

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



**“DESARROLLO DE UN SOFTWARE GERENCIAL BAJO
PLATAFORMA WEB PARA EL CONTROL ACADÉMICO DEL
ÁREA DE POSTGRADO DE LA UNIVERSIDAD DE ORIENTE”**

Realizado por:

DÍAZ VELÁSQUEZ JEAN CARLOS.
TIRADO CABRERA NORELKYS DEL CARMEN.

Trabajo de grado presentado en la Universidad de Oriente
como requisito parcial para optar al título de
INGENIERO EN COMPUTACIÓN

Barcelona, Octubre 2009

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



**“DESARROLLO DE UN SOFTWARE GERENCIAL BAJO
PLATAFORMA WEB PARA EL CONTROL ACADÉMICO DEL
ÁREA DE POSTGRADO DE LA UNIVERSIDAD DE ORIENTE”**

Asesor:

Msc. Francy Ríos

Asesor Académico

Barcelona, Octubre 2009.

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



**“DESARROLLO DE UN SOFTWARE GERENCIAL BAJO
PLATAFORMA WEB PARA EL CONTROL ACADÉMICO DEL
ÁREA DE POSTGRADO DE LA UNIVERSIDAD DE ORIENTE”**

Jurado Calificador

Ing. Rhonald Rodríguez

(Jurado)

Ing. Claudio Cortínez

(Jurado)

Msc. Francy Ríos

(Asesor Académico)

Barcelona, Octubre 2009.

RESOLUCIÓN

De acuerdo al Artículo N° 44 Del Reglamento de Trabajo de Grado

“Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario”.

DEDICATORIAS

Dedico este proyecto y toda mi carrera universitaria a Dios por ser el máximo Ingeniero creador del universo, y permitirme alcanzar esta meta. Por los triunfos y los momentos difíciles que me han enseñado a valorar la vida cada día más.

A mi Hija, por ser mi razón de ser en la vida, con una mirada y una sonrisa tierna me llenaban de perseverancia para continuar.

A mis Padres, a ti Madre Por haberme educado y soportar mis errores. Por tus consejos, por el amor que siempre me has brindado, por cultivar e inculcar ese sabio don de la responsabilidad. A ti Padre, mi mejor amigo y maestro de la vida, que con todo su cariño, comprensión, paciencia y apoyo que me motivaban a seguir.

A mi Esposo, mi fiel compañero por apoyarme en este largo recorrido, por ser mí amigo, mi confidente y por saber comprenderme.

A mis Familiares, Mis sobrinos, que directamente me impulsaron para llegar hasta esta etapa de mi vida. Mis hermanos, que siempre hemos estado unidos y me han apoyado para lograr esta meta. Mi abuelita, que con sus sabios consejos me llena de amor y sabidurías.

A unas personas, que fueron hacer un papel importante en el cielo al lado de dios y desde allá velaron por el final de la meta. Facunda (mi madrina), Livia (mi tía), Gladey (mi coma) y Gloribeth (mi sobrina).

A mi incondicional amigo Jean Carlos, por ser esa persona tan especial (Dios nos puso juntos en este camino para lograr alcanzar el triunfo).

¡Norelkys Tirado!

DEDICATORIAS

Este trabajo de grado está dedicado a todas aquellas personas que hicieron posible su culminación con sus valiosas ayudas.

En primer lugar a mis padres Elena de Díaz y Rafael Díaz cuyo esfuerzo por ayudarme a estudiar nunca ha mermado...

A mis hermanos por ser un motivo para estudiar y lograr metas...

A mis tíos y tías paternos y maternos por ser la segunda opción en muchos momentos...

A mi abuela Juliana Velásquez y a mi abuela Juana Díaz por ser especiales...

A mi tío Aquiles Díaz quien no tiene la oportunidad de estar con nosotros en estos momentos pero que seguramente estará feliz de este logro, este proyecto está dedicado a él.

A mis tutores de la universidad por enseñarme todo lo que sé, en especial a los de carrera que siempre están tan empeñados en mantener el nivel académico en alto nivel...

A mis amigos de la universidad por ser la primera opción en muchos momentos, entre los especiales Norelkys, Eliumar, Dimas, Carol y todos aquellos que llegaron a conocerme un poco...

¡Jean Carlos Díaz!

AGRADECIMIENTOS

Dios Todopoderoso por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante rompiendo todas las barreras que se me presenten. *¡Gracias Señor!*

Gracias a mi porción de cielo, mi consentida que nunca pensé que de tan pequeñito cuerpecito emanara tanta fuerza y entusiasmo para continuar.

¡Te amo Susej!

Agradezco a mi madre Gisela por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa. Agradezco mi padre León por ser el mejor papá del mundo ejemplos de perseverancia y constancia que lo caracterizan y que me ha inculcado siempre. Sus esfuerzos, se convirtieron en su triunfo y el mío. *¡Los amo!*

Gracias, a mi esposo Jesús por ser la persona que ha compartido el mayor tiempo a mi lado, porque en su compañía las cosas malas se convierten en buenas, la tristeza se transforma en alegría y la soledad no existe. Este también es tu victoria.

¡Te Adoro Gordo!

Gracias a mis familiares mis sobrinos Carlos y Ronald, hermanos Noryolis, Alexander y Aidenys y mi abuelita (chencha) que con su amor, cariño y motivación me convirtieron en su inspiración. *¡Los Quiero!*

Gracias a mis amigos Jean Carlos y Dimas (compadritos) por el equipo que formamos logramos llegar hasta el final del camino y que hasta el momento, seguimos siendo los mejores amigos. *¡Los Aprecio!*

Al profesor Víctor Mújica por el apoyo en mi formación profesional,

A todos mis amigos pasados y presentes; pasados por ayudarme a crecer y madurar como persona y presentes por estar siempre conmigo apoyándome en todas las circunstancias posibles, también son parte de esta alegría,

¡Norelkys Tirado!

AGRADECIMIENTOS

Gracias a mis padres por darme la oportunidad de estudiar una carrera universitaria, a dios por mantenerme con buena salud la mayor parte de mi carrera.

Gracias a la Universidad de Oriente por todo lo que aprendí de ella, a mis profesores y amigos.

Gracias a Norelkys Tirado por tanta paciencia que tuvo conmigo y por las tantas veces que me ayudó a dar un paso más.

Gracias a la profesora Francy Ríos por la oportunidad de ser su tesista y por la enorme paciencia que tuvo con nosotros, muchas gracias profe, a usted y a su esposo Andrés Martínez.

¡Jean Carlos Díaz!

RESUMEN

En este proyecto de investigación se desarrolla un sistema automatizado como soporte de las tareas diarias de postgrado en el área académica de la Universidad de Oriente. El software está desarrollado bajo el lenguaje de programación PHP el cuál brinda un buen nivel de seguridad y facilidad de uso en la interfaz de los usuarios, que sumado al manejador de bases de datos Oracle conforman una herramienta muy potente y segura para los estudios de postgrado. Las funciones principales de esta aplicación web es administrar todo lo referente a los Programas de Postgrado, líneas de investigación, grupos de investigación, datos y notas de estudiantes, datos de profesores, datos del personal administrativo, preinscripciones, generar los reportes de transacciones, emitir constancias y actas, y además generar los archivos OPSU codificados con el formato que este organismo impone de las diferentes variables estadísticas que se generan en el área de postgrado. Las herramientas necesarias para desarrollar este software son el editor de páginas Web Macromedia Dreamweaver 8.0, el servidor Web Apache 2.0, el módulo PHP 5.0 y el administrador de bases de datos Oracle 8i. En la construcción del sistema (SISGAP) se utilizó el Proceso Unificado de Desarrollo de Software, el Modelado de UML y sus extensiones que son WebML.

CONTENIDO

RESOLUCIÓN.....	IV
DEDICATORIAS	V
DEDICATORIAS	VI
AGRADECIMIENTOS.....	VII
AGRADECIMIENTOS.....	VIII
RESUMEN	IX
CONTENIDO.....	X
INDICE DE TABLAS.....	XVII
INDICE DE FIGURAS.....	XIX
CAPITULO I.....	24
EL PROBLEMA	24
1.1 INTRODUCCIÓN.....	24
<i>1.1.1 Problema</i>	26
<i>1.1.2 El Propósito</i>	27
<i>1.1.3 Importancia</i>	28
<i>1.1.4 Alcance</i>	28
<i>1.1.5 Originalidad</i>	29
1.2 OBJETIVOS.....	29
<i>1.2.1 Objetivos Generales</i>	29

1.2.2	<i>Objetivos Específicos</i>	30
CAPITULO II	31
MARCO TEÓRICO	31
2.1	ANTECEDENTES.....	31
2.2	SISTEMA DE INFORMACIÓN.....	33
2.2.1	<i>Concepto [5]</i>	33
2.2.2	<i>Actividades básicas de los sistemas de información [5]</i>	33
2.2.3	<i>Análisis y diseño del sistema [5]</i>	34
2.2.4	<i>Desarrollo del diseño del sistema [5]</i>	35
2.2.4.1	<i>Investigación preliminar</i>	35
2.2.4.2	<i>Determinación de los requerimientos del sistema</i>	37
2.2.4.3	<i>Diseño del sistema</i>	39
2.3	INGENIERÍA DE SOFTWARE.....	40
2.3.2	<i>Software [25]</i>	40
2.3.3	<i>Ciclo de Vida Del Software [25]</i>	41
2.3.4	<i>Etapas de la Ingeniería de Software [25]</i>	41
2.4	PROGRAMACIÓN ORIENTADA A OBJETOS	43
2.4.1	<i>Concepto [14]</i>	43
2.4.2	<i>Fundamentos de lo Orientado a Objetos [4]</i>	45
2.5	PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE	47
2.5.1	<i>Concepto [15]</i>	47
2.5.2	<i>Características [15]</i>	48
2.5.3	<i>Proceso Unificado Rational (RUP) [15]</i>	50
2.5.4	<i>Flujos de Trabajos para una Iteración [15]</i>	52
2.5.5	<i>Ventajas del Proceso Unificado de Desarrollo de Software [15]</i>	53

2.6 LENGUAJE UNIFICADO DE MODELADO (UML).....	54
2.6.1 <i>Concepto</i> [3].....	54
2.6.2 <i>Modelos</i> [3].....	56
2.6.3 <i>Diagramas</i> [17].....	58
2.6.3.1 Diagrama de Casos de Uso [17]	58
2.6.3.2 Diagrama de Secuencia [17].....	60
2.6.3.3 Diagrama de Colaboración [24].....	62
2.6.3.4 Diagrama de Estado [7]	63
2.6.3.5 Diagrama de Clases [22].....	66
2.6.3.6 Diagrama de Componentes [9]	67
2.6.3.7 Diagramas en el diseño de aplicaciones Web [6].....	68
2.7 TECNOLOGÍAS WEB.....	68
2.7.1 <i>Internet</i> [16].....	68
2.7.2 <i>Aplicación Cliente - Servidor</i> [16].....	69
2.7.3 <i>Servidor Web</i> [11].....	69
2.7.4 <i>Lenguaje de Etiquetas por Hipertexto</i> [12].....	72
2.7.5 <i>Técnica de desarrollo web AJAX</i> [6].....	73
2.7.5.1 Tecnologías incluidas en AJAX	74
2.7.6 <i>Aplicaciones Web</i> [22]	75
2.7.6.1 Anatomía de una Página Web [2].....	77
2.7.7 <i>Concepto general de WebML</i> [20]	78
2.7.7.1 Sumario de Estereotipos del Modelo de Hipertexto WebML [20].....	84
2.8 LENGUAJE DE PROGRAMACIÓN PHP.....	92
2.8.1 <i>Definición</i> [1].....	92
2.9 BASES DE DATOS	93
2.9.1 <i>Definición</i> [8].....	93
2.9.2 <i>Sistema De Gestión De Base De Datos (SGBD)</i> [8]	94

2.9.3	<i>Arquitectura de un SGBD [8]</i>	95
2.9.4	<i>Modelo de datos [8]</i>	96
2.9.5	<i>Arquitectura Cliente – Servidor [8]</i>	97
2.9.6	<i>Administrados de Base de Datos (DBA) [8]</i>	97
2.9.7	<i>Concepto Generales Sobre la Arquitectura del SGBD ORACLE [10]</i>	98
2.9.8	<i>Usuarios, profiles, roles, privilegios</i>	102
2.9.8.1	<i>Creación De Usuarios [10]</i>	102
CAPITULO III		105
FASE DE INICIO		105
3.1	INTRODUCCIÓN.....	105
3.2	REQUISITOS CANDIDATOS.....	107
3.3	CONTEXTO DEL SISTEMA.....	108
3.3.1	<i>Modelo de Dominio</i>	109
3.3.3	<i>Diagrama de Clases de Dominio</i>	111
3.4	RIESGOS DEL SISTEMA.....	112
3.5	MODELO DE CASOS DE USO.....	114
3.5.1	<i>Descripción de los Actores</i>	116
3.5.2	<i>Descripción de los casos de uso</i>	116
3.5.3	<i>Diagrama detallado del caso de uso “Autenticar”</i>	118
3.5.4	<i>Diagrama Detallado del Caso de uso “Generar Reportes”</i>	119
3.5.5	<i>Diagrama Detallado del Caso de uso “Generar Archivos OPSU”</i>	120
3.5.6	<i>Diagrama Detallado del Caso de uso “Cargar Notas”</i>	121
3.5.7	<i>Diagrama Detallado del Caso de uso “Procesar Inscripción”</i>	122
3.5.8	<i>Diagrama Detallado del Caso de uso “Consultar Notas, Horarios, Datos personales”</i>	124

3.5.9	<i>Diagrama Detallado del Caso de uso “Administrar datos”</i>	125
3.5.10	<i>Diagrama Detallado del Caso de uso “Configurar Sistema”</i>	127
3.7	MODELO DE ANÁLISIS	129
3.7.1	<i>Diagramas de Clases de Análisis</i>	129
3.7.1.1	Clases de Interfaz	129
3.7.1.2	Clases de Control	131
3.7.1.3	Clase de Entidad	132
3.7.1.4	Diagrama de Análisis: Autenticación	133
3.7.1.5	Diagrama de análisis: Ingresar datos Programas Postgrado, Profesores, Personal Administrativo	134
3.7.1.6	Diagrama de análisis: Preinscripción Nuevos Alumnos	136
3.7.1.7	Diagrama de análisis: Generar Reportes	137
3.7.2	<i>Paquete de Análisis</i>	140
3.7.2.1	Paquete de Análisis Autenticar Usuario	141
3.7.2.2	Paquete de Análisis Administrar Datos	141
3.7.2.3	Paquete de Análisis Administrar Sistema	142
3.8	IDENTIFICACIÓN DE LOS SUBSISTEMAS	143
3.9	EVALUACIÓN DE LA FASE DE INICIO	144
	CAPITULO IV	145
	FASE DE ELABORACIÓN	145
4.1	INTRODUCCIÓN	145
4.2	REQUISITOS DEL SISTEMA	146
4.2.1	<i>Requisitos funcionales</i>	146
4.2.2	<i>Requisitos No Funcionales</i>	146
4.2.3	<i>Modelo de caso de uso</i>	147
4.2.3.1	Identificación de Actores	147

4.2.3.2 Diagrama detallado del caso de uso “Modificar recuperar clave de acceso”	148
4.2.3.3 Diagrama Detallado del Caso de uso “Generar Lista de Asistencias”	150
4.2.4 Prototipo de Interfaz de Usuario	151
4.2.4.1 Prototipo de Interfaz Ingresar al Sistema	152
4.2.4.2 Prototipo Interfaz de Servicios	153
4.2.5 Modelo de Hipertextos	153
4.2.5.1 Identificación de Sites Views	154
4.3 ANÁLISIS	157
4.3.1 Clases de análisis	157
4.3.2 Diagrama de análisis del caso de uso Modificar, recuperar clave de acceso	158
4.3.3 Diagrama de análisis del caso de uso Generar lista de asistencias.....	159
4.4 DISEÑO.....	160
4.4.1 Base de Datos del Sistema SISGAP	161
4.4.1.1 Identificación de las tablas de la Base de Datos del Sistema SISGAP.....	161
4.4.2 Modelo físico de la Base de Datos del Sistema SISGAP	186
4.5 MODELO DE GESTIÓN DE CONTENIDO	186
4.5.1 Página de Inicio de Sesión o Ingresar al Sistema SISGAP	188
4.5.2 Páginas Agregar Datos al Sistema SISGAP.....	188
4.5.3 Páginas de Modificar Datos en el Sistema SISGAP.....	190
4.5.4 Páginas de Eliminar Datos	191
4.5.5 Páginas de Reportes en el sistema SISGAP	192
4.6 IMPLEMENTACIÓN DEL SISTEMA SISGAP	194
4.6.1 Identificación de Componentes de la Arquitectura y Subsistemas	194
4.7 EVALUACIÓN DE LA FASE DE ELABORACIÓN	195
CAPITULO V	196
FASE DE CONSTRUCCIÓN.....	196

5.1 INTRODUCCIÓN.....	196
5.2 FLUJO DE TRABAJO IMPLEMENTACIÓN.....	197
5.2.1 <i>Implementación del Modelo de Hipertexto</i>	197
5.2.1.1 Computación de una página dinámica.....	197
5.2.1.2 Codificación de las páginas Web.....	200
5.3 PRUEBAS DE INTERACCIÓN.....	211
5.3.1 <i>Diseño de Casos de Prueba</i>	211
5.4 EVALUACIÓN DE LA FASE DE CONSTRUCCIÓN.....	215
CAPITULO VI.....	216
FASE DE TRANSICIÓN.....	216
6.1 INTRODUCCIÓN.....	216
6.2 LANZAMIENTO DE LA VERSIÓN BETA.....	216
6.3 EVALUACIÓN DE LA FASE DE TRANSICIÓN.....	217
CONCLUSIONES.....	218
RECOMENDACIONES.....	220
BIBLIOGRAFÍA.....	221
A.5.4.1 Base de Datos.....	242
A.5.4.2 Sesiones.....	242
A.5.4.3 Aplicación.....	243
A.5.4.4 Servidor Web.....	243
A.5.4.5 Link's a sesiones de nivel inferior.....	243
A.6 OBSERVACIONES.....	243
METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:.....	245

INDICE DE TABLAS

EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (2/8)	85
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (3/8)	86
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (4/8)	87
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (5/8)	88
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (6/8)	89
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (7/8)	90
EN LA TABLA 2.1 SE DESCRIBE CADA ESTEREOTIPO WEBML (8/8)	91
TABLA 3.1.- RIESGOS DEL SISTEMA (1/2).....	113
TABLA 3.1.- RIESGOS DEL SISTEMA (2/2).....	114
TABLA 3.2 DESCRIPCIÓN DEL LOS ACTORES.....	116
TABLA 3.3 DESCRIPCIÓN DE LOS CASOS DE USO.....	117
TABLA 3.4 DESCRIPCIÓN DE LAS CLASES DE INTERFAZ (1/2).....	130
TABLA 3.4 DESCRIPCIÓN DE LAS CLASES DE INTERFAZ (2/2).....	131
TABLA 3.5 DESCRIPCIÓN DE LAS CLASES DE CONTROL.....	132
TABLA 3.6 DESCRIPCIÓN DE LAS CLASES DE ENTIDAD (1/2).....	132
TABLA 3.6 DESCRIPCIÓN DE LAS CLASES DE ENTIDAD (2/2).....	133
TABLA 4.1 DESCRIPCIÓN DE LOS NUEVOS CASOS DE USO.....	147
TABLA 4.2. SITE VIEW INGRESAR DATOS.....	154

TABLA 4.3. SITE VIEW CONSULTAR DATOS.....	155
TABLA 4.4. SITE VIEW ELIMINAR DATOS	156
TABLA 4.5. SITE VIEW CONFIGURAR SISTEMA	157
TABLA 5.1. INGRESAR AL SISTEMA. FUENTE AUTORES. 2009	212
TABLA 5.2. INGRESAR NUEVO PROGRAMA DE POSTGRADO. FUENTE AUTORES. 2009	213
TABLA 5.3. GENERAR ARCHIVO OPSU. FUENTE AUTORES. 2009.....	214

INDICE DE FIGURAS

FIGURA 2.1. FASES DEL CICLO DE DESARROLLO. FUENTE: JACOBSON I. (2000)	51
FIGURA 2.2. FLUJOS DE TRABAJO. FUENTE: JACOBSON I. (2000).....	52
FIGURA 2.3.- CASOS DE USO DE UNA MÁQUINA DE CAFÉ. FUENTE: POPKIN S. (1997).....	59
FIGURA 2.4. DIAGRAMA DE SECUENCIA PARA UN ESCENARIO. FUENTE: POPKIN S. (1997)	61
FIGURA 2.5.- EJEMPLO DIAGRAMA DE COLABORACIÓN. FUENTE: VILAS A. (2002).....	63
FIGURA 2.6.- DIAGRAMA DE ESTADO. FUENTE: CONSULTORÍA S. (2004)	64
FIGURA 2.7.- DIAGRAMA DE CLASES. FUENTE: POPKIN S. (1997)	67
FIGURA 2.8.- EJEMPLO DE DIAGRAMA DE COMPONENTES. FUENTE: ROA P. (1997)	FIGUE 67
FIGURA 2.9 ANATOMÍA DE UNA PÁGINA WEB. FUENTE: APPLEBY M. (1998) ...	77
FIGURA 3.1: FASE DE INICIO (UML) FUENTE: JACOBSON I. (2000)	105
FIGURA 3.2: DIAGRAMA DE CLASES DE DOMINIO. FUENTE: AUTORES. 2009	112
FIGURA 3.3. DIAGRAMA DE CASOS DE USO DEL SISTEMA SISGAP. FUENTE: AUTORES. 2009	115

FIGURA 3.4 DIAGRAMA DE CASOS DE USO AUTENTICAR. FUENTE: AUTORES. 2009..... 119

FIGURA 3.5 DIAGRAMA DETALLADO GENERAR REPORTES. FUENTE: AUTORES. 2009 120

FIGURA 3.6. DIAGRAMA GENERAR ARCHIVOS OPSU. FUENTE: AUTORES. 2009 120

FIGURA 3.7 DIAGRAMA DETALLADO CARGAR NOTAS. FUENTE: AUTORES. 2009..... 122

FIGURA 3.8 DIAGRAMA DETALLADO INSCRIPCIÓN. FUENTE: AUTORES. 2009 123

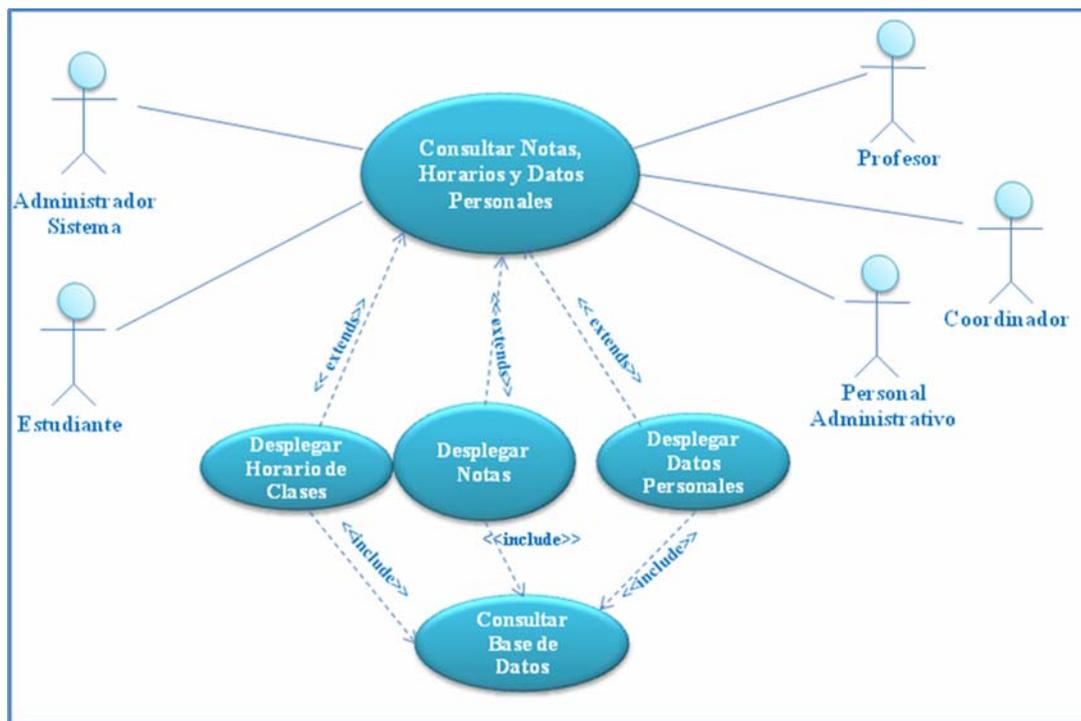


FIGURA 3.9 DIAGRAMA DETALLADO CONSULTAR NOTAS, HORARIOS, DATOS PERSONALES..... 125

FIGURA 3.10 DIAGRAMA DETALLADO ADMINISTRAR DATOS ESTUDIANTES, PROFESORES, PERSONAL ADMINISTRATIVO. FUENTE: AUTORES. 2009	126
FIGURA 3.11 DIAGRAMA DETALLADO CONFIGURAR SISTEMA. FUENTE: AUTORES. 2009	128
FIGURA 3.12 DIAGRAMA DE ANÁLISIS DEL CASO DE USO AUTENTICAR. FUENTE: AUTORES 2009	133
FIGURA 3.13 DIAGRAMA DE ANÁLISIS DEL CASO DE USO INGRESAR DATOS. FUENTE: AUTORES. 2009	134
FIGURA 3.14 DIAGRAMA DE COLABORACIÓN: INGRESAR DATOS. FUENTE: AUTORES. 2009	135
FIGURA 3.15 DIAGRAMA DE ANÁLISIS: PREINSCRIPCIÓN. FUENTE: AUTORES. 2009	136
FIGURA 3.16. DIAGRAMA DE COLABORACIÓN: PREINSCRIPCIÓN. FUENTE: AUTORES. 2009	137
FIGURA 3.17. DIAGRAMA DE ANÁLISIS: GENERAR REPORTES. FUENTE: AUTORES. 2009	138
FIGURA 3.18. DIAGRAMA DE COLABORACIÓN: GENERAR REPORTES. FUENTE: AUTORES. 2009	139
FIGURA 3.19 PAQUETE DE ANÁLISIS AUTENTICAR USUARIO. FUENTE: AUTORES. 2009	141
FIGURA 3.20 PAQUETE DE ANÁLISIS ADMINISTRAR DATOS. FUENTE: AUTORES. 2009	142

FIGURA 3.21 PAQUETE DE ANÁLISIS ADMINISTRAR SISTEMA. FUENTE: AUTORES. 2009	142
FIGURA 3.22 DIAGRAMA DE CAPAS. FUENTE: AUTORES. 2009.....	143
FIGURA 4.1: FASE DE ELABORACIÓN (UML). FUENTE: JACOBSON I. (2000)....	145
FIGURA 4.2. DIAGRAMA DE CASOS DE USO DEL SISTEMA SISGAP. FUENTE: AUTORES. 2009	148
FIGURA 4.3. DIAGRAMA DE CASO DE USO MODIFICAR-RECUPERAR CLAVE. FUENTE: AUTORES. 2009.....	150
FIGURA 4.4. DIAGRAMA DETALLADO GENERAR LISTAS DE ASISTENCIAS. FUENTE: AUTORES. 2009.....	151
FIGURA 4.5. PROTOTIPO DE INTERFAZ DE USUARIO “INGRESAR AL SISTEMA SISGAP”.	152
FIGURA 4.6 ESTRUCTURA DE LAS PÁGINAS WEB DE LA APLICACIÓN SISGAP. FUENTE: AUTORES. 2009.....	153
FIGURA 4.7. DIAGRAMA DE ANÁLISIS DEL CASO DE USO MODIFICAR, RECUPERAR CLAVE.....	158
FIGURA 4.8. DIAGRAMA DE COLABORACIÓN DEL CASO DE USO MODIFICAR, RECUPERAR CLAVE. FUENTE: AUTORES. 2009	158
FIGURA 4.9. DIAGRAMA DE ANÁLISIS DEL CASO DE USO GENERAR LISTA DE ASISTENCIAS.	159
FIGURA 4.10. DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GENERAR LISTA DE ASISTENCIAS. FUENTE: AUTORES. 2009	160
FIGURA 4.11.....	187

FIGURA 4.12. DIAGRAMA DE GESTIÓN DE CONTENIDO INICIO DE SESIÓN.	
FUENTE: AUTORES. 2009	188
FIGURA 4.13. DIAGRAMA DE GESTIÓN DE CONTENIDO INGRESAR ESTUDIANTE.....	189
FIGURA 4.14. DIAGRAMA DE GESTIÓN DE CONTENIDO MODIFICAR DATOS ESTUDIANTE.....	191
FIGURA 4.15. DIAGRAMA DE GESTIÓN DE CONTENIDO ELIMINAR DATOS ESTUDIANTE.....	192
FIGURA 4.16. DIAGRAMA DE GESTIÓN DE CONTENIDO GENERAR REPORTE	
.....	193
FIGURA 4.17. DIAGRAMA DE GESTIÓN DE CONTENIDO DE CREAR LISTAS ASISTENCIA O DE CLASES. FUENTE: AUTORES. 2009	194
FIGURA 4.18. DIAGRAMA DE DESPLIEGUE DEL SISTEMA SISGAP.....	195
FIGURA 5.1: FASE DE CONSTRUCCIÓN (UML). FUENTE: JACOBSON I. (2000).	196
FIGURA 5.2. PROCESAMIENTO DE UNA PÁGINA CON CONTENIDO EXTRAÍDO DE UNA BASE DE DATOS. FUENTE: AUTORES. 2009	198
FIGURA 5.3. INTERFAZ BIENVENIDA. FUENTE: AUTORES. 2009	206

CAPITULO I

EL PROBLEMA

1.1 Introducción

La tecnología a nivel mundial, siempre ha sido parte importante dentro de la sociedad y la educación, donde las universidades forman parte integral de este proceso, en el ámbito de la información y la comunicación, las instituciones utilizan sistemas y recursos para el desarrollo y difusión digitalizada de la información. El desarrollo de aplicaciones Web como complemento al proceso de aprendizaje se convierte en un recurso imprescindible en nuestros días. En la actualidad, se incorporan estas nuevas tecnologías al proceso educativo como el área de Postgrado con la finalidad de variar y flexibilizar las oportunidades de aprender sin restricciones de lugar, tiempo. La Universidad de Oriente, específicamente se perfila en concordancia con estos nuevos avances en el área de la ciencia y la tecnología, por medio de la incorporación de nuevos proyectos enmarcados al desarrollo de aplicaciones usando plataforma Web, que permitirán dar a conocer y satisfacer las necesidades de información a toda la comunidad universitaria.

La Universidad de Oriente (UDO); es una Institución inscrita en el ministerio de Educación Superior, se encuentra distribuida en la Región Nor-Oriental del país, comprendiendo los Estados Anzoátegui, Bolívar, Monagas Nueva Esparta y Sucre.

Creada el 21 de Noviembre de 1958, fundada por el **Dr. Edgar Sanabria**, presidente de la Junta de Gobierno, Decreto Ley 59, como Ministro de Educación **Dr. Rafael Pisan**.

Es indudable que la excelencia de los estudios universitarios está muy relacionada con los estudios de Postgrado y la investigación que realizan convirtiéndose en las expresiones más altas de la vida académica de las universidades. La Universidad de Oriente comenzó los estudios de Postgrado en el 1971, en sus inicios con un solo programa y actualmente ofrece una amplia variedad en áreas que durante años se han venido consolidando hasta los más altos niveles de excelencia.

El área de Postgrado de la Universidad de Oriente, se encuentra conformada principalmente por: Decano, Coordinadores, Profesores y estudiantes y por supuesto la principal dependencia que es el Departamento de Admisión y Control de Estudios (D.A.C.E), esta área ofrece una amplia gama de especialidades, maestría, doctorado y otros, que contribuye a la superación profesional del estudiante.

En la presente investigación se desarrolla un software gerencial bajo plataforma Web para el control de las actividades académicas que se llevan a cabo en el área de Postgrado de la Universidad de Oriente.

1.1.1 Problema

El control de actividades y procesos académicos actual del área de Postgrado de la Universidad de Oriente, se maneja de forma manual e independiente según la escuela y el departamento al cual pertenezca el programa, dando como resultado la entrega de los documentos que el estudiante requiere como: constancias, planilla de pre-inscripción, registros etc., de forma tardía, debido a que no hay acceso rápido a la información, además no existe un mecanismo de captura de la información estadística detallada necesarias para las Oficina de Planificación del Sector Universitario (OPSU) para la inspección de las instancias responsables de gerencial el sector.

La presente investigación se basa fundamentalmente en solventar las dificultades que se presentan en el área de Postgrado con respecto al proceso de inscripción de los estudiantes y los archivos de información necesarios para la Oficina de Planificación del Sector Universitario(OPSU), con una propuesta que se inicia con un sistema que permita realizar el proceso de pre-inscripción y reportes empleando tecnología Web, esto con el fin de proporcionar un excelente servicio al gremio universitario y mejorar los procesos para adquirir los registros correspondientes.

Con la data que genera este sistema se proporciona información veraz a todos los entes involucrados, debido a que podrán acceder al mismo y obtener información respetando siempre el nivel en que se clasifican los usuarios.

1.1.2 El Propósito

El principal propósito de este proyecto fue el desarrollo de un software gerencial que permita:

- La utilización de un único software para toda el área de Postgrado, que facilite las tareas diarias a todos los usuarios que acceden a la información con altos niveles de seguridad de forma estandarizada.
- Reducir el tiempo de espera en todos los niveles de usuarios mediante la utilización de una plataforma Web, la cual permitirá acceder y realizar las consultas de una manera rápida y eficiente.
- Proporcionar una interfaz amistosa, que permita a los usuarios interactuar sin complejidad alguna.
- Facilitar al área de Postgrado una interfaz que permita tener un control más eficiente de la información que maneja, lo que facilitará el proceso de tomas de decisiones a nivel gerencial.

1.1.3 Importancia

Es importante resaltar de una manera general, que la mayor parte de estos procesos automatizados les permitirá mantener, una adecuada planificación y control de las actividades para así brindar un manejo veraz de la información, usando tecnología Web y proporcionando soluciones a corto o mediano plazo.

La implantación y utilización de este sistema traerá consigo una notable mejora en la relación Tiempo-Trabajo en cuanto a los procesos de Inscripción y la estandarización con los demás núcleos, que permitirá un mejor desempeño en la funcionalidad del área del Postgrado de la Universidad de Oriente.

1.1.4 Alcance

El alcance de este proyecto estuvo delimitado para el diseño y el desarrollo de un software bajo plataforma Web, donde físicamente la aplicación abarcará todos los servicios necesarios para el usuario, además los requerimientos implementados por la (OPSU) en un medio de fácil manejo. Una vez la aplicación ejecutada en un servidor se podrá acceder a la misma en todo momento y en cualquier lugar.

El desarrollo del sistema, se siguieron los lineamientos de la metodología del Proceso Unificado de Desarrollo de Software, a través de extensiones del Lenguaje

Unificado de Modelado para aplicaciones WEB, el cual ayudó a especificar, construir, visualizar y documentar los objetos de un sistema; a su vez trabajamos bajo un entorno Web debido a que puede ser accedido desde lugares remotos, utilizando como herramienta el lenguaje **PHP** que constituye software libre (decreto 3390), donde el enfoque será mucho más amplio a la hora de la implementación del sistema, utilizando el paradigma de la **POO** (Programación Orientada a Objetos). En la codificación y motor de la base de datos de este nuevo sistema se utilizó una única herramienta con licencia siendo el más confiable, seguro y de última generación como es el manejador de base de datos **ORACLE 8i**.

1.1.5 Originalidad

Es el primer proyecto que se desarrolló en el área de Postgrado la Universidad de Oriente mediante un entorno Web.

Además este proyecto es unos de los primeros que cubren las expectativas y requerimientos de la Oficina de Planificación del Sector Universitario (OPSU) a nivel Nacional.

1.2 Objetivos

1.2.1 Objetivos Generales

Desarrollar un software gerencial bajo plataforma Web para el control de las actividades académicas que se llevan a cabo en el área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente.

1.2.2 Objetivos Específicos

- Recopilar información sobre el funcionamiento de las actividades académicas del área de Postgrado.
- Determinar los requerimientos del sistema.
- Definir la arquitectura del sitio Web que mejor se ajuste a las necesidades.
- Diseñar la Base de Datos para el almacenamiento de la información relacionada con el sistema.
- Diseñar una interfase que permita al usuario un desenvolvimiento de manera intuitiva al momento de interactuar con el software.
- Realizar los diferentes módulos que conforman el software.
- Realizar pruebas para la detección y corrección de fallas.
- Elaborar manual de usuarios del software.

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes

El proyecto planteado pretende buscar una solución a la problemática presentada, y para su elaboración se consultó a los trabajos de grado que se han realizado con anterioridad, los cuales son similares o guardan alguna relación con el sistema desarrollado. Entre ellos están:

- **(Katuska Núñez, 2004). “Desarrollo de un software para manejar los servicios académicos del Departamento de Mecánica”.** En este proyecto de investigación se desarrolló una gestión de los datos académicos del Departamento de Ingeniería Mecánica, utilizando un ambiente cliente/servidor donde la interacción con los usuarios se lleva a cabo a través de páginas Web dinámicas. [19]

- **(Víctor Mújica y Elena García, 2005). “Desarrollo de un Software para la automatización de las actividades administrativas del departamento de Computación y Sistemas, de la universidad de Oriente Núcleo de Anzoátegui”.** Utilizando la metodología del proceso unificado

de Desarrollo de Software (Jacobson, Booch y Rumbaugh), manipulación de bases de datos robusta que manejan gran cantidad de información, cuya arquitectura fue desarrollada de acuerdo al patrón cliente/servidor. El software fue creado utilizando el lenguaje Visual Basic. [18]

- **(Dulmar Tovar, 2008). “Desarrollo de un software que permita automatizar las actividades asociadas al proceso de inscripción por el Departamento de Computación y Sistemas, vías Internet”.** Consiste en un sistema cuyo acceso se realiza por Internet. Éste proporciona las herramientas adecuadas tanto a alumnos como al Departamento CYS. [23]

- **(Kenia Campos y Ángela Hernández, 2008). “Desarrollo de un software para el proceso de suministro de hemoderivados realizados por el banco de sangre del hospital “Dr. Luís Razetti” de Barcelona a los diferentes pacientes que necesitan transfusiones de sangre”.** Este proyecto permite desarrollar un registro confiable, seguro y al día de los procesos realizados por el banco de sangre, trabajando bajo una plataforma Web y utilizando PHP para el desarrollo del mismo [5].

2.2 Sistema de información

2.2.1 Concepto [5]

Un sistema, es un conjunto de componentes que interactúan entre sí para lograr un objetivo común. Una organización es un sistema, donde sus componentes: mercadotecnia, manufactura, ventas, investigación, embarques, contabilidad y personal; trabajan juntos para crear utilidades que beneficien tanto a los empleados como a los accionistas de la compañía. Cada uno de estos componentes es a su vez un sistema. Todo sistema depende en mayor ó menor medida, de una entidad abstracta denominada sistema de información. Este sistema es el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas hasta sistemas de cómputo que generan reportes periódicos para varios usuarios. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que estos trabajen con eficiencia para alcanzar el mismo objetivo.

2.2.2 Actividades básicas de los sistemas de información [5]

Según (Peralta, 1997) el sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información:

- **Entrada de Información:** es el proceso mediante el cual el Sistema de Información toma los datos que requiere para

procesar la información. Las entradas pueden ser manuales o automáticas.

- **Almacenamiento de información:** el almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior.
- **Procesamiento de Información:** es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.
- **Salida de Información:** la salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes.

2.2.3 Análisis y diseño del sistema [5]

El análisis y diseño del sistema se refiere al proceso de explorar la situación de la organización con el propósito de mejorarla con métodos y procedimientos más

adecuados, en el caso de este proyecto es utilizando el *Lenguaje de Modelado Unificado* (UML) de manera novedosa.

El análisis de sistemas, es el proceso de clasificación e interpretación de hechos, diagnósticos de problemas y empleo de la información para recomendar mejoras al sistema. El análisis especifica que es lo que el sistema debe hacer.

El diseño del sistema, es el proceso de planificar, reemplazar ó complementar un sistema organizacional existente. Es como los planos de un edificio: especifica todas las características del producto terminado, es decir, establece como alcanzar el objetivo.

2.2.4 Desarrollo del diseño del sistema [5]

Es un proceso que consta de las siguientes actividades:

2.2.4.1 Investigación preliminar

La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones; sin importar cuales sean éstas, el proceso se inicia siempre con la petición de una persona (administrador, empleado, ó especialista en sistemas). Cuando se formula la solicitud, comienza la primera actividad: la investigación preliminar. Esta tiene tres partes: Aclaratoria de la solicitud, estudio de factibilidad y aprobación de la solicitud.

- **Aclaración de la solicitud:** Muchas solicitudes que provienen de empleados y usuarios no están formulados de manera clara. Por consiguiente antes de Considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea. Si el solicitante pide ayuda sin saber qué es lo que está mal ó donde se encuentra el problema, la aclaración del mismo se vuelve más difícil. En cualquier caso, antes de seguir adelante, la solicitud del proyecto debe estar claramente planteada.

- **Estudio de factibilidad:** El sistema solicitado debe ser factible en tres aspectos:
 - **Factibilidad técnica:** El trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible?, si se necesita una nueva tecnología ¿cuál es la posibilidad de desarrollarla?
 - **Factibilidad económica:** Al crear el sistema, ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿los costos asociados con la decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto?

- **Factibilidad operacional:** Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios?

- **Aprobación de la solicitud:** No todos los proyectos solicitados son deseables o factibles. Los que son deben incorporarse a los planes. Muchas organizaciones desarrollan sus planes para sistemas de información con el mismo cuidado con el que planifican nuevos productos y programas de fabricación o la expansión de las instalaciones. Después de aprobar la solicitud del proyecto se estima su costo, el tiempo necesario para terminarlo y las necesidades de personal; con esta información se determina donde ubicarlo dentro de la lista existente de proyectos.

2.2.4.2 Determinación de los requerimientos del sistema

El aspecto fundamental del análisis del sistema es comprender todas las facetas importantes de la parte de la organización que se encuentra bajo estudio. Los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de la organización para dar respuesta a las siguientes preguntas claves:

- 1)¿Qué es lo que se hace?
- 2)¿Cómo se hace?
- 3)¿Con que frecuencia se presenta?
- 4)¿Qué tan grande es el volumen de transacciones o de decisiones?
- 5)¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
- 6)¿Existe algún problema?
- 7)Si existe un problema, ¿qué tan serio es?
- 8)Si existe un problema, ¿cuál es la causa que lo origina?

Para contestar estas preguntas, el analista conversa con varias personas para reunir detalles relacionados con los procesos de la organización, sus opiniones sobre por qué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso. Se emplean cuestionarios para obtener esta información cuando no es posible entrevistar, en forma personal, a los miembros de grupos grandes dentro de la organización. Asimismo, las investigaciones detalladas requieren el estudio de manuales y reportes, la observación en condiciones reales de las actividades del trabajo y, en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad.

Conforme se reúnen los detalles, los analistas estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles de procesamiento, tiempo de respuesta y métodos de entradas y salidas.

2.2.4.3 Diseño del sistema

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la de desarrollo de software a la que denominan diseño físico.

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida.

El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas, y símbolos especiales). La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software.

En este trabajo de investigación el diseño del sistema de información se efectuará utilizando para ello el *Lenguaje de Modelado Unificado* (UML), herramienta novedosa para el desarrollo de sistemas.

2.3 Ingeniería de software

Es una disciplina que integra proceso, métodos y herramientas para el desarrollo del software de computadora. Se han propuesto varios paradigmas diferentes, cada uno exhibiendo ventajas e inconvenientes, pero todos tienen una serie de fases genéricas en común.

2.3.2 Software [25]

Son procedimientos y reglas lógicas escritas en la forma de programas y aplicaciones, que definen el modo de operación de la computadora. Tienen carácter virtual (en contraposición con el hardware) y están almacenadas en los diferentes tipos de memoria de lectura/escritura. Comprende todo tipo de programas, utilidades, aplicaciones, sistemas operativos, drivers que hacen posible que el usuario pueda trabajar con la máquina.

Se pueden diferenciar dos grandes niveles a la hora de abordar que metodología emplearemos en la construcción de programas, estos son: el relativo a los pequeños programas, y el que se refiere al desarrollo de grandes sistemas de software, el primero normalmente lo realizan programadores individuales; y el segundo involucran a un gran equipo de programadores. Lógicamente, entre estos dos grandes niveles se encuentra toda una gama de situaciones intermedias. La programación a gran escala necesita apoyarse en unos principios sólidos y firmes que

faciliten el desarrollo de la actividad de programación. Estos principios los facilita la Ingeniería del Software.

2.3.3 Ciclo de Vida Del Software [25]

La Ingeniería del Software establece una metodología para el desarrollo de un buen sistema software conocida como el *ciclo de vida*, siendo éste el período de tiempo que se extiende desde la concepción inicial del sistema, pasando por su uso y mantenimiento, hasta su posterior retirada o sustitución por un nuevo producto de programación. Es una metodología que se desarrolla en cascada, parte de la base de que no se puede pasar a cada una de las etapas sin haber dado por concluidas todas las anteriores.

2.3.4 Etapas de la Ingeniería de Software [25]

La ingeniería de software requiere llevar a cabo muchas tareas, sobre todo las siguientes:

- **Análisis de requisitos:** Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de

software para reconocer requisitos incompletos, ambiguos o contradictorios.

- **Especificación:** Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.

- **Diseño y Arquitectura:** Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. (Edward Yourdon dice que consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc.

- **Programación:** Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga.

- **Prueba:** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una

técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral.

- **Documentación:** Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.
- **Mantenimiento:** Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o bugs. La mayor parte consiste en extender el sistema para hacer nuevas cosas. De manera similar, alrededor de 2/3 de toda la ingeniería civil, arquitectura y trabajo de construcción es dar mantenimiento.

2.4 Programación orientada a objetos

2.4.1 Concepto [14]

La programación orientada a objetos se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos

de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

Ventajas de un lenguaje orientado a objetos:

- Fomenta la reutilización y extensión del código.
- Relacionar el sistema al mundo real.
- Facilita la creación de programas visuales.
- Construcción de prototipos.
- Agiliza el desarrollo de software.
- Facilita el mantenimiento del software.

- **Objetos:** Es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados.
- **Las Clases:** Las clases son declaraciones de objetos, son abstracciones de objetos, es decir que la definición de un objeto es la clase. Cuando programamos un objeto y definimos sus características y funcionalidades en realidad lo que estamos haciendo es programar una clase.

- **Herencia:** Sirve para crear objetos que incorporen propiedades y métodos de otros objetos. Así podremos construir unos objetos a partir de otros sin tener que reescribirlo todo.
- **Mensaje:** Un mensaje en un objeto es la acción de efectuar una llamada a un método.
- **Encapsulamiento y Ocultación:** Cada objeto es una estructura compleja en cuyo interior hay datos y programas, todos ellos relacionados entre sí, como si estuvieran encerrados conjuntamente en una cápsula. Esta propiedad se denomina *encapsulamiento*. Los objetos son inaccesibles, e impiden que otros objetos, los usuarios, o incluso los programadores conozcan cómo está distribuida la información o qué información hay disponible. Esta propiedad de los objetos se denomina *ocultación de la información*.

2.4.2 Fundamentos de lo Orientado a Objetos [4]

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 su 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. 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:** En 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).

2.5 Proceso unificado de desarrollo de software

2.5.1 Concepto [15]

Es el proceso de desarrollo de un conjunto de actividades necesarias, para transformar los requisitos de usuarios en un software, es muy flexible, porque se puede especializar para variedad de sistemas y áreas de aplicación, se basa en componentes que permiten describir al sistema software en construcción. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso unificado es soportado

por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

- Interacción con el usuario desde un inicio.
- Mitigación de riesgos antes de que ocurran.
- Liberaciones frecuentes.
- Aseguramiento de la calidad.
- Involucramiento del equipo en todas las decisiones del proyecto.
- Anticiparse al cambio de requerimientos.

2.5.2 Características [15]

- **Iterativo e Incremental:** El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición (figura 2.1). Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada una de estas

iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

- **Dirigido por los casos de uso:** En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración coja un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

- **Centrado en la arquitectura:** El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.

- **Enfocado en los riesgos:** El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

2.5.3 Proceso Unificado Rational (RUP) [15]

El RUP es un producto de Rational. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo, dichos ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados.

En la figura 1 se muestra el flujo de cada fase del ciclo de desarrollo de forma esquematizada:

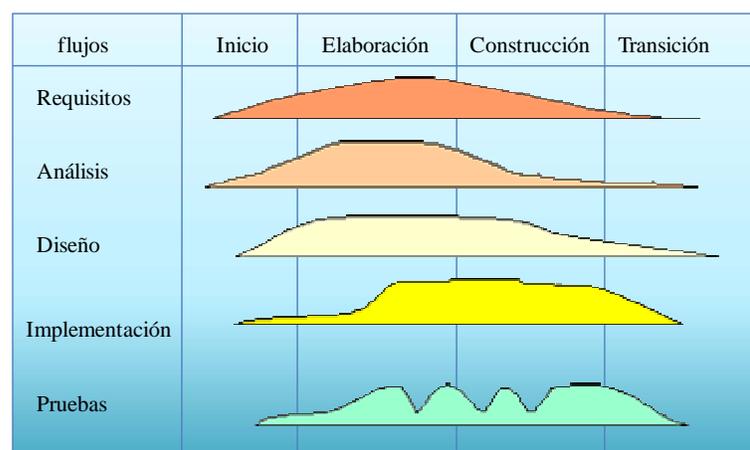


Figura 2.1. Fases del Ciclo de Desarrollo. Fuente: **Jacobson I. (2000)**

2.5.4 Flujos de Trabajos para una Iteración [15]

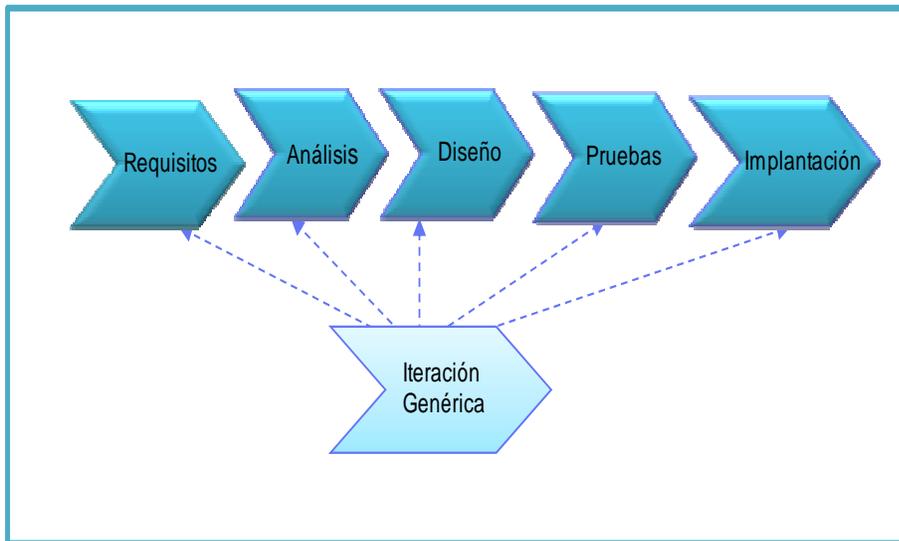


Figura 2.2. Flujos de trabajo. Fuente: Jacobson I. (2000)

Cada iteración está formada por cinco grupos de flujos de trabajos fundamentales, estos flujos se adaptan según sea la fase para la cual se está desarrollando. En la figura 2.2, se observa cómo, para una iteración de cualquier fase, coexisten los cinco flujos de trabajos.

2.5.5 Ventajas del Proceso Unificado de Desarrollo de Software [15]

La Iteración controlada acelera el ritmo de esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo.

La iteración controlada reconoce una realidad que a menudo ignora que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio. Mediante proceso unificado racional los requisitos se adquieren y refinan en sucesivas iteraciones.

En cada fase del proceso unificado se genera un producto; así mismo el desarrollo de cada fase puede contener algunos de los siguientes modelos: requisitos, análisis, diseño, implementación, despliegue y pruebas.

Los modelos son ideales porque mediante éstos se representa el trabajo evolutivo que desarrollan los ingenieros de software; y no incluye un producto final, como lo es la versión Beta del sistema; es decir no se requiere esperar la culminación del trabajo para observar un producto.

Los modelos juntos representan al sistema como un todo.

El proceso unificado está basado en componentes; y utiliza el estándar del lenguaje unificado de modelado para llevar a cabo sus modelos.

2.6 Lenguaje unificado de modelado (UML)

2.6.1 Concepto [3]

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos “business”.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.

- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grade

Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares.

En 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de un metamodelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de 1997, los seis promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML, llamado UML versión 1.1. Este documento fue aprobado por el OMG en Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El

OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el 1 de abril de 1999.

2.6.2 Modelos [3]

Un modelo es la representación en un cierto medio de algo en el mismo u otro medio. El modelo captura los aspectos importantes del ente que será modelado desde un cierto punto de vista, simplificando u omitiendo el resto.

- **Modelo de Casos de Uso:** El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos. Contiene actores, casos de uso y sus relaciones.
- **Modelo de Análisis:** El modelo de análisis se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. La utilización de otros paquetes de análisis es por tanto una forma de organizar el modelo de análisis en partes más manejables que representan abstracciones de subsistemas y posiblemente capas completas del diseño del sistema.
- **Modelo de Diseño:** El modelo de diseño es un modelo de objetos que describe la realización física de los casos de usos

centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

- **Modelo de Despliegue:** El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos del cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.
- **Modelo de Implementación:** El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros.

- **Modelo de Prueba:** El modelo de prueba describe principalmente cómo se prueban los componentes ejecutables (como las construcciones) en el modelo de implementación con pruebas de integración y de sistema. El modelo de pruebas puede describir también cómo han de ser probados aspectos específicos del sistema.

2.6.3 Diagramas [17]

Un diagrama es la representación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

2.6.3.1 Diagrama de Casos de Uso [17]

Es la técnica más efectiva y a la vez más simple para modelar los requisitos del sistema desde la perspectiva del usuario. Los casos de uso se utilizan para modelar cómo un sistema o negocio funciona actualmente, o cómo los usuarios desean que funcione. No es realmente una aproximación a la orientación a objetos; es realmente una forma de modelar procesos. Es, sin embargo, una manera muy buena de dirigirse hacia el análisis de sistemas orientado a objetos. Los casos de uso son generalmente el punto de partida del análisis orientado a objetos con UML.

Elementos Principales

El modelo de casos de uso consiste en actores y casos de uso (figura 2.2). Los actores representan usuarios y otros sistemas que interaccionan con el sistema. Se dibujan como “muñecos de palo”. Actualmente representan el tipo de usuario, no una instancia del mismo.

Los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor. Se dibujan como elipses.

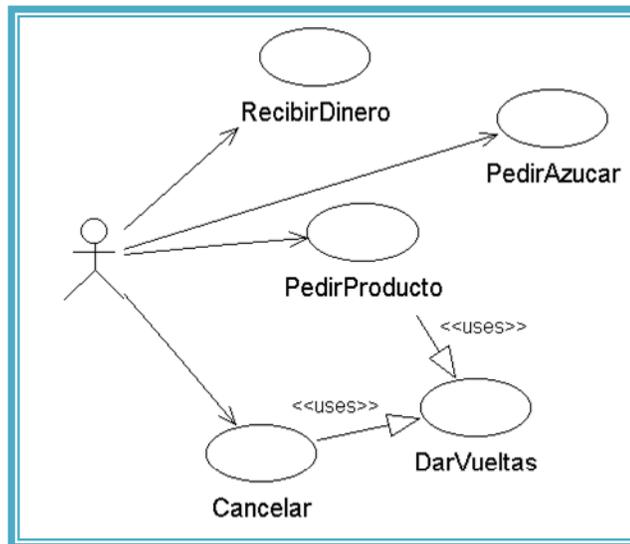


Figura 2.3.- Casos de uso de una máquina de café. Fuente: Popkin S. (1997)

Relaciones

- *Asociación*: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación. Dicha relación se denota con una flecha simple.
- *Dependencia o Instanciación*: Es una forma muy particular de relación entre las clases, en la cual una clase depende de otra, es

decir, se instancia. Dicha relación se denota con una flecha punteada.

- *Generalización:* Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de uso (<<include>>) o de herencia (<<extends>>). Este tipo de relación está orientado exclusivamente para casos de uso, y no para actores.

2.6.3.2 Diagrama de Secuencia [17]

El Diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso. Mientras que el diagrama de caso de uso permite el modelado de una vista “business” del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se tiene modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede “caminar sobre” esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria (figura 2.3).

Durante el análisis inicial, el modelador típicamente coloca el nombre “business” de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre “business” es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

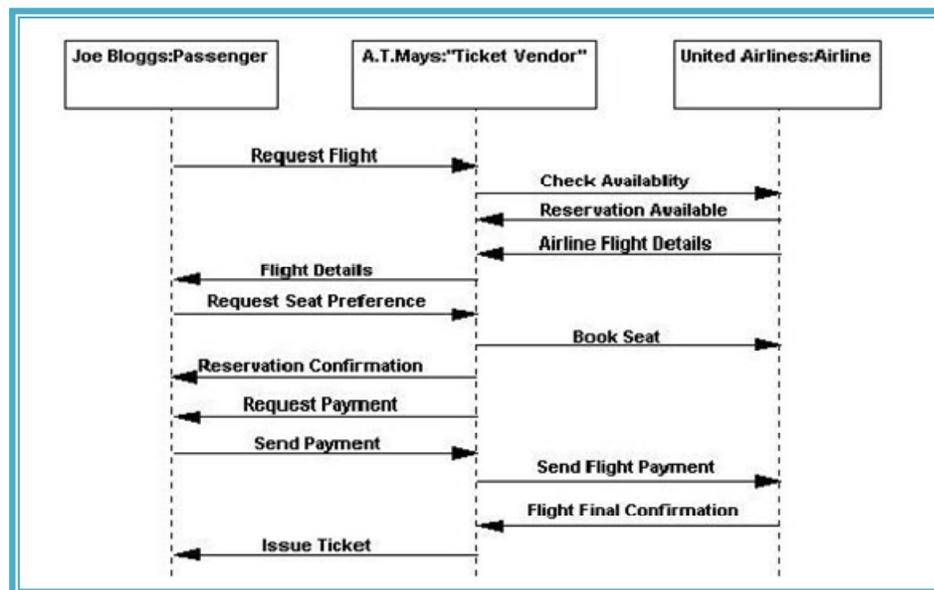


Figura 2.4. Diagrama de Secuencia para un escenario. Fuente: Popkin S. (1997)

2.6.3.3 Diagrama de Colaboración [24]

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas en torno a los 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 contra 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 más frecuentemente en la fase de diseño, es decir, cuando estamos diseñando la implementación de las relaciones. La siguiente figura 2.5 muestra un ejemplo de diagrama de colaboración.

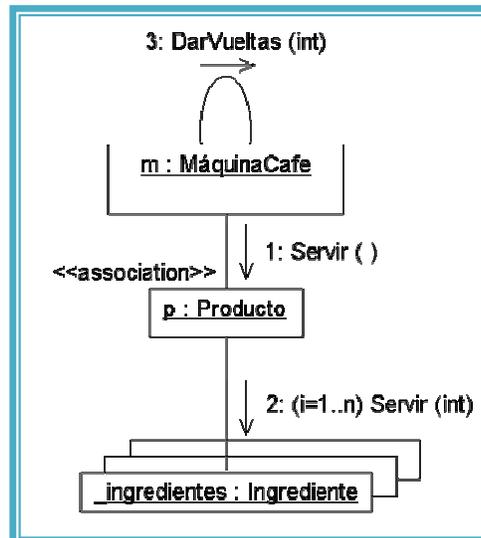


Figura 2.5.- Ejemplo diagrama de colaboración. Fuente: Vilas A. (2002)

A diferencia de otras notaciones que muestran tanto el estado y el comportamiento de la clase en el diagrama de clases, UML separa el comportamiento de las clases en los diagramas de colaboración. Los diagramas de clase de UML no incluyen flujo de mensajes entre clases, es por esto que los diagramas de colaboración se deben crear en paralelo con los diagramas de clases. Aunque se puede indicar el orden del flujo de mensajes en un diagrama de colaboración numerando los mensajes, no se suele hacer, ya que para este propósito son mejores los diagramas de secuencia.

2.6.3.4 Diagrama de Estado [7]

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro.

Los Diagramas de Estado representan autómatas de estados finitos, desde el p.d.v. de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El

estado está caracterizado parcialmente por los valores algunos de los atributos del objeto. El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase (figura 2.5). Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos, son grafos dirigidos y deterministas. La transición entre estados es instantánea y se debe a la ocurrencia de un evento.

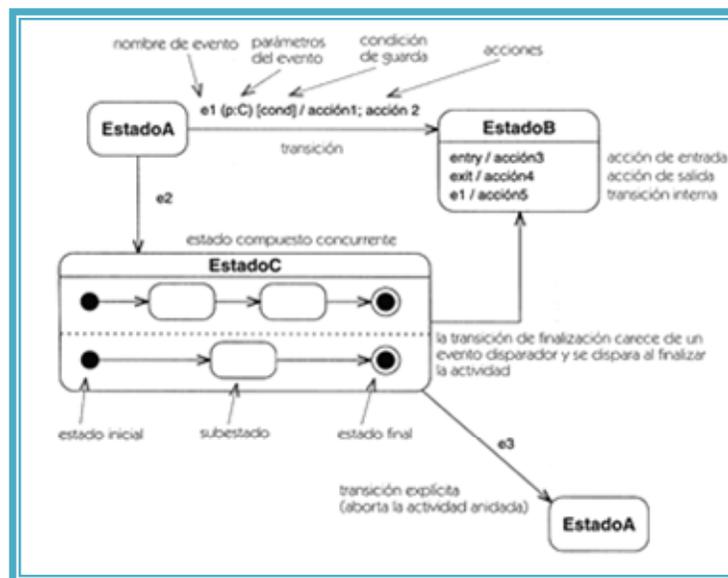


Figura 2.6-. Diagrama de Estado. Fuente: Consultoría S. (2004)

Estado:

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes

redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente).

Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de verdadero o falso.
- Recepción de una señal de otro objeto en el modelo.
- Recepción de un mensaje.
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular
- El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

Envío de mensajes

Además de mostrar y transición de estados por medio de eventos, puede representarse el momento en el cual se envían mensajes a otros objetos. Esto se realiza mediante una línea punteada dirigida al diagrama de estados del objeto receptor del mensaje.

2.6.3.5 Diagrama de Clases [22]

Es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño se usa el mismo diagrama y se modifica para satisfacer los detalles de las implementaciones.

Aproximación a un Caso de Uso guiado

En una aproximación a un Caso de Uso guiado hacia el análisis orientado a objetos, el diagrama de clases se desarrolla a través de información obtenida en los Casos de Uso, Diagramas de Secuencia y Diagramas de Colaboración (figura 2.6).

Los objetos encontrados durante el análisis son modelados en términos de la clase a la que instancian, y las interacciones entre objetos son referenciados a relaciones entre las clases instanciadas.

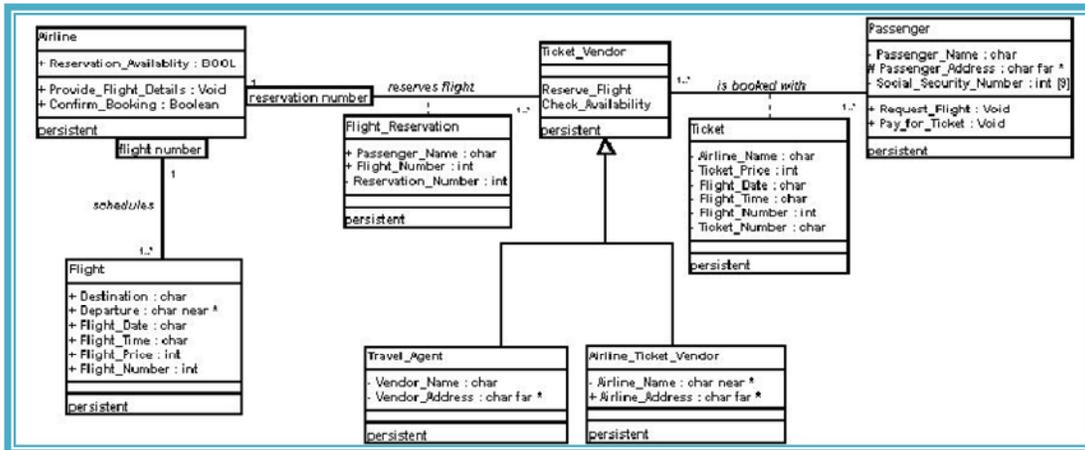


Figura 2.7.- Diagrama de clases. Fuente: Popkin S. (1997)

2.6.3.6 Diagrama de Componentes [9]

Un diagrama de componentes muestra un conjunto de componentes y sus relaciones. Los diagramas de componentes se utilizan para describir la vista e implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases en que un componente generalmente se compone de una o más clases, interfaces o colaboraciones.

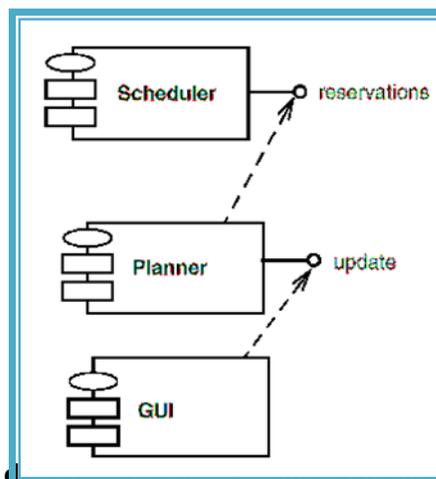


Figura 2.8.- Ejemplo de Diagrama de Componentes. Fuente: Figueroa P. (1997)

2.6.3.7 Diagramas en el diseño de aplicaciones Web [6]

Mientras que los objetos están siendo partidos, las páginas también están siendo definidas. Esta actividad involucra descubrir las páginas web y sus relaciones de unas con otras y con los objetos del sistema. Este paso depende mucho del patrón de arquitectura de la aplicación. Por ejemplo, es enteramente posible para una aplicación de tipo Web de entrega usar solamente una página Web. Esta página particular, ciertamente será cargada con objetos y Applets complejos y todavía para ciertas situaciones, sería la mejor solución.

Para describir las aplicaciones Web, se utilizan diferentes diagramas de UML, como diagramas de clases, de secuencia, entre otros. Existe además un diagrama muy importante, el diagrama de vista lógica, donde se presentan en conjunto las páginas Web conceptuales y los objetos del negocio. Las páginas Web son similares a objetos, lógicamente tienen un propósito simple y relaciones con otros objetos. En este tipo de diagrama, ellos coexisten con clases suministradas por el sistema y aún más importante, tienen su relación expresada en el diagrama.

2.7 Tecnologías Web

2.7.1 Internet [16]

Es un método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única. De ahí que Internet se conozca comúnmente con el nombre de "red de redes", pero es importante destacar que Internet no es un nuevo tipo de red física, sino un método de interconexión.

2.7.2 Aplicación Cliente - Servidor [16]

Programa compartido en toda una red. El programa se encuentra almacenado en un servidor de red y puede ser utilizado simultáneamente por más de un cliente.

2.7.3 Servidor Web [11]

Un servidor web es un programa que implementa el *protocolo HTTP* (*HyperText Transfer Protocol*). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (*HyperText Markup Language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una *página web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Servidor web internet Information Server

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Servidor web Apache

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

Los Servicios de Internet Information Services (IIS) proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor

Web seguro. Si ha pensado alojar un sitio Web y FTP (File Transfer Protocol, Protocolo de transferencia de archivos) con IIS, configure el servidor como un servidor de aplicaciones.

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

2.7.4 Lenguaje de Etiquetas por Hipertexto [12]

El lenguaje de marcas de hipertexto que se utiliza para documentos del World Wide Web. HTML es una aplicación de SGML que utiliza etiquetas para marcar los elementos, como texto y gráficos, en un documento para indicar como deberían visualizar los exploradores Web estos elementos al usuario y como deberían responder a las acciones del usuario, como la activación de un enlace presionando una tecla o haciendo clic con el ratón. HTML 2.0, definido por el Internet Engineering Task Force (IETF), incluye características del HTML común a todos los exploradores Web alrededor de 1995 y fue la primera versión de HTML ampliamente utilizado en el WWW.

En 1994, se propuso HTML+ para ampliar el HTML 2.0 pero nunca se implementó. HTML 3.0, que nunca se estandarizó ni fue implementado totalmente por ningún desarrollador de exploradores Web, introdujo las tablas. HTML 3.2, el último estándar propuesto, incorpora características ampliamente implementadas en 1996.

La mayoría de los exploradores, especialmente Netscape Navigator e Internet Explorer, reconocen más etiquetas de HTML que las incluidas en el estándar actual.

HTML 4, la última especificación, soporta hojas de estilo y lenguajes de script e incluye características de internacionalización y de accesibilidad.

2.7.5 Técnica de desarrollo web AJAX [6]

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

2.7.5.1 Tecnologías incluidas en AJAX

AJAX es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.

2.7.6 Aplicaciones Web [22]

El creciente desarrollo del comercio electrónico así como el desplazamiento de las distintas organizaciones hacia la Web ha traído en la actualidad una constante evolución de las aplicaciones Web. Cada día se incrementan las transacciones financieras, la transferencia de información confidencial y ejecución de procesos online, entre otros, las cuales exigen funcionalidad, confiabilidad, usabilidad y eficiencia por mencionar algunas características de calidad. Esta relevancia de la economía genera grandes desafíos en las aplicaciones Web que son los de controlar y mejorar su calidad.

Aunque las aplicaciones Web están creciendo rápidamente. La mayoría de los desarrolladores Web ponen poca atención en el análisis de requisitos, así como en las metodologías y procesos de desarrollo. Además los desarrolladores de aplicaciones confían excesivamente en el conocimiento y experticia de los desarrolladores individuales y sus prácticas de desarrollo individual más bien que en las prácticas estándar. No obstante, son las mismas metodologías de desarrollo las que no tratan de manera adecuada y profunda los atributos de calidad. Estas situaciones traen como consecuencia que los atributos de calidad de los sistemas basados en la Web tales como la funcionalidad, confiabilidad, mantenibilidad, usabilidad y portabilidad no se les da la debida consideración que se merecen durante el proceso de desarrollo.

La revisión de los modelos de calidad de la IEEE y la ISO/IEC y de esta forma tomarlos como referencia en el análisis de las metodologías. Después se revisa que es una aplicación Web así como su diferencia de las aplicaciones convencionales para poder identificar y comprender cuales son aquellas características particulares que las hacen diferentes y así saber a qué nuevos aspectos enfrentar en el tratamiento de los requisitos de calidad. También se describen los atributos de calidad de las aplicaciones Web que actualmente varios investigadores consideran en sus trabajos y se analiza cual es el punto de vista del autor, se muestran resumidamente todas aquellas propuestas que tratan los atributos de calidad así como las técnicas que utilizan y que atributos son los que considera. Por último se hace una clasificación de los atributos que trata cada metodología en base al estándar IEEE 1061 y es mediante este estándar y dicha clasificación que se construye la taxonomía general de los atributos de calidad. Con esta taxonomía se pretende mostrar a los desarrolladores de sistemas Web cuales son aquellas metodologías, técnicas, actividades y requisitos de calidad que se deben considerar en la etapa de ingeniería de requisitos para dar mayor garantía de calidad a las aplicaciones Web.

No obstante cabe destacar que hay mucho campo de investigación en el área de ingeniería de requisitos de los sistemas Web puesto que la mayoría de las metodologías no consideran los atributos de calidad y aquellas que lo hacen no lo tienen muy claro o lo hacen de forma limitada, la ingeniería Web es una disciplina que continua con su proceso de madurez.

2.7.6.1 Anatomía de una Página Web [2]

Según Mas (2005) una página Web es superficialmente parecida a cualquier otro documento: un texto, unas imágenes, todo compuesto de una determinada manera. Una página Web es un tipo de fichero que tiene poco de particular: se trata simplemente de un fichero de texto, con una extensión .htm o .html (de hypertext markup language - lenguaje de hipertexto.) Este fichero contiene el texto más una serie de códigos que permiten dar formato a la página en el navegador: por ejemplo, distribuir en columnas, poner letras en negrita, asignar colores, rodear una imagen con texto... El programa navegador (normalmente Internet Explorer o Navigator) interpreta los códigos del html para mostrar en pantalla la información contenida y del modo que se ha especificado aquellos códigos. En la figura 2.8 se observan los elementos principales que conforman una página web.

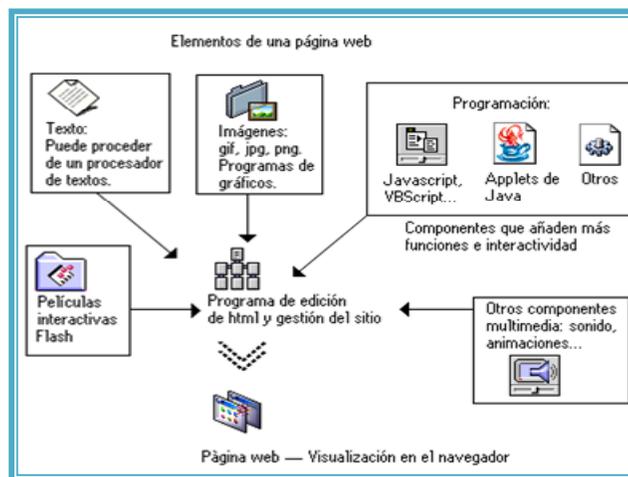


Figura 2.9 Anatomía de una página Web. Fuente: Appleby M. (1998)

Principales componentes de la página Web típica, como se muestra en la ilustración anterior:

Texto. El texto editable se muestra en pantalla con alguna de las fuentes que el usuario tiene instaladas (a veces se utiliza una tecnología de fuentes incrustadas, con lo que vemos en el monitor una fuente que realmente no se tiene, pero es poco frecuente).

Gráficos. Son ficheros enlazados desde el fichero de la página propiamente dicho. Se puede hablar de dos formatos casi exclusivamente: GIF y JPG.

Formularios. Son una mezcla de texto y a veces gráficos, que permiten enviar información por parte del visitante, por ejemplo, consultando un catálogo, solicitando más información, comunicando su opinión, votando en una encuesta. Existen diferentes modelos de formulario; algunos simplemente se envían por correo electrónico; otros funcionan ejecutando un programa guión en el servidor.

2.7.7 Concepto general de WebML [20]

Es un lenguaje de modelado de alto nivel para la especificación de hipertexto, el cual consiste en simples conceptos visuales para expresar un hipertexto como un conjunto de páginas compuestas de unidades de contenido y operaciones enlazadas, y para unir tales unidades y operaciones a la data que hacen referencia. WebML sigue

el estilo reconocido de lenguajes de modelado conceptual como Entidad – Relación y UML: cada concepto tiene una representación gráfica y las especificaciones son diagramas. Así como el modelo Entidad – Relación construye, también los diagramas de WebML se pueden representar usando la sintaxis de UML, posiblemente con pérdida de concisión pero no del poder expresivo.

En general, WebML es un audaz e impresionante logro. Reúne elegantemente y parece ser capaz de construir cualquier cosa a través del uso apropiado de composición y la construcción de bloques. Es impresionante que WEBML ya tenga un lenguaje y modelo de diseño visual bien trabajado. Se basa en tres modelos fundamentales para el desarrollo de aplicaciones Web, como lo son:

- **Modelo de Datos:** El modelo de datos de WebML es una adaptación conveniente de los modelos conceptuales de diseño de datos que se emplea en otras disciplinas como diseño de bases de datos, ingeniería de software y representación del conocimiento. El modelo de datos de WebML es compatible con el modelo de datos Entidad – Relación usado en el diseño conceptual de bases de datos, también es compatible con los diagramas de clase UML empleados en el modelado orientado a objetos.

- El elemento fundamental del modelo de datos son las entidades, definidas como contenedores de elementos de datos, y sus relaciones definidas como las conexiones semánticas entre entidades. Las entidades tienen propiedades, llamadas atributos, con un tipo asociado. Las entidades pueden ser organizadas de manera jerárquica y sus relaciones pueden restringirse por medio de la cardinalidad.

- **Modelo de Hipertexto:** El modelo de hipertexto especifica cómo se compone el sitio y la navegación en el sitio. La composición del sitio describe las páginas que forman parte del hipertexto y las unidades de contenido que constituyen cada página. Las páginas de un sitio Web son catalogadas como contenedores de información que es enviada al usuario. Las unidades son elementos atómicos de contenido, empleadas para publicar información descrita en el modelo de datos. WebML contiene siete (7) tipos de unidades predefinidas para desarrollar páginas Web: data, multi-data, index (y sus variantes jerárquicas y selección múltiple), entry, scroller. Cada unidad está asociada a una entidad subyacente, de la cual se obtiene el contenido. La

especificación de una entidad subyacente determina el tipo de objeto del cual se deriva el contenido de la unidad.

- La navegación del sitio se realiza a través de enlaces, los cuales se definen entre unidades que se encuentran en una misma página, en diferentes páginas o entre páginas completas. La información transportada a través de los enlaces se conoce como contexto de navegación o simplemente contexto. Los enlaces que transportan información contextual se denominan enlaces contextuales mientras que los que no transportan información son conocidos como enlaces no contextuales. La información contextual generalmente es necesaria para asegurar las operaciones de computación de las unidades.

- El modelo de hipertexto debería estar al nivel correcto de abstracción; la especificación del hipertexto debe ser mantenida a un nivel conceptual, el cual significa que no debe comprometer mucho el diseño y la implementación de los detalles, tales como la distribución actual de la funcionalidad entre las varias capas de la aplicación Web.

- **Modelo de Gestión de Contenido:** Las aplicaciones Web con frecuencia realizan operaciones sobre datos. Ejemplos de esto es llenar formularios de un perfil personal, la adición de ítems al carrito de compras, o la actualización de contenido publicado en la Web por medios de aplicaciones de manejo de contenidos. En todos estos casos, las acciones llevadas a cabo por medio de interfaces Web, tienen efectos secundarios, por ejemplo ellos cambian el contenido de algunos orígenes de datos conectados al sitio Web. Además de actualizar los datos, las aplicaciones Web pueden invocar programas definidos externamente, enviándoles entradas de datos, el cual depende del contenido de la página y de lo que escoja el usuario. Ejemplos de tales operaciones de propósito general es la identificación de un usuario, enviar un mail, entre otros

Estas operaciones en WebML no afectan el modelo de datos, y requiere dos simples e intuitivas extensiones del modelo de hipertexto. La primera extensión es la notación de unidades operacionales, las cuales son usadas para expresar algún proceso ejecutado como el resultado de navegar un enlace; una unidad operacional puede denotar una manipulación de datos o la ejecución de un servicio externo genérico.

La segunda extensión se aplica a unidades operacionales de los enlaces de salida, los cuales están distinguidos en OK-links y KO-links. OK y KO-links capturan el concepto de operación exitosa y fallida, respectivamente, y permiten al diseñador tomar rutas alternativas después de la ejecución de una operación, dependiendo del resultado de la ejecución.

WebML incluye muchas unidades operacionales predefinidas, las cuales ofrecen la mayoría de las primitivas más usadas para actualizar instancias de las entidades y relaciones de la aplicación, creando, modificando, y borrando objetos, y conectándose y desconectándose entre relaciones, además otras operaciones útiles como ingreso de usuario, salida de usuario y envío de e-mails.

Las operaciones de manejo de contenido predefinidas pueden ser agrupadas en transacciones, las cuales son secuencias de actualizaciones ejecutadas automáticamente; cuando una transacción es especificada, o toda la secuencia de operaciones que la constituyen se ejecuta exitosamente, o la secuencia toda no se ejecuta.

2.7.7.1 Sumario de Estereotipos del Modelo de Hipertexto WebML [20]

La herramienta de modelado WebML ofrece un amplio conjunto de notaciones y estereotipos que permiten representar la definición, estructuración e interpretación de aplicaciones web.

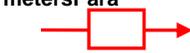
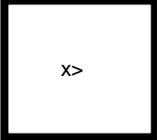
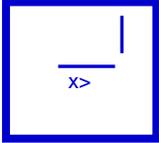
En la tabla 2.1 Se describe cada estereotipo WebML (1/8)

Elemento WebML	Descripción	Propiedades
<p data-bbox="412 795 605 827">Entity (Entidad)</p> 	<p data-bbox="670 795 1015 961">Una entidad es la representación de un objeto o concepto del mundo real que se describe en una base de datos.</p>	<p data-bbox="1154 795 1252 827">Ninguna</p>
<p data-bbox="375 1056 594 1129">ISA Hierarchy (Jerarquías ISA)</p> 	<p data-bbox="670 1056 1015 1423">Jerarquías Isa: Se dice A isa B si el conjunto de entidades B es una generalización del conjunto de entidades A. $\text{attrib}(B) \subseteq \text{attrib}(A)$. A de B (en el mismo sentido de la programación orientada a objetos).</p>	<p data-bbox="1154 1056 1252 1087">Ninguna</p>
<p data-bbox="399 1495 574 1568">Relations... (Relaciones)</p> 	<p data-bbox="670 1495 1015 1610">Una relación entre dos o más entidades describe alguna interacción entre las mismas.</p>	<p data-bbox="1154 1495 1252 1526">Ninguna</p>

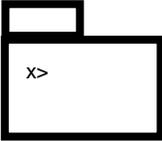
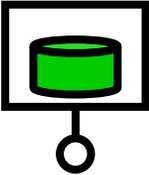
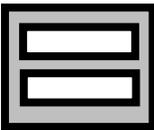
En la tabla 2.1 Se describe cada estereotipo WebML (2/8)

Elemento WebML	Descripción	Propiedades
<p>Link (Enlace)</p> <p>Para</p> 	<p>Es una conexión orientada entre dos unidades o páginas. Abstrae el concepto de ancla y permite portar información (por medio de parámetros entre unidades)</p> <p>Pueden ser definidos como:</p> <ul style="list-style-type: none"> » Enlace de transporte » Enlace automático 	<p>Enlaces normales, automáticos y de transporte.</p> <ul style="list-style-type: none"> • Nombre • Elemento origen • Elemento destino • Tipo de enlace • Parámetros de enlace
<p>Transport Link (Enlace de transporte)</p> <p>Para</p> 	<p>Enlace de Transporte: no funcionan como un ancla, pero son capaces de pasar parámetros.</p>	<p>Parámetros de enlace</p> <ul style="list-style-type: none"> • Nombre • Valor origen
<p>Automatic Link (Enlace automatico)</p> <p>Para</p> 	<p>Enlace Automático: son navegados sin la intervención del usuario.</p>	<p>Parámetros de enlace</p> <ul style="list-style-type: none"> • Nombre • Valor origen
<p>Link OK (Enlace OK)</p> <p>Para</p> 	<p>Enlace OK: enlace de operación distinguida, se ejecuta en caso de que la operación haya sido exitosa</p>	<ul style="list-style-type: none"> • Nombre • Elemento origen (unidad de operación) • Elemento destino <p>Parámetros de enlace</p>

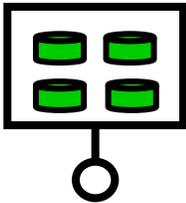
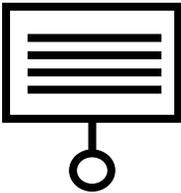
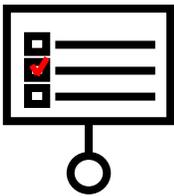
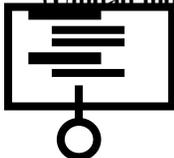
En la tabla 2.1 Se describe cada estereotipo WebML (3/8)

Elemento WebML	Descripción	Propiedades
<p>Link KO (Enlace KO)</p> <p>Para</p> 	<p>Enlace KO: enlace de operación distinguida, se ejecuta en caso de que la operación falle.</p>	<ul style="list-style-type: none"> • Nombre • Elemento origen (unidad de operación) • Elemento destino • Parámetros de enlace.
<p>Window (Ventana)</p> 		<p>Ninguna</p>
<p>Web Page (Pagina Web)</p> 	<p>Representa la interfaz actual buscada por el usuario. Esta contiene sub-paginas AND/OR</p>	<ul style="list-style-type: none"> • Nombre • Punto de referencia <p>Contenido: unidades o subpáginas</p>
<p>Area (Área)</p> 	<p>Es un contenedor de páginas o recursivamente de sub-áreas, donde cada una puede ser usada para obtener una organización jerárquica del hipertexto</p>	<ul style="list-style-type: none"> • Nombre • Punto de referencia <p>Contenido: páginas, sub-áreas y página por defecto o subpágina</p>

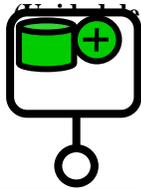
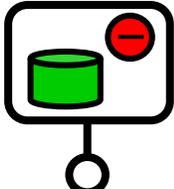
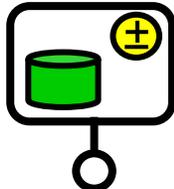
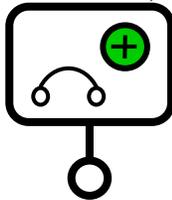
En la tabla 2.1 Se describe cada estereotipo WebML (4/8)

Elemento WebML	Descripción	Propiedades
<p data-bbox="402 453 615 527">Siteview (Vista del sitio)</p> 	<p data-bbox="670 453 1015 527">Representa una vista de hipertexto</p>	<ul data-bbox="1044 453 1382 575" style="list-style-type: none"> • Nombre • Contenido: páginas, áreas y página inicial
<p data-bbox="391 774 578 848">Transaction (Transacción)</p> 	<p data-bbox="670 774 1015 1073">Es una secuencia de operaciones ejecutadas atómicamente, lo que significa que todas las operaciones individuales se ejecutan exitosamente o toda la secuencia entera no es hecha</p>	<p data-bbox="1117 774 1211 800">Ninguna</p>
<p data-bbox="386 1188 634 1262">Data Unit (Unidad de datos)</p> 	<p data-bbox="670 1188 1015 1310">Publica un objeto obtenido de una entidad determinada</p>	<ul data-bbox="1044 1188 1365 1360" style="list-style-type: none"> • Nombre • Entidad de Origen • Selector (opcional) <p data-bbox="1154 1331 1360 1356">Atributos incluidos</p>
<p data-bbox="380 1556 641 1629">Entry Unit (Unidad de entrada)</p> 	<p data-bbox="670 1556 1015 1677">Esta soporta el ingreso de datos basado en un formulario</p>	<ul data-bbox="1044 1556 1300 1635" style="list-style-type: none"> • Nombre • Punto de referencia <p data-bbox="1044 1650 1382 1730">Contenido: unidades o subpáginas</p>

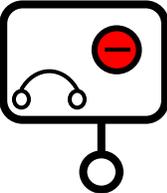
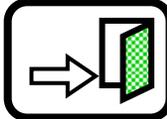
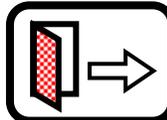
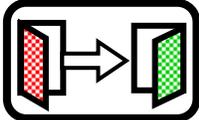
En la tabla 2.1 Se describe cada estereotipo WebML (5/8)

Elemento WebML	Descripción	Propiedades
<p>Multidata Unit (Unidad de datos múltiples)</p> 	<p>Presenta múltiples objetos de una entidad juntos, repitiendo la presentación de muchas unidades de dato</p>	<ul style="list-style-type: none"> • Nombre • Entidad de Origen • Selector (opcional) • Atributos incluidos • Cláusula de orden (opcional)
<p>Index Unit (Unidad índice)</p> 	<p>Presentan múltiples objetos de una entidad como una lista</p>	<ul style="list-style-type: none"> • Nombre • Entidad de Origen • Selector (opcional) • Atributos incluidos • Cláusula de orden (opcional)
<p>Multichoice (Unidad índice de múltiples elecciones)</p> 	<p>Una variante del anterior, donde cada elemento de la lista está asociado con un “checkbox” permitiendo al usuario seleccionar múltiples objetos</p>	<ul style="list-style-type: none"> • Nombre • Entidad de Origen • Selector (opcional) • Atributos incluidos • Cláusula de orden (opcional)
<p>Hierarchical Index Unit (Unidad índice)</p> 	<p>Una variante del “index unit”, en cual las entradas de los índices están organizadas en un árbol multi - nivel</p>	<ul style="list-style-type: none"> » Nombre » Por cada nivel: <ul style="list-style-type: none"> » Entidad de Origen » Selector (opcional) » Atributos incluidos » Cláusula de orden (opcional)

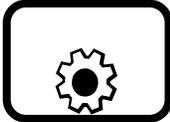
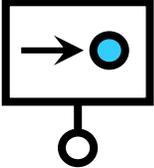
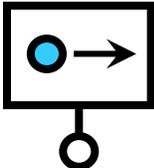
En la tabla 2.1 Se describe cada estereotipo WebML (6/8)

Elemento WebML	Descripción	Propiedades
<p>Create Unit (Unidad de creación)</p> 	<p>Habilita la creación de una nueva instancia de una entidad</p>	<ul style="list-style-type: none"> • Nombre • Entidad de origen • Conjunto de agnaciones de valores
<p>Delete Unit (Unidad de eliminación)</p> 	<p>Elimina uno o más objetos de una entidad dada</p>	<ul style="list-style-type: none"> • Nombre • Entidad de origen • Selector
<p>Modify Unit (Unidad de modificación)</p> 	<p>Modifica o actualiza uno o más objetos de una entidad dada</p>	<ul style="list-style-type: none"> • Nombre • Entidad de origen • Selector • Conjunto de asignaciones de valores
<p>Connect Unit (Unidad de conexión)</p> 	<p>Crea nueva instancia de una relación</p>	<ul style="list-style-type: none"> • Nombre • Rol de la relación • Selector de la entidad de origen • Selector de la entidad objetivo

En la tabla 2.1 Se describe cada estereotipo WebML (7/8)

Elemento WebML	Descripción	Propiedades
<p>Disconnect Unit (Unidad de desconexión)</p> 	<p>Elimina una instancia de una relación</p>	<ul style="list-style-type: none"> • Nombre • Rol de la relación • Selector de la entidad de origen • Selector de la entidad objetivo
<p>Login Unit (Unidad de entrada al sistema)</p> 	<p>Verifica la identidad de un usuario de acceda al sitio</p>	<ul style="list-style-type: none"> • Nombre de Usuario • Contraseña
<p>Logout Unit (Unidad de cierre de sesión)</p> 	<p>Lleva al usuario a la página principal sin control de acceso</p>	<ul style="list-style-type: none"> • Ninguno
<p>Change Group Unit (Unidad de cambio de grupo)</p> 	<p>Verifica la entidad de un usuario accediendo al sitio por medio de otro grupo de usuarios</p>	<ul style="list-style-type: none"> • Nombre de Usuario • Contraseña

En la tabla 2.1 Se describe cada estereotipo WebML (8/8)

Elemento WebML	Descripción	Propiedades
<p>Sendmail Unit (Unidad de envío de correo o mensajes)</p> 	<p>Provee la capacidad de enviar mensajes de email</p>	<ul style="list-style-type: none"> • Remitente • Recipiente • Asunto • Cuerpo • Adjuntos
<p>Generic Operation Unit (Unidad de Operación Genérica)</p> 	<p>Define una operación genérica: la contribución y el producto. Por el que los parámetros deben ser definidos por el diseñador</p>	<p>Las propiedades son definidos por el diseñador</p>
<p>Set Unit (Unidad de Asignación)</p> 	<p>Asigna un valor a un parámetro global</p>	<ul style="list-style-type: none"> • Parámetro global
<p>Get Unit (Unidad de Extracción)</p> 	<p>Retira el valor de un parámetro global</p>	<ul style="list-style-type: none"> • Parámetro global

2.8 Lenguaje de programación PHP

2.8.1 Definición [1]

Es uno de los lenguajes de lado servidor más extendido en la web y que ha sido creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3.

Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión

son su rapidez -gracias a que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código-, su mayor independencia del servidor Web -creando versiones de PHP nativas para más plataformas- y un API más elaborado y con más funciones.

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones. No es un lenguaje de marcado como podría ser HTML, XML o WML. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

2.9 Bases de datos

2.9.1 Definición [8]

Una base de datos es una colección de datos relacionados. Por datos, se quiere decir hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades implícitas:

Una base de datos representa algunos aspectos del mundo real, en ocasiones denominado mini mundo o Universo del Discurso. Los cambios en el mini mundo se reflejan en la base de datos.

Una base de datos es una colección de datos lógicamente coherentes, con algunos significados inherentes. Un conjunto aleatorio de datos no puede considerarse como una base de datos.

Las bases de datos se diseñan, construyen y pueblan con datos para un propósito específico. Está destinada a un grupo de usuarios y tiene algunas aplicaciones preconcebidas de interés para dichos usuarios.

2.9.2 Sistema De Gestión De Base De Datos (SGBD) [8]

Es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Por lo tanto un SGBD es un sistema de software de propósito general que facilita los procesos de definición, construcción, y manipulación de base de datos para distintas aplicaciones. Ahora veamos descrito cada uno de estos procesos:

La definición: consiste en especificar los tipos de de datos, las estructuras y restricciones para los datos que se van a almacenar en dicha base de datos.

La construcción: es el proceso de almacenar los datos concretos sobre algún medio de almacenamiento controlado por el SGBD.

La manipulación: incluye funciones como consultar la base de datos para recuperar unos datos específicos, actualizar la base de datos para reflejar los cambios ocurridos en el mini mundo, y generar informes a partir de los datos.

2.9.3 Arquitectura de un SGBD [8]

Una arquitectura apropiada para los sistemas de base de datos es la llamada arquitectura de tres esquemas también conocida como la arquitectura ANSI/SPARC, la cual tiene como objetivo separar las aplicaciones del usuario y la base de datos física. Esta arquitectura se define en los tres niveles siguientes:

El nivel interno: Tiene un esquema interno, que describe la estructura física de almacenamiento de la base de datos. El esquema interno emplea un modelo de datos físico y describe todos los detalles para su almacenamiento, así como los caminos de acceso para la base de datos.

El nivel conceptual: Tiene un esquema conceptual, que describe la estructura de la base de datos completa para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras físicas de almacenamiento y se concentra en describir entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones. En este nivel podemos usar un modelo de datos de alto nivel o uno de implementación.

El nivel externo o de vistas: Incluye varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado, y oculta a ese grupo el resto de la base de datos.

2.9.4 Modelo de datos [8]

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos. El concepto de estructura de una base de datos hace referencia a los tipos de datos, los vínculos y las restricciones que deben cumplirse para esos datos.

La mayoría de los modelos de datos contienen un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la base de datos. Los modelos de datos más utilizados son los conocidos modelos de datos de representación o de implementación y entre ellos están los más comunes que son:

El modelo de datos relacional.

El modelo de red.

El modelo jerárquico.

2.9.5 Arquitectura Cliente – Servidor [8]

Se usa para caracterizar un SGBD cuando la aplicación se ejecuta físicamente en una máquina, llamada cliente, y otra, el servidor, se encarga del almacenamiento y el acceso a los datos. Los proveedores ofrecen diversas combinaciones de clientes y servidores; por ejemplo, un servidor para varios clientes.

2.9.6 Administrados de Base de Datos (DBA) [8]

Es el responsable para el buen funcionamiento de toda una estructura de datos y todo su entorno de trabajo. Esta persona deberá tener un alto conocimiento de todo el sistema que envuelve al SGBD. Sobre este recae una alta responsabilidad, debido a las graves repercusiones que puede traer una mala gestión de la información.

Las funciones que se le asignan a un DBA Oracle son las siguientes:

- Instalación y actualización del software del SGBD.
- Analizar e instalar en la base de datos las aplicaciones con las que van a interactuar los usuarios. Esto, básicamente lleva consigo:
- Asignación de espacios (tablespaces) en la base de datos y ubicación de los mismos en los discos de la máquina.
- Creación de todos los objetos (tablas, índices, vistas, etc.) requeridos por cualquiera de las aplicaciones, así como su dimensionamiento dentro de la base de datos.

- Actualización de los parámetros del sistema para el correcto funcionamiento de las aplicaciones.
- Creación de los usuarios y asignación de privilegios, con el fin de controlar la seguridad de acceso de estos a la información de la base de datos.
- Controlar y monitorizar los accesos de los usuarios a la base de datos. Excesivas conexiones pueden provocar cuellos de botella en la red y bloqueos en algunos de los recursos de la máquina.
- Gestionar y optimizar el rendimiento de los procesos contra la base de datos. Esta será una de las funciones más importantes del administrador, porque de nada sirve tener una buena infraestructura montada si los colapsos son continuos y la lentitud en los accesos es constante.
- Planificación de los sistemas de backup y recuperación de los datos en caso de que sea necesario.
- Control del crecimiento de los archivos y procesos generados en el sistema por parte del mismo SGBD.

2.9.7 Concepto Generales Sobre la Arquitectura del SGBD ORACLE [10]

- **Archivo de configuración INIT.ORA:** Archivo de inicialización de valores para el arranque de la instancia y para el

establecimiento de configuraciones Oracle. Se especifican parámetros predeterminados con el valor que se desee, con el fin de optimizar los rendimientos o adecuar los recursos. Será donde se establezcan todos los parámetros para que Oracle trabaje de una u otra forma. Estos archivos son independientes por instancia, es decir, habrá tantos archivos de configuración como instancias activas en el sistema de información.

- **Archivos de control (Control Files):** Conjunto de archivos, de tamaño reducido y estable, encargados de mantener la consistencia del resto de los archivos de la base de datos. Mantendrán información como :
 - Contador de Commit almacenado en cada archivo.
 - Número de Redo Log actual.
 - Información adicional que indicará a la base de datos si los archivos son todos del mismo momento o se han recuperado de backup, el punto de arranque de la recuperación, etc. Aunque pequeños y generalmente olvidados, son trascendentales para trabajar con la base de datos.
- **Archivos históricos (Redo Log File):** Son archivos donde se irán almacenando los cambios provocados por las transacciones sobre la base de datos. Son utilizados exclusivamente en modo

escritura, excepto cuando se produce una recuperación de la base de datos.

- **Sentencias de Commit:** Es el punto de decisión de la transacción, indicando que los cambios hechos sobre la base de datos son validados y realizados. Se trata de un punto sin retorno para una transacción. Cuando se produce un Commit, éste provoca una escritura en los archivos históricos (redo log files), con el objetivo de que quede grabada la decisión del Commit. Los datos nuevos no tienen por qué ser escritos en disco de forma inmediata. En realidad, permanecen en memoria hasta que sean limpiados por falta de espacio, se produzca un Checkpoint o entren en los timeouts de limpieza de memoria.
- **Segmentos de Rollback:** Dan la posibilidad a la transacción de no hacer Commit sino Rollback. Esto quiere decir que, mientras que no se realice un commit, el resto de las transacciones deberán ver los datos anteriores al comienzo de la misma. Para ello, en los segmentos de rollback se guarda el valor anterior de los datos modificados por una transacción para la consistencia en lectura en entornos multi-usuario.
- **Puntos de ruptura (Checkpoint):** Eventos que se activan en un tiempo determinado para la limpieza de los bloques sucios de memoria. Un punto de ruptura se provocará porque :

- Ha habido un cambio en los archivos de redo log.
- Se ha llegado al límite previamente establecido por el parámetro `log_checkpoint_interval` del archivo de configuración `INIT.ORA`.
- Se ha llegado al límite previamente establecido por el parámetro `checkpoint_timeout`. El objetivo de los puntos de ruptura es poder tener marcas de recuperación basadas en los archivos históricos actuales (redo log files).
- **Archivos de base de datos:** Componentes físicos de la base de datos a nivel de sistema operativo, los cuales contendrán los datos generados por parte de los usuarios del sistema.
- **Espacios de tabla (tablespaces):** Agrupaciones lógicas de uno o más archivos de la base de datos. Un tablespace puede estar formado por uno o más archivos de la base de datos, y un archivo de la base de datos sólo puede estar englobado en un único tablespace. En los tablespaces será donde se ubiquen los datos que se creen en la base de datos, y físicamente se almacenarán en los archivos de base de datos que componen el tablespace.

2.9.8 Usuarios, profiles, roles, privilegios

Debemos tener en cuenta que al momento de crear la base de datos se crean automáticamente dos usuarios con todos los privilegios. Ellos son SYS y SYSTEM, para crear algún usuario o rol, o asignar privilegios o roles debemos conectarnos con alguno de estos dos usuarios.

2.9.8.1 Creación De Usuarios [10]

En la creación de usuarios se piden los siguientes requisitos:

- **Nombre:** nombre del usuario.
- **Password:** se debe especificar la clave del usuario.
- **Profile:** corresponde a un perfil que se le puede asignar al usuario el cual contiene:
 - Tiempo de CPU: se indica en seg/100 el tiempo máximo que el usuario puede usar la cpu.
 - CPU/Call: se indica en seg/100 el tiempo máximo para llamados al sistema.
 - Tiempo de Conexión: se indica el tiempo máximo de conexión en minutos.
 - Tiempo Desocupado: se indica en minutos el tiempo máximo que el usuario puede permanecer desocupado.

- Sesiones concurrentes: la cantidad máxima de sesiones conectadas a la base de datos.
- **Temporary- TableSpace:** generalmente se coloca el TableSpace temporal de la creación de la base de datos, pero puede usarse cualquiera. Hay que tener en cuenta que aquí se hacen las consultas y ordenaciones de este usuario.
- **Roles:** son un conjunto de privilegios que se asignan a los usuarios para trabajar en su entorno. Ya vienen predefinidos una serie de roles, pero también se pueden crear roles personalizados, a continuación se detallan los roles predefinidos:
- **DBA:** corresponde al súper usuario, es el que tiene asignado implícitamente todos los privilegios del sistema y demás roles. Los usuarios SYS y SYSTEM tienen este rol. Puede realizar todas las operaciones sobre almacenamiento, usuarios, objetos, backups, etc.

CONNECT: este rol se le asigna automáticamente a cada usuario que se cree en la base de datos. Quiere decir que tiene privilegios sobre sus objetos como conectarse a la base de datos y abrir una sesión, crear sus tablas, borrarlas y actualizarlas; pero no para otros usuarios.

RESOURCE: este rol le posibilita al usuario tener un conjunto de privilegios de sistema un poco más avanzado que CONNECT como crear triggers, procedimientos, secuencias, etc.

- **EXP_FULL_DATABASE:** este rol se asigna a usuarios para que puedan realizar backups de la base de datos usando la utilidad EXP.
- **IMP_FULL_DATABASE:** este rol se asigna a usuarios para que puedan restaurar backups usando la utilidad IMP.
- **System Privilegios** (privilegios del sistema): estos están asignados básicamente al tratamiento de objetos como crear tablas, crear vistas, etc. Cabe destacar que la mayoría de estos privilegios se encuentran en los roles del sistema.
- **Quotas:** se le indica al usuario el espacio en Kb o Mb que podrá manipular, se puede indicar un tamaño específico o ilimitado, por defecto no puede manipularlo. No solo se puede asignar espacio a sus tablespaces sino que a otros que no le pertenezcan.

CAPITULO III

FASE DE INICIO

3.1 Introducción

El Lenguaje de Modelado Unificado es una especificación de notación orientada a objetos, se compone de diagramas que representan las diferentes etapas del desarrollo del proyecto que personifican las diferentes vistas del mismo. Estos diagramas juntos son los que representa la arquitectura del proyecto.

La Figura 3.1 muestra cada una de las fases representadas en el Lenguaje de Modelado Unificado.

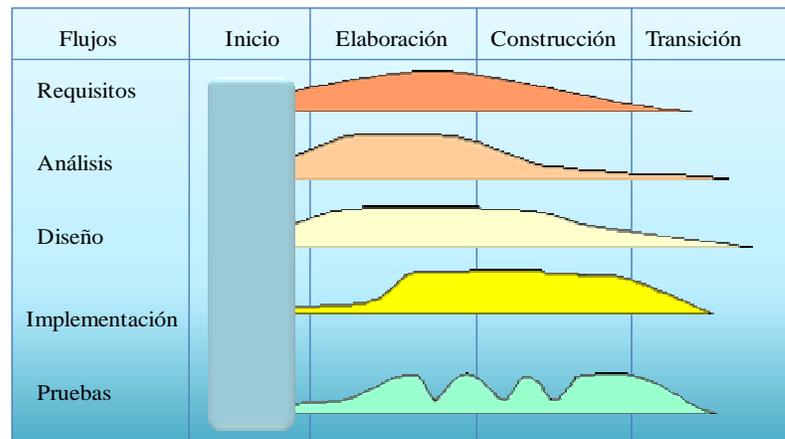


Figura 3.1: Fase de Inicio (UML) Fuente: Jacobson I. (2000)

El objetivo de la fase de inicio es la puesta en marcha del proyecto, en donde se describe una propuesta de arquitectura, se identifican los riesgos críticos y se

construye un prototipo que muestre que se pueden solucionar los problemas del cliente y de los usuarios finales.

En la fase de Inicio se obtienen los requisitos fundamentales del sistema, ya que se necesita adquirir conocimientos con respecto al proceso manual y a los posibles riesgos que se originarían con la automatización. En esta etapa se realizan los primeros modelos del sistema, tales como casos de usos y análisis.

Para el desarrollo de la fase de inicio los requerimientos se obtuvieron a través de observaciones, entrevistas, recolección de datos y análisis de información existente, (como entradas de datos, el flujo de información, control y salidas requeridas) lo que permitió modelar el sistema para la obtención de una arquitectura base. Se establece la forma en que los usuarios llevan a cabo los procesos, permitiendo determinar los requisitos adicionales para el funcionamiento del sistema e identificar los riesgos del mismo, lo que conlleva a una vista inicial del diseño, por ende, se aplican los siguientes conceptos:

1. Requisitos Candidatos.
2. Contexto del sistema.
3. Modelo del Dominio
 - ④ Clases del Dominio.
 - ④ Diagrama de clases del Dominio.

4. Requisitos funcionales.
 - Ⓜ Riesgos del Sistema.
 - Ⓜ Modelo de Casos de Usos
5. Requisitos no funcionales
 - Ⓜ Modelo de análisis.
 - Ⓜ Clases de Análisis
 - Ⓜ Paquete de Análisis
 - Ⓜ Diagrama de Colaboración
6. Identificación de los subsistemas.

3.2 Requisitos candidatos

Los requisitos candidatos son un conjunto de ideas donde se percibe un bosquejo general del software que se quiere desarrollar, estos se originan según las necesidades de cada usuario y aparecen durante la vida de un sistema.

Aplicando este concepto se tiene como requisitos Candidatos:

- Capacidad para gestionar el proceso de inscripción del área de Postgrado.
- Controlar el flujo de data asociado a los procesos.
- Proporcionar una sección para planificar los programas relacionado a cada Postgrado y emitiendo los reportes respectivos.

- El sistema debe ser capaz de permitir consultas y un menú de ayuda.
- Contar con una interfaz on-line amigable, clara y sencilla para que los usuarios puedan interactuar con el sistema sin complicaciones.
- Ser compatible con navegadores como FireFox, Opera, Internet Explorer, etc.

3.3 Contexto del sistema

El contexto del sistema viene dado por una visión general que abarque las características principales del sistema y la forma en que se realizan las tareas diarias dentro del área de Postgrado de la Universidad de Oriente. Para lograr tener esta perspectiva de los procesos que allí se ejecutan fue necesario realizar encuestas al personal que labora en ella, obteniendo información precisa el personal administrativo (secretarias y asistentes) y otras fuentes de información como tesis de grado realizadas con anterioridad, que contemplan sistemas de automatización de esta área, esto permitió tener una descripción de los procesos que se llevan a cabo a diario, las transacciones necesarias en cada proceso, las necesidades de información para el personal administrativo, docente, estudiantil y para la elaboración de los archivos cifrados de la OPSU.

El Software Gerencial para el Control de Actividades Académicas del área de Postgrado propone automatizar los procesos relacionados con la planificación de los períodos académicos de Postgrado, el proceso de inscripciones de nuevos estudiantes y estudiantes ya activos dentro del sistema, control de la data de estudiantes, profesores y personal administrativo además de computarizar el proceso de transacciones de notas y calificaciones de cada materia impartida en Postgrado.

3.3.1 Modelo de Dominio

El modelo de dominio representa las relaciones entre objetos adstratos o físicos como personas, procesos o procedimientos del sistema real, en ningún momento se refiere a fragmentos de código fuente o partes del programa computacional.

La función principal de este modelo es dar una mejor comprensión del funcionamiento real de las transacciones realizadas en el área de Postgrado y que son tomadas en cuenta para el desarrollo de este proyecto de grado. Además de ejemplificar algunas de las operaciones que se hacen a diario se pretende establecer claramente las relaciones entre las entidades del sistema, todo esto se logra gracias a la etapa de investigación previa que se realizó para conocer el contexto del sistema.

3.3.2 Clase de Dominio

El diagrama de clase de dominio permite representar los objetos y sus relaciones las dependencias que existen entre ellos, la herencia de una clase a otra, las asociaciones y composiciones además de una visión general del funcionamiento del sistema.

- Las clases de dominio mostradas a continuación nos brindan información pertinente sobre el sistema gerencial del área de postgrado, estas se listan a continuación:
- **Estudiante:** Alumno del área de postgrado, profesional en ramas distintas de ingeniería, administración, medicina, etc.
- **Personal Administrativo:** Relativo al personal de oficina dentro del área de postgrado como lo son secretarias, asistentes, coordinadores entre otros.
- **Personal Docente:** Abarca todo el profesorado que labora en esta prestigiosa institución.
- **Coordinador:** Representa los coordinadores de programas de postgrado.
- **Administrador de Sistema:** El personal encargado del mantenimiento de sistema.
- **Autoridades:** las autoridades de toda la universidad.
- **Programas de Postgrado:** Representa los programas de postgrado que son impartidos en la institución.

- **Grupos de Investigación:** Todos aquellos grupos formados por personal docente de postgrado.
- **Línea de Investigación:** Las investigaciones que realizan los grupos de investigación en una rama de la ciencia específica.
- **Trabajos de Investigación:** Comprende varias líneas de investigación que se realicen sobre un producto, un instrumento o una teoría determinada.
- **Base de Datos:** Representa todos los atributos de las bases de datos que pueda utilizar el sistema, normalmente por cada núcleo existe una y es idéntica a la de los otros núcleos.
- **Sesión:** Gestiona las sesiones de todos los usuarios.
- **Institución:** Envuelve los datos relativos a cada núcleo como sus teléfonos departamentos, etc.
- **Archivos OPSU:** gestiona la creación de dichos archivos extrayendo los datos de la base de datos.
- **Control de Estudios:** Representa la oficina de control de estudios de la universidad.

3.3.3 Diagrama de Clases de Dominio

Basándose en las clases de dominio obtenidas, se elabora un diagrama (figura 3.2) que muestra las asociaciones, interacciones, composiciones y herencia entre

Tabla 3.1.- Riesgos del Sistema (1/2)

Módulo	Descripción	Nivel	Contingencia
Hardware	<ul style="list-style-type: none"> Bajo rendimiento (lentitud al realizar operaciones) por insuficiencia de memoria RAM y/o poca capacidad de procesador. 	Crítico	<ul style="list-style-type: none"> Instalar memorias RAM y procesador de más capacidad.
	<ul style="list-style-type: none"> Tarjeta de red de baja potencia o inexistente 	Crítico	<ul style="list-style-type: none"> Instalar nueva tarjeta de red.
	<ul style="list-style-type: none"> Insuficiencia de espacio en disco duro 	Crítico	<ul style="list-style-type: none"> Instalar HDD de más capacidad.
Software	<ul style="list-style-type: none"> Fallas al generar reportes 	Crítico	<ul style="list-style-type: none"> Revisar código del sistema.
	<ul style="list-style-type: none"> Incorrecta manipulación del sistema 	No Crítico	<ul style="list-style-type: none"> Entrenar al Personal Generar Ayudas, manuales.
Base de Datos	<ul style="list-style-type: none"> No hay conexión entre aplicación y Base de datos. 	Muy Crítico	<ul style="list-style-type: none"> Revisar código del sistema
	<ul style="list-style-type: none"> Insuficiente espacio para datos en la base de datos. 	Crítico	<ul style="list-style-type: none"> Revisar Base de Datos Revisar HHDD Probar otros manejadores de B.D.

planes de prueba del sistema y controlar las iteraciones e integración del mismo. En este tipo de modelos existen dos componentes fundamentales: El **Actor**, rol que un usuario juega con respecto al sistema (que no necesariamente representa a una persona en particular, sino más bien la labor que realiza), y el **Caso de Uso**, que es la operación específica que se realiza tras una orden de algún agente externo. En la figura 3.3 se muestra el diagrama de casos de uso, el cual esquematiza las operaciones más relevantes del sistema al igual que la abstracción de los usuarios.

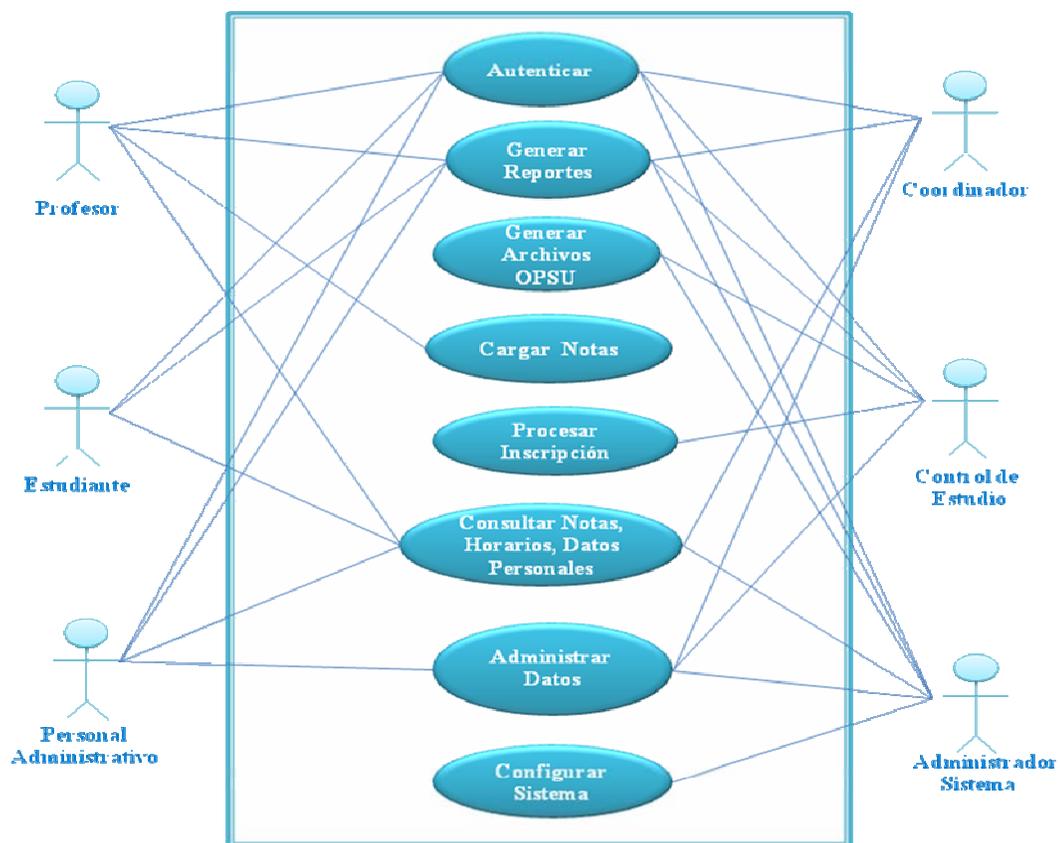


Figura 3.3. Diagrama de Casos de Uso del Sistema SISGAP. Fuente: Autores. 2009

3.5.1 Descripción de los Actores

En la tabla 3.2 se describen las características o roles principales de los actores que forman parte del sistema.

Tabla 3.2 Descripción del los Actores.

Actor	Descripción
 Coordinador	Representa al Coordinador de Programa de Postgrado, Coordinador de Postgrado de núcleo y al Coordinador General de Programa de Postgrado. Todos ellos necesitan información sobre el personal académico, administrativo y estudiantil de los Programas de Postgrado para realizar reportes periódicos y elevarlos a instancias superiores.
 Profesor	Persona encargada de impartir las cátedras de los Programas de Postgrado, asesorar estudiantes sobre sus cargas académicas y proyectos de grado así como realizar proyectos de investigación conjuntamente con otros profesores de su afinidad profesional.
 Administrador de Sistemas	Persona que mantiene operativo el sistema, además de realizar ajustes en caso de fallas y copias permanentes de la base de datos.
 Estudiantes	Persona que cursa un programa de postgrado en sus categorías de regular, libre, especial, oyente o condicional.
 Personal Administrativo	Engloba todo aquel personal administrativo de funciones rutinarias como secretarias, asistentes, asesores que trabajan en el área de postgrado.
 Control de estudio	Engloba todo aquel personal administrativo de funciones rutinarias como secretarias y demás personal que labora en control de estudios.

Fuente: Autores 2009

3.5.2 Descripción de los casos de uso

La tabla 3.3 contiene la descripción de los casos de uso del sistema.

Tabla 3.3 Descripción de los Casos de uso.

Caso de Uso	Descripción
Autenticar	Este caso de uso es necesario de cumplir para todo aquel que desee ingresar al sistema por el portal Web, se basa en la corroborar si un usuario tiene permiso para acceder al sistema, dependiendo de su categoría como actor tendrá un nivel de seguridad para acceder a todo el sistema o a sólo parte de él.
Generar Reportes	Está diseñado para generar constancias, Actas y reportes estadísticos de uso del sistema.
Generar Archivos OPSU	Permite al personal involucrado con el caso de uso generar archivos cifrados sobre datos del postgrado según exigencia de esta oficina gubernamental.
Cargar notas	El usuario ingresa en la base de datos la información referente a las notas de estudiantes.
Procesar Inscripción	El personal involucrado de control de estudio accede al sistema para formalizar la inscripción de los estudiantes seleccionados.
Consultar Notas, Horarios, Datos Personales	Los usuarios que ingresan a este caso de uso tienen acceso a información registrada en la base de datos referente a las notas, datos personales de sí mismo y los horarios de los programas de postgrado.
Administrar Datos	El personal administrativo accede al sistema para ingresar, modificar o eliminar datos de estudiantes, profesores, personal administrativo y programas.
Configurar Sistema	El administrador de sistema ingresa al sistema para ajustar los parámetros del programa según se requiera en la ocasión.

Fuente: Autores. 2009

3.5.3 Diagrama detallado del caso de uso “Autenticar”

1. Actores: Estudiantes, profesores, personal administrativo, coordinadores, Control de Estudio y administrador Sistema.
2. Descripción: Permite al usuario ingresar al sistema.
3. Pseudocódigo:

Flujo Principal:

- El usuario invoca al caso de uso entrar al sistema.
- El usuario ingresa su nombre de usuario y su contraseña.
- El sistema verifica los datos y dependiendo del grupo en el que se encuentre el sistema le proporciona los permisos de acceso o deniega el mismo.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar el ingreso al sistema.

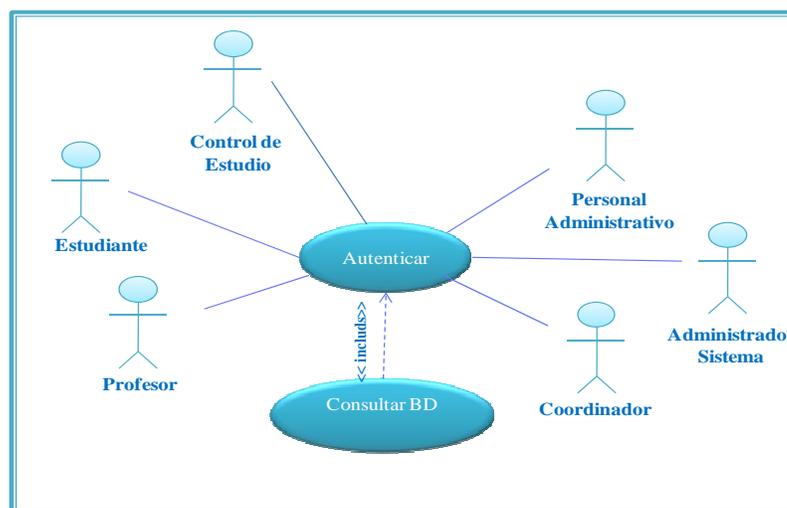


Figura 3.4 Diagrama de Casos de Uso Autenticar. Fuente: Autores. 2009

3.5.4 Diagrama Detallado del Caso de uso “Generar Reportes”

- Actores: Personal Administrativo, Profesores, Estudiantes, Coordinador, Control de Estudio y Administrador de Sistema.
- Descripción: Les permite a los distintos usuarios generar reportes, constancias, actas o estadísticas relacionadas con postgrado.
- Pseudocódigo:

Flujo Principal:

1. El usuario invoca al caso de uso generar reportes.
2. El usuario selecciona el documento a generar.
3. El sistema extrae los datos necesarios de la base de datos y construye el documento.
4. Finaliza el caso de uso.

Flujo Alternativo:

1. El usuario puede salir o cancelar la creación del documento.

En la figura 3.5 se muestra el diagrama de caso de uso “Generar Reportes”.

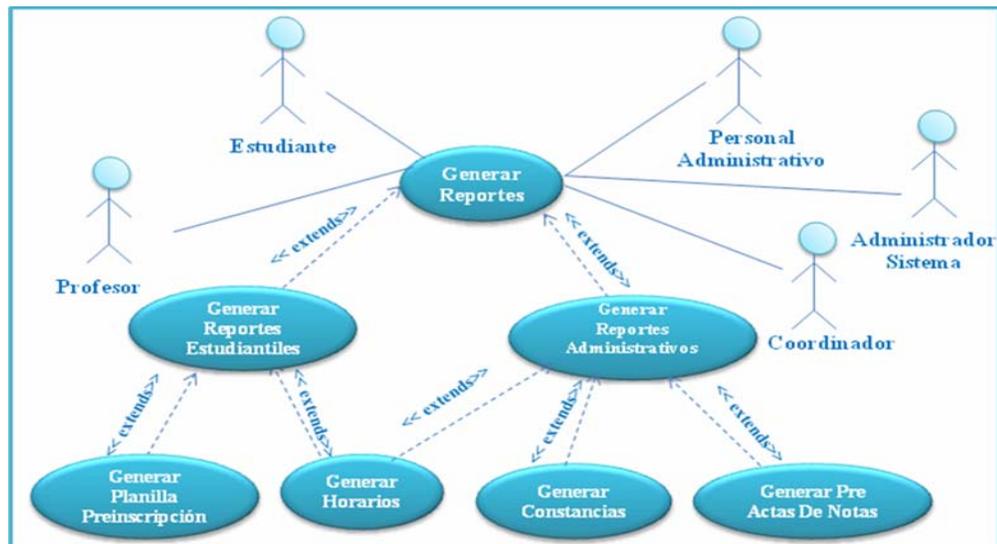


Figura 3.5 Diagrama Detallado Generar Reportes. Fuente: Autores. 2009

3.5.5 Diagrama Detallado del Caso de uso “Generar Archivos OPSU”.

En la figura 3.6 se muestra el diagrama de caso de uso “Generar Archivos OPSU”.

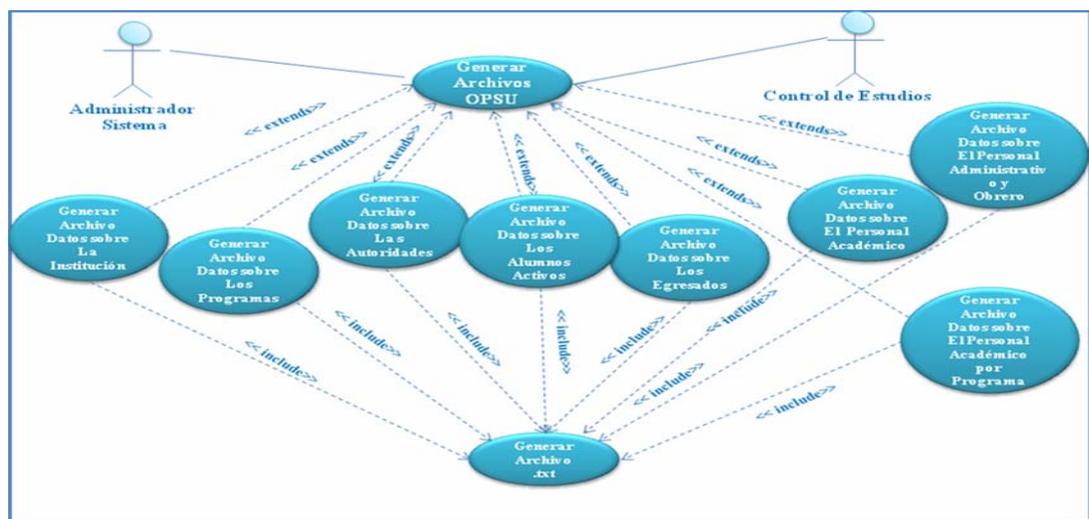


Figura 3.6. Diagrama Generar Archivos OPSU. Fuente: Autores. 2009

- **Actores:** Administrador sistema y Control de Estudio.

- **Descripción:** Permite a los usuarios generar los archivos estadísticos del Postgrado cifrados con formatos de la OPSU.
- **Pseudocódigo:**

Flujo Principal:

- El usuario invoca al caso de uso generar archivos OPSU.
- El usuario selecciona el archivo a generar.
- El sistema extrae los datos necesarios de la base de datos y construye el documento en formato documento.txt y cifrado según parámetros exigidos por este ente.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar la creación del documento.

3.5.6 Diagrama Detallado del Caso de uso “Cargar Notas”

- **Actores:** Profesor.
- **Descripción:** Permite al personal administrativo ingresar datos sobre la notas de las asignaturas correspondientes a cada estudiante además de ingresar en la base de datos la información sobre los programas de postgrado.
- **Pseudocódigo:**

Flujo Principal:

- El usuario invoca al caso de uso cargar datos de programas de postgrado y notas de estudiantes.
- El sistema valida datos según sea necesario.
- El sistema guarda los datos en la BD.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar el ingreso de datos a la base de datos.

En la figura 3.7 se muestra el diagrama de caso de uso “Cargar Datos de Programas de Postgrado, Notas”.

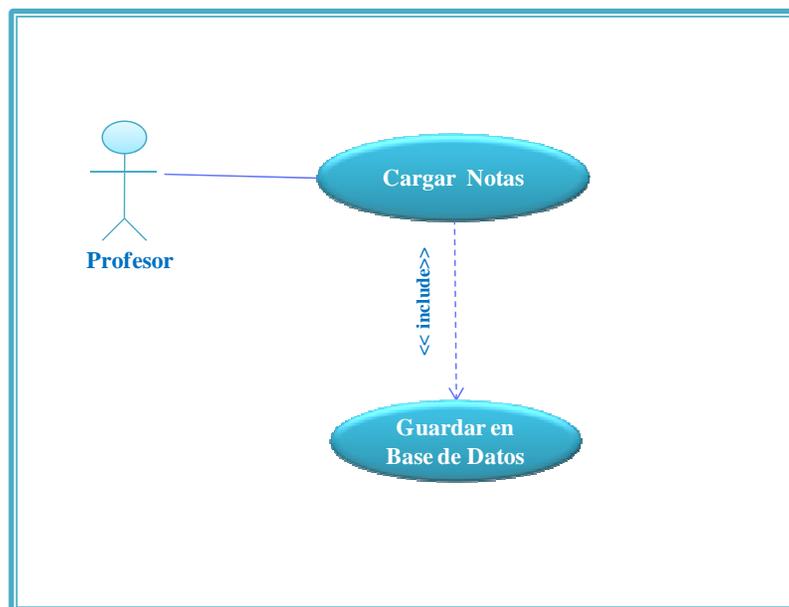


Figura 3.7 Diagrama Detallado Cargar Notas. Fuente: Autores. 2009

3.5.7 Diagrama Detallado del Caso de uso “Procesar Inscripción”

En la figura 3.8 se muestra el diagrama de caso de uso “Inscripción”

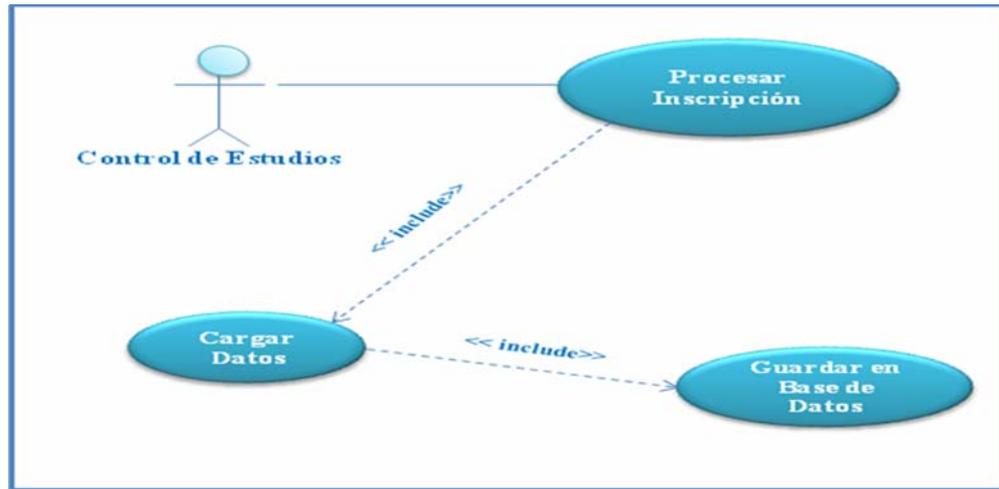


Figura 3.8 Diagrama Detallado Inscripción. Fuente: Autores. 2009

- **Actores:** control de Estudio.
- **Descripción:** Permite formalizar la inscripción de los estuantes seleccionados por la comisión Coordinadores del área de Postgrado.
- **Pseudocódigo:**

Flujo Principal:

- El usuario invoca al caso de uso Inscripción.
- Carga los datos principales del estudiante.
- El sistema extrae de la base de datos la información ya almacenada del estudiante y la muestra por pantalla.
- Guarda los datos finales.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar la inscripción al sistema.

3.5.8 Diagrama Detallado del Caso de uso “Consultar Notas, Horarios, Datos personales”.

- Actores: Personal Administrativo, estudiantes, profesores, coordinadores y administrador de sistema.
- Descripción: Permite al realizar consultas a la base de datos sobre los puntos mencionados.
- Pseudocódigo:

Flujo Principal:

- El usuario invoca al caso de uso consultar notas, horarios y datos personales.
- El sistema extrae de la base de datos la información solicitada y la muestra por pantalla.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar la consulta a la base de datos.

En la figura 3.9 se muestra el diagrama de caso de uso “Consultar Notas, Datos Personales, Horarios”.

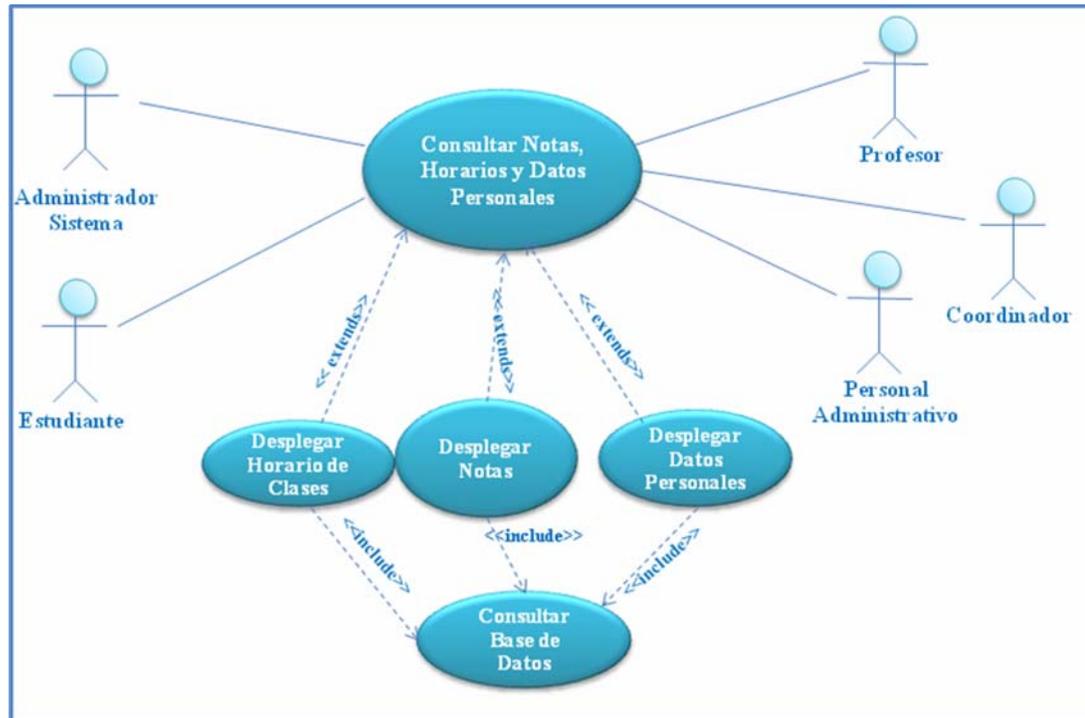


Figura 3.9 Diagrama Detallado Consultar Notas, Horarios, Datos Personales.

3.5.9 Diagrama Detallado del Caso de uso “Administrar datos”

- Actores: Personal Administrativo, coordinadores, Control de Estudio administrador de sistema.
- Descripción: Permite ingresar, modificar y eliminar los datos del personal ya mencionado.
- Pseudocódigo:

Flujo Principal:

- El usuario invoca al caso de uso administrar datos estudiantes, profesores y personal administrativo.

- En caso de consulta el sistema extrae de la base de datos la información solicitada y la muestra por pantalla.
- En caso de modificación o ingreso de datos el sistema valida los mismos antes de ingresarlos a la BD.
- En caso de eliminar datos el sistema advierte sobre los riesgos de esta operación y procede a borrar los datos de la BD.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar la Administración de Datos en la base de datos del personal referido.

En la figura 3.10 se podemos observar el diagrama de caso de uso “Administrar datos Estudiantes, Profesores y Personal Administrativo”.

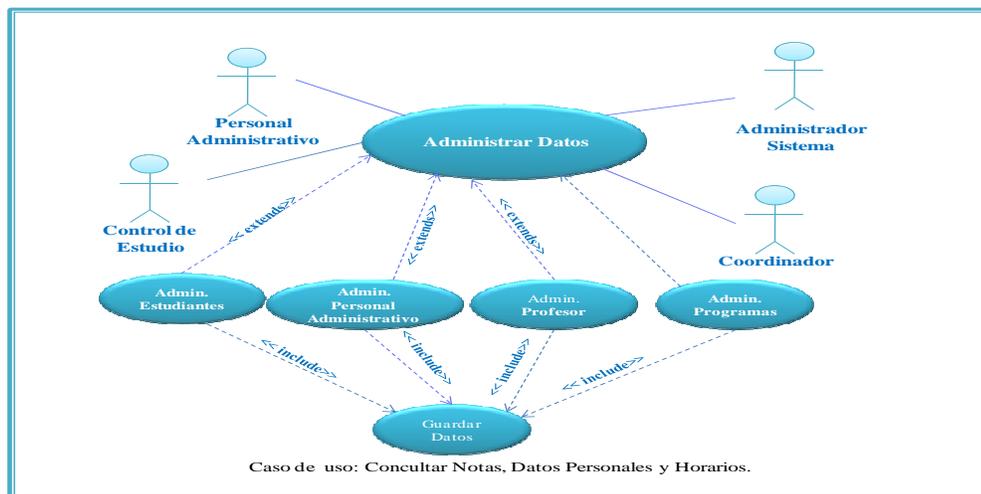


Figura 3.10 Diagrama Detallado Administrar Datos Estudiantes, Profesores, Personal Administrativo. Fuente: Autores. 2009

3.5.10 Diagrama Detallado del Caso de uso “Configurar Sistema”

- **Actores:** Administrador de sistema.
- **Descripción:** Permite ajustar los parámetros de funcionamiento del sistema.
- **Pseudocódigo:**

Flujo Principal:

- El administrador invoca al caso de uso configurar sistema.
- El administrador selecciona el módulo a configurar.
- El administrador establece los nuevos parámetros de funcionamiento del sistema.
- Finaliza el caso de uso.

Flujo Alternativo:

- El administrador puede salir o cancelar la configuración de sistema.

En la figura 3.11 se podemos observar el diagrama de caso de uso “Configurar Sistema”.

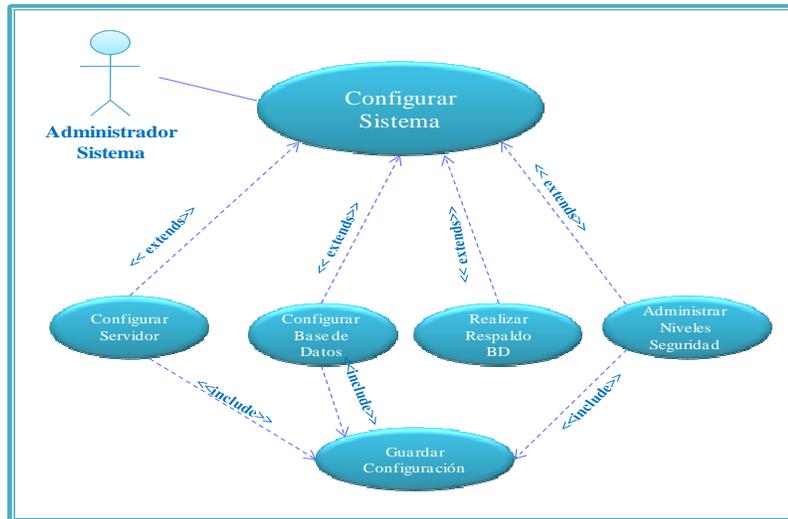


Figura 3.11 Diagrama Detallado Configurar Sistema. Fuente: Autores. 2009

3.6 Requisitos no funcionales

Los requisitos no funcionales son aquellos necesarios para la ejecución de algún caso de uso específico o de varios de ellos pero no compromete el funcionamiento general del sistema. Alguno de estos requisitos no funcionales se lista a continuación.

- El sistema funcionará en mejor medida de acuerdo con la capacidad del servidor de manejar una base de datos extensa y muchas conexiones entrantes al mismo tiempo.
- Se necesita contar con los niveles de seguridad de los usuarios para direccionarlos a cada interfaz relacionada con sus privilegios dentro del sistema.

- El código fuente del programa administrativo debe ser lo más sencillo posible para un fácil entendimiento y un mantenimiento rápido.

3.7 Modelo de análisis

El análisis permite expresar la estructura del sistema, centrándose tanto en su arquitectura como en sus requisitos, de tal forma que éste pueda ser comprendido y refinado de la manera más correcta posible, pues éste modelo proporciona las herramientas necesarias para la ejecución de las etapas posteriores (diseño, construcción).

3.7.1 Diagramas de Clases de Análisis

Se utilizan para describir un caso de uso en su flujo de ejecución normal, además de sus objetos relacionados utilizando herramientas como los estereotipos básicos de Interfaz, de Control y de Identidad.

3.7.1.1 Clases de Interfaz

Una interfaz es una colección de operaciones que especifican un servicio de una determinada clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento, puede mostrar el comportamiento completo o sólo una parte del mismo. Una interfaz describe un conjunto de especificaciones de

operaciones (o sea su signatura) pero nunca su implementación. En la tabla 3.4 se muestra las clases de interfaz.

Tabla 3.4 Descripción de las clases de Interfaz (1/2).

Clase	Definición
:Autenticación	Permite a los usuarios tener acceso al sistema una vez que el sistema comprueba su nombre y clave de acceso
:Ingresar datos Programas de Postgrado, Profesores, Personal Administrativo.	El usuario Personal Administrativo ingresa datos referentes a los programas de postgrado, profesores y personal administrativo. El administrador de sistema puede ingresar datos básicos del personal administrativo cuando se instala el software.
:Preinscripción	Esta clase le permite a un aspirante realizar la preinscripción en el postgrado de su preferencia con dicha función activada.
:Inscripción	Permite al personal administrativo completar el proceso de inscripción de un estudiante en un programa de postgrado, luego de ser aceptado.
:Consultar Horarios	Esta clase permite al usuario navegar a través del horario de clases de un programa de postgrado.

Tabla 3.4 Descripción de las clases de Interfaz (2/2).

Clase	Definición
:Modificar datos en la BD	Permite al personal administrativo actualizar datos de cualquier actor incluyendo programas.
:Generar Reportes	Facilita a los usuarios generar archivos de texto como constancias, actas de notas, reportes estadísticos, archivos OPSU, etc.
:Eliminar Datos	Esta clase permite el borrado de datos cuando estos son erróneos.
:Configuración	Permite la modificación de funcionamiento del sistema por parte del administrador del sistema.

Fuente: Autores. 2009

3.7.1.2 Clases de Control

Su objetivo es representar la coordinación, secuencias, transacciones y control de otros objetos, es decir, el modelo dinámico del sistema. Las clases de control permiten (aunque no siempre) desacoplar las clases de interfaz y entidad. Estas son necesarias cuando el conjunto de eventos que se producen es complejo y el comportamiento dinámico puede cambiar. En este sistema se ha detectado tres clases de control, descritas a continuación en la tabla 3.5.

Tabla 3.5 Descripción de las clases de control.

Clase	Descripción
:Validar Usuario	Permite comprobar si el usuario tiene suficientes privilegios para ingresar y mantenerse en el sistema.
:Validar Datos	Esta clase permite verificar que los datos ingresados por un usuario sean verídicos a la hora de ingresar fechas, edades y otros datos.
:Crear Reportes	Esta clase permite crear los distintos documentos que emite el sistema en su diversidad de formatos.

Fuente: Autores. 2009

3.7.1.3 Clase de Entidad

Las clases de entidad modelan información persistente y en ocasiones pueden tener un comportamiento complejo y métodos relativos a la información que representa. Estas no son específicas de ningún caso de uso y son independientes de los actores. Seguidamente se muestra en la tabla 3.6 las clases entidad relacionadas con el sistema.

Tabla 3.6 Descripción de las clases de Entidad (1/2).

Clase	Descripción
Estudiante	Contiene los datos referentes a un estudiante de postgrado.
Programa de Postgrado	Representa toda la información relativa a un programa de postgrado.
Profesor	Contiene los datos del personal académico de esta área de trabajo.
Personal Administrativo	Guarda la información de este personal de trabajo en todas las jerarquías en que este se esquematiza.

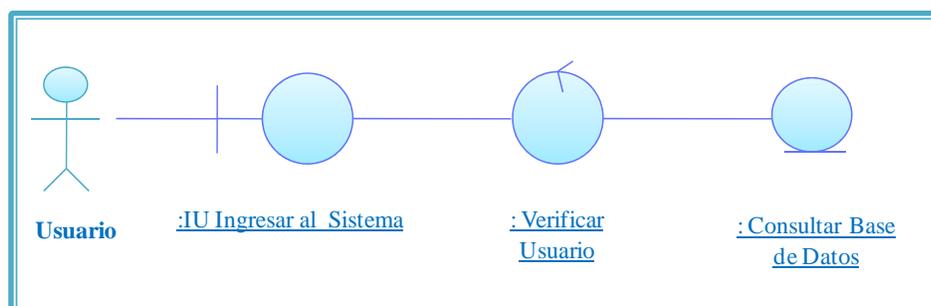
Tabla 3.6 Descripción de las clases de Entidad (2/2).

Clase	Descripción
Coordinadores	Contiene los datos de este personal de coordinación de los programas de postgrado.
Administrador de sistema	Almacena la información del personal de mantenimiento del sistema.
Preinscripciones	Contiene los datos de estudiantes que han realizado una preinscripción en postgrado.

Fuente: Autores. 2009

3.7.1.4 Diagrama de Análisis: Autenticación

Este caso de uso es utilizado por todos los usuarios sin importar su nivel de acceso, es por tanto de interés detallar su funcionamiento con los elementos UML descritos anteriormente. En la figura 3.12 se muestra el diagrama de análisis de este caso de uso.

**Figura 3.12 Diagrama de Análisis del caso de uso Autenticar. Fuente: Autores 2009**

3.7.1.5 Diagrama de análisis: Ingresar datos Programas Postgrado, Profesores, Personal Administrativo

En la figura 3.13 se puede observar el diagrama de análisis del caso de uso descrito, donde el personal administrativo y control de estudio ingresa al sistema datos sobre estos actores y programas de postgrado. El actor administrador de sistema puede ingresar datos de personal administrativo al instalar en programa por primera vez, esto es así, debido a que debe darles el nivel de usuario correspondiente, los demás usuarios son ingresados por el personal administrativo y al guardarlos en la base de datos con el cargo que poseen en postgrado el software le asigna su nivel de usuario automáticamente.

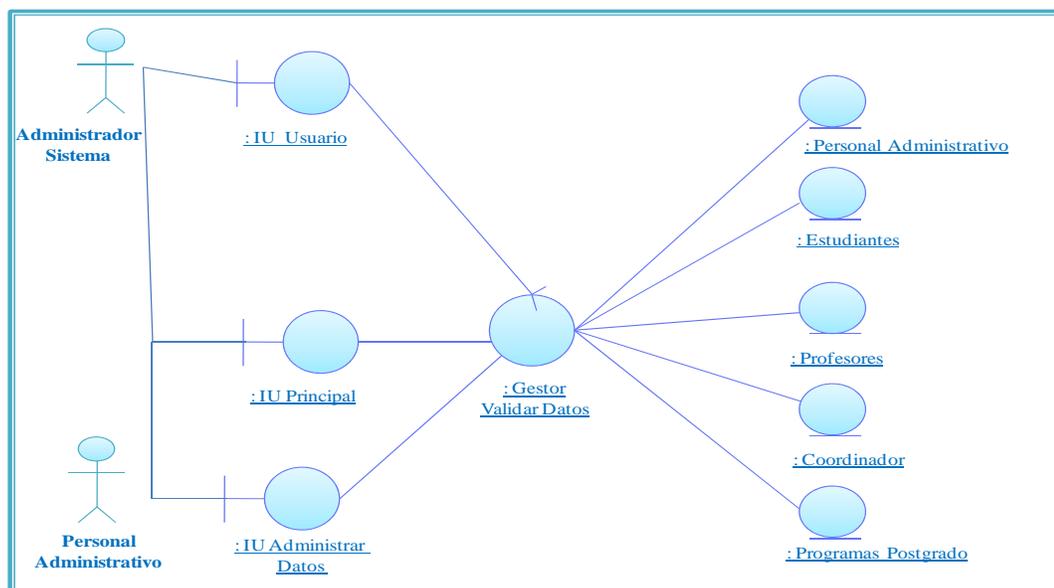


Figura 3.13 Diagrama de Análisis del caso de uso Ingresar Datos. Fuente: Autores. 2009

En la figura 3.13 se detalla el diagrama de colaboración del diagrama de análisis “Ingresar Datos”.

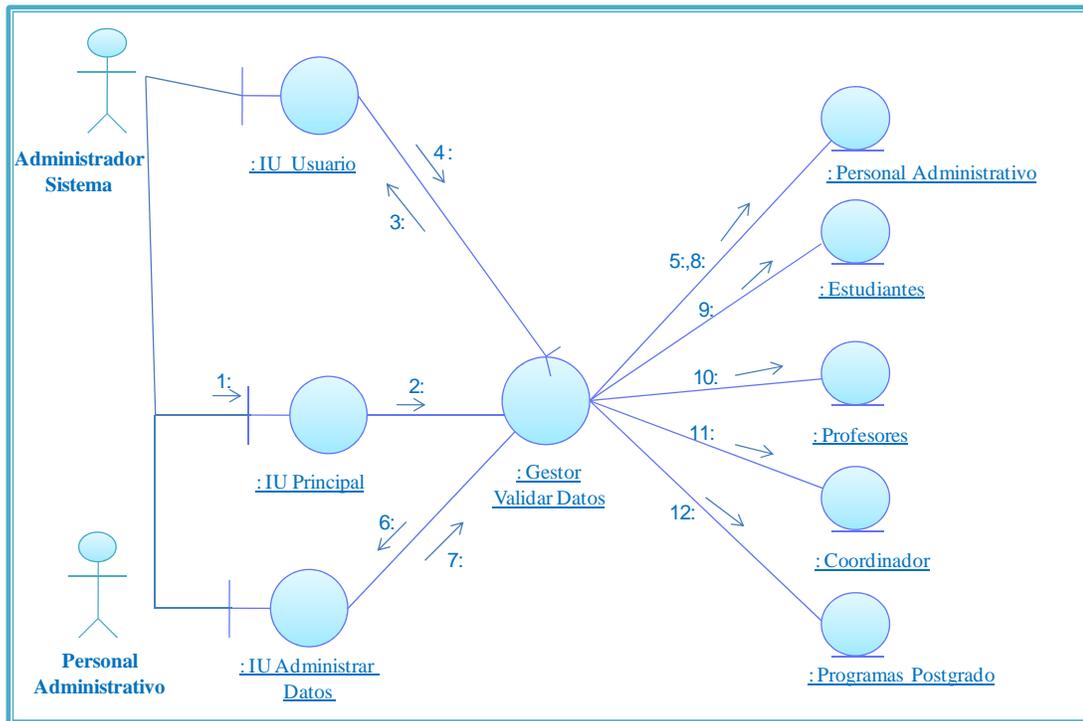


Figura 3.14 Diagrama de colaboración: Ingresar Datos. Fuente: Autores. 2009

Leyenda

- **1: Solicitar Ingresar Dato:** Administrador de sistema accede a la interfaz.
- **2: Valida la activación:** Se activa el gestor validar datos.
- **3: Activa la interfaz:** Capturar Usuario.
- **4: Procesa y Valida datos.** Personal registrado.
- **5: Guardar Datos:** Almacena datos de cualquier actor o programa de postgrado.

- **6: Activa la interfaz:** Capturar Datos
- **7: Procesa y Valida datos.** Personal registrado.
- **8: Almacenar Datos:** Se guardan en la BD los datos de personal administrativo.
- **9: Almacenar Datos:** Se guardan en la BD los datos de estudiantes.
- **10: Almacenar Datos:** Se guardan en la BD los datos de profesores.
- **11: Almacenar Datos:** Se guardan en la BD los datos de coordinadores.
- **12 Almacenar Datos:** Se guardan en la BD los datos de programas de postgrado.

3.7.1.6 Diagrama de análisis: Preinscripción Nuevos Alumnos

Esta operación se realiza por aspirantes a estudiante de los Programas de Postgrado donde deben ingresar sus datos personales y cumplir con otros requisitos necesarios para ejecutar este caso de uso. En la figura 3.15 se muestra este diagrama.

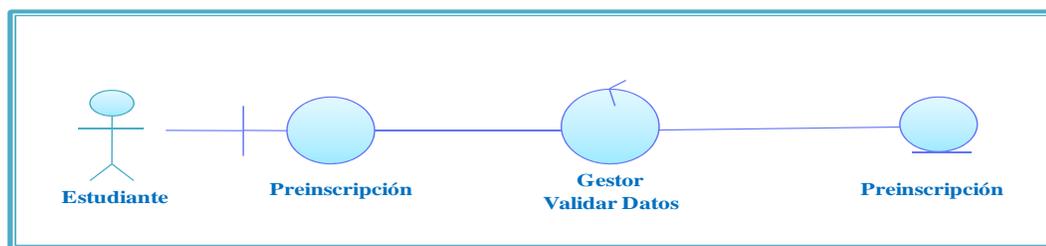


Figura 3.15 Diagrama de análisis: Preinscripción. Fuente: Autores. 2009

En la figura 3.16 se muestra el diagrama de colaboración de este caso de uso.

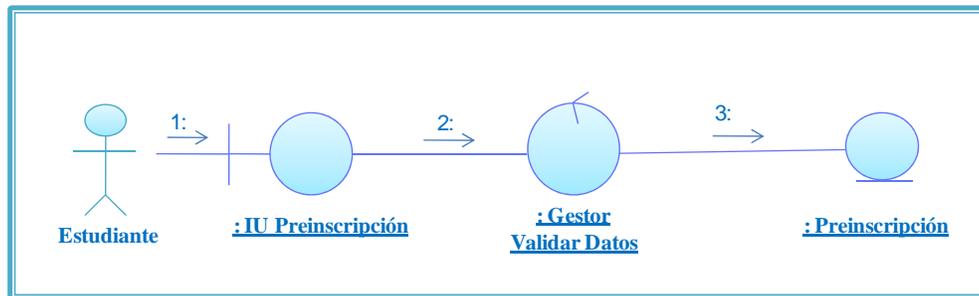


Figura 3.16. Diagrama de colaboración: Preinscripción. Fuente: Autores. 2009

Leyenda:

- **1: Interfaz Preinscripción:** Estudiante accede a la interfaz.
- **2: Guardar Datos:** Se activa el gestor validar datos.
- **3: Almacenar Datos:** Se guardan en la BD los datos de la preinscripción.

3.7.1.7 Diagrama de análisis: Generar Reportes

Permite a los distintos usuarios obtener archivos como constancias, actas, archivos OPSU, etc. En la figura 3.17 se despliega dicho caso de uso.

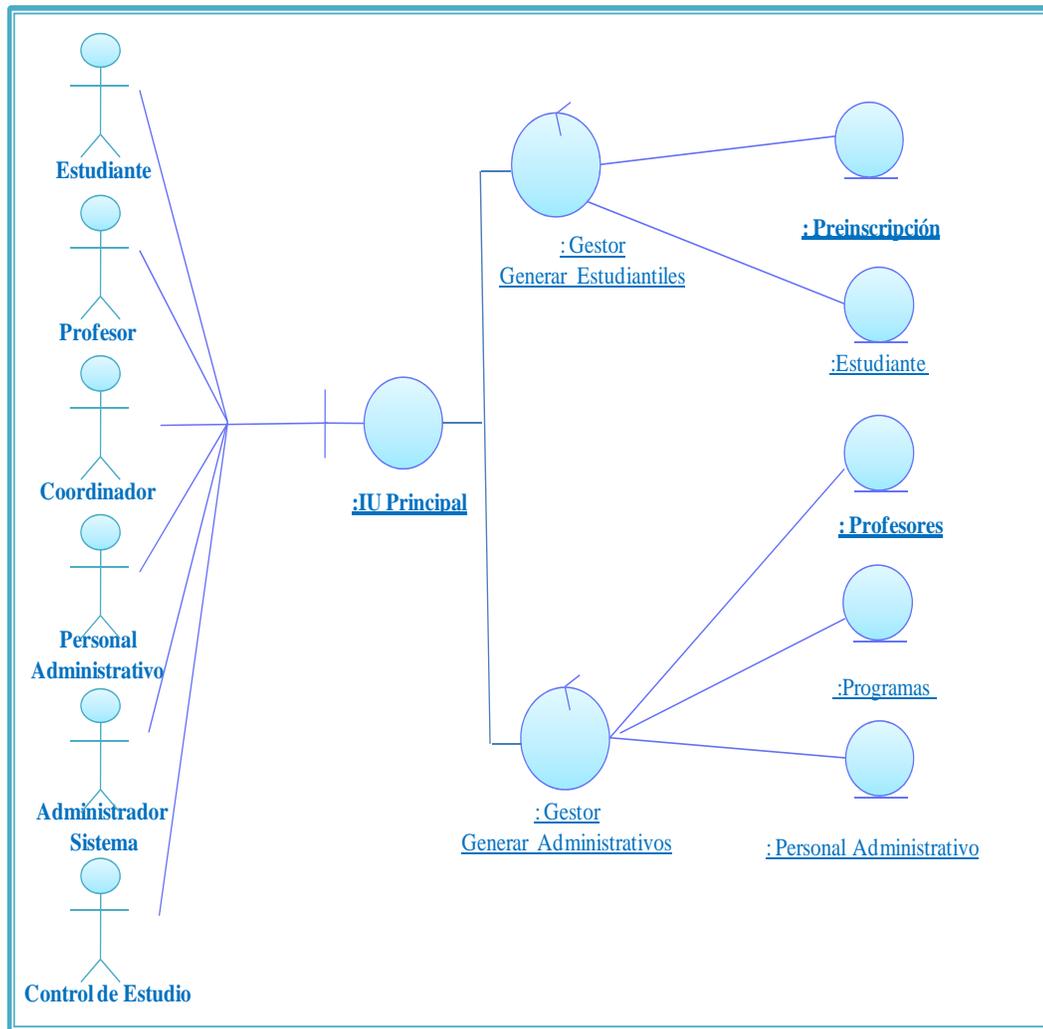


Figura 3.17. Diagrama de análisis: Generar Reportes. Fuente: Autores. 2009

A continuación se muestra en la figura 3.17 el diagrama de colaboración asociado con este caso de uso.

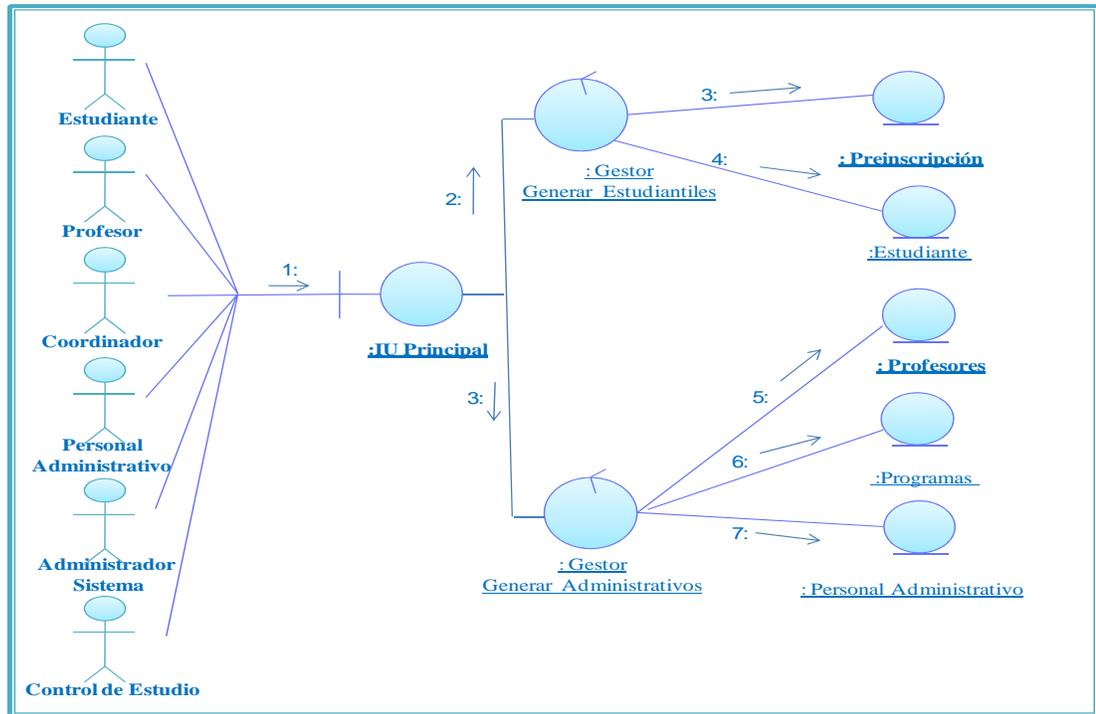


Figura 3.18. Diagrama de colaboración: Generar Reportes. Fuente: Autores. 2009

Leyenda:

- **1: Solicita Generar Reporte:** Estudiante accede a la interfaz.
- **2: Gestor Generar Reporte:** Se activa el gestor generar reporte Estudiantil.
- **3: Gestor Generar Reporte:** Se activa el gestor generar reporte Administrativo.
- **4: Preinscripciones:** Se consulta los datos necesarios para crear el reporte.
- **5: Estudiantes:** Se consulta los datos necesarios para crear el reporte.

- **6: Profesores:** Se consulta los datos necesarios para crear el reporte.
- **7: Programas postgrado:** Se consulta los datos necesarios para crear el reporte.
- **8: Personal Administrativo:** Se consulta los datos necesarios para crear el reporte.

3.7.2 Paquete de Análisis

Los paquetes de análisis proporcionan un medio para organizar y manejar la complejidad en modelos grandes. Agrupa los elementos como casos de usos, clases de análisis, actividades, procesos, estados, entre otros, (los cuales deben estar fuertemente relacionados y depender en lo mínimo de otros paquetes) asociados de manera que puedan ser referidos en su totalidad.

El paquete de análisis se basa en los requisitos funcionales del sistema. Los paquetes de análisis se crearon basándonos en los requisitos funcionales y en el dominio del problema. A continuación se identifican algunos paquetes de análisis a partir de sus casos de uso.

A continuación se presentan los paquetes de análisis específicos del sistema, los cuales están relacionados con los casos de uso ya analizados.

3.7.2.1 Paquete de Análisis Autenticar Usuario

Este paquete está relacionado con los casos de uso consultar permisos o nivel de seguridad y evaluar acceso como se ve en la figura 3.19

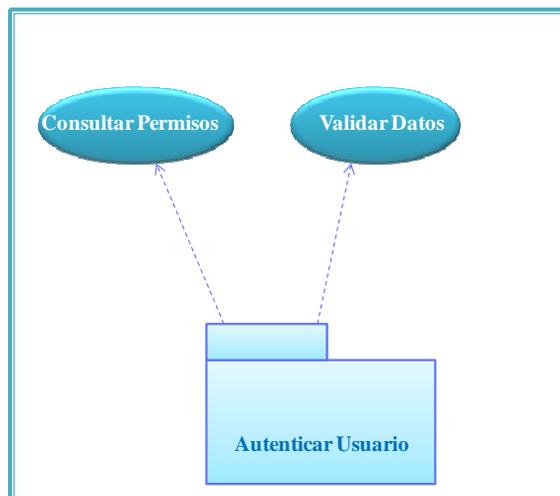


Figura 3.19 Paquete de análisis Autenticar Usuario. Fuente: Autores. 2009

3.7.2.2 Paquete de Análisis Administrar Datos

Este paquete aglomera todos aquellos casos de uso que hacen posible ingresar, modificar o eliminar datos del sistema, independiente del usuario que ejecute la acción. El paquete se muestra en la figura 3.20

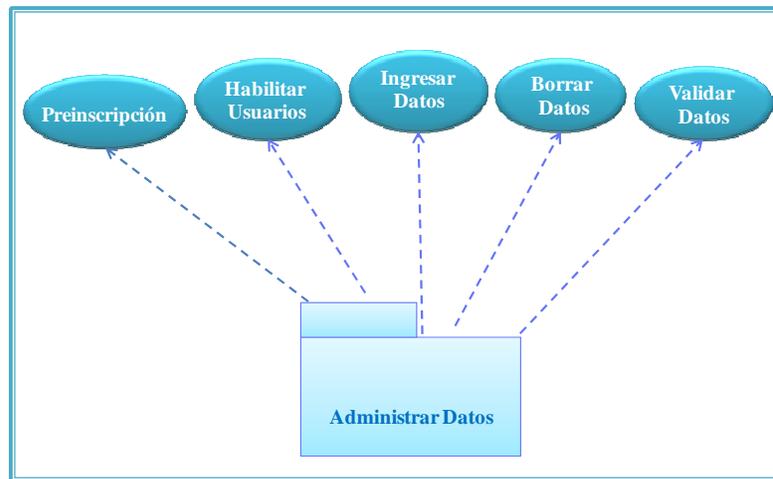


Figura 3.20 Paquete de análisis Administrar Datos. Fuente: Autores. 2009

3.7.2.3 Paquete de Análisis Administrar Sistema

Este paquete aglomera todos aquellos casos de uso que hacen posible ingresar datos al sistema independiente del usuario. Esto se muestra en la figura 3.21

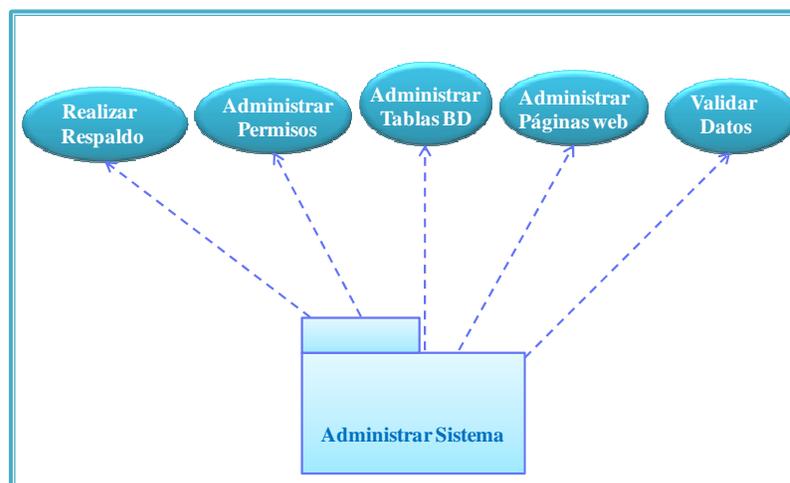


Figura 3.21 Paquete de análisis Administrar Sistema. Fuente: Autores. 2009

3.8 Identificación de los subsistemas

El sistema fue desarrollado bajo la plataforma de Windows XP SP2, el lenguaje de interpretado PHP 5.0.2 utilizando como herramienta de programación Dreamweaver 8.0 y el manejador de bases de datos Oracle 8i, al ser una aplicación Web utiliza un servidor HTML que en este caso es el programa Apache 2.0. Esta combinación de software limitará su aplicación en ambientes Linux, a menos que se use el manejador Oracle 10g. El programa se puede llevar de Oracle 8i a Oracle 10g, no así en caso contrario, por esta razón prevaleció la versión más antigua. En la figura 3.22 se muestra el diagrama de capas del sistema.

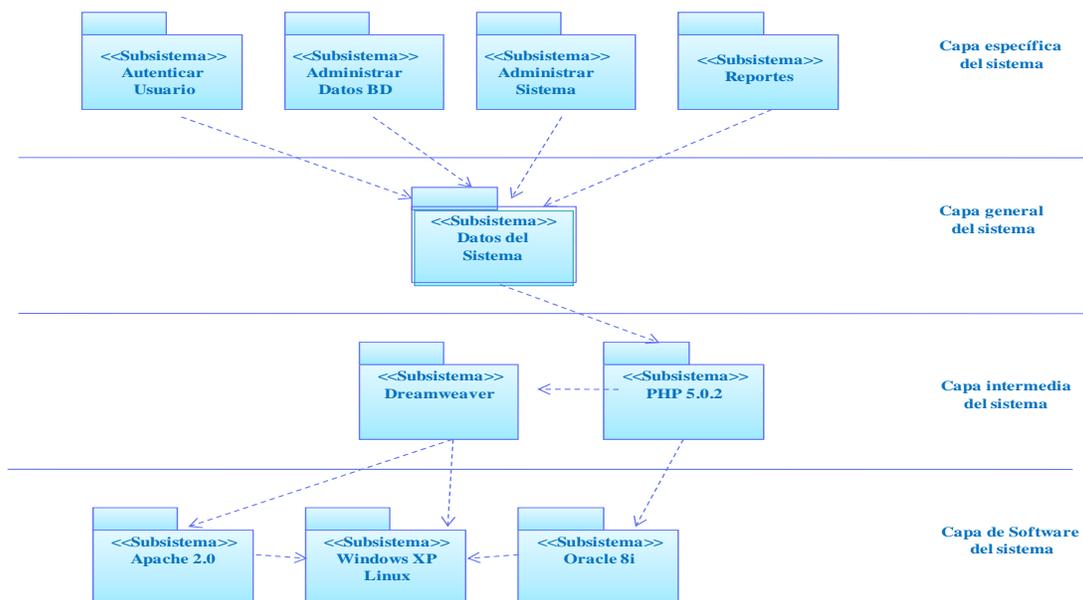


Figura 3.22 Diagrama de Capas. Fuente: Autores. 2009

3.9 Evaluación de la fase de inicio

Es esta etapa se desarrollaron como puntos principales los casos de uso, análisis y diagramas asociados dando una primera visión general del sistema, que será desarrollado con más profundidad en los capítulos subsiguientes. En esta fase se describió las necesidades principales del área de postgrado en el campo de la administración académica de esta importante escuela de formación profesional. Se desarrollaron los casos de uso más relevantes pudiendo ser expandidos y complementados con nuevos casos de uso según la ocasión para darle completitud a la documentación del sistema.

CAPITULO IV

FASE DE ELABORACIÓN

4.1 Introducción

La fase de elaboración es la encargada de determinar la solución técnica del proyecto. Así como durante la fase de inicio se determinó el *qué*, ahora es necesario el *cómo*. Al igual que la fase anterior se realizaron los pasos necesarios para el desarrollo de esta etapa de construcción del software, se explotarán los casos de uso descritos en la fase anterior y se determinará si es necesario complementarlos con nuevos casos de uso.

La Figura 4.1 muestra los pasos que se darán en esta fase además de visualizar el progreso alcanzado en el trabajo realizado.

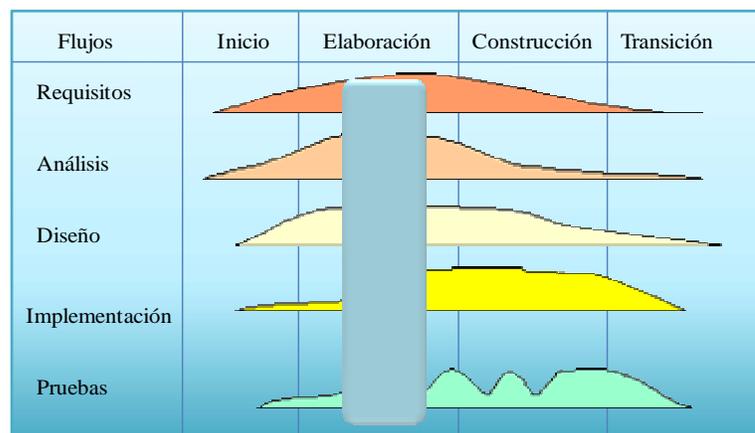


Figura 4.1: Fase de Elaboración (UML). Fuente: Jacobson I. (2000)

La fase de elaboración plantea la arquitectura para el ciclo de vida del sistema. Se construye un modelo de la arquitectura, que se desarrolla en iteraciones sucesivas hasta obtener el producto final, este prototipo debe contener los casos de uso críticos que fueron identificados en la fase de inicio. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable proyecto en general.

4.2 Requisitos del sistema

El análisis de requisitos en la fase de elaboración se lleva a cabo con la finalidad de refinar los requisitos ya obtenidos, contemplando nuevos requisitos que fueron pasados por alto en la fase de inicio.

4.2.1 Requisitos funcionales

El sistema debe permitir consultar las cargas de profesores y generar reportes de las mismas, esta carga proporciona información a cada uno de los profesores activos de las materias que dictaran durante el semestre académico.

4.2.2 Requisitos No Funcionales

En esta fase no se identificaron nuevos requisitos no funcionales debido a que los requerimientos necesarios fueron identificados en la fase de inicio.

4.2.3 Modelo de caso de uso

Se han identificados tres nuevos casos de uso en esta fase, lo que permite conocer más a fondo la estructura del sistema.

4.2.3.1 Identificación de Actores

En esta fase no se identificaron nuevos actores, ya que los actores que interactúan con los nuevos casos de uso ya se encuentran identificados en la fase de inicio. En la fase de inicio describieron los casos de uso del sistema y sus relaciones con actores o con ellos mismos, realizando un estudio más profundo de estos se detectaron nuevos casos de uso complementarios a los ya existentes. En la tabla 4.1 se describen estos casos de uso.

Tabla 4.1 Descripción de los nuevos Casos de uso.

Caso de Uso	Descripción
Restituir - modificar clave de acceso	El caso de uso es realizado por aquel usuario que olvida su clave de acceso al sistema, o desea cambiarla por seguridad.
Generar Listas de asistencias	El personal administrativo y coordinadores realizan listas de asistencias de una asignatura. Este caso de uso es una expansión del caso de uso Generar Reportes.

Fuente: Autores. 2009

En la figura 4.2 se muestra los nuevos casos de uso resaltantes en esta etapa del diseño del sistema.

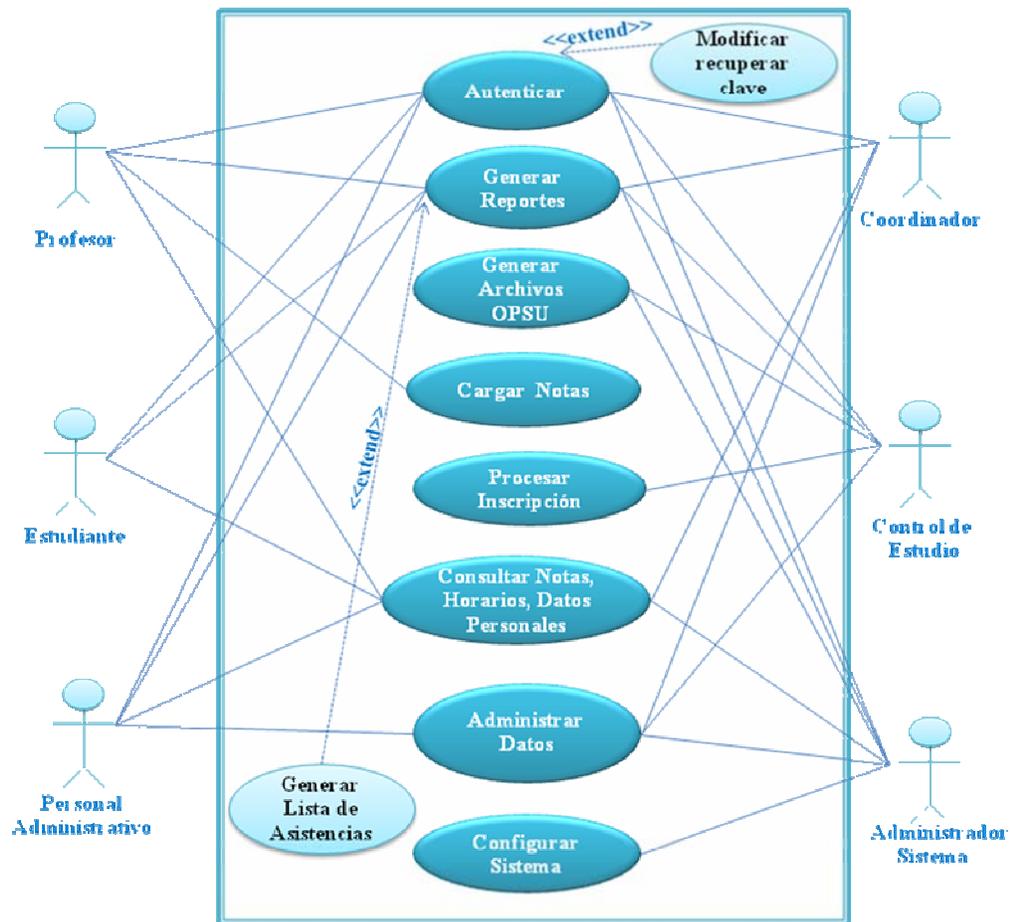


Figura 4.2. Diagrama de Casos de Uso del Sistema SISGAP. Fuente: Autores. 2009

4.2.3.2 Diagrama detallado del caso de uso “Modificar recuperar clave de acceso”

- **Actores:** Estudiantes, profesores, personal administrativo, coordinadores, Control de Estudio y administrador de sistema.

- **Descripción:** Permite al usuario modificar o recuperar la clave de acceso al sistema.
- **Pseudocódigo:**

Flujo Principal:

- El usuario invoca al caso de uso entrar al sistema.
- El usuario invoca al caso de uso modificar-recuperar su clave de acceso.
- El usuario ingresa respuestas a cuestionario.
- El sistema genera nueva clave y la envía al correo electrónico del usuario.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar el ingreso al sistema.

En la figura 4.3 se detalla el diagrama de este caso de uso.

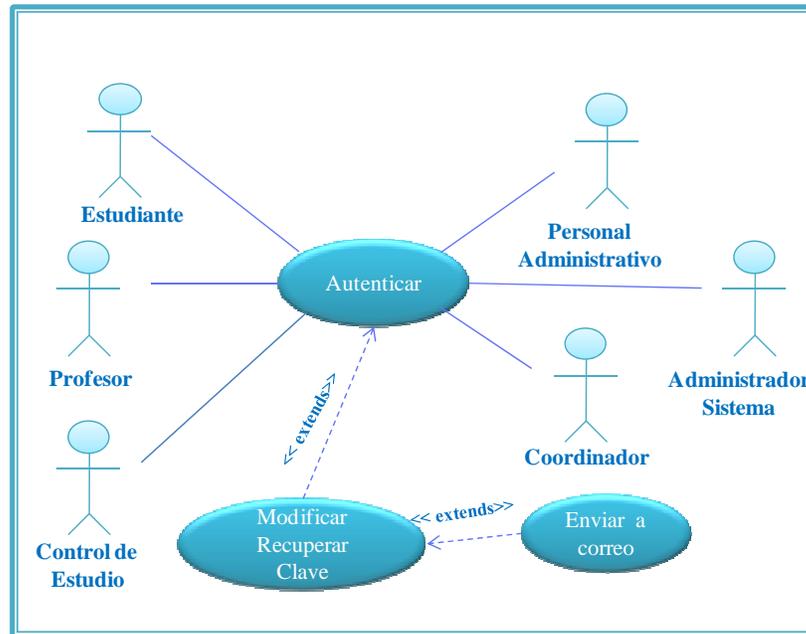


Figura 4.3. Diagrama de Caso de Uso Modificar-Recuperar clave. Fuente: Autores. 2009

4.2.3.3 Diagrama Detallado del Caso de uso “Generar Lista de Asistencias”

- **Actores:** Personal Administrativo, Profesores, Coordinador, Control de Estudio y Administrador de Sistema.
- **Descripción:** Les permite a los distintos usuarios generar lista de asistencias de las distintas secciones de postgrado.
- **Pseudocódigo:**

Flujo Principal:

- El usuario invoca al caso de uso generar lista de asistencias.
- El usuario selecciona la sección.

- El sistema extrae los datos necesarios de la base de datos y construye el documento.
- Finaliza el caso de uso.

Flujo Alternativo:

- El usuario puede salir o cancelar la creación del documento.

En la figura 4.4 se muestra el diagrama de caso de uso “Generar Listas de Asistencias”.

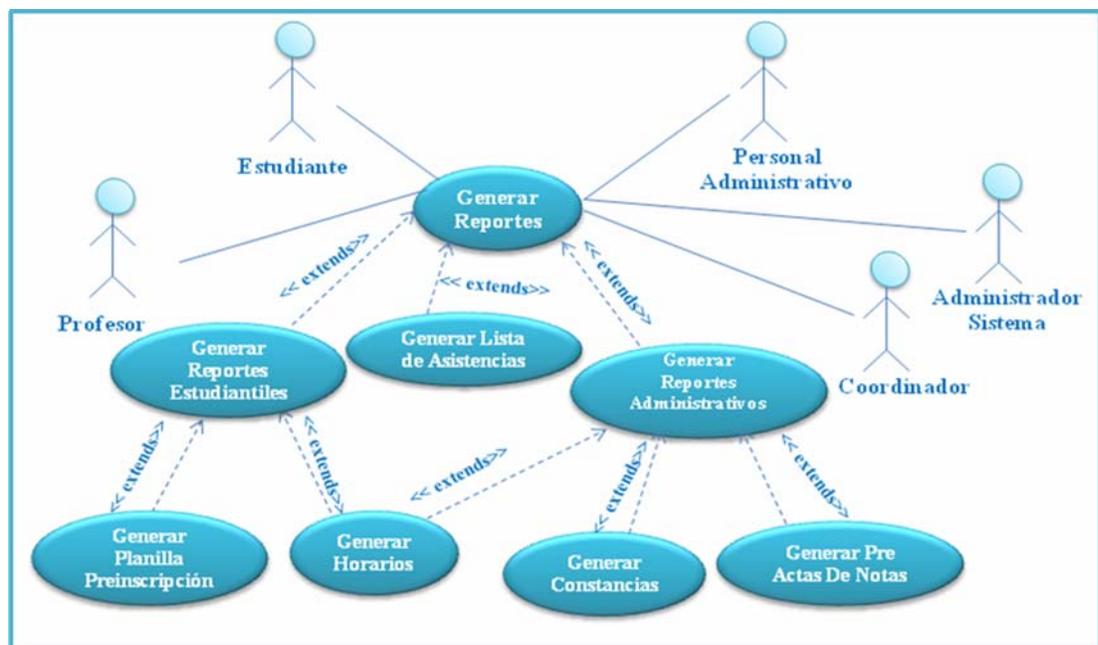


Figura 4.4. Diagrama Detallado Generar Listas de Asistencias. Fuente: Autores. 2009

4.2.4 Prototipo de Interfaz de Usuario

La interfaz de usuario es la ventana con la que interactúa el usuario en el sistema, y es a través de ella donde se realiza la comunicación o intercambio de información entre ambas partes.

4.2.4.1 Prototipo de Interfaz Ingresar al Sistema

En esta interfaz permite a los usuarios ingresar sus números de cédula y claves para tener acceso a los servicios del sistema SISGAP. Cada usuario registrado posee un nivel de seguridad que le permite disfrutar de distintos servicios en ese dicho nivel. A continuación se muestra en la figura 4.5 el prototipo de la interfaz ingresar al sistema.



The image shows a web interface prototype for the SISGAP system at the University of Oriente. The header features the SISGAP logo (Sistema Gerencial del Área de Postgrado) on the left, the University of Oriente name and logo in the center, and a small photo of people on the right. The main content area has a blue gradient background on the left with the text 'PostGrado UDO'. On the right, there is a circular logo of the University of Oriente. Below the logo is a blue box titled 'Inicio de Sesión' containing two input fields: 'Cedula:' and 'Contraseña:'. A button labeled 'Abrir Sesión' is positioned below the fields. Underneath the button is the text 'Olvidé mi contraseña'. At the bottom of the page, a small note reads: 'Primera vez que inicia sesión ingrese 4 últimos dígitos de cédula como contraseña'.

Figura 4.5. Prototipo de interfaz de usuario “Ingresar al Sistema SISGAP”.

4.2.4.2 Prototipo Interfaz de Servicios

Representa la estructura de las interfaces en forma genérica identificando las zonas de actividad comunes para las interfaces de servicios. En la siguiente figura se muestra la estructura de las páginas Web del sistema SISGAP.

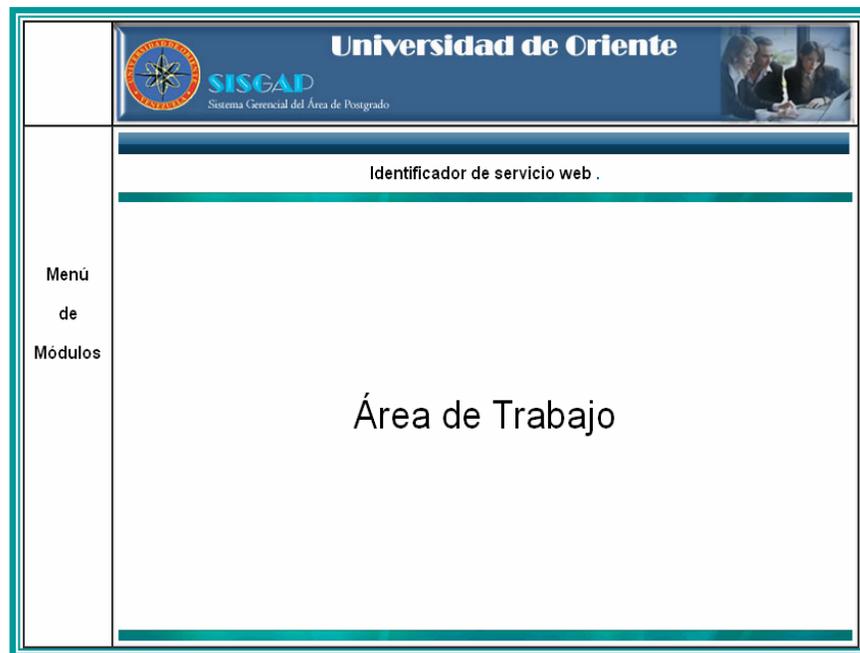


Figura 4.6 Estructura de las páginas Web de la aplicación SISGAP. Fuente: Autores. 2009

4.2.5 Modelo de Hipertextos

El modelo de hipertexto pretende describir como se organizan las interfaces de usuario en aplicación por cada tarea, operación o caso de uso que pretende realizar el usuario en el sistema. Se debe tomar en cuenta que las operaciones de ingresar datos, cualquiera que sea su tipo, la operación de modificar datos y la operación de eliminar datos se realiza en un esquema de ordenamiento de interfaces de usuario y archivos

Web muy parecidos para cada una de estas operaciones. A continuación se mostrarán los esquemas de ordenamiento de interfaces de usuario y archivos Web para cada operación en el sistema.

4.2.5.1 Identificación de Sites Views

➤ Site View Ingresar Datos

En la tabla 4.2 se despliega la descripción de esquema de ordenamiento de archivos web e interfaces de usuario de esta operación.

Tabla 4.2. Site View Ingresar Datos

Nombre del Site View	Ingresar Datos
Descripción	Al ingresar datos en el sistema el usuario realiza la operación a través de formularios que envían los datos a otro archivo web que realiza la tarea y se redirecciona a otra interfaz de usuario.
Grupo de Usuarios	Personal Administrativo, Coordinador, Profesores y Administrador de sistema.
Casos de Uso Involucrados	Cargar datos Programas de Postgrado, Cargar Notas de Programas, Administrar datos de Estudiantes, Profesores, Administrador de Sistema y Personal Administrativo.

Fuente: Autores. 2009

➤ Site View Consultar Datos

En las operaciones de consulta al sistema las interfaces de consulta son distintas a cada usuario debido a que hay información sensible que algunos usuarios no deben manejar y sólo los usuarios autorizados tienen acceso a ella, sin embargo, la operación de consulta se realiza de la misma forma para cualquier usuario. La tabla 4.3 detalla ese aspecto.

Tabla 4.3. Site View Consultar Datos

Nombre del Site View	Consultar Datos
Descripción	Las operaciones de consulta se realizan a través de interfaces que extraen los datos necesarios de la base de datos.
Grupo de Usuarios	Personal Administrativo, Coordinador, Profesores, Estudiantes y Administrador de sistema.
Casos de Uso Involucrados	Consultar Notas, Horarios y Datos personales

Fuente: Autores. 2009

➤ Site View Eliminar Datos

Tabla 4.4. Site View Eliminar Datos

Nombre del Site View	Eliminar Datos
Descripción	Las operaciones de eliminación de datos se realizan a través de interfaces que atrapan los datos necesarios para identificar el registro a eliminar y luego borrarlo de la base de datos.
Grupo de Usuarios	Personal Administrativo, Coordinador y Administrador de sistema.
Casos de Uso Involucrados	Administrar datos Estudiantes, profesores y personal administrativo.

Fuente: Autores

2009

➤ Site View Configurar Sistema

En estas operaciones se especifican el comportamiento del sistema. Las interfaces Web permiten atrapar las configuraciones deseadas y son registradas en archivos de texto, por cuanto en estas configuraciones la(s) base(s) de dato(s) no escapa a estas configuraciones.

Tabla 4.5. Site View Configurar sistema

Nombre del Site View	Configurar sistema
Descripción	Estas operaciones modifican el comportamiento del servidor Web, el servidor de base de datos y los niveles de seguridad de los usuarios.
Grupo de Usuarios	Administrador de sistema.
Casos de Uso Involucrados	Configurar Sistema.

Fuente: Autores.

2009

4.3 Análisis

El análisis de esta fase del proyecto busca expandir aquellos diagramas de UML desarrollados en la fase de inicio que ha sido modificados o expandidos en esta fase con el objetivo de precisar el funcionamiento de alguno de ellos y concretar el desarrollo de los módulos o componentes de software.

4.3.1 Clases de análisis

En esta fase se agregaron algunos nuevos casos de uso, los cuales fueron descritos al principio de la misma, a estos casos de uso se hace necesario desarrollar las distintas clases de análisis para precisar el funcionamiento de los mismos. Tanto

los casos de uso como sus clases de análisis de la fase de inicio que no sufrieron cambios en esta, serán considerados tal cual fueron descritos en esa fase.

4.3.2 Diagrama de análisis del caso de uso Modificar, recuperar clave de acceso

Este caso de uso resulta útil al sistema por cuanto permite al usuario modificar su clave de acceso al sistema o recuperarla en caso de olvidarla. En la figura 4.7 se detalla el diagrama de análisis de este caso de uso.

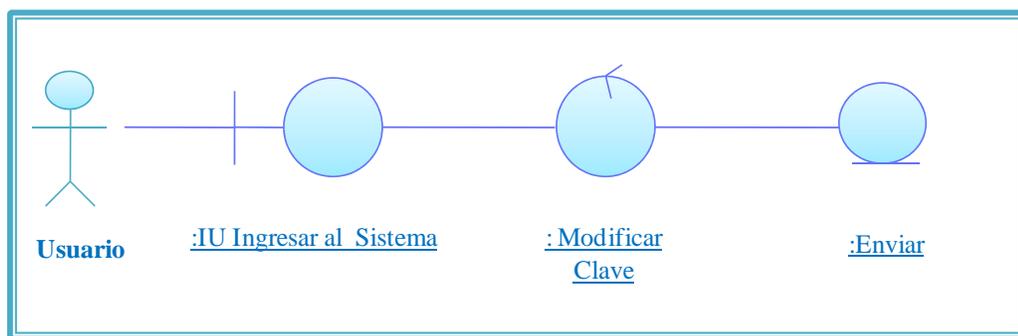


Figura 4.7. Diagrama de análisis del caso de uso Modificar, recuperar clave.

En la figura 4.8 se describe el diagrama de colaboración de este caso de uso.

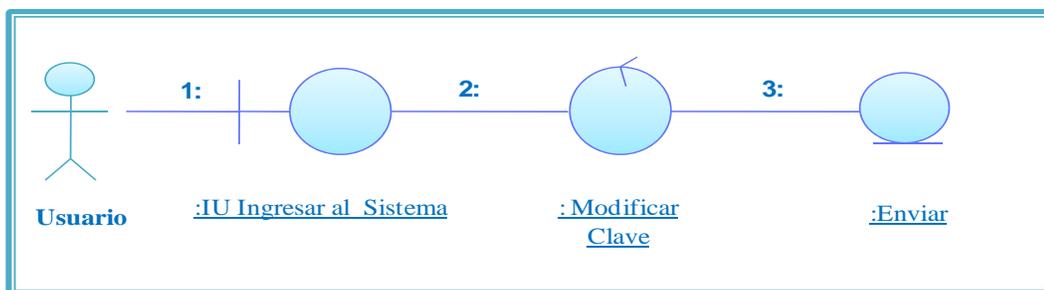


Figura 4.8. Diagrama de Colaboración del caso de uso Modificar, recuperar clave.

Fuente: Autores. 2009

➤ Leyenda:

1: Interfaz Autenticar: Usuario accede a la interfaz Ingresar al Sistema.

2: Modificar Datos: Se realiza la operación modificar datos realizando cuestionario a usuario.

3: Almacenar Datos: Se envía al correo del usuario el resultado de la operación.

4.3.3 Diagrama de análisis del caso de uso Generar lista de asistencias

En este caso de uso se genera las listas de asistencias de las secciones por cohorte y programa para llevar el control de asistencia. En la figura 4.9 se muestra el diagrama de análisis de este caso de uso.

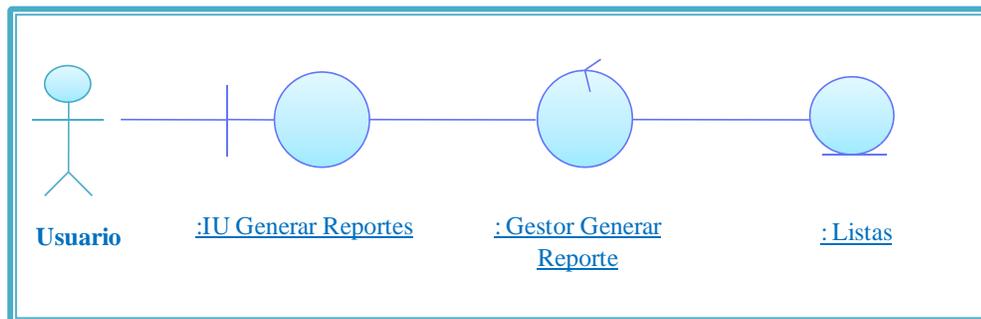


Figura 4.9. Diagrama de análisis del caso de uso Generar lista de asistencias.
Fuente: Autores. 2009

Seguida mente en la figura 4.10 se describe el diagrama de colaboración de este caso de uso.

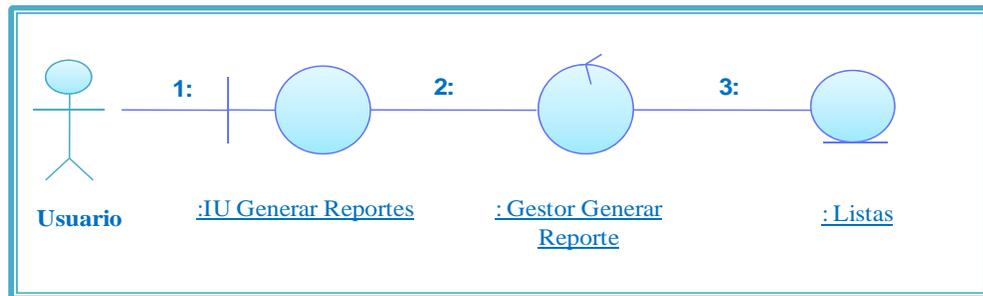


Figura 4.10. Diagrama de Colaboración del caso de uso Generar lista de asistencias.
Fuente: Autores. 2009

➤ Leyenda:

1: Interfaz Generar reporte: Usuario accede a la interfaz Generar reportes.

2: Escoger Programa, Cohorte, sección: El usuario especifica la sección de la cual quiere generar la lista de asistencia.

3: Generar archivo: Se visualiza el resultado de la operación.

4.4 Diseño

En el flujo de trabajo de diseño se modela el sistema y se encuentra su representación para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones. Aquí se toma como entrada el análisis de los requisitos. En el diseño se realizan actividades delicadas y cuidadosas que dan como resultado los componentes que se implementarán en la siguiente fase. Entre ellos tenemos, el diseño de la base de datos, diseño de la arquitectura, diseño de interfaces y Páginas Web.

4.4.1 Base de Datos del Sistema SISGAP

Como se puede observar en los modelos de gestión diseñados en la sesión anterior, se hacía un llamado a la base de datos del sistema, bien sea para ingresar o consultar. La base de datos para este sistema debe recoger las tablas de codificación de datos de la OPSU y con ella modelar las demás tablas de la base de datos para que guarden la data que se origina en los Postgrados de forma compatible con la data de las tablas de la OPSU ya mencionadas.

4.4.1.1 Identificación de las tablas de la Base de Datos del Sistema SISGAP

En el diseño de la base de datos del sistema se identificaron las tablas necesarias para el funcionamiento del mismo, listándose a continuación cada una de ellas.

Tablas OPSU: En estas tablas se registran los códigos asociados a los términos que se utilizan en la base de datos, estos códigos son asignados por la OPSU para tener un conjunto de datos comunes entre las entidades educativas y esta oficina de control educativo.

➤ Tabla Países

En ella se registra los nombres de países y los códigos asignados por la OPSU.

CODIGO varchar2(10), PAIS varchar2(50).

➤ Tabla Entidades Federales

En esta tabla se registra los nombres de las entidades federales o estados con sus respectivos códigos.

CODIGO varchar2(10), NOMBRE_ENTIDAD varchar2(50).

➤ Tabla Municipios

En esta tabla se guarda los municipios de cada estado y el código asignado a cada uno de ellos.

CODIGO varchar2(10), NOMBRE_MUNICIPIO varchar2(50).

➤ Tabla Parroquias

Aquí se registran las parroquias de los municipios y sus códigos.

CODIGO varchar2(10), PARROQUIA varchar2(60).

➤ Tabla Profesión

En esta tabla se registran todas las profesiones existentes en el país y sus respectivos códigos.

CODIGO varchar2(10), PROFESION varchar2(150).

➤ Tabla Unidad de Créditos

Esta tabla registra el tipo de unidad de crédito válido en la institución, bien sea crédito, horas, módulos, etc., cada uno de ellos tiene un valor de código que lo identifica

CODIGO number(2,0), UNIDAD_CREDITO_EQUIV varchar2(50).

➤ Tabla Categorías de Profesores

En ella se aloja la distinción que se le da a cada personal académico, a saber, asociado, agregado, contratado, etc., cada distinción posee su código.

CODIGO varchar2(10), CATEGORIA varchar2(50).

Tabla Etnias

Las distintas etnias indígenas del país son registradas en esta tabla con su respectivo código.

CODIGO varchar2(10), ETNIA varchar2(50).

➤ Tabla Tiempo Dedicación

Registro del tiempo que dedica el personal académico a su jornada de trabajo, que se clasifica en completo, medio tiempo, convencional o exclusiva, cada uno de estas clasificaciones tiene su código asignado

CODIGO varchar2(10), TIEMPO_DEDICACION varchar2(50).

➤ Tabla Régimen de Estudio

Los programas de postgrado se clasifican según su régimen de estudio: anual, semestral, trimestral u otro, estas clasificaciones tienen su dual numérico que las identifica ante la OPSU.

CODIGO varchar2(10), REGIMEN_ESTUDIO varchar2(50).

➤ Tabla Salida Intermedia

Algunos programas de postgrado poseen salida intermedia y otros no, al ente regulador de las instituciones educativas le interesa saber esta información.

CODIGO varchar2(10), SALIDA_INTERMEDIA varchar2(50).

➤ Tabla Acreditación

En esta tabla se guarda los dos casos posibles, que un programa esté acreditado o no, esta información tiene su código según el caso.

CODIGO varchar2(10), ACREDITACION varchar2(50).

➤ Tabla Privado de Libertad:

En ella se guarda las dos repuestas que podría dar un estudiante, (si / no) cuando se le pregunte si tiene antecedentes.

CODIGO varchar2(10), PRIVADO_LIBERTAD varchar2(50).

➤ Tabla Tipo de Dependencia

La tabla registra los tipos de dependencias que puede clasificar a una universidad: oficial, privada o mixta.

CODIGO varchar2(10), TIPO_DEPENDENCIA varchar2(50).

➤ Tabla Condición Académico Administrativa

Las universidades pueden tener más de un núcleo, en esos casos uno de ellos será la sede y los otros simplemente núcleos, extensiones o ampliaciones, esos casos tienen sus códigos OPSU.

CODIGO varchar2(10), CONDICION_ACAD_ADMINIST varchar2(50).

➤ Tabla Turno

Los programas de postgrado pueden ser dictados en distintos turnos, esas opciones se guardan en esta tabla.

CODIGO varchar2(10), TURNO_ESTUDIO varchar2(50).

➤ Tabla Modalidad de Estudio

Las opciones serían: presencial, semi presencial, a distancia u otros. Esta tabla guarda esas opciones con sus códigos.

CODIGO varchar2(10), MODALIDAD_ESTUDIO varchar2(50).

➤ Tabla Grado Académico

Grado académico que se le otorga a los cursantes del postgrado

Luego de culminar los estudios.

CODIGO varchar2(10), TITULO_GRADO_ACADEMICO varchar2(60).

➤ Tabla Requerimientos

El requisito final que se exige para otorgar el título al cursante del postgrado: tesis, pasantía, etc.

CODIGO varchar2(10), REQUERIMIENTOS varchar2(60).

➤ Tabla Escala de Medición

La escala de medición de las evaluaciones hechas en el postgrado.

CODIGO varchar2(10), ESCALA_MEDICION varchar2(50).

➤ Tabla Condición de Residencia

El estudiante puede ser residente o transeúnte según CNU.

CODIGO varchar2(10), CONDICION_RESIDENCIA varchar2(50).

➤ Tabla Modalidad de Ingreso

La forma en que el estudiante ingresó al programa de postgrado.

CODIGO varchar2(10), MODALIDAD_IINGRSO varchar2(50).

➤ Tabla Condición Laboral Docentes

CODIGO varchar2(10), CONDICION_LABORAL varchar2(50).

➤ Tabla Condición Laboral Administración

CODIGO varchar2(10), CONDICION_LABORAL varchar2(50).

➤ Tabla Tipo de Personal

El personal administrativo y obrero tienen clasificaciones que se representan en esta tabla.

CODIGO varchar2(10), TIPO_PERSONAL varchar2(50).

➤ Tabla Becado por Institución

Los estudiantes podrían estar o no becados por la institución donde estudian.

CODIGO varchar2(10), BECADO varchar2(50).

- Tabla Estudiante en Trabajo de Grado

CODIGO varchar2(10), EN_TRABAJO_DE_GRADO varchar2(50).

- Tabla Nivel Académico

El nivel académico del personal docente tiene clasificaciones que son alojadas en esta tabla.

CODIGO varchar2(10), NIVEL_ACADEMICO varchar2(50).

- Tabla Mención Honorífica

Las distintas menciones honoríficas que un estudiante puede obtener

CODIGO varchar2(10), Mension_Honorifica varchar2(50).

- Tablas del Área de Postgrado UDO

Estas tablas alojan la información de la institución y del personal que forma parte de esta área de estudio.

➤ Tabla Datos de la Institución

Codigo_Institucion number(10,0)
,Codigo_Nucleo number(10,0)
,Nombre_Institucion varchar2(50)
,Siglas varchar2(10)
,Dependencia varchar2(30)
,Condicion_Acad_Admin varchar2(20)
,Entidad_Federal varchar2(50)
,Municipio varchar2(60)
,Parroquia varchar2(100)
,Gaceta_Oficial number(10,0)
,Comentario varchar2(100)

➤ Tabla programas de Postgrado

Codigo_Programa number(10,0),
Nombre_Programa varchar2(200),
Mencion varchar2(100),
Regimen_Estudio varchar2(100),

Unidad_Credito varchar2(50),
 Total_unid_credit number(10,0),
 Duracion number(5,0),
 Modalidad_Estudio varchar2(50),
 Turno varchar2(50),
 Titulo_Grado_Academico varchar2(100),
 Salida_Intermedia varchar2(20),
 Requerimientos varchar2(100),
 Escala_Medicion_Calificaciones varchar2(10),
 Numero_Gaceta_Oficial number(15,0),
 Arancel number(15,0),
 Acreditacion varchar2(20),
 Nivel_Academico varchar2(20),
 Nucleo_Institucion varchar2(50),
 Area_Conocimiento varchar2(100),
 fecha_creacion date,
 Costo_Preinscripcion number(15,3),
 Costo_Inscripcion number(15,3),
 Costo_Modulo number(15,3),
 Comentario varchar2(300),

➤ Tabla Cohortes

Codigo_Programa number(10,0), Id_Cohorte varchar2(5), Inicio
 date,Fin date.

➤ Tabla Secciones

Codigo_Programa number(10,0), Id_Cohorte varchar2(5), Id_Seccion
varchar2(5).

➤ Tabla Asignaturas

Codigo_Programa number(10,0)
 ,Id_Cohorte varchar2(5)
 ,Id_Seccion varchar2(5)
 ,Codigo_Asignatura number(10,0)
 ,Nombre_Asignatura varchar2(100)
 ,HorasTeoricas number(3,0)
 ,HorasPracticas number(3,0)
 ,Identificacion_Profesor number(10,0)
 ,Creditos number(5,0)
 ,Entrega_Material date
 ,Inicio date
 ,Fin date
 ,Presencial1 date
 ,Presencial2 date
 ,Presencial3 date

➤ Tabla Horarios

Codigo_Asignatura number(10,0)
 ,Dia varchar2(15)
 ,Aula varchar2(50)
 ,Hora_inicio varchar2(10)

,Hora_Fin varchar2(10)

➤ Tabla Líneas de Investigación

Codigo varchar(10) not null, Nombre_Lineainvest varchar2(100),
CedulaCoordinador number(10,0).

➤ Tabla Integrantes Líneas de Investigación

Codigo varchar(10), Cedula_Integrante number(10,0).

➤ Tabla Proyectos líneas de Investigación

Codigo varchar(10), Nombre_Project_Invest
varchar2(100), Cedula_Responsable number(10,0).

➤ Tabla Trabajos de Investigación

CODIGO NUMBER(10), NOMBRE_TRABAJO_INVEST
VARCHAR2(100), NUMERO_ID_RESPONSABLE NUMBER(10),
FASE_DEL_TRABAJO VARCHAR2(150), DESCRIPCION
VARCHAR2(150).

➤ Tabla Grupos de Investigación

Codigo varchar(10), Nombre_Grupo varchar2(100),
Cedula_Responsable number(10,0).

➤ Tabla Integrantes Grupos de Investigación

Codigo varchar(10),Cedula_Integrante number(10,0).

➤ Tabla Autoridades de la Institución

Pais_Nacimiento varchar2(100),
Numero_Pasaporte varchar2(10),
Numero_Cedula number(10,0) ,
Apellidos varchar2(50),
Nombres varchar2(50),
Sexo varchar2(20),
Fecha_Nacimiento date,
Cargo varchar2(100),
Título_Grado_Academico varchar2(100),
Categoria varchar2(100),
Ubicacion varchar2(100),
Fecha_Ingreso date,
Fecha_Egreso date,
Nucleo_Institucion varchar2(50),
Discapacidad varchar2(50).

➤ Tabla Historial Académico de Autoridades de la Institución

Numero_Identificacion number(10,0) not null ,
 Estudio_Realizado varchar2(100),
 Empresa_Institucion varchar2(100),
 Promedio_Notas number(5,3),
 MENSION_HONORIFICA varchar2(20),
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Titulo_Certificado varchar2(100).

➤ Tabla Direcciones de Autoridades de la Institución

Numero_Identificacion number(10,0) not null, Direccion_1
 Varchar2(100), Direccion_2 varchar2(100), Direccion_3 varchar2(100),
 Direccion_4 varchar2(100).

➤ Tabla Teléfonos de Autoridades de la Institución

Numero_Identificacion number(10,0) not null, Telefono_1
 number(15,0), Telefono_2 number(15,0),
 Telefono_3 number(15,0), Telefono_4 number(15,0).

➤ Tabla Correos de Autoridades de la Institución

Numero_Identificacion number(10,0) not null, Correo_1
 varchar2(30), Correo_2 varchar2(30), Correo_3 varchar2(30),
 Correo_4 varchar2(30).

➤ Tabla Datos Extra-laborales de Autoridades de la Institución

Numero_Identificacion number(10,0) not null, Nombre_Empresa
 varchar2(100), Cargo_Empresa varchar2(100).

➤ Tabla Personal Docente

Pais_Nacimiento varchar2(100),
 Numero_Pasaporte varchar2(10),
 Numero_cedula number(10,0) not null ,
 Apellidos varchar2(50),
 Nombres varchar2(50),
 Sexo varchar2(50),
 Fecha_Nacimiento date,
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Categoria varchar2(50),
 Categoria_Inicial varchar2(50),
 Tiempo_Dedicacion varchar2(50),
 Titulo_Grado_Academico varchar2(100),
 Condicion_Laboral varchar2(50),
 Profesion varchar2(100),
 Unidad_Adscripcion varchar2(100),

Nivel_Academico varchar2(100),
 Nucleo_Institucion varchar2(50),
 Discapacidad varchar2(50),
 Estado_Laboral varchar2(50),
 Tiempo_Permission_Laboral number(5,0),

➤ Tabla Historial Académico de Personal Docente

Numero_Identificacion number(10,0) not null ,
 Estudio_Realizado varchar2(100),
 Empresa_Institucion varchar2(100),
 Promedio_Notas number(5,3),
 MENSION_HONORIFICA varchar2(20),
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Titulo_Certificado varchar2(100).

➤ Tabla Direcciones de Personal Docente

Numero_Identificacion number(10,0) not null, Direccion_1
 Varchar2(100), Direccion_2 varchar2(100), Direccion_3 varchar2(100),
 Direccion_4 varchar2(100).

➤ Tabla Teléfonos de Personal Docente

Numero_Identificacion number(10,0) not null, Telefono_1
 number(15,0), Telefono_2 number(15,0), Telefono_3 number(15,0),
 Telefono_4 number(15,0).

➤ Tabla Correos de Personal Docente

Numero_Identificacion number(10,0) not null, Correo_1 varchar2(30),
 Correo_2 varchar2(30), Correo_3 varchar2(30), Correo_4 varchar2(30).

➤ Tabla Datos Extra-Laborales de Personal Docente

Numero_Identificacion number(10,0) not null, Nombre_Empresa
 varchar2(100), Cargo_Empresa varchar2(100).

➤ Tabla de Personal Administrativo

Pais_Nacimiento varchar2(50),
 Numero_Pasaporte varchar2(10),
 Numero_cedula number(10,0) not null ,
 Apellidos varchar2(50),
 Nombres varchar2(50),
 Sexo varchar2(20),
 Fecha_Nacimiento date,
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Dependencia_Administrativa varchar2(50),

Tipo_Personal varchar2(50),
 Condicion_Laboral varchar2(20),
 Nucleo_Institucion varchar2(50),
 Discapacidad varchar2(50),
 Correo_Electronico varchar2(50).

➤ Tabla Historial Académico de Personal Administrativo

Numero_Identificacion number(10,0) not null ,
 Estudio_Realizado varchar2(100),
 Empresa_Institucion varchar2(100),
 Promedio_Notas number(5,3),
 MENSION_HONORIFICA varchar2(20),
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Titulo_Certificado varchar2(100).

➤ Tabla Direcciones de Personal Administrativo

Numero_Identificacion number(10,0) not null, Direccion_1
 Varchar2(100), Direccion_2 varchar2(100), Direccion_3 varchar2(100),
 Direccion_4 varchar2(100).

➤ Tabla Teléfonos de Personal Administrativo

Numero_Identificacion number(10,0) not null, Telefono_1
number(15,0),

Telefono_2 number(15,0), Telefono_3 number(15,0), Telefono_4
number(15,0).

➤ Tabla Correos de Personal Administrativo

Numero_Identificacion number(10,0) not null, Correo_1 varchar2(30),
Correo_2 varchar2(30), Correo_3 varchar2(30), Correo_4 varchar2(30).

➤ Tabla Datos Extra-Laborales de Personal Administrativo

Numero_Identificacion number(10,0) not null, Nombre_Empresa
varchar2(100),

Cargo_Empresa varchar2(100).

➤ Tabla Teléfonos

Numero_Identificacion number(10,0) not null,

Telefono_1 number(15,0),

Telefono_2 number(15,0),

Telefono_3 number(15,0),

Telefono_4 number(15,0).

➤ Tabla Direcciones

Numero_Identificacion number(10,0) not null,
 Direccion_1 Varchar2(100),
 Direccion_2 varchar2(100),
 Direccion_3 varchar2(100),
 Direccion_4 varchar2(100).

➤ Tabla Correos Electrónicos

Numero_Identificacion number(10,0) not null,
 Correo_1 varchar2(30),
 Correo_2 varchar2(30),
 Correo_3 varchar2(30),
 Correo_4 varchar2(30).

➤ Tabla Datos Personales de Estudiantes

Pais_Nacimiento varchar2(100),
 Condicion_Residencia varchar2(50),
 Numero_Pasaporte varchar2(10),
 Numero_cedula number(10,0) not null ,
 Apellidos varchar2(50),
 Nombres varchar2(50),
 Sexo varchar2(50),
 Edo_Civil varchar2(20),
 Fecha_Nacimiento date,
 Tratamiento varchar2(20),
 Especialidad varchar2(50),

Etnia varchar2(20),
 Privado_De_Libertad varchar2(20),
 Discapacidad varchar2(50),
 Nucleo_Institucion varchar2(40).

➤ Tabla Datos Académicos de Estudiantes

Numero_Identificacion number(10,0) not null ,
 Codigo_Programa number(10,0),
 CODIGO_COHORTE varchar2(10),
 CODIGO_SECCION varchar2(10),
 Mension_Programa varchar2(100),
 Modalidad_De_Ingreso varchar2(50),
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Fecha_Defensa date,
 Ultimo_Periodo_Incrit date,
 Turno varchar2(50),
 Creditos_Aprobados number(5,0),
 Promedio_Notas number(5,3),
 Inscripcion_Condicional number(1,0),
 MENSION_HONORIFICA varchar2(20),
 Becado varchar2(10),
 INSTITUCION_OTORGA_BECA varchar2(100),
 En_Trabajo_De_Grado varchar2(20),
 TESIS_APROBADA varchar2(10),
 Nucleo_Institucion varchar2(40),
 Categoria_Académica varchar2(50),

EstadoAcademico varchar2(50),
 Fecha_Inicio_Permission date,
 Fecha_Fin_Permission date.

➤ Tabla Historial Académico de Estudiantes

Numero_Identificacion number(10,0) not null ,
 Estudio_Realizado varchar2(100),
 Empresa_Institucion varchar2(100),
 Promedio_Notas number(5,3),
 MENSION_HONORIFICA varchar2(20),
 Fecha_Ingreso date,
 Fecha_Egreso date,
 Titulo_Certificado varchar2(100),

➤ Tabla Direcciones de Estudiantes

Numero_Identificacion number(10,0) not null,
 Direccion_1 Varchar2(100),
 Direccion_2 varchar2(100),
 Direccion_3 varchar2(100), Direccion_4 varchar2(100).

➤ Tabla teléfonos de Estudiantes

Numero_Identificacion number(10,0) not null,
 Telefono_1 number(15,0),
 Telefono_2 number(15,0),

Telefono_3 number(15,0),
Telefono_4 number(15,0).

➤ Tabla Correos de Estudiantes

Numero_Identificacion number(10,0) not null, Correo_1 varchar2(30),
Correo_2 varchar2(30), Correo_3 varchar2(30), Correo_4 varchar2(30).

➤ Tabla Datos Laborales de Estudiantes

Numero_Identificacion number(10,0) not null, Nombre_Empresa
varchar2(100), Cargo_Empresa varchar2(100).

➤ Tabla Usuarios de Sistema

Numero_Identificacion number(10,0), Clave varchar2(15),
Nivel_Seguridad varchar2(10).

➤ Tabla Sesiones Administrativas

Codigo_Sesion varchar2(50) not null,
Numero_Identificacion number(10,0) not null,
Fecha_Conexion varchar2(10),
Fecha_Desconexion varchar2(10),
Hora_Conexion varchar2(8),

Hora_Desconexion varchar2(8),
DIRECCION_IP varchar2(25),
Comentario varchar2(50).

➤ Tabla Sesiones de Estudiantes

Codigo_Sesion varchar2(50) not null,
Numero_Identificacion number(10,0) not null,
Fecha_Conexion date,
Fecha_Desconexion date,
Hora_Conexion varchar2(50),
Hora_Desconexion varchar2(50),
Numero_Transacciones number(5,0),
Comentario varchar2(50).

➤ Tabla Sesiones de Profesores

Codigo_Sesion varchar2(50) not null,
Numero_Identificacion number(10,0) not null,
Fecha_Conexion date,
Fecha_Desconexion date,
Hora_Conexion varchar2(50),
Hora_Desconexion varchar2(50),
Numero_Transacciones number(5,0),
Comentario varchar2(50).

4.4.2 Modelo físico de la Base de Datos del Sistema SISGAP

En la figura 4.11 se muestra el modelo físico de datos de la Base de Datos del sistema donde se hace una representación de l modelo relacional de las tablas.

4.5 Modelo de gestión de contenido

El diseño de hipertextos es una de las actividades que consume más tiempo en esta fase. Se requiere poseer ya un conocimiento absoluto de cómo funciona el sistema y de lo que se requiere específicamente. Durante esta actividad se diseñarán las páginas o la operación de las páginas que integrarán los site views que se requieren.

4.5.1 Página de Inicio de Sesión o Ingresar al Sistema SISGAP

La página contiene un formulario de nombre y contraseña. El usuario introduce el nombre y la clave y presiona el botón entrar. Si el usuario ingresa correctamente sus datos, se le da acceso al sistema. En caso de que el nombre y contraseña no coincidan, se regresa a la página de entrada con un mensaje de error. Observe la figura 4.12.

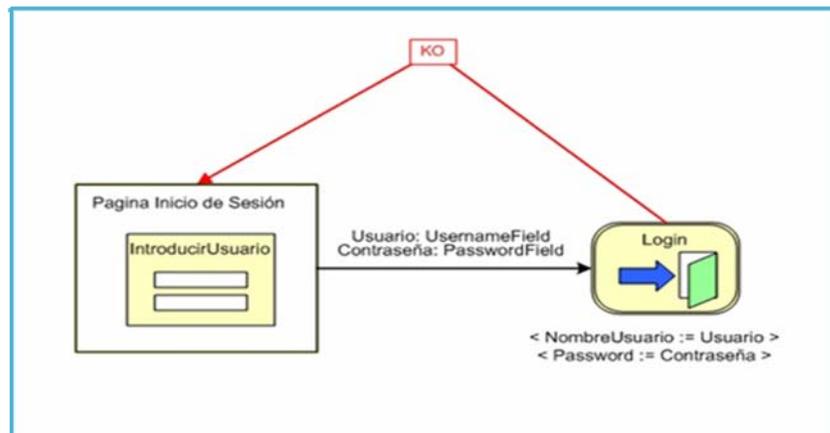


Figura 4.12. Diagrama de Gestión de contenido Inicio de sesión. Fuente: Autores. 2009

4.5.2 Páginas Agregar Datos al Sistema SISGAP

Par agregar contenido en el sistema, se utiliza un modelo de gestor de contenido generalizado que consiste en los siguientes pasos:

- El usuario introduce los datos a ser guardados en el sistema en un formulario de entrada.
- Al finalizar de agregar los datos, presiona el botón aceptar.

- Una unidad de creación se encarga de grabar los datos en la unidad de datos correspondientes.
- Otra unidad de creación, del registro del sistema, recibe el nombre de usuario de una variable global de sesión, y lo guarda en el registro del sistema.
- Si en cualquier momento falla la ejecución, se regresa a la misma página con un mensaje de error, por el contrario, si se ejecuta sin problemas, se regresa a la misma página con un mensaje de éxito.

A continuación se muestra una serie de diagramas de gestión de contenido de las páginas que usan este modelo generalizado de agregar contenido al sistema. En la figura 4.13 se muestra estos pasos.

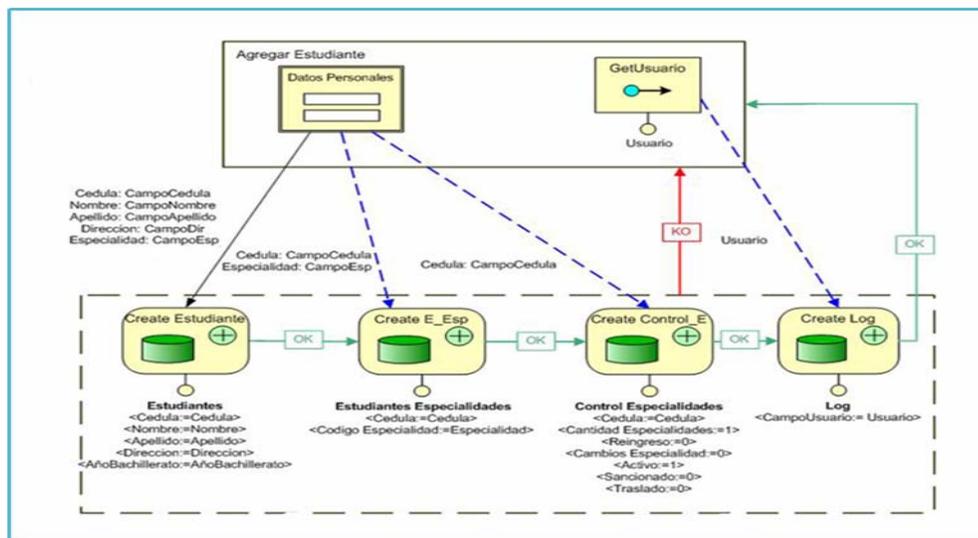


Figura 4.13. Diagrama de gestión de contenido Ingresar estudiante.

Fuente: Autores. 2009

4.5.3 Páginas de Modificar Datos en el Sistema SISGAP

Para modificar los datos del sistema, se diseñó un modelo de gestor de contenido muy similar al de agregar los datos. Consiste en los siguientes pasos:

- El usuario llena el formulario de búsqueda donde se localizará el registro a ser modificado.
- El sistema muestra el resultado de la búsqueda en otro formulario donde puede ser editada la información.
- El usuario presiona el botón aceptar al finalizar las modificaciones.
- Una unidad de modificación se encarga de realizar el cambio de la información en la unidad de datos adecuada.
- Otra unidad de creación, del registro del sistema, recibe el nombre de usuario de una variable global de sesión, y lo guarda en el registro del sistema.
- Si en cualquier momento falla la ejecución, se regresa a la misma página con un mensaje de error, por el contrario, si se ejecuta sin problemas, se regresa a la misma página con un mensaje de éxito.

Las páginas modificación de datos siguen el esquema representado a través de la figura 4.14.

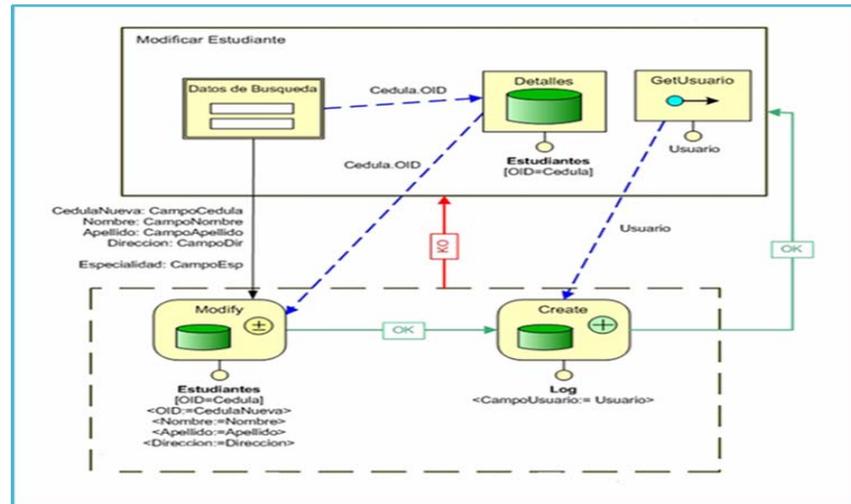


Figura 4.14. Diagrama de gestión de contenido Modificar datos estudiante.
Fuente: Autores. 2009

4.5.4 Páginas de Eliminar Datos

Las páginas de eliminar datos son muy similares todas. Básicamente muestra todos los registros almacenados en el sistema, ordenados según algún criterio; el usuario puede seleccionar los registros en específico y luego eliminarlo. Los pasos que ejecuta el sistema son los siguientes:

- El sistema muestra automáticamente los registros almacenados en el sistema.
- El usuario puede seleccionar algunos registros a eliminar o también buscar alguno en particular usando un formulario de búsqueda.
- Luego de escoger los registros a borrar, el usuario presiona el botón de eliminar.

- La unidad de eliminación correspondiente se encarga de borrar el o los registros seleccionados por el usuario.
- Otra unidad de creación, del registro del sistema, recibe el nombre de usuario de una variable global de sesión, y lo guarda en el registro del sistema.
- Si en cualquier momento falla la ejecución, se regresa a la misma página con un mensaje de error, por el contrario, si se ejecuta sin problemas, se regresa a la misma página con un mensaje de éxito.

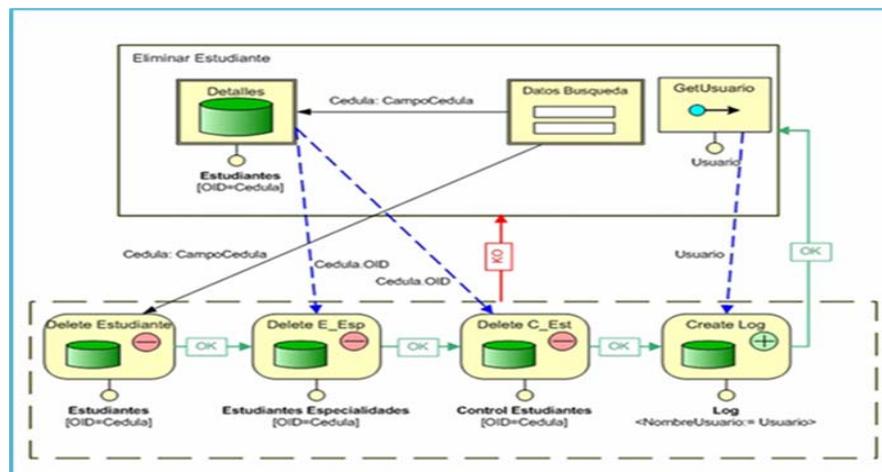


Figura 4.15. Diagrama de gestión de contenido eliminar datos estudiantile.

En la figura 4.15 se muestran una serie de páginas que siguen este esquema:

4.5.5 Páginas de Reportes en el sistema SISGAP

Estás páginas están basadas en consultas que se realizan al sistema. Algunas pueden requerir de un formulario de búsqueda, mientras que para otras esto no es necesario.

En el caso de los reportes que no requieran datos de búsqueda se muestra inmediatamente el resultado de la consulta realizada en un formato adecuado para impresión. Por el contrario, en las páginas que necesitan parámetros de búsqueda, primero se llena un formulario con los datos requeridos y luego se muestra el resultado de la consulta en una página nueva con el formato apropiado para la impresión.

A continuación en la figura 4.16 algunos diagramas de las páginas que realizan reportes.

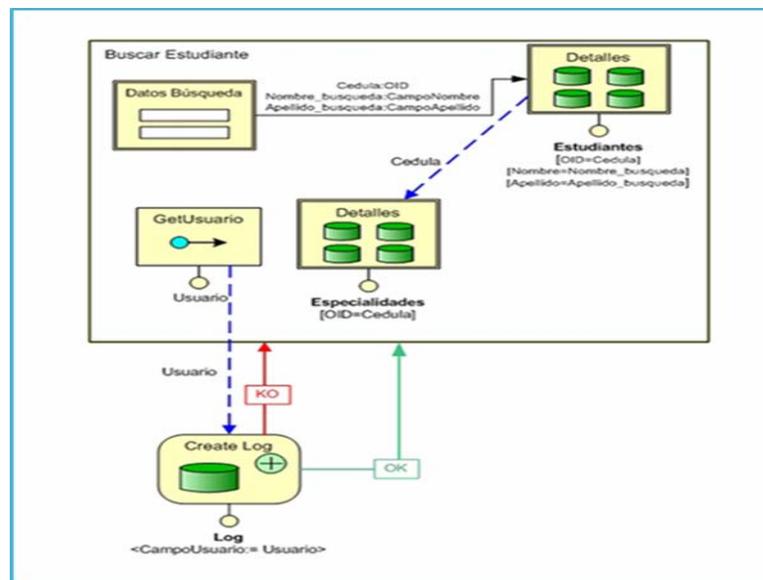


Figura 4.16. Diagrama de gestión de contenido generar reportes

La figura 4.17 describe el proceso de crear listas de asistencias o de clases.

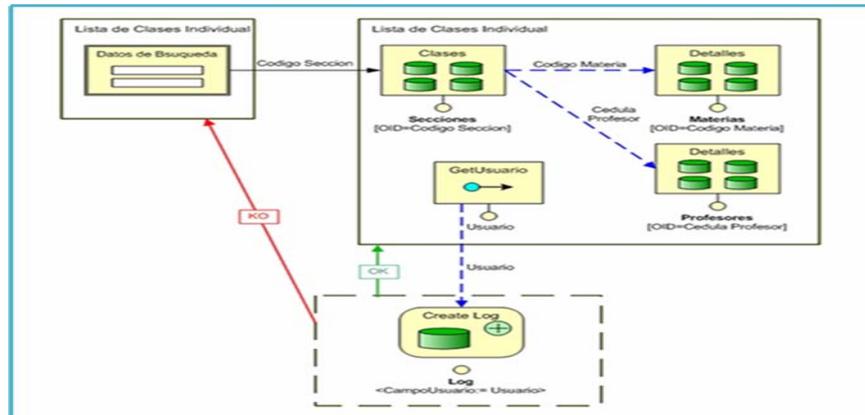


Figura 4.17. Diagrama de gestión de contenido de crear listas asistencia o de clases.
Fuente: Autores. 2009

4.6 Implementación del sistema SISGAP

En la implementación del sistema se hace necesario describir los subsistemas necesarios para el funcionamiento del sistema y los componentes que los utilizan.

4.6.1 Identificación de Componentes de la Arquitectura y Subsistemas

Existe un servidor donde se alojan un conjunto de páginas Web que representan el sistema como tal, y que se ejecutarán a través del motor de scripts PHP, y serán distribuidas por la intranet e Internet mediante el servidor Web Apache por el protocolo TCP/IP. Adicional a estos componentes existe el servidor de base de datos Oracle que es el encargado de manejar las transacciones que solicite el cliente a través del navegador Web. La figura 4.18 muestra el diagrama de despliegue de componentes con los subsistemas de servicio.

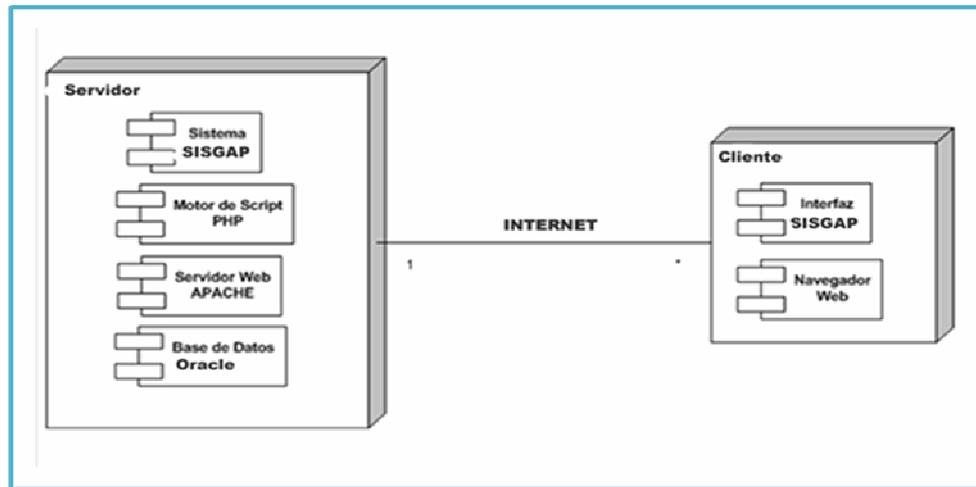


Figura 4.18. Diagrama de despliegue del Sistema SISGAP.

4.7 Evaluación de la fase de elaboración

En esta fase se logró precisar los casos de uso de la fase de inicio y agregar otras características al sistema, además se logró tener una estructura básica que será ampliada en la siguiente fase del proyecto.

Para efectos de funcionamiento el sistema logra realizar las operaciones descritas en los casos de uso y sus diagramas de análisis y diseño, pero será en la fase de construcción donde se realcen los componentes de software que realizan las tareas planificadas de estos casos de uso y diagramas antes mencionados.

CAPITULO V

FASE DE CONSTRUCCIÓN

5.1 Introducción

En la fase de elaboración se realizó el diseño de la arquitectura del sistema, se hace necesario poner en marcha la construcción de los elementos diseñados para fortalecer la estructura de la misma, creando un software con capacidad operativa que pueda ser puesto en marcha lo más prontamente posible, sin olvidar que tal software debe cumplir con todos los requerimientos obtenidos durante las fases anteriores.

En el capítulo anterior se desarrollaron los flujos de trabajo Requisitos, Análisis y Diseño del proyecto SISGAP, y parcialmente el flujo de trabajo Implementación. La Figura 5.1 muestra los pasos que se darán en esta fase además de visualizar el progreso alcanzado en el trabajo realizado.

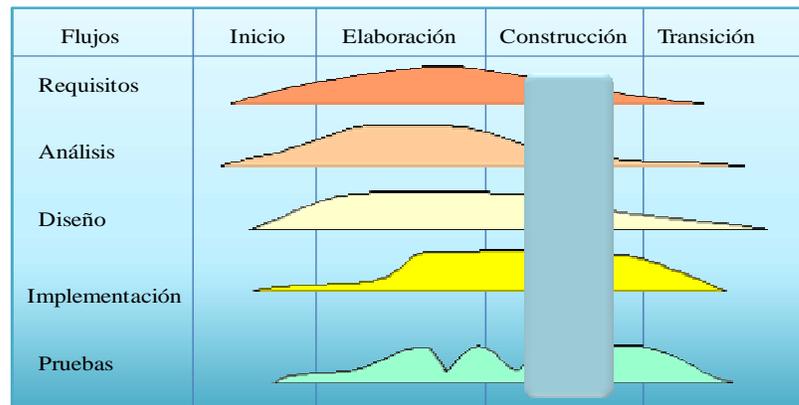


Figura 5.1: Fase de Construcción (UML). Fuente: Jacobson I. (2000)

El objetivo principal de esta fase es construir la aplicación basándose en la arquitectura obtenida, realizando ajustes y modificaciones si el sistema lo requiere (siempre y cuando respetando la arquitectura base y los requerimientos ya estudiados), probando los distintos códigos y la comunicación con las bases de datos, diseñando y codificando cada una de las interfaces con el usuario.

5.2 Flujo de trabajo implementación

El flujo de trabajo implementación del proyecto fue ya iniciado en la fase anterior, en esta será desarrollado totalmente para lograr obtener un software ejecutable y operativo. Para la codificación de las páginas se usará el lenguaje de programación PHP a través del ambiente de desarrollo que ofrece Dreamweaver 8.0. Las operaciones de base datos serán gestionadas por el administrador de base datos Oracle 8i. Se usará Easy PHP1.8 como servidor local de pruebas. La integración de todos estos elementos formará la arquitectura de trabajo estable que se diseñó en la fase anterior.

5.2.1 Implementación del Modelo de Hipertexto

5.2.1.1 Computación de una página dinámica

El flujo de trabajo para procesar una página dinámica de un contenido guardado en una base de datos se esquematiza en la figura 5.2.

En primer lugar la figura 5.2 describe lo que sucede cuando un usuario realiza cualquier consulta al sistema. Para iniciar el procesamiento de una consulta el sistema extrae de la solicitud HTTP los parámetros de la consulta, realiza la conexión al manejador de bases de datos y luego los parámetros se ordenan en comandos llamados queries y se envían al sistema manejador de bases de datos para realizar diversas comprobaciones y ejecutar el query. Una vez se tiene el resultado del query, éste se envía al manejador de scripts para ordenarlos e imprimirlos en la página Web junto con los otros elementos Web de esta.

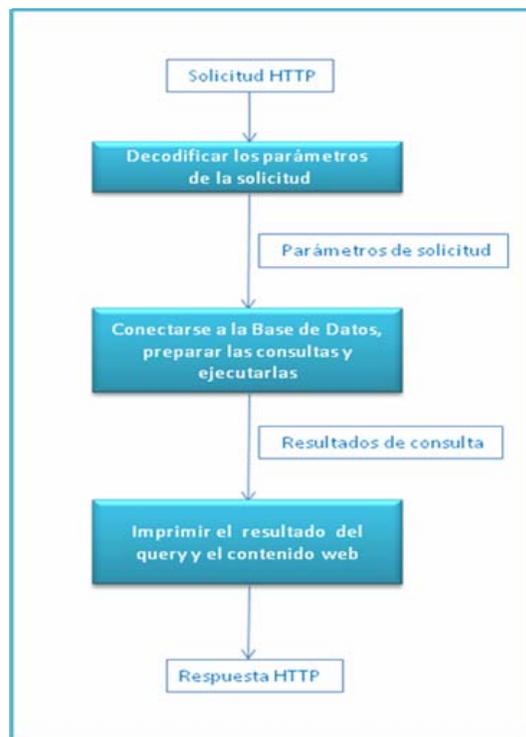


Figura 5.2. Procesamiento de una página con contenido extraído de una base de datos.
Fuente: Autores. 2009

La parte esencial del procesamiento de las páginas son la construcción y

ejecución de las consultas de retiro de información y la producción del código HTML. Cada unidad en la página tiene sus propias reglas para retirar el contenido y producir el código HTML, y cuando la página contiene múltiples unidades enlazadas, el orden en el cual las consultas son ejecutadas es importante, porque una unidad puede requerir entrada de otras unidades. Es en la solución de este tipo de problemas que la especificación WebML de las páginas ayuda: WebML clasifica los elementos de contenido que pueden aparecer en la página en categorías bien definidas, correspondientes a las diferentes tipo de unidades, y establece reglas bien definidas para el orden en el cual las unidades son procesadas, representado por el procedimiento de procesamiento de alto nivel de páginas. Por lo tanto, el esquema general de actividades ilustrado en la figura 5.2, puede ser especializado al caso de una página WebML, consistiendo en muchas unidades enlazadas, para obtener un esquema general de implementación como el descrito a continuación.

Parte 1: Extraer parámetros de la petición HTTP.

Parte 2: Conectar a la base de datos.

Parte 3: Preparar y ejecutar las consultas:

- Construir la declaración de la consulta de la unidad actual.
- Ejecutar la consulta.
- Si existen unidades dependientes, enlazar la salida de la unidad actual a la entrada de las unidades dependientes y repetir los pasos 1-3.

Parte 4: Procesar el contenido de la página dinámica:

- Construir el marco HTML para renderizar la unidad actual, del resultado de la consulta asociada a el.
- Construir los enlaces de salida de la unidad:
- Construyendo la parte fija del URL.
- Construyendo los parámetros asociados con el enlace.

Parte 5: Eliminar recursos temporales.

5.2.1.2 Codificación de las páginas Web

Desde este punto del proyecto se comienza a presentar el código fuente del diseño del sistema logrado hasta ahora. Para realizar esta tarea se utiliza scripts del lenguaje PHP, este lenguaje soporta la programación orientada a objeto, el cuál se pretende utilizar desde el principio del proyecto. A continuación se tomara alguna interfaz y se mostrará de ella, el código fuente que la crea o imprime en el navegador Web.

- Interfaz de usuario ingresar al sistema o inicio de sesión.

Esta interfaz es utilizada por todo usuario que desee ingresar al sistema, el archivo php que la imprime en el navegador es “InicioSesion.php” y a continuación se muestra su código fuente:

InicioSesion.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Universidad de Oriente</TITLE>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-
1"><LINK
href="archivos/estilos.css"
type=text/css rel=stylesheet>
<style type="text/css">
.Estilo8 {color: #000066}
.Estilo106 {color: #3FADCDCD}
.Estilo108 {
font-weight: bold;
font-size: 12pt;}--></style>
<META content="MSHTML 6.00.2900.3059" name=GENERATOR>
<script language=JavaScript>
function inhabilitar(){
alert ('Esta función está inhabilitada.\n\n')
return false }
document.oncontextmenu=inhabilitar
</script>
<script language="JavaScript" type="text/JavaScript"><!--
function MM_findObj(n, d) { //v4.01
var p,i,x; if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.frames.length) {
d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];
for(i=0;!x&&i<d.layers&&i<d.layers.length;i++)
x=MM_findObj(n,d.layers[i].document);

```

```

if(!x && d.getElementById) x=d.getElementById(n); return x; }
function fx_completa() {
    var par_usuario = "";
    par_usuario = document.form1.ci_index.value;
    par_usuario = '0000000' + par_usuario;
    inicio = par_usuario.length - 8 ;
    par_usuario = par_usuario.substring( inicio - 1);
    document.form1.ci_index.value = par_usuario;
}
function MM_validateForm() { //v4.0
    var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
        if (val) { nm=val.name; if ((val=val.value)!="") {
            if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
                if (p<1 || p==(val.length-1)) errors+='- '+nm+' Contiene una Dirección de
internet.\n';}
            else if (test!='R') { num = parseFloat(val);
                if (isNaN(val)) errors+='- '+ ' La cédula debe ser numérica!\n';
                if (test.indexOf('inRange') != -1) {
                    p=test.indexOf(':');
                    min=test.substring(8,p); max=test.substring(p+1);
                    if (num<min || max<num) errors+='- '+nm+' must contain a number   between
'+min+' and '+max+'.\n';} } }
            else if (test.charAt(0) == 'R') errors += '- '+nm+' es Imprescindible.\n'; } }
        if (errors) alert('Error:\n'+errors);
        document.MM_returnValue = (errors == "");} //-->
</script>

```

```

<body ONLOAD="document.form1.ci_index.focus();" topmargin="0"
leftmargin="0" oncontextmenu="return false" location="no" ondragstart="return
false" onselectstart="return false" >
<TABLE cellSpacing=0 cellPadding=0 width=727 align=center border=2>
<TBODY>
<TR>
<TD bgColor=#f9f2f0><IMG
src="Iconos e Imagenes/titulo2.jpg" alt="jc"
width=725 height=97></TD>
</TR>
<TR>
<TD vAlign=top align=top bgColor=#ffffff height=461> <table
width="100%" height="459" border="0" cellpadding="3" cellspacing="1" ><tr>
<td width="22%" valign="top">
<p></p>
</td>
<td width="78%" height="457" valign="top">
<FORM action=login.php method=post enctype="application/x-www-form-
urlencoded" name=form1 target="_self"
onsubmit="MM_validateForm('cedula','R','clave','R');return
document.MM_returnValue" >
<div align="center">
<p>&nbsp;</p>
<p></p>
<p>&nbsp;</p>
<table width="43%" height="122" border="0" cellpadding="0"
cellspacing="0">
<tr>

```



```

</tr>
</table>
<br>
<table width="445" border="0" cellpadding="0" cellspacing="0">
<tr>
<td align="center" class="Estilo8"><a
href="OlvideClave.php">Olvidé; mi contraseñ;a</a> </td>
</tr>
<tr>
<td><p class="Estilo106">&nbsp;</p>
<p class="Estilo106">Primera vez que inicia sesión ingrese 4
&uacute;ltimos dígitos de cédula como contraseña</p></td>
</tr>
</table>
</div>
</FORM></td>
</tr>
</table>
<!-- Fin tabla de opciones--></TD>
</TR>
</TBODY></TABLE>
</TD></TR>
<TR>
<TD class=Estilo28 vAlign=top align="center">
<div align="center"></div></TD></TR>
</BODY></HTML>

```

➤ Página de logueo exitoso o Página de Bienvenida

Para esta interfaz se muestra la imagen de ella en la figura 5.3 además de su código fuente.



Figura 5.3. Interfaz Bienvenida. Fuente: Autores. 2009

Código Fuente de la Interfaz

```
<?php
session_start();
if( isset($_SESSION['password'] )){
if ( isset($_SESSION['user']) ) {}
else
{header("Location:InicioSesionAdministracion.php");}}
else{ header("Location:InicioSesionAdministracion.php");
echo " no enviaste ni clave ni usuario";}
// Atrapar las variables Usuario y Clave de la sesión.
```

```

    $usuario = $_SESSION['user'];
    $clave = $_SESSION['password'];
    // Incluir Librerías y otros archivos necesarios.
    include_once("PeticonesAdministracion.php");
    include_once("ClaseEstudiante.php");
    include_once("ClaseBaseDeDatos.php");
    $NuevoObjetoBD = new BaseDeDatos();
    $conn = $NuevoObjetoBD->ConectarBD($usuario,$clave);?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Universidad de Oriente</TITLE>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-
1"><LINK
href="archivos/estilos.css" type=text/css rel=stylesheet>
<STYLE type=text/css>
.Estilo105 {FONT-SIZE: 18px}
.Estilo107 {font-size: 10pt}
.Estilo108 {
color: #333333;
font-size: 12px;}
.Estilo110 {
font-size: 16px;
font-weight: bold;}
.Estilo111 {color: #006699}
</STYLE><META content="MSHTML 6.00.2900.3059"
name=GENERATOR>
<script language=JavaScript>
// Método para evitar desplegar menús con el botón derecho del ratón.
function inhabilitar(){
alert ('Esta función está Deshabilitada.\n\n')

```

```

return false }
document.oncontextmenu=inhabilitar
// Fin de Método Bloquear Ratón.
</script></HEAD>
<BODY vLink=#000000 aLink=#000000 link=#000000 topMargin=0
oncontextmenu="return false" location=no>
<TABLE cellSpacing=0 cellPadding=0 width=845 align=center border=0>
<TBODY><TR>
<TD colspan="2" bgColor=#ffffff>&nbsp;</TD>
<TD width="731" bgColor=#f9f2f0><IMG height=94
src="Iconos e Imagenes/titulo2.jpg"
width=729></TD>
</TR><TR>
<TD width="20" vAlign=top bgColor=#eeeeee></TD>
<TD width="94" vAlign=top bgColor=#eeeeee><TABLE height = 284
cellSpacing=1 cellPadding=3 width="100%" bgColor=#cccccc border=0>
<TR>
<TD class=Estilo33 id=MSlinks bgColor=#000066 height=21>&nbsp;</TD>
</TR><TBODY><TR>
<TD height="12" bgColor=#ffffff id=MSlinks><div align="center"><A
href="SistemaProgramasPostgrado.php"
class="Estilo107">Programas</A></div></TD>
</TR><TR>
<TD height="24" bgColor=#ffffff id=MSlinks><div align="center"><A
href="SistemaLineasInvestigacion.php" class="Estilo107">Lineas De
Investigaci&oacute;n </A></div></TD>
</TR><TR><TD height="24" bgColor=#ffffff id=MSlinks><div
align="center"><A

```

```

        href="SistemaGruposInvestigacion.php"      class="Estilo107">Grupo      De
Investigaci&oacute;n</A></div></TD></TR><TR>
        <TD height="24" bgColor=#ffffff id=MSlinks><div align="center"><A
        href="SistemaTrabajosDeGrado.php"  class="Estilo107">Trabajos De Grado
</A></div></TD></TR><TR>
        <TD class=Estilo33 id=MSlinks bgColor=#000066 height=21>&nbsp;</TD>
</TR><TR>
        <TD height="5" bgColor=#ffffff id=MSlinks><div align="center"><A
        href="SistemaAdministrarProfesores.php"
class="Estilo107">Profesores</A></div></TD></TR><TR>
        <TD height="12" bgColor=#ffffff id=MSlinks><div align="center"><a
        href="SistemaAdministrarPersonalAdministrativo.php"
class="Estilo107">Personal Administrativo </a></div></TD></TR><TR>
        <TD height="25" bgColor=#ffffff id=MSlinks><div align="center"><A
        href="SistemaAdministrarEstudiantes.php"      class="Estilo107">Estudiantes
Regulares </A></div></TD></TR><TR>
        <TD height="25" bgColor=#ffffff id=MSlinks><div align="center"><A
        href="GuardarDatosPersonalesEstudiante.php"
class="Estilo107">Inscripciones</A></div></TD></TR><TR>
        <TD height="11" bgColor=#ffffff id=MSlinks><div align="center"><span
class="Estilo107"><a
href="SistemaMostrarDatosInstitucion.php">Institucion</a></span></div></TD>
</TR><TR>
        <TD height="12" bgColor=#ffffff id=MSlinks><div
align="center"><a href="SistemaArchivosCCNPG.php"
class="Estilo107">ArchivosOPUSU </a></div></TD></TR><TR>
        <TD height="12" bgColor=#ffffff id=MSlinks><div align="center"><span
class="Estilo107"><a href="Directorios/index.php">Directorios
</a></span></div></TD></TR><TR>
        <TD class=Estilo33 id=MSlinks bgColor=#000066 height=21>&nbsp;</TD>

```

```

</TR><TR> <TD height="19" bgColor=#ffffff id=MSlinks><A
href="SistemaCerrarSesion.php" class="Estilo107">Cerrar Sesi&oacute;n
</A></TD></TR></TBODY></TABLE></TD>
<TD vAlign=top bgColor=#eeeeee height=362>
<TABLE height=418 width="100%" border=0><TBODY><TR><TD
width="77%" height="414" valign="top" bgColor=#ffffff
style="PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-
BOTTOM: 0px; PADDING-TOP: 0px"><table width="721" height="459"
border="0" bgcolor="#FFFFFF"><tr>
<td width="349" height="23"></td></tr><tr>
<td height="28" class="Estilo105 Estilo108"><div align="center">
<p class="Estilo100 Estilo110 Estilo111">Sitio administrativo de los servicios
de Postgrado de la Universidad De Oriente.</p>
</div></td></tr><tr>
<td height="7" ></td></tr><tr>
<td height="373" ><div align="center">
<table width="209" height="196" border="0" bordercolor="#009999"><tr>
<td width="365" height="31"><div align="center"><span class="Estilo100
Estilo110 Estilo111"> Personal Administrativo</span></div></td></tr><tr>
<td><div align="center"></div></td></tr></table> </div></td></tr><tr><td
height="12" >
</td>
</tr></table> </TD></TR></TBODY></TABLE></TD>
</TR></TABLE></BODY></HTML>

```

5.3 Pruebas de interacción

En este flujo de trabajo se realizan las pruebas necesarias para determinar cualquier anomalía en los casos de uso, se ingresan datos en las respectivas interfaces para lograr este objetivo. Se prueban las rutas lógicas del software y la colaboración entre los componentes, al proporcionar casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos. Esta técnica de evaluación de software es denominada pruebas de caja blanca.

Estas pruebas se aplicaron a todos los componentes de software desarrollados, comprobando así la exitosa integración de la aplicación. Además, se probaron los enlaces de las páginas, que no hubiesen enlaces rotos o vacíos, ni enlaces que diseccionarían hacia alguna página indebida. A continuación se presentan ejemplos de algunos casos de prueba de integración que se aplicaron al sistema.

5.3.1 Diseño de Casos de Prueba

Los casos de prueba deben realizarse sabiendo como funciona el programa, es decir, como es su desempeño internamente. A continuación se presenta un conjunto de tablas que describen los casos de prueba diseñados y la relación entre cada uno de ellos. En la tabla 5.1 se muestra el ingreso de datos al usuario para acceder a sistema.

Tabla 5.1. Ingresar al sistema. Fuente Autores. 2009

	Iniciar sesión en el sistema.
Entrada	Nombre de usuario 15548152, contraseña 8152.
Resultado	El usuario es autenticado e ingresa al sistema.
Condiciones	El usuario está creado en la base de datos del sistema.
Procedimiento	<p>Escriba el nombre de usuario y la contraseña en los cuadro de texto correspondiente.</p> <p>Presione el botón entrar. Se ingresa al sistema.</p>

Tabla 5.2. Ingresar Nuevo programa de Postgrado. Fuente Autores. 2009

Ingresar Programa de Postgrado.	
Entrada	<p>(Estos datos se extrajeron de ¡Error! Referencia de hipervínculo no válida.)</p> <p>Código Programa: 1745 Nombre Programa: Maestría en Agricultura tropical. Mención: Mejoramiento de Plantas. Grado Académico: Magister Scientiarum. Área de conocimiento: Ciencia del Agro y del Mar. Dirección: Campus Juanico. Urb. Juanico. Maturín. Edo. Monagas. Núcleo: Monagas. Modalidad: Presencial. Acreditado: Sí. Gaceta Oficial: 36.363. Fecha Creación: 29/12/1997.</p>
Resultado	El programa es registrado en el sistema.
Condiciones	El programa no debe estar registrado y el usuario debe ser Persona Administrativo.
Procedimiento	Acceder al menú de tareas y escoger "Programas" luego escoger de menú desplegable la opción Nuevo Programa. Ingrese los datos del programa en el formulario y presione aceptar para guardarlo en la base de datos

Una vez que el usuario está logueado correctamente en el sistema puede escoger entre las tareas usuales de postgrado, para continuar con el caso de prueba se ingresará al sistema un nuevo programa de postgrado. En la tabla 5.2 se puede observar la operación.

Un caso de prueba importante en este sistema es la comprobación de la creación correcta de los archivos OPSU, mecanismo implementado por este organismo para llevar un control estadístico de las instituciones universitarias que dictan cursos de postgrado. Estos archivos OPSU presentan un formato estricto de códigos según la información que se esté recopilando, en total son 7 tipos de archivos y tomaremos el caso de prueba anterior para generar el archivo OPSU correspondiente.

Tabla 5.3. Generar Archivo OPSU. Fuente Autores. 2009

Generar Archivo OPSU Programas de Postgrado.	
Entrada	<p>Escoger el tipo de archivo “Datos sobre los Programas de Postgrado”.</p> <p>Escoger el Año de consulta, o sea, los programas activos para ese año específico.</p> <p>Ingresar la oficina o dependencia que está creando el archivo OPSU.</p>
Resultado	<p>Se crea el archivo de texto 836_201_2009.txt en la bandeja de personal administrativo “Directorios” donde puede ser descargado.</p>
Condiciones	<p>Deben existir programas de postgrado registrados en el año de consulta o antes del mismo.</p>
Procedimiento	<p>Acceder al menú de tareas y escoger “Archivos OPSU” luego escoger del menú desplegable la opción Tipo de Archivo.</p> <p>Ingrese el año de consulta.</p> <p>Ingrese el nombre de la oficina o dependencia que está generando el archivo.</p> <p>Presione el botón “Vista previa” para ver los datos que van a ser codificados en el archivo.</p> <p>Presione el botón “Crear Archivo” para construir el archivo y alojarlo en la carpeta Directorios de Administración.</p> <p>Descargar el Archivo 836_201_2009.txt desde la carpeta Directorios haciendo clic sobre él.</p>

5.4 Evaluación de la fase de construcción

Esta fase culminó exitosamente, se logró desarrollar todos los casos de uso diseñados, además de integrar las funciones de soporte a cada uno de ellos. La codificación de los casos de uso en lenguaje php ha sido la tarea más larga de todas debido a la gran cantidad de archivos php que se necesitaron la lograr los objetivos propuestos. Las pruebas realizadas resaltaron algunos errores en la programación lo que llevo a profundizar el ordenamiento del código fuente en cada archivo php para obtener una visión clara de las tareas que realiza cada archivo y lograr un estándar en archivos de similar comportamiento.

CAPITULO VI

FASE DE TRANSICIÓN

6.1 Introducción

La fase de transición del Proceso Unificado de desarrollo de software abarca las últimas etapas de la actividad genérica de construcción y la primera parte de la actividad genérica de despliegue. El software se entrega a los usuarios para realizar pruebas beta, y la retroalimentación del usuario reporta tanto defectos como cambios necesarios. Además, el equipo de software crea la información de soporte necesaria (por ejemplo, manuales de usuario, guías de resolución de problemas, procedimientos de instalación) para el lanzamiento. Al final de la fase de transición, el incremento de software se convierte en un lanzamiento de software utilizable.

6.2 Lanzamiento de la versión beta

Para la preparación del lanzamiento de la versión beta de la aplicación, se ha planeado instalar el software en un área de trabajo, seleccionando un grupo de usuarios para que utilicen el software e instalar la estructura de hardware necesaria para que éste funcione. Entre los usuarios se requiere que estén presentes al menos, el jefe del Centro de Computación Académica y algunos Coordinadores de Postgrado. A cada uno de ellos se le suministrará la documentación que indica lo siguiente:

- Como se realiza el acceso al software.

- Requisitos mínimos para el uso del sistema.
- Los niveles de permiso por usuario, es decir, las restricciones o limitaciones de acceso de cada usuario.
- Como utilizar el software.

6.3 Evaluación de la fase de transición

Los objetivos propuestos se han alcanzado al culminar con esta fase, se ha llegado a obtener un software estable que administra los servicios del área académica de postgrado y a la vez captura la información necesaria para crear los archivos de estadísticas de la Oficina de Planificación del Sector Universitario (OPSU) colocando a la Universidad de Oriente en la vanguardia tecnológica al poseer una herramienta que le ayuda a administrar sus operaciones diarias en el área de postgrado y a la vez cumplir con las exigencias de entes del estado.

CONCLUSIONES

Se concretaron los objetivos propuestos al principio del proyecto de obtener un sistema administrativo del área de Postgrado de la Universidad de Oriente bajo plataforma Web desarrollado bajo el paradigma Orientado a Objetos, para ello se aplicó la metodología del Proceso Unificado de Desarrollo la cual permitió desarrollar el proyecto de una manera organizada y metódica obteniendo los modelos y artefactos del sistema.

Se logró construir un módulo del software que recopilara información de la base de datos y la cifrara bajo el formato de la Oficina de Planificación del Sector Universitario (OPSU), creando así todos los tipos de archivos que este ente exige.

El desarrollo de una estructura bien organizada de la base de datos bajo el esquema o modelo relacional permitió obtener interacciones rápidas para los usuarios de mayor demanda del sistema.

El diseño de los módulos del software utilizando el lenguaje de modelado WebML permitió una fácil construcción de la aplicación, comprensión del funcionamiento del sistema, integración de los módulos y mantenimiento de los mismos.

Por último el manual de usuario brinda el adecuado soporte al uso del sistema por parte del personal de postgrado.

RECOMENDACIONES

- Realizar una actualización del sistema de redes para un mejor desempeño del sistema debido a que la alta demanda que tiene la universidad en accesos a servicios Web puede ralentizar las operaciones sobre el sistema.

- Realizar un mantenimiento periódico a la base de datos, como por ejemplo respaldos diarios y optimizaciones a las tablas para que se mantengan actualizados los índices de búsqueda y el buen rendimiento de la misma.

- Crear un módulo de software que permita dar soporte estándar a sistemas manejadores de bases de datos por parte del sistema, como por ejemplo ADO.

BIBLIOGRAFÍA

- [1] Álvarez M., Álvarez R., Cuenca C. (2004) **“Programación en PHP – Manual Completo”**. <http://www.desarrollo.com/manuales/12>.
- [2] Appleby, D., Vandekopple, J. (1998). **“Lenguajes De Programación. Paradigma y Practica”**. Segunda Edición. Editorial McGraw-Hill. México.
- [3] Anónimo. (2005). **“Guía de UML”**.
<http://www.osmosislatina.com/Lenguaje/uml/casos.html>.
- [4] Anónimo. (2005). **“Introducción a la Programación Orientada a Objetos”**.
http://Pisuerga.inf.ubu.es./lsi/Invest/Java/Tutor/I_1.html.
- [5] CAMPOS, K. Y HERNÁNDEZ, Á. **“Desarrollo de un software para el proceso de suministro de hemoderivados realizados por el bando de sangre del hospital “Dr. Luís Razetti” de Barcelona a los diferentes pacientes que necesitan transfusiones de sangre”**

[6] Ceri S., Fraternali P., Bongio A., Brambilla M., Comai S. y Matera M., **“Designing Data-Intensive Web Applications”**. Morgan Kaufmann Publishers. Primera Edición. Estados Unidos. (2003).

[7] Consultoría de Seguridad. **Diagramas de Estado**. Disponible en:
<http://www.creangel.com/uml/estado.php>

[8] Elmasri R., Navathe S. (2000). **“Sistemas de Bases de Datos. Conceptos Fundamentales”**. Editorial Pearson Educación. Segunda Edición. México.

[9] Figueroa, P.: **“Elementos Rotacionales de UML 1.0”**. (1997). Disponible en:
<http://www.cs.ualberta.ca/~pfiguero/soo/uml/>

[10] García, J. **Introducción a ORACLE** (2001). Disponible en:
<http://www.webestilo.com/oracle/intro.phtml>.

[11] IBM **“Ingeniería de Software”**. (2005). (Manual Técnico) volumen 1. IBM Learning Services. Venezuela.

[12] IBM **“Lenguaje de Mercado de Hipertexto:HTML”** . (2005). (Manual Técnico) volumen 1. IBM Learning Services. Venezuela.

[13] IBM **“PHP Hypertext Preprocessor”**. (2006). (Manual Técnico) volumen 1. IBM Learning Services. Venezuela.

[14] IJELCHUK, C, **“Lenguaje de modelado Unificado”**, (2.003).

<http://www.clikear.com/manuales/uml>.

[15] JACOBSON, I. y BOOCH, G. **“El proceso Unificado de Desarrollo de Software”**, (2000). Pearson Educación, S.A. Madrid.

[16] **“Microsoft Diccionario de Informática e Internet”**. (2001). Editorial McGraw-Hill. Primera Edición. Madrid.

[17] Mora, F. (2003). **“UML: Lenguaje Unificado de Modelado”**. Primera Edición, Universidad de Alicante-DCCIA, España.

[18] MÚJICA, V. y GARCÍA, E. **“Desarrollo de un Software para la automatización de las actividades administrativas del departamento de Computación y Sistemas, de la universidad de Oriente Núcleo de Anzoátegui”**. Trabajo de Grado. Universidad de Oriente, Estado Anzoátegui, Venezuela (2005).

[19] NÚÑEZ, K. **“Desarrollo de un software para manejar los servicios académicos del Departamento de Mecánica”**. Trabajo de Grado. Universidad de Oriente, Estado Anzoátegui, Venezuela (2004).

[20] Página oficial WebML **“WebML”**.
<http://www.webml.org/webml/page1.do>

[21] Página oficial OPSU **“OPSU”**. (2005),
http://www.ccnpg.gob.ve/directorio_nacional/busqueda_det.asp?id=1745).

[22] Popkin Software and System: **Modelado de Sistemas con UML**.
Disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>

[23] TOVAR, D. **“Desarrollo de un software que permita automatizar las actividades asociadas al proceso de inscripción por el Departamento de Computación y Sistemas, vías Internet”**. Trabajo de Grado. Universidad de Oriente, Estado Anzoátegui, Venezuela (2008).

[24] Vilas, A.: **Diagramas de Colaboración**. Disponible en:
<http://www-gris.det.uvigo.es/~avilas/UML/node43.html>

[25] ZAVALA, R. “**Ingeniería de Software**”, Disponible en:
www.angelfire.com/scifi/jzavalas/apuntes/IngSoftware.html (2000).

Manual de usuario

A.1 Introducción

Bienvenidos a SISGAP, Software para la Automatización del Área Académica de Postgrado de la Universidad de Oriente. El sistema SISGAP es un Software diseñado para la gestión de todas las actividades asociadas a la gestión de datos académicos de los programas de postgrado, tales como datos de los programas de postgrado, datos del personal académico del área, datos de los estudiantes de postgrado, emisión de reportes, generación de archivos OPSU entre otras actividades; todo esto a través de Internet, permitiendo así que cada usuario tenga acceso al sistema desde sitios remotos e internos de la universidad.

A.2 Requisitos del sistema

Para el buen funcionamiento del sistema se requieren los siguientes requisitos:

Requisitos Mínimos

- Procesador INTEL Pentium III 256Mb RAM o similar.
- Monitor SVGA Resolución 800x600 píxeles o superior.
- Navegador IExplorer 5.0 o similar + Pluggin JavaScript
- Se requiere tener instalado Adobe Acrobat Reader.

Mínimo Recomendado

- Procesador INTEL Pentium IV 512Mb RAM o similar
- Monitor SVGA Resolución 1024x768 píxeles o superior.
- Navegador IExplorer 6.0, similar o superior + Pluggin JavaScript
- Adobe Acrobat Reader 7 o superior.
- Se recomienda el uso de impresoras.

A.3 Área de trabajo

De acuerdo a la figura A.1, el área de trabajo de SISGAP está dividida en tres secciones o partes, cada una de ellas cumpliendo funciones específicas:

A.3.1 Área de Menú

Es el área o marco izquierdo de la pantalla. Está compuesta por una serie de enlaces a las opciones que varían según el usuario.



Figura A.1- Áreas de Trabajo de SISGAP Fuente: Autores 2009

A.3.2 Área de Navegación

Área o marco inferior izquierdo. Su contenido varía de acuerdo al tipo de usuario: Si no está registrado o no ha iniciado sesión, el usuario puede acceder desde la página InicioSesionSISGAP el cual remite a un login que verifica y redirecciona al usuario. Si el usuario está registrado, aparecen las opciones asociadas a su nivel de acceso, incluyendo enlaces para cambio de contraseña y cierre de sesión:

- **Usuario de nivel 1 (estudiante):** Consultar Notas, consultar datos personales, imprimir constancia.
- **Usuario de nivel 2 (profesores):** Consultar horarios, consultar datos personales, generar listas asistencias, ingresar notas.
- **Usuario de nivel 3 (personal administrativo):** Administración general de los datos de programas de postgrado, estudiantes y profesores, administración de preinscripciones, generar reportes.
- **Usuario de nivel 4 (personal Control Estudios):** Administración de inscripciones, generar archivos OPSU, generar constancias.

A.3.3 Área de Contenido

Área o marco inferior derecho. Es el área de mayor tamaño y, como su nombre lo indica, es el área donde se muestra la información solicitada por el usuario, realizándose las operaciones asociadas a cada perfil de usuario.

A.4 Inicio

Al iniciar el sistema se mostrará la pantalla de la figura A.2, para los usuarios Personal Administrativos, pero para poder acceder a las opciones del mismo se requiere que ingrese el usuario y contraseña. En este caso el usuario es el número de cédula y la contraseña es, por defecto, la misma cédula truncada en los últimos 4 dígitos de ella (la contraseña puede ser cambiada posteriormente).



Figura A.2- Bienvenida de Sesión de SISGAP Fuente: Autores 2009

La página InicioSesion sólo puede verse si no hay sesión abierta. Si se desea iniciar una nueva sesión, debe cerrarse la activa, dando clic al enlace “cerrar sesión” ubicado en el área de navegación, al final de las opciones de los usuarios. Una vez cerrada la sesión, el usuario es redirigido a la pantalla principal del sistema.

A.5 Sesiones

Las sesiones comprenden distintas actividades de acuerdo al tipo de usuario, conservando el formato de pantalla.

A.5.1 Nivel de Acceso 1: Estudiante

La sesión más utilizada del sistema, puesto que puede ser accedida por muchos estudiantes de manera simultánea.



Figura A.3- Interfaz principal de usuario nivel 1. Fuente: Autores

Ésta es a su vez la más sencilla de todas las sesiones, ya que proporciona básicamente las opciones que un alumno necesita para realizar consultas al sistema, a saber:

A.5.1.1 Consultar Notas

Es un formulario de entrada de datos que permite al estudiante realizar solicitudes de información sobre notas de materias inscritas. Para ellos se deben realizar los siguientes pasos:

- Seleccionar la materia de la que desea obtener información de notas en el menú desplegable.
- Presione el botón “enviar”. Si se presenta un error en el envío de los datos se mostrará un mensaje de alertas y el usuario es redirigido al formulario.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.1.2 Consultar Horarios

Se trata de un formulario de selección que permite al alumno consultar el horario de clases de una sección ó de una materia de postgrado. Su procedimiento es sencillo, consistiendo en los siguientes pasos:

- Seleccionar la sección o materia a consultar horario.
- Presione el botón “aceptar.”, para desplegar en pantalla el horario escogido. Si no se ha hecho ninguna selección, se muestra un mensaje de error y el usuario es redirigido al formulario.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.1.3 Programación Académica

Es una herramienta que permite al estudiante consultar la programación académica de la cohorte, módulo o asignatura desde que se programa la entrega de material hasta la presentación de evaluaciones del módulo de programa de postgrado. Los pasos para acceder a esta información se listan a continuación.

- Ingresar el identificador de módulo o asignatura de la cual se desea saber la programación académica.
- Presionar el botón aceptar, para mostrar en pantalla la planificación académica del módulo de postgrado.

- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.1.4 Modificar Datos de Usuarios

Consiste en un formulario de entrada de datos que permite al estudiante modificar la clave de usuario, además de datos de recuperación de clave.

- Ingresar los datos de recuperación de clave.
- Ingresar la clave de usuario nueva.
- Presione el botón “enviar”.
- Para limpiar el formulario, presione el botón “borrar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.2 Nivel de Acceso 2: Profesores

Esta sesión se restringe sólo para el personal académico del área de postgrado. En dicha sesión el usuario puede gestionar las notas de materias que dicta, guardando las notas definitivas en la base de datos, además de realizar consultas a la base de datos y generar reportes como listas de asistencias. A continuación se describen las

funciones que el personal académico puede utilizar en el sistema. En la figura A.4 se despliega la interfaz principal de la sesión personal académico.



Figura A.4- Interfaz principal de usuario nivel 2. Fuente: Autores 2009

A.5.2.1 Cargar Notas

En este formulario los usuarios deben reingresar la clave de usuario para activar la función. En esta función del sistema el usuario carga notas definitivas de una asignatura o módulo, y se genera un acta preliminar de notas que el usuario puede entregar en control de estudio. A continuación se observa los pasos que se siguen en este procedimiento.

- Ingresar clave y nombre de usuario.
- Ingresar el Listado de notas definitivas de la materia o asignatura.
- Generar Acta de Notas preliminar.
- Presionar botón guardar notas.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de profesores.

A.5.2.2 Consultas

Es un conjunto de reportes que se muestran en el navegador Web. Entre las consultas que se realizan dentro de esta sesión tenemos los horarios de clase donde el usuario imparte clases.

El procedimiento para realizar consultas es el que sigue (similar para todos):

- Seleccionar la materia de la cual se desea generar la consulta de horario.
- Una vez seleccionado, se visualizan los datos de horas académicas de la sección seleccionada.
- Presionar el botón “imprimir” para abrir la versión del reporte en Adobe Acrobat Reader.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de profesores.

A.5.2.3 Programación Académica

Es una herramienta que permite al estudiante consultar la programación académica de la cohorte, módulo o asignatura desde que se programa la entrega de material hasta la presentación de evaluaciones del módulo de programa de postgrado. Los pasos para acceder a esta información se listan a continuación.

- Ingresar el identificador de módulo o asignatura de la cual se desea saber la programación académica.
- Presionar el botón aceptar, para mostrar en pantalla la planificación académica del módulo de postgrado.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.
- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.2.4 Modificar Datos de Usuarios

Consiste en un formulario de entrada de datos que permite al estudiante modificar la clave de usuario, además de datos de recuperación de clave.

- Ingresar los datos de recuperación de clave.
- Ingresar la clave de usuario nueva.
- Presione el botón “enviar”.
- Para limpiar el formulario, presione el botón “borrar”.

Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de Estudiantes.

A.5.2.5 Generar Listas Asistencias

Este formulario le permite al personal académico generar un archivo de texto para el control de asistencias.

- Seleccionar la sección de la cual se quiere generar la lista de asistencia.
- Presionar el botón aceptar para crear el archivo, presionar el botón imprimir para obtener la lista de asistencia en físico.
- Para limpiar (resetear) el formulario, presione el botón “limpiar”.

- Para cancelar presionar el botón cancelar, será redirigido a la interfaz principal de profesores.

A.5.3 Nivel de Acceso 3: Personal Administrativo

Sesión dedicada al personal administrativo de postgrado. En esta sesión se manipulan todos los datos relacionados a los programas de postgrado y el personal relacionado a los mismos. En la figura A.5 se puede observar la interfaz principal de esta sesión.



Figura A.5- Interfaz principal de usuario nivel 3. **Fuente:** Autores 2009

A.5.3.1 Programas

Los formularios relacionados a esta función del software recogen la información mínima necesaria para crear archivos OPSU y manipular una información de los programas de postgrado relativamente completa. Aquí se puede realizar todas las operaciones sobre los datos de programas de postgrado exceptuando la eliminación de programas de postgrado.

A.5.3.2 Líneas de Investigación

Las líneas de investigación son un conjunto de investigaciones en un rama de ciencia específica, o sobre un producto, artículo u objeto donde se desarrollen investigaciones científicas sobre él. Al igual que en la administración de programas de postgrado se tienen distintos formularios para manipular los datos de las líneas de investigación teniendo la opción de eliminar una línea de investigación además de las otras operaciones administrativas que se pueden realizar.

A.5.3.3 Grupos de Investigación

Agrupar a profesores de postgrado que estén desarrollando investigaciones sobre una línea de investigación, sobre ellos se pueden realizar todas las operaciones administrativas de datos dentro de esta sesión.

A.5.3.4 Trabajos de Grado

Administra los trabajos de grado que se estén realizando o se hallan culminado en el área de postgrado.

A.5.3.5 Profesores

Desde esta función se puede administrar los datos del personal académico, desde los link siguientes se logra administrar los datos de personal Administrativo, Estudiantes regulares de Postgrado y datos de la institución.

A.5.3.6 Cartelera Virtual

Este link despliega una interfaz donde se publica informaciones referentes a los programas de postgrado.

A.5.3.7 Directorios

Desde aquí se puede acceder a archivos generados en la aplicación Web para ser descargados.

A.5.4 Nivel de Acceso 4: Administrador de sistema

En esta sesión se realizan las distintas configuraciones del sistema SISGAP. En la grafica A.5 se observa las opciones disponibles para esta sesión.

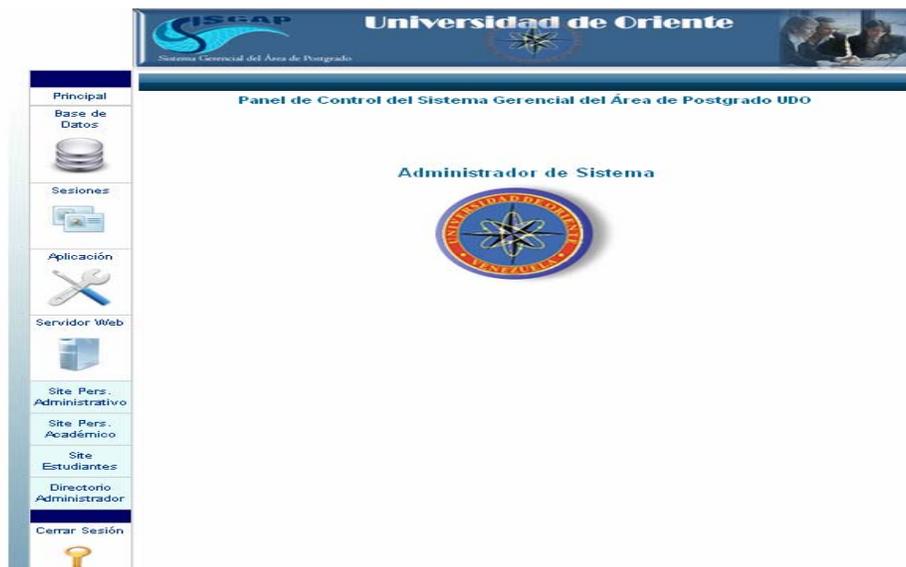


Figura A.6- Interfaz principal de usuario nivel 4. Fuente: Autores 2009

A.5.4.1 Base de Datos

La función base de datos se utiliza para configurar todo lo referente a la base de datos incluso para registrarlas cuando sea instalado el sistema en un servidor Web para así obtener conexión con las bases de datos. Dentro de este link se pueden realiza operaciones de Backup y recuperación de información de las bases de datos, además de realizar operaciones de consulta y correcciones sobre datos de la OPSU.

A.5.4.2 Sesiones

Desde este link se ingresa los usuarios de sistema clasificándolos según sus privilegios, también se configura los privilegios de los usuarios por nivel.

A.5.4.3 Aplicación

Desde aquí se configura el funcionamiento de la aplicación, configurando los enlaces de las páginas y activación de funciones como cargar notas en la sesión de nivel 2 o personal académico.

A.5.4.4 Servidor Web

Permite configurar el funcionamiento del servidor Web, como el tiempo de conexión, ancho de banda de archivos Web y cierre automáticos de sesión.

A.5.4.5 Link's a sesiones de nivel inferior

Permite al administrador de sistema verificar el funcionamiento de las sesiones de nivel de seguridad inferior a 4.

A.6 Observaciones

- Sólo los alumnos pertenecientes al área de Postgrado pueden registrarse como usuarios del sistema.

- Los usuarios de nivel 2 (Personal Administrativo) o 3 (Administrador de Sistema) sólo pueden ser registrados por el operador actual, empleando la administración de usuarios.
- Los enlaces de “agregar” y “eliminar” solicitudes de inscripción sólo permanecerán activos durante el período de preinscripciones. Una vez culminado, ambos enlaces serán deshabilitados.

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

TÍTULO	DESARROLLO DE UN SISTEMA AUTOMATIZADO BAJO ENTORNO WEB PARA EL CONTROL DE LA PROGRAMACIÓN ACADÉMICA EN LA UNIVERSIDAD DE ORIENTE NÚCLEO DE ANZOÁTEGUI
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
DIAZ V., JEAN C.	CVLAC: 15.548.152 E MAIL: jeancdv9@hotmail.com
TIRADO C., NORELKYS DEL C.	CVLAC: 15.789.190 E MAIL: norelkys53@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Coordinación

Postgrado

Estadísticas

OPSU

WEBML

Gerencial ORACLE PHP

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

RESUMEN (ABSTRACT):

En este proyecto se desarrolla un sistema automatizado como soporte de las tareas diarias de postgrado en el área académica de la Universidad de Oriente. El software está desarrollado bajo el lenguaje de programación PHP sumado al manejador de bases de datos Oracle para conformar una herramienta muy potente y segura para el área de postgrado. Las funciones principales de esta aplicación web es administrar todo lo referente a los Programas de Postgrado, líneas de investigación, grupos de investigación, datos y notas de estudiantes, datos de profesores, datos del personal administrativo, preinscripciones, generar los reportes de transacciones, emitir constancias y actas, y además generar los archivos OPSU codificados con el formato que este organismo impone de las diferentes variables estadísticas que se generan en el área de postgrado. Las herramientas necesarias para desarrollar este software son el editor de páginas Web Macromedia Dreamweaver 8.0, el servidor Web Apache 2.0, el módulo PHP 5.0 y el administrador de bases de datos Oracle 8i. En la construcción del sistema (SISGAP) se utilizó el Proceso Unificado de Desarrollo de Software, el Modelado de UML y sus extensiones que son WebML.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Ríos, Francy	ROL	CA	AS X	TU	JU
	CVLAC:	V-12.291.177			
	E_MAIL	francyrios75@gmail.com			
	E_MAIL				
Cortínez N., Claudio A.	ROL	CA	AS	TU	JU X
	CVLAC:	12.155.334			
	E_MAIL	cl_cortinez@cantv.net			
	E_MAIL				
Rodríguez M., Rhonald E.	ROL	CA	AS	TU	JU X
	CVLAC:	14.077.185			
	E_MAIL	rhoen@hotmail.com			
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	10	22
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.sistemasisgap.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 en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente, Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

De acuerdo al artículo 44 del reglamento de trabajo de grado de la Universidad de Oriente.

“Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario”

Díaz v., Jean C.

AUTOR

Tirado C., Norelkys del C.

AUTOR

Rios, Francy

TUTOR

Rodríguez, Rhonald

JURADO

Cortínez, Claudio

JURADO

Jose Luis Bastardo

POR LA SUBCOMISION DE TESIS