diferencia de que posee una modalidad diurna o nocturna para un mismo programa, la distribución

de las unidades curriculares es a través de trayectos y tramos, un trayecto puede contener 2 ó 3 tramos, los tramos equivalen a un semestre. Estos programas son impartidos en espacios denominados “aldea universitaria”. Cada coordinador está asociado a un Eje, éste representa un conjunto de estados de Venezuela.

A continuación se describen las tablas del modelo relacional de la base de datos véase figura 4.9.

**Tabla: usuario**

Descripción: Registra los diferentes tipos de usuarios, esta tabla hereda atributos de las tablas alumno, coordinador, docente y administrador.

**Tabla: eje\_ubv**

Descripción: Almacena los Ejes UBV que representan una distribución geográfica.

**Tabla: estado\_vzla**

Descripción: Almacena los códigos y nombres de estados de Venezuela, así como su relación con un Eje UBV.

**Tabla: municipio\_vzla**

Descripción: Almacena los códigos y nombres de los municipios de Venezuela.

**Tabla: aldea**

Descripción: Almacena las aldeas, los códigos y nombres de los espacios donde se imparten los diferentes programas.

**Tabla: programa**

Descripción: Almacena los códigos y el nombre del programa o “carrera” ofrecido por la universidad.

**Tabla: aldea\_programa**

Descripción: Registra la relación aldea-programa, dado que varios programas son impartidos en las diferentes aldeas.

**Tabla: alumno\_programa**

Descripción: registra la relación alumno-programa, dado que un alumno puede cursar varios programas.

**Tabla: estado\_académico**

Descripción: registra la relación entre estudiante, programa y malla curricular, así como el estatus del alumno en ese programa-malla\_curricular.

**Tabla: estatus**

Descripción: Almacena los códigos y el nombre del estado del estudiante en una relación programa-malla\_curricular, ésta puede ser estatus “activo” o “inactivo”.

**Tabla: malla\_curricular**

Descripción: Almacena los códigos y el nombre de las mallas\_curriculares de cada programa así como la relación con un programa, una malla representa al pensum de un programa pero puede tener modalidad diurna o nocturna.

**Tabla: turno**

Descripción: Almacena los códigos y el nombre del turno de la malla curricular, ésta puede ser una malla diurna o nocturna.

**Tabla: programa**

Descripción: Almacena los códigos y el nombre del programa o “carrera” ofrecido por la universidad.

**Tabla: oferta\_académica**

Descripción: Almacena las unidades curriculares ofrecidas a las aldeas por los programas para un período lectivo, adicionando sección, cupo y docente.

**Tabla: período\_académico**

Descripción: Almacena los códigos y el nombre de los períodos académicos que se dispongan para periodos lectivos.

**Tabla: inscripción**

Descripción: Almacena los códigos de unidades curriculares que va cursar un estudiante en un período lectivo, asociando el programa, malla\_curricular, sección, docente y aldea.

**Tabla: unidad\_curricular**

Descripción: Almacena los códigos y el nombre de cada unidad curricular, así como el tipo de unidad curricular, las horas y turno. Cada unidad curricular posee un código único que la identifica.

**Tabla: tipo\_uc**

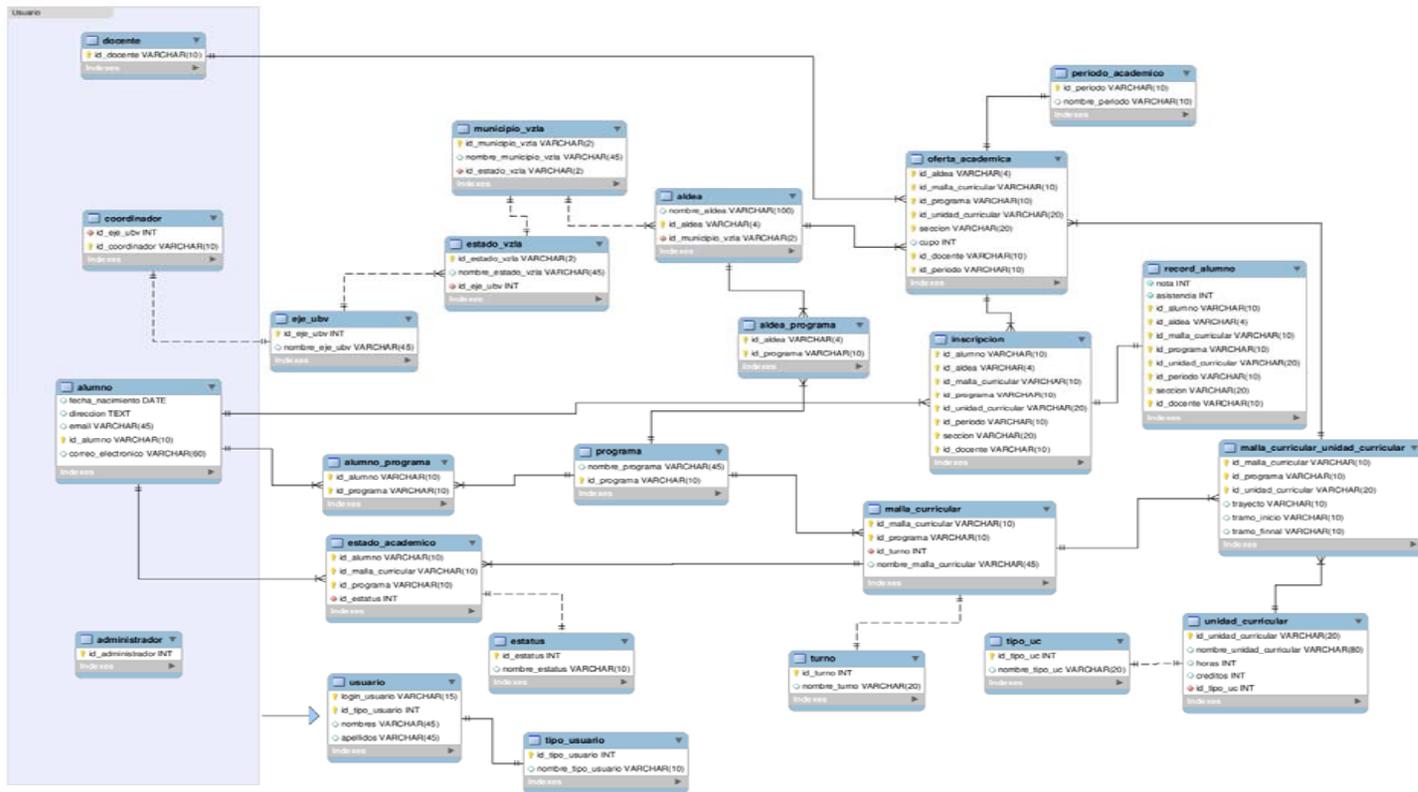
Descripción: Almacena los códigos y el nombre del programa de los tipos de unidades curriculares, estas pueden ser obligatoria o electiva.

**Tabla: malla\_curricular\_unidad\_curricular**

Descripción: registra la relación malla\_curricular-unidad\_curricular permite construir los diferentes pensum según la malla.

**Tabla: record\_alumno**

Descripción: Almacena las unidades curriculares cursadas por el estudiante, así como la calificación, asistencia y docente que impartió la unidad curricular.



**Figura 4.9** Modelo Relacional de la Base de Datos para del Sistema SURUBV.  
**Fuente:** elaboración propia. Año:2009

### **4.3 Diseño de la interfaz del sistema.**

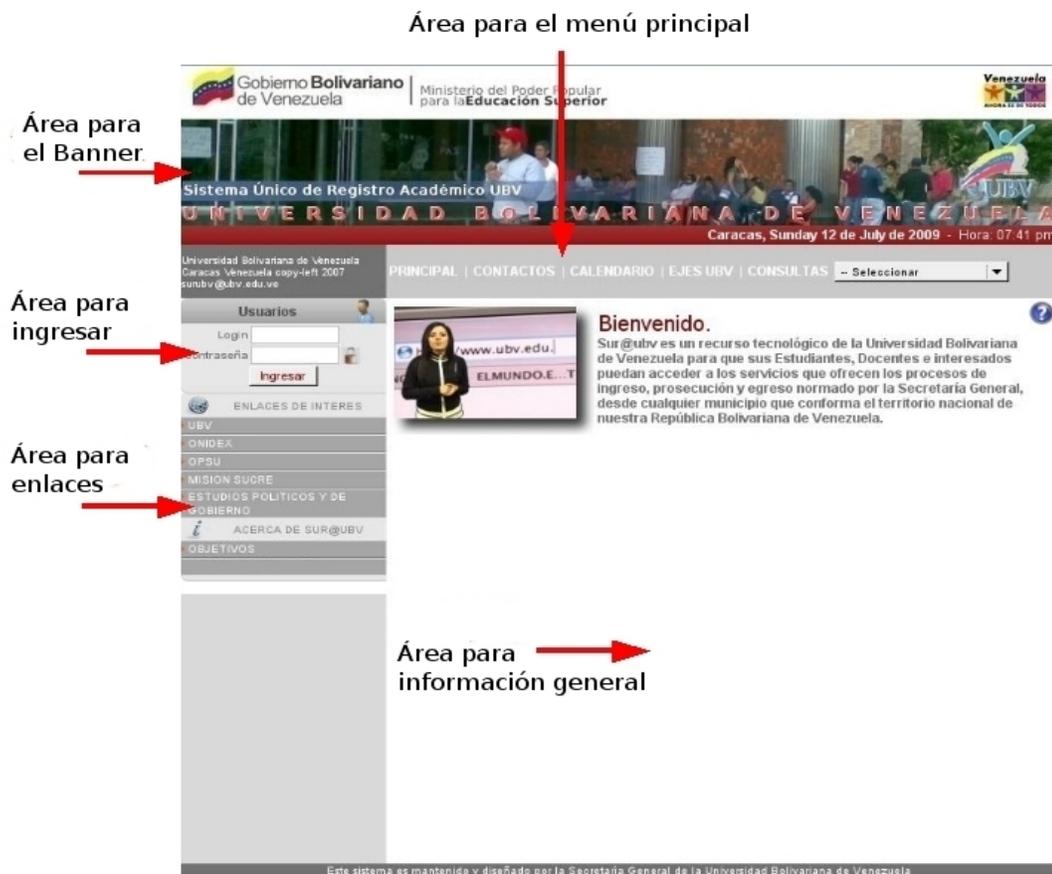
La interacción con el sistema será a través de un navegador Web o Browser, es importante señalar que las interfaces del sistema son páginas Web generadas por un programa en el lado del servidor, el lenguaje de programación utilizado para elaborar las páginas es el HTML, el cual a pesar de ser un estándar es interpretado de forma distinta por algunos navegadores Web, es por ello que se debe recurrir a técnicas que permitan solventar estas situaciones, un ejemplo es el uso del CSS (*Cascading Style Sheets, hojas de estilo en cascada*), su uso en páginas Web minimiza el riesgo de que la interfaz altere su apariencia en caso de utilizar navegadores Web distintos a los tradicionales como *Internet Explorer* o *Mozilla Firefox*.

Debido a que el acceso al sistema es a través de un red pública como lo es la Internet, es necesario el uso de un certificado de seguridad entre el servidor Web y el cliente navegador Web, esto impide que los datos como login o password puedan ser interceptados por terceros.

Otro aspecto que se consideró para el diseño de la interfaz, es que el sistema va ser utilizado por usuarios con distintos niveles de conocimiento, esto demanda un interfaz que permita escalabilidad, intuitiva y de fácil acceso al usuario. Existen varios punto de vista en el diseño de las interfaces, del usuario, del programador y la del diseñador gráfico, cada uno tiene un modelo mental propio de la interfaz que contiene conceptos y expectativas de la misma.

#### **4.3.1 Diseño de la Interfaz Principal del Sistema.**

El la figura 4.10 se muestra el diseño preliminar de la interfaz de inicio del sistema, mostrando áreas que permitirán manejar información y mejorar su aspecto sin que se altere el diseño básico.



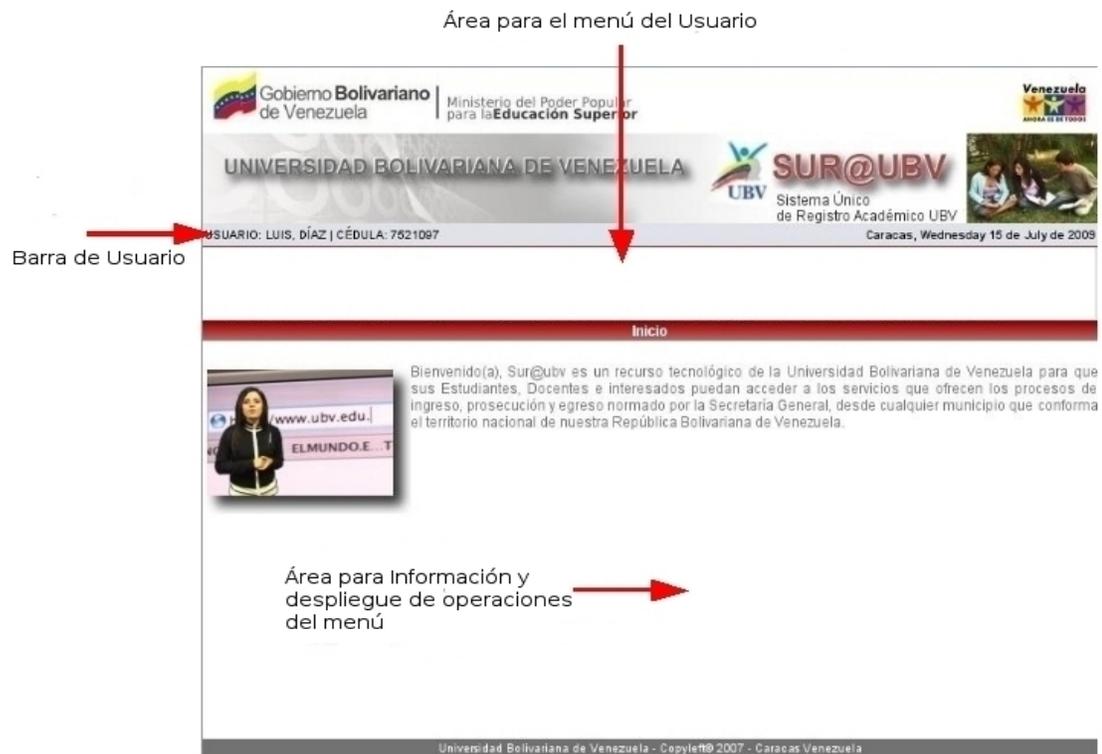
**Figura 4.10** Diseño Preliminar de Interfaz de Inicio del Sistema SURUBV.  
**Fuente:** elaboración propia. Año:2009

- **Área para el Banner:** Esta área es destinada para colocar logos que identifiquen a la institución.
- **Área para el Ingresar:** En esta zona es donde los usuarios ingresan el login y su clave para el ingreso al sistema, dependiendo el tipo de usuario el sistema mostrará una interfaz
- **Área para Enlaces:** Esta área es destinada para colocar *links* o enlaces o otras páginas Web.

- **Área para información general:** Esta área es destinada para colocar información de interés a la institución.

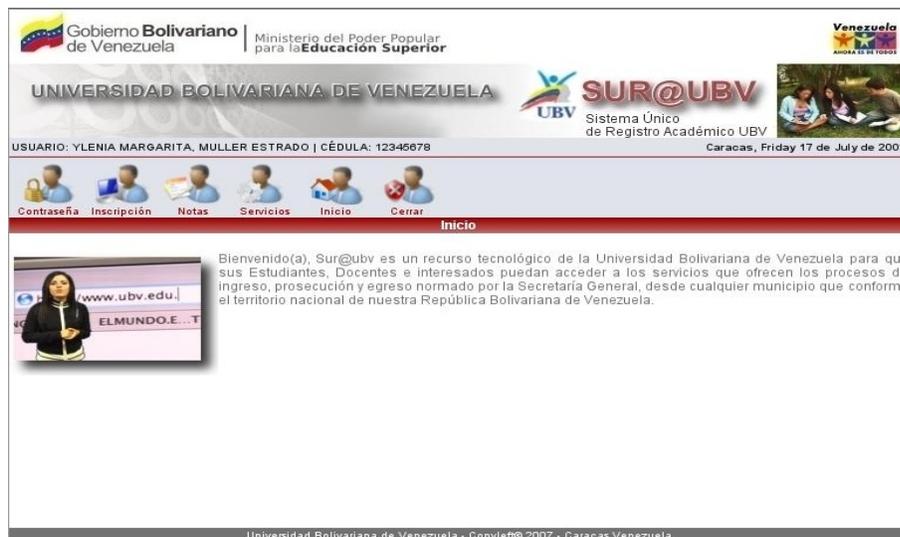
### 4.3.2 Diseño de la Interfaz General de Usuario.

En la figura 4.11 se muestra la interfaz general que servirá como base para cada tipo de usuario, esta interfaz posee un área para colocar el menú de opciones correspondiente a cada tipo de usuario, así como una barra donde se identifica el usuario y un área para colocar información, el contenido del área para colocar información cambia de acuerdo a las opciones seleccionadas en el menú de usuario. El diseño general consta de un encabezado, un cuerpo y un pie de página.



**Figura 4.11** Interfaz General de Usuario.  
**Fuente:** elaboración propia. Año:2009

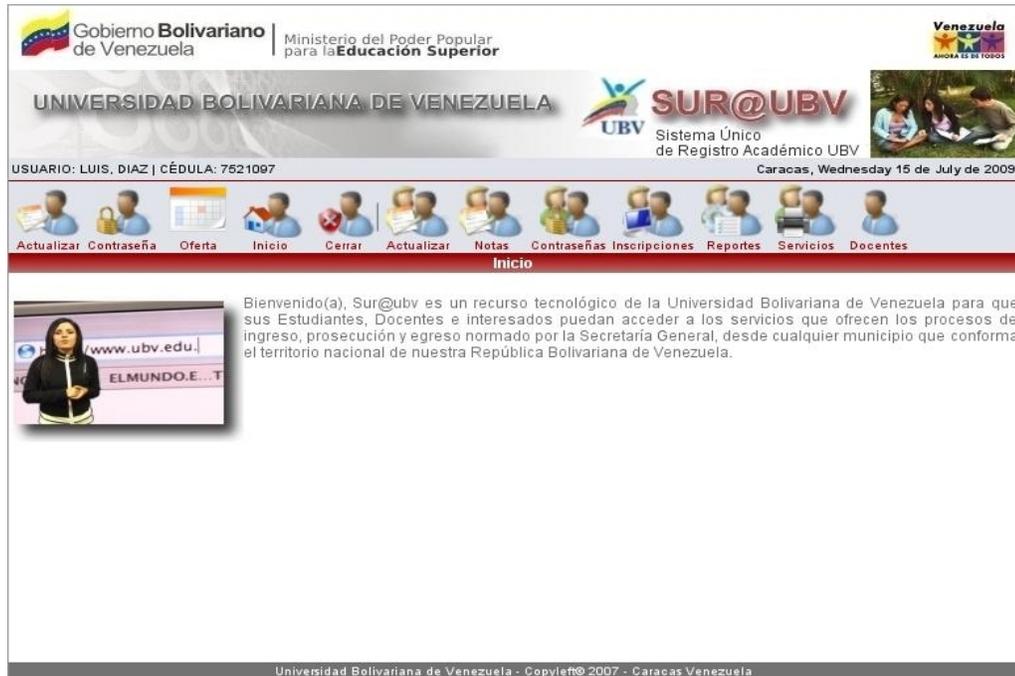
En las figuras 4.12, 4.13, 4.14 y 4.15 se muestran los diferentes diseños preliminares de las interfaces de usuarios del sistema, las cuales hacen uso del diseño de la interfaz general de usuarios vista en la figura 4.11. Una vez que el usuario selecciona una opción del menú, la información es desplegada en el centro de la interfaz, utilizando un área bastante amplia para desplegar formularios e información.



**Figura 4.12** Interfaz del Usuario Estudiante.  
**Fuente:** elaboración propia. Año:2009



**Figura 4.13** Interfaz del Usuario Docente.  
**Fuente:** elaboración propia. Año:2009



**Figura 4.14** Interfaz del Usuario Coordinador Eje UBV.  
**Fuente:** elaboración propia. Año:2009



**Figura 4.15** Interfaz del Usuario Administrador.  
**Fuente:** elaboración propia. Año:2009

### 4.3.3 Diseños para Reportes Impresos.

Durante los procesos de inscripción, registro de notas docente y servicios académico se generarán documentos impresos como resultado de consultas a la base de datos. A continuación en las figuras 4.16, 4.17 y 4.18 se muestran los modelos para estos documentos.



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD BOLIVARIANA DE VENEZUELA  
SISTEMA ÚNICO DE REGISTRO ACADÉMICO UBV

**REGISTRO DE INSCRIPCIÓN**

Fecha: 18-07-2009  
Hora: 10:22:46

---

CÉDULA: 12345678  
NOMBRES: YLENIA MARGARITA  
APELLIDOS: MULLER ESTRADO  
PROGRAMA DE FORMACIÓN: INFORMÁTICA PARA LA GESTIÓN SOCIAL  
ESTADO: FALCÓN  
MUNICIPIO: CARIRUBANA  
TIPO ALDEA: SEDE  
NOMBRE ALDEA: UBV FALCÓN

---

**UNIDADES CURRICULARES** PERIODO ACADÉMICO: I-2009

COD MALLA	TRAY	TRAM INICIO	TRAM FINAL	CÓDIGO	Hrs UC	NOMBRE UNIDAD CURRICULAR	SECCIÓN	TURNO	HORARIO
INF-DIA	TRAY1	TRAMI	TRAMI	FEP-EPI-100	2	BASES DEL CONOCIMIENTO	I-111	DIURNO	<input type="checkbox"/>
INF-DIA	TRAY1	TRAMI	TRAMI	FPI-ETI-100	2	ÉTICA	I-111	DIURNO	<input type="checkbox"/>
INF-DIA	TRAY1	TRAMI	TRAMI	INF-NET-100	5	INTERNET Y LA WEB	I-111	DIURNO	<input type="checkbox"/>
INF-DIA	TRAY1	TRAMI	TRAMI	INF-SW-100	6	ALGORITMOS Y PROGRAMACIÓN	I-111	DIURNO	<input type="checkbox"/>
INF-DIA	TRAY1	TRAMI	TRAMI	INF-TEC-100	4	INTRODUCCIÓN A LA INFORMÁTICA	I-111	DIURNO	<input type="checkbox"/>
INF-DIA	TRAY1	TRAMI	TRAMI	MAT-ALG-100	4	MATEMÁTICAS DISCRETAS	I-111	DIURNO	<input type="checkbox"/>

COD MALLA	CÓDIGO	CALIFICACIÓN NÚMERO	CALIFICACIÓN LETRA	DOCENTE	FIRMA DOCENTE
INF-DIA	FEP-EPI-100				
INF-DIA	FPI-ETI-100				
INF-DIA	INF-NET-100				
INF-DIA	INF-SW-100				
INF-DIA	INF-TEC-100				
INF-DIA	MAT-ALG-100				

Sello y Firma

.....

No es válido sin el Sello y la Firma.

**Figura 4.16** Diseño de Constancia de Inscripción.  
Fuente: elaboración propia. Año:2009

En la figura 4.16, se asignó un espacio para que el estudiante consigne las calificaciones finales directamente con el docente, y quede como registro para el estudiante.

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD BOLIVARIANA DE VENEZUELA  
SISTEMA ÚNICO DE REGISTRO ACADÉMICO UBV  
COORDINACIÓN REGIONAL DE LOS ESTADOS FALCÓN, LARA, YARACUY,  
COORDINACIÓN DE INGRESO, PROSECUCIÓN Y EGRESO ESTUDIANTIL



#### CONSTANCIA DE ESTUDIOS

Quien suscribe, LIC. LUIS DIAZ, Coordinador(a) de Ingreso, Prosecución y Egreso Estudiantil de la Universidad Bolivariana de Venezuela, por medio de la presente, hace constar que el ciudadano (a): **YLENIA MARGARITA, MULLER ESTRADO**, titular de la cédula de identidad N° **12345678**, es estudiante activo de esta universidad. Actualmente cursa el período académico **I-2009** comprendido entre el 30-03-2009 y 07-08-2009, cursando el Tramo **TRAM1**, en el Programa de Formación de grado **INFORMÁTICA PARA LA GESTIÓN SOCIAL**, en el estado **FALCÓN**, municipio **CARIRUBANA**.

Constancia que se expide a petición de la parte interesada, a los 18 días del mes de Julio de Dos mil Nueve.

Atentamente,

Coordinación de Ingreso, Prosecución y Egreso Estudiantil

No es válido sin el sello.

**Figura 4.17** Diseño de Constancia de Estudios.  
**Fuente:** elaboración propia. Año:2009

Los documentos constancia de inscripción, constancia de estudios y notas registradas pueden ser generados por el usuario Estudiante y el usuario Coordinador Eje UBV como puede apreciarse en los diseños de clases detalladas de la sección 4.1.1.

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD BOLIVARIANA DE VENEZUELA  
SISTEMA ÚNICO DE REGISTRO ACADÉMICO UBV

**REGISTRO DE NOTAS**

Cédula: 12345678  
NOMBRES: YLENIA MARGARITA  
APELLIDOS: MULLER ESTRADO  
PROGRAMA DE FORMACIÓN: INFORMÁTICA PARA LA GESTIÓN SOCIAL  
MALLA: INF-DIA  
ESTADO: FALCÓN  
MUNICIPIO: CARIRUBANA  
TIPO ALDEA: SEDE  
NOMBRE ALDEA: UBV FALCÓN



Fecha: 19-07-2009

Cédula: 12345678 / Página: 1

TRAY	TRAMO INICIO	TRAMO FINAL	TIPO UC	CÓDIGO	Hrs UC	NOMBRE UNIDAD CURRICULAR	(0-20) NOTA	SECCIÓN	ESTADO UC
<b>PERIODO: I-2004</b>									
TRAY1	TRAM1	TRAM1	OBLIGATORIA	INF-SW-100	6	ALGORITMOS Y PROGRAMACIÓN	12	IGS1	APROBADA
<b>PERIODO: II-2004</b>									
TRAYA	TRAMA	TRAMA	ELECTIVA	IDM-ING-400	3	INGLÉS I	12	UNICA	APROBADA
<b>PERIODO: I-2005</b>									
TRAY2	TRAM4	TRAM4	OBLIGATORIA	INF-ORG-100	4	INTERACCIÓN HUMANO - COMPUTADOR	18	IGS4	APROBADA
TRAY1	TRAM2	TRAM2	OBLIGATORIA	INF-TEC-105	2	LABORATORIO DE INFORMÁTICA	12	ABC	APROBADA

Coordinación CIPEE  
Sello y Firma

-----

No es válido sin el Sello y la Firma.

**Figura 4.18** Diseño de Notas Registradas.  
**Fuente:** elaboración propia. Año:2009

A continuación en la figura 4.19 se muestra el diseño para las actas finales que el usuario docente debe imprimir y consignar ante el control de estudios (CIPEE) coordinación de ingreso prosecución y egreso estudiantil de la aldea. Esta acta es

generada luego que el docente registre las notas de la sección en una unidad curricular.

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD BOLIVARIANA DE VENEZUELA  
SISTEMA ÚNICO DE REGISTRO ACADÉMICO UBV  
COORDINACIÓN DE INGRESO PROSECUCIÓN Y EGRESO ESTUDIANTIL

**ACTA DE NOTAS**



COD MALLA: INF-DIA  
COD UC: INF-NET-105  
NOMBRE UC: LABORATORIO DE APLICACIONES WEB  
SECCIÓN: 1  
PERÍODO: I-2009

CÉDULA: 10835937  
NOMBRES: ANGEL RAMÓN  
APELLIDOS: PÉREZ  
EJE UBV: MATURÍN

Nro	CÉDULA	NOMBRES	APELLIDOS	NOTA FINAL [0-20]Pts	ASISTENCIA [0-100]%	ESTATUS UNIDAD CURRICULAR
1	10308346	YOLANDA JOSEFINA	MICELI TORRES	18 Pts	90 %	APROBADA
2	11338436	KARINA DEL VALLE	MEDINA	10 Pts	90 %	APROBADA
3	13813524	FRANCISCO JAVIER	CHATAING OLIVEROS	18 Pts	100 %	APROBADA
4	15115113	NA YELIS JOSEFINA	BONAFINE PALACIOS	20 Pts	90 %	APROBADA
5	15723934	EGLIS MARGARITA	MÁRQUEZ GÓMEZ	16 Pts	90 %	APROBADA
6	16176102	BILLIS JOSÉ	ORTEGA SOLANO	16 Pts	100 %	APROBADA
7	16375937	MAYRIS CAROLINA	MENDOZA ROMERO	11 Pts	90 %	APROBADA
8	16516159	DARWIN ANTONIO	JIMENEZ CHACON	15 Pts	100 %	APROBADA
9	16517170	PEDRO C.	LARROSA H.	10 Pts	90 %	APROBADA
10	17090645	JOSE	ROCA	18 Pts	90 %	APROBADA
11	17090796	CARLOS EDUARDO	GIMENEZ CHACON	18 Pts	80 %	APROBADA
12	17241685	KEILA DEL VALLE	AGOSTINI ROSILLO	12 Pts	100 %	APROBADA
13	17547540	VANESSA JOSE	MARTINEZ ARREDONDO	13 Pts	95 %	APROBADA
14	17712262	MIGUEL ALEXANDER	HERRERA CONDE	12 Pts	88 %	APROBADA
15	17713048	LAURIS DELFINA	BRITO SANTIL	17 Pts	75 %	APROBADA
16	17746219	EDGAR IGNACIO	LOPEZ GARCIA	12 Pts	100 %	APROBADA
17	18079412	MARIA BEATRIZ	CORVO MUÑOZ	10 Pts	86 %	APROBADA
18	18462323	JEAN CARLOS	MAITA RUIZ	10 Pts	87 %	APROBADA
19	19258131	ROSIBETH DE LOS ANGELES	MAYORGA REYES	10 Pts	75 %	APROBADA
20	19603913	CARLOS HUMBERTO	SIFONTE GUZMAN	19 Pts	90 %	APROBADA
21	19782519	ELIANNY JOSE	CANACHE REINALES	18 Pts	100 %	APROBADA
22	19875498	ROXANA YULUEN	ASTUDILLO MARCHAN	12 Pts	100 %	APROBADA
23	20002391	MARIATZI ALEJANDRA	RODRIGUEZ MONTAÑO	12 Pts	100 %	APROBADA
24	22974350	MILAGROS COROMOTO	HERNÁNDEZ RAMÍREZ	11 Pts	90 %	APROBADA
25	4026357	ARMANDO RENÉ	FREITES SERRANO	14 Pts	80 %	APROBADA
26	82287491	MARTIN YACIEL	GARRIDO MARTINEZ	18 Pts	90 %	APROBADA
27	9658503	NHORYS JOSEFINA	ORTIZ MARIN	18 Pts	80 %	APROBADA

Fecha y Hora de Impresión: 22-07-2009 11:34:41  
Fecha y Hora del registro: 19-07-2009 10:14:19

-----  
Firma docente

-----  
Firma y sello  
Coordinador de PFG

-----  
Firma y sello  
Coordinación de Ingreso Prosección y  
Egreso Estudiantil

**Figura 4.19** Diseño de Acta de Notas.

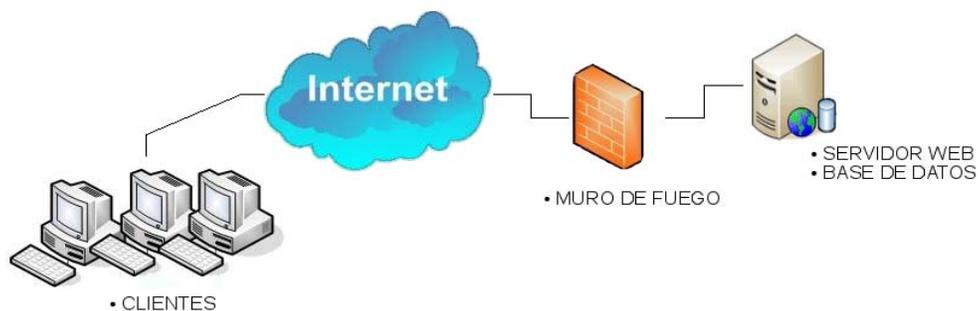
**Fuente:** elaboración propia. Año:2009

#### 4.4 Plataforma de soporte para el sistema propuesto.

El sistema SURUBV está basado en una arquitectura Cliente-Servidor, las sedes de la Universidad Bolivariana de Venezuela se encuentran en los estados Monagas, Zulia, Falcón, Distrito capital y Bolívar. Cada estado comprende un Eje UBV, véase tabla 3.2 del capítulo anterior, en cada Eje existen un gran número de aldeas, donde se imparten los programas de la universidad, a estos programas se les denominan programas municipalizados.

El acceso al sistema debe ser desde todo lugar de Venezuela donde se cuente con un navegador Web y se utilizará la red pública Internet, la universidad cuenta con su propio dominio en Internet, [www.ubv.edu.ve](http://www.ubv.edu.ve), éste permitiría crear la dirección para el sistema, de forma siguiente, [www.surubv.ubv.edu.ve](http://www.surubv.ubv.edu.ve). El sistema debe contar con un computador servidor con las características señaladas en el capítulo anterior sección 3.3, en el cual estará alojado el servidor Web y la aplicación.

El computador servidor estará ubicado en la sede UBV Distrito capital, ya que esta sede cuenta con un enlace de 1GB, lo cual permitirá un manejar el tráfico generado por las consultas y acceso a aplicación durante los diferentes procesos, en la figura 4.20 se muestra la estructura que debe poseer el sistema.



**Figura 4.20** Diseño de la Plataforma.  
**Fuente:** elaboración propia. Año:2009

## CONCLUSIONES

Una vez realizado el análisis de la situación actual y su posterior solución a través del diseño de un sistema informático, se puede concluir:

1. El proceso de ingreso, prosecución y egreso que dispone actualmente la Universidad Bolivariana de Venezuela llevado a cabo por la fundación Misión Sucre no permite la obtención de información en línea, para la toma de decisiones oportuna.
2. Debido a que los Programas de Formación municipalizados poseen una dinámica diferente a los impartidos en las sedes, las calificaciones definitivas de los estudiantes no se obtienen a tiempo para el próximo período de inscripción.
3. Las unidades curriculares al no poseer códigos únicos, eran registradas en diferentes trayectos y tramos, para las diferentes mallas curriculares de los Programas de Formación.
4. En el sistema propuesto la inscripción es centralizada y se garantiza el registro con un código único para las unidades curriculares de las mallas respectivas en todo el territorio nacional.
5. El sistema propuesto permite obtener información *on line* de las inscripciones y la prosecución del estudiante a nivel nacional.
6. El diseño del sistema en un entorno Web (cliente-servidor), permitirá que los diferentes tipos de usuarios puedan acceder rápida y fácilmente a la aplicación desde Internet.

7. Los estereotipos definidos por la extensión del UML para aplicaciones Web (WAE), fueron de gran utilidad en la elaboración del modelo de diseño, ayudando a visualizar claramente las diferentes páginas Web necesarias para la representación del sistema.
  
8. Para el diseño de las pantallas se tomó en cuenta las necesidades de usuario, así como el acceso fácil a las diferentes funcionalidades de cada interfaz utilizando un menú, encabezado y pie de páginas diferentes para cada uno.

## **RECOMENDACIONES**

Una vez culminado el diseño del sistema propuesto, se recomienda lo siguiente:

1. Que la Universidad Bolivariana de Venezuela considere el sistema propuesto como una alternativa para mejorar la toma de decisiones en relación a los procesos de ingreso, prosecución y egreso de los Programas de Formación tanto municipalizados como los de sede.
2. La creación de aplicaciones con tecnología Web para la automatización de los procesos académico-administrativos de la UBV, aprovechando la plataforma tecnológica que existe hoy en día.
3. Dotar a las Coordinaciones de Ingreso, Prosecución y Egreso Estudiantil (CIPEE) de cada sede de la universidad con la conectividad necesaria para acceder a la Internet.
4. El control académico-administrativo de los estudiantes en Programas de Formación municipalizados por parte de la universidad, se debe llevar a cabo por el mismo personal que controla los estudiantes de las sedes.
5. Se recomienda el uso del sistema operativo GNU/Linux como plataforma de desarrollo de la aplicación debido a su comprobada confiabilidad, buena seguridad, poderosa funcionalidad, buen desempeño y bajo costo en consonancia con el decreto 3390, que establece el uso prioritario de software libre en la administración pública.

## BIBLIOGRAFÍA

1. Álvarez G., (1997) . **“Web Seguro”**. [www.iec.csic.es/cryptonomicon/web.html](http://www.iec.csic.es/cryptonomicon/web.html).
2. Bendahan M., (1997). **“Proceso de Desarrollo de Software”**.  
<http://www.monografias.com/trabajos5/desof/desof.shtml>.
3. Booch, G., (1996). **“Análisis y Diseño Orientado a Objetos”**, 2da edición. Ed. Addison-Wesley / Diaz de Santos, México.
4. Castillo P., (2004). **“Páginas Web Estáticas vs. Paginas Web Dinámicas”**.  
<http://www.articulo.org/idx/15/2039/Negocios-en-Internet/article/Paginas-Web-Estaticas-vs-Paginas-Web-Dinamicas.html>
5. CETTICO (Centro de Transferencia Tecnológica en Informática y Comunicaciones), (1997). **"Enciclopedia de Informática y Comunicación Teleinformática"**, 1era edición, Editorial CULTURAL S.A., España.
6. **Elmansri R. y Navathe S.**, (1994). **"Fundamentos de Bases de Datos"**. Editorial Adison -Wesley Iberoamericana, España.
7. **Enciclopedia Wikipedia**,(2008).  
[http://es.wikipedia.org/wiki/Desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_de_software).
8. Galantón A. y Arocha O., (1999). **“Modernización de los Sistemas Automatizados de Admisión, Inscripción de Estudiantes de Nuevo Ingreso y Validación de la Programación Académica del Núcleo de Anzoátegui de la Universidad de Oriente”**, Trabajo de Grado de Ingeniería en Computación, Universidad de Oriente, Venezuela.

9. Guevara J., (1998). **“Desarrollo e Implantación de los Servicios Académicos del Departamento de Computación y Sistemas, usando Tecnología WWW”**, Trabajo de Grado de Ingeniería en Computación, Universidad de Oriente, Venezuela.
10. Jacobson I, Booch G, Rumbaugh J., (1999). **“El Proceso Unificado de Desarrollo de Software”**, Ed. Addison Wesley, México.
11. James R, Ivar J, Grody B., (2002). **“El Lenguaje Unificado de Modelado. Manual de Referencia”**, 1era edición, Editorial Prentice Hall, México.
12. Jesús T. y Kronos., (1997). **“Sistemas de bases de datos y SGBD”**.  
<http://tramullas.com/documatica/2.html>.
13. Kon M.,(1997). **“El Software Libre”**.  
<http://www.monografias.com/trabajos12/elsoflib/elsoflib.shtml>.
14. Larman C., (1999). **"UML y patrones. Introducción al Análisis y Diseño Orientado a Objetos"**, 1era edición, Editorial Prentice Hall, España.
15. Marcias C y Orozco S. (sd), **“Uso de UML en aplicaciones Web: páginas y relaciones”**.  
[http://www.milestone.com.mx/articulos/uso\\_de\\_uml\\_en\\_aplicaciones\\_web.htm](http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm)
16. Martínez M., (2005). **“Páginas Web Dinámicas”**.  
[http://www.mati.unam.mx/index.php?option=com\\_content&task=view&id=100&Itemid=50](http://www.mati.unam.mx/index.php?option=com_content&task=view&id=100&Itemid=50).
17. Molero A., (2001). **“Diseño de la Intranet de la Escuela de Medicina de la Universidad de Oriente Núcleo de Anzoátegui”**. Trabajo de Grado de Escuela de Medicina, Universidad de Oriente, Venezuela.

18. Montes B. y Navarro A, (2002). **“Estudio de la Aplicación de UML en el Modelado de Sistemas Organizacionales Inteligentes”**. Trabajo de Grado Ingeniería de Sistemas, Universidad de Oriente, Venezuela.
19. Orallo E., (2007). **“El lenguaje Unificado de Modelado (UML)”**. <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
20. Puglieser I., (1995). **“Sistema de Gestión de Datos para el Departamento de Compras del Núcleo Anzoátegui de la Universidad de Oriente”**, Trabajo de Grado de Ingeniería en Computación, Universidad de Oriente, Venezuela.
21. Reyes A., (2005). **“Conceptos y principios orientado a objetos”**. [http://www.elguille.info/colabora/NET2005/Percynet\\_Conceptosyprincipiosorientadoaobjetos.htm](http://www.elguille.info/colabora/NET2005/Percynet_Conceptosyprincipiosorientadoaobjetos.htm).
22. Tanenbaum A., (1997). **"Reses de Computadoras"**. 3era edición, Editorial Prentice Hall, México.
23. Valle J., (2005). **"Definición Modelo Cliente Servidor"** [.http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml](http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml).

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

<b>TÍTULO</b>	“DISEÑO DE UNA APLICACIÓN WEB PARA LA GESTIÓN EN LÍNEA DE LOS SERVICIOS ACADÉMICOS DE UNA INSTITUCIÓN DE EDUCACIÓN SUPERIOR”
<b>SUBTÍTULO</b>	

### AUTOR (ES):

<b>APELLIDOS Y NOMBRES</b>	<b>CÓDIGO CULAC / E MAIL</b>
Oscar E. Rondón G.	<b>CVLAC:</b> 10.835.937 <b>E MAIL:</b> hostkar@gmail.com
	<b>CVLAC:</b> <b>EMAIL:</b>

### PALÁBRAS O FRASES CLAVES:

Software Libre

---

Cliente-Servidor

---

Aplicación Web

---

UML

---

WAE

---

Universidad Bolivariana de Venezuela

---

Desarrollo en Cascada

---

Modelo Relacional

---

ÁREA	SUBÁREA
Ingeniería y Ciencias Aplicadas	Ingeniería de sistemas

### RESUMEN (ABSTRACT):

El presente Trabajo se refiere al diseño de una aplicación informática utilizando tecnología Web. Este permitirá la gestión en línea de los servicios académicos de la Universidad Bolivariana de Venezuela (UBV), la cual está distribuida en 5 sedes en todo el territorio nacional. La UBV ofrece Programas de Formación de Grado que se imparten no sólo en la sede, sino también en otras instalaciones denominadas “aldeas”. Para la recolección de información acerca de los procesos que dan soporte a los servicios académicos como son: las inscripciones, solicitud de documentos, registro de notas, prosecución del estudiante, entre otros, se emplearon técnicas como la entrevista y observación directa. El modelado de los procesos se realizó mediante el Lenguaje de Modelado Unificado (UML). En la fase de análisis se elaboró un modelo general de casos de uso, lo cuales se desarrollaron a través de diagramas de colaboración empleándose clases de análisis, estas últimas representan las clases que se usan en el modelo de diseño de clases. Estos modelos se construyeron por medio de los estereotipos del UML para aplicaciones Web denominada “Extensión para aplicaciones Web” (WAE). El diseño de la base de datos se elaboró siguiendo el modelo relacional, el cual permitió establecer las relaciones entre las distintas entidades participantes. Las interfaces se diseñaron considerando los diversos tipos de usuarios que van a interactuar con el sistema de acuerdo a su nivel de conocimiento tecnológico.

**CONTRIBUIDORES:**

<b>APELLIDOS Y NOMBRES</b>	<b>ROL / CÓDIGO CVLAC / E_MAIL</b>				
<b>Carrasquero, Manuel</b>	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU x</b>	<b>JU</b>
	<b>CVLAC:</b>	7.374.987			
	<b>E_MAIL</b>	mcarrasquero@interlink.net.ve			
	<b>E_MAIL</b>				
<b>Siso, Felysol</b>	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU</b>	<b>JU x</b>
	<b>CVLAC:</b>	8.635.705			
	<b>E_MAIL</b>	fviso24@anz.udo.edu.ve			
	<b>E_MAIL</b>				
<b>Martínez, Andrés</b>	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU</b>	<b>JU x</b>
	<b>CVLAC:</b>	12.576.853			
	<b>E_MAIL</b>	amartinez@anz.udo.edu.ve			
	<b>E_MAIL</b>				

**FECHA DE DISCUSIÓN Y APROBACIÓN:**

<b>2009</b>	<b>10</b>	<b>15</b>
<b>AÑO</b>	<b>MES</b>	<b>DÍA</b>

**LENGUAJE. SPA**

**ARCHIVO (S):**

<b>NOMBRE DE ARCHIVO</b>	<b>TIPO MIME</b>
TESIS.Diseño de sistema surubv. doc	application/msword

**CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS:** A B C D E F G H I J  
K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x y  
z. 0 1 2 3 4 5 6 7 8 9.

**ALCANCE**

**ESPACIAL:** \_\_\_\_\_ (OPCIONAL)

**TEMPORAL:** \_\_\_\_\_ (OPCIONAL)

**TÍTULO O GRADO ASOCIADO CON EL TRABAJO:**

Ingeniero de Sistemas  
\_\_\_\_\_

**NIVEL ASOCIADO CON EL TRABAJO:**

Pre-grado  
\_\_\_\_\_

**ÁREA DE ESTUDIO:**

Departamento de Computación y Sistemas

---

**INSTITUCIÓN:**

Universidad de Oriente, Núcleo Anzoátegui

---

**DERECHOS**

De acuerdo al Artículo 44 del reglamento de Trabajos de Grados de la Universidad de Oriente:

“Los trabajos de grado son de exclusiva propiedad de la universidad y solo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al consejo universitario”

---

**Oscar E. Rondón G.**  
**AUTOR**

---

**M.Sc. Manuel Carrasquero**  
**TUTOR**

---

**M.Sc. Felysol Siso**  
**JURADO**

---

**M.Sc. Andrés Martínez**  
**JURADO**

---

**POR LA SUBCOMISIÓN DE TESIS**