



Mon  
003  
R175  
2012

**Universidad Nacional de Ingeniería**  
***Facultad de Ciencia y Sistemas***

**Trabajo Monográfico para optar al Título de:**  
**Ingeniero de sistemas.**

**Título:**

**Diseño de un sistema de registro académico y arancelario automatizado  
en Compulab-Estelí**

**Presentado por:**

**Br. René Josué Ramos Barreda**

**Docentes:**

**Tutor:** Msc. Ing. Luis María Dicovski R.

**Asesor:** Ing. José Manuel Poveda R.

**Septiembre - 2012**

**Managua, Nicaragua.**



## **DEDICATORIA**

Dedicado a Dios, por haber iluminado mi camino y por este regalo que me ha dado, que es la capacidad de aprender y crecer.

A mi familia, mi mamá, Itzel Barreda, mi papá, Hiraldo, por todo el esfuerzo que han realizado para que realice mis estudios, por su confianza, apoyo y todo el amor que siempre me han dado.

A mi esposa Doris Ráudez y a mi hijo por su apoyo incondicional.

## **RESUMEN**

El presente trabajo describe la elaboración de una investigación realizada en el centro de estudio Compulab - Estelí, para la obtención de un sistema de información que ayude a mejorar los procesos que se efectúan en esta institución.

Para realizar esta investigación se hicieron entrevistas y observaciones que dieron pase al análisis de la información, para determinar qué se debía automatizar, seguidamente se leyeron teorías que ayudan a fundamentar el presente documento.

Este sistema de información tiene como objetivo fundamental automatizar el proceso de inscripciones de estudiantes, registro de solvencias y control de notas, además almacena toda esta información en una base de datos la cual también almacena datos personales de los estudiantes y docentes.

Con el sistema se obtendrá mayor rapidez en los procesos de búsquedas y confiabilidad de los datos.

Finalmente el sistema permite el análisis basado en reportes que ayudan a tener un mejor conocimiento de las inscripciones, exámenes y notas.

## ÍNDICE

### DEDICATORIA

### RESUMEN

INTRODUCCIÓN.....	1
PLANTEAMIENTO DEL PROBLEMA .....	3
OBJETIVOS .....	5
JUSTIFICACIÓN .....	6
CAPITULO I      MARCO TEORICO .....	7
1. Sistemas de Información .....	8
1.1 Sistema de información (SI).....	9
1.2 Tipos de sistemas de información .....	9
1.2.1 Sistema de procesamiento de transacciones (TPS).....	9
1.2.2 Sistemas de trabajo de conocimientos (KWS).....	10
1.2.3 Sistemas de automatización de oficinas (OAS).....	10
1.2.4 Sistemas de información gerencial (MIS).....	11
1.2.5 Sistemas de soporte a decisiones (DSS). .....	12
1.2.6 Sistemas de información ejecutiva (EIS).....	13
1.3 Proceso del sistema de información.....	14
1.3.1 Entrada de información.....	14
1.3.2 Almacenamiento de información. ....	14
1.3.3 Procesamiento de información. ....	14
1.3.4 Salida de información. ....	15
1.4 Sistema Automático.....	15
1.4.1 Automatización.....	15
2. Registro académico.....	16
2.1 Sistema de información de registro académico.....	16
2.2 Características del Registro Académico.....	17
2.3 Registro académico y arancelario basado en el centro educativo.	17
2.4 Registro académico basado en la necesidad del docente. ....	17
2.5 Funcionalidad del registro académico .....	18

2.6	Proceso de registro académico .....	18
2.6.1	Matricula.....	18
2.6.2	Actualización de expediente académico. ....	19
2.6.3	Elaboración de informe de notas .....	19
2.6.4	Elaboración de cuadros estadísticos .....	19
3.	Ingeniería Web .....	19
3.1	El proceso de la ingeniería Web .....	19
3.2	Control y garantía de la calidad .....	21
3.3	Control de la configuración.....	21
3.4	La gestión del proceso .....	22
3.5	Diferencia con la ingeniería de software.....	23
4.	Arquitectura web .....	23
4.1	Aplicaciones multinivel .....	23
4.2	Herramientas de Programación .....	26
4.2.1	El lenguaje HTML.....	26
4.2.2	JavaScript .....	27
4.2.3	WAMP .....	28
4.2.4	Apache Web Server .....	29
4.2.5	PHP .....	29
4.2.6	PhpMyAdmin.....	31
4.2.7	MySql .....	31
4.2.8	Macromedia Dreamweaver .....	32
4.3	Estándares de diseño web .....	32
4.3.1	Desarrollar con estándares tiene muchas ventajas .....	33
5.	La Orientación a Objetos, OO.....	35
5.1	¿Qué es un Objeto? .....	35
5.2	¿Qué es una Clase? .....	37
5.3	¿Qué es la Herencia?.....	38
5.4	¿Qué es una Interfaz?.....	39
6.	El Lenguaje Unificado de Modelado, UML.....	40
6.1	Bloques básicos de construcción de UML .....	40

6.1.1	Elementos .....	42
6.1.2	Relaciones.....	45
6.1.3	Diagramas .....	46
6.2	¿Cómo utilizar UML? .....	49
7.	Aspectos metodológicos .....	53
7.1	Proceso de lweb.....	53
7.2	Paradigma espiral orientado a la web (modelo del proceso lweb)	53
7.2.1	Formulación .....	55
7.2.2	Análisis.....	55
7.2.3	Diseño.....	56
7.2.3.1	Diseño Arquitectónico .....	57
7.2.3.2	Diseño de navegación .....	58
7.2.3.3	Diseño de la interfaz .....	59
7.2.4	Pruebas .....	60
8.	Metodología: OOHDM (object oriented hypermedia design methodology) .....	61
8.1	Introducción OOHDM.....	61
8.2	Fases de la metodología OOHDM.....	62
8.2.1	Determinación de Requerimientos .....	63
8.2.2	Diseño Conceptual .....	64
8.2.3	Diseño Navegacional.....	65
8.2.4	Diseño de Interfaz Abstracta .....	68
8.2.5	Implementación .....	69
8.3	VENTAJAS DE OOHDM .....	69
<b>CAPITULO II</b>		
	<b>ANALISIS Y DISEÑO DEL SISTEMA.....</b>	<b>71</b>
1.	Estudios de factibilidad. ....	72
1.1	Factibilidad técnica.....	72
1.2	Factibilidad operacional. ....	75
1.3	Factibilidad económica.....	77
1.3.1	Estimación de Costo del Software .....	79

1.3.1.1	Definición.....	79
1.3.1.2	Metodología .....	80
1.3.1.3	Cálculo de Costos.....	95
2.	Descripción de Procesos de Negocios.....	97
2.1	Definición de la situación Actual.....	97
2.2	Descripción del entorno. ....	98
2.2.1	Proceso de Matricula.....	98
2.2.2	Control de notas .....	101
2.2.3	Elaboración de certificados de notas. ....	102
2.2.4	Creación de Grupos de Clases.....	104
2.2.5	Control de Solvencias.....	106
3.	Análisis y Diseño UML .....	107
3.1	Flujos de Trabajos Persistentes. ....	107
3.2	Análisis de los casos de usos y sus escenarios.....	108
3.2.1	Actores encontrados en el Análisis.....	108
3.2.2	Diagramas de Casos de Uso. ....	109
3.3	Diagramas de Secuencia.....	121
3.4	Diagramas de Colaboración.....	126
3.5	Diagrama de Clases .....	130
3.6	Diagramas de Estados.....	131
3.7	Diagrama de componentes .....	135
3.8	Modelo de datos.....	136
3.8.1	Diccionario de datos. ....	137
3.8.2	Relaciones de la base de datos.....	142
3.9	Diagrama Navegacional.....	143
3.10	Interfaces .....	144
3.10.1	Login.....	144
3.10.2	Menú Resumen .....	144
3.10.3	Menú Grupos .....	145
3.10.3.1	Menú Nuevo Grupo .....	145
3.10.3.2	Menú Borrar Grupo .....	146

3.10.4	Menú Estudiante.....	146
3.10.4.1	Menú Nuevo Estudiante.....	147
3.10.4.2	Menú Editar Estudiante.....	147
3.10.4.3	Menú Eliminar Estudiante.....	148
3.10.5	Menú Docente .....	149
3.10.5.1	Menú Inscribir Docente.....	149
3.10.5.2	Menú Eliminar Docente.....	150
3.10.6	Menú Clases.....	150
3.10.6.1	Menú Nueva Clase.....	151
3.10.6.2	Menú Borrar una Clase .....	151
3.10.7	Menú Horarios .....	152
3.10.7.1	Menú Nuevo Horario .....	152
3.10.7.2	Menú Borrar Horario .....	153
3.10.8	Menú Notas .....	153
3.10.9	Menú Reportes.....	154
4.	Pruebas al sistema .....	155
5.	Implantación del sistema .....	156
	CONCLUSIONES.....	158
	RECOMENDACIONES .....	159
	BIBLIOGRAFÍA.....	160

## ÍNDICE DE TABLAS

Tabla 1 Matriz Causas y Consecuencias .....	2
Tabla 2 Herramientas de Programación.....	26
Tabla 3: Elementos UML.....	44
Tabla 4: Relaciones UML.....	45
Tabla 5: Diagramas UML .....	48
Tabla 6: Especificaciones de los equipos .....	72
Tabla 7: Habilidades mínimas para usuarios del sistema. ....	76
Tabla 8: Costos de Sistema .....	78
Tabla 9: Factores de Escala.....	81
Tabla 10: Factores de Escala 2 .....	82
Tabla 11: Factores de Costo .....	86
Tabla 12: Factores de Costo 2.....	92
Tabla 13: Tabla Estudiante.....	137
Tabla 14: Tabla Exámenes.....	137
Tabla 15: Tabla Grupos.....	138
Tabla 16: Tabla Curso .....	138
Tabla 17: Tabla Horarios.....	139
Tabla 18: Tabla Materias .....	139
Tabla 19: Tabla Docentes.....	140
Tabla 20: Tabla Notas .....	140
Tabla 21: Tabla Cartera .....	141
Tabla 22: Tabla Administradores.....	141
Tabla 23: Relaciones de la base de datos .....	142

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Arquitectura Multinivel. ....	24
Ilustración 2: Arquitectura Web de tres niveles. ....	25
Ilustración 3: Funcionamiento de PHP .....	30
Ilustración 4: Objetos comunicándose. ....	36
Ilustración 5: La herencia. ....	39
Ilustración 6: El modelo del proceso IWeb .....	53
Ilustración 7: Modelo Conceptual para una Tienda de CD's. ....	65
Ilustración 8: Esquema básico navegacional para my.yahoo.com. ....	66
Ilustración 9: Diagrama de Configuración para los nodos ADV.....	68
Ilustración 10: Diseño de red lógico .....	74
Ilustración 11: Diseño de red físico. ....	75
Ilustración 12: Diagrama de Actividad – Proceso de Matricula.....	100
Ilustración 13: Diagrama de Actividad – Control de Notas .....	102
Ilustración 14: Diagrama de Actividad – Certificado de Notas .....	103
Ilustración 15: Diagrama de Actividades – Grupos de Clases.....	105
Ilustración 16: Diagrama de Actividad – Solvencias.....	106
Ilustración 17: Diagrama de Casos de uso – Grupos. ....	109
Ilustración 18: Diagrama de Caso de Uso - Clases.....	111
Ilustración 19: Diagrama de Casos de Uso - Docentes .....	113
Ilustración 20: Diagrama de Casos de Uso - Estudiantes .....	115
Ilustración 21: Diagrama de Casos de Usos – Notas.....	118
Ilustración 22: Diagrama de Casos de Uso – Informes.....	119
Ilustración 23: Diagrama de Secuencia – Casos de uso – Grupos. ....	121
Ilustración 24: Diagrama de Secuencia – Caso de Uso - Clases.....	122
Ilustración 25: Diagrama de Secuencia 3 – Casos de Uso - Docentes.....	123
Ilustración 26: Diagrama de Secuencia – Casos de Uso - Estudiantes .....	124
Ilustración 27: Diagrama de Secuencia – Casos de Uso – Notas.....	125
Ilustración 28: Diagrama de Secuencia - Caso de Uso - Informes .....	125
Ilustración 29: Diagramas de Colaboración – Grupo.....	126
Ilustración 30: Diagrama de Colaboración – Clase .....	126
Ilustración 31: Diagrama de Colaboración – Docente.....	127
Ilustración 32: Diagrama de Colaboración – Estudiante.....	128
Ilustración 33: Diagrama de Colaboración – Notas .....	129
Ilustración 34: Diagrama de Colaboración – Informes .....	129
Ilustración 35: Diagrama de Clases .....	130
Ilustración 36: Diagramas de Estado - Estudiante.....	131
Ilustración 37: Diagramas de Estado - Notas .....	132
Ilustración 38: Diagramas de Estado - Grupo.....	132
Ilustración 39: Diagramas de Estado - Cursos .....	133
Ilustración 40: Diagramas de Estado - Materia .....	133

Ilustración 41: Diagramas de Estado - Docente.....	134
Ilustración 42: Diagrama de componentes.....	135
Ilustración 43: Modelo Entidad – Relación UML .....	136
Ilustración 44: Diagrama Navegacional .....	143
Ilustración 45: Login.....	144
Ilustración 46: Menú Resumen .....	144
Ilustración 47: Menú – Grupos.....	145
Ilustración 48: Menú Nuevo Grupo.....	145
Ilustración 49: Menú – Borrar Grupo.....	146
Ilustración 50: Menú Estudiante.....	146
Ilustración 51: Menú Nuevo Estudiante .....	147
Ilustración 52: Menú Editar Estudiante .....	147
Ilustración 53: Menú Eliminar Estudiante .....	148
Ilustración 54: Menú Eliminar Estudiante 2.....	148
Ilustración 55: Menú Docente.....	149
Ilustración 56: Menú Inscribir Docente .....	149
Ilustración 57: Menú Eliminar Docente .....	150
Ilustración 58: Menú Clases .....	150
Ilustración 59: Menú Nueva Clase.....	151
Ilustración 60: Menú Borrar una Clase .....	151
Ilustración 61: Menú Horarios .....	152
Ilustración 62: Menú Nuevo Horario.....	152
Ilustración 63: Menú Borrar Horario .....	153
Ilustración 64: Menú Notas.....	153
Ilustración 65: Menú Reportes .....	154

## **Anexos**

<b>Anexos 1 Entrevista y guía de observación .....</b>	<b>2</b>
<b>Anexos 2 Guía de Observación.....</b>	<b>5</b>
<b>Anexos 3 Glosario .....</b>	<b>7</b>

## **INTRODUCCIÓN**

Compulab-Estelí es una institución de educación técnica que ayuda a jóvenes y adultos a alcanzar un mejor nivel de vida, a través de una educación integral en diferentes carreras técnicas y ligadas a las tecnologías de la informática.

Toda esta labor formativa se realiza dentro del marco técnico del INATEC.

Compulab tiene 3 sucursales a nivel nacional, y cuenta en total con casi 800 estudiantes en sus 3 centros, siendo 250 estudiantes específicamente en la sucursal de Estelí.

Este estudio aborda una de las problemáticas que afectan a instituciones educativas a nivel nacional, lo cual es involucrar el avance de la tecnología en sus procesos administrativos.

En Compulab – Estelí el control de registro académico y arancelario es manual, se observa que la mayoría de los procedimientos tienen inconvenientes, lo que hace al trabajo pesado y muchas veces deficiente por la cantidad de estudiantes y procedimientos manuales que se utilizan.

Entre ellos se pueden destacar: pérdida o duplicación de información debido a la transcripción manual y /o forma de almacenamiento, retardo en el comienzo de las clases y actividades regulares.

Por lo tanto, con el fin de mejorar los procesos de registro académico y arancelario se propone desarrollar un sistema de información que permita llevar un control preciso que involucran los departamentos de administración, secretaria y registro académico. Además el presente proyecto está enfocado en el desarrollo de una aplicación Web utilizando PHP y MySql como herramientas para lograr la automatización de los procesos en un 90%.

### Matriz Causas y Consecuencias

<b>Causas</b>	<b>Consecuencias</b>
Falta de recursos técnicos	Los trabajadores no explotan los recursos técnicos que tienen a su disposición por ignorancia o miedo.
Poco conocimiento de la automatización de los recursos tecnológicos	Se mira a los recursos tecnológicos como enemigos, y no como formas de hacer un trabajo rápido y eficiente.
Poca capacitación del personal	Inestabilidad laboral, poco interés de invertir en la capacitación o contratación de personas aptas para estos cargos.
Poco interés de la institución en buscar alternativas administrativas con el uso de la tecnología.	La institución se queda embancada en procesos de administración y registros rutinarios.
Conformidad en el uso de procesos de registro académicos rutinarios y lentos.	Perdida o duplicación de la información
Creencias equivocadas en el hecho de que este tipo de tecnología es cara y conllevan gastos innecesarios.	Subutilización de recursos que se tienen a la mano y al contrario, pérdida de materiales.

**Tabla 1 Matriz Causas y Consecuencias**

## **PLANTEAMIENTO DEL PROBLEMA**

En Compulab Estelí, el registro académico y manejo de pagos y aranceles se ha venido realizando de forma manual a través de los años.

Lo que se ha convertido en un proceso rutinario y atenuante, acarreado problemas de administración y control en el mismo por la duplicación o pérdida de información necesaria.

Entre las causas encontramos el mal uso de los recursos físicos y recursos humanos. También influye el poco interés de los propietarios de buscar alternativas de solución basadas en el uso de la tecnología por la creencia equivocada de que la implementación y cambio de estos procesos resulta ser costosa.

En Compulab – Estelí se demuestra la ineficacia a través de la observación de los procesos que se realizan en forma manual y que originan un alto grado de desorganización, lentitud en la realización de reportes, duplicación de inscripciones y demora en la entrega de notas, que a su vez ocasiona agotamiento del personal.

El Centro Educativo actualmente cuenta con un número de matrículas de superior a los 250 estudiantes, lo que implica una gran cantidad de trabajo para llevarse de forma manual, influyendo de manera directa en provocar el insuficiente e incompetente desarrollo del registro, atrasando a su vez las actividades de otros departamentos dependientes del proceso tales como el departamento de administración y de registro académico.

En todo este proceso, el planificar y controlar la inscripción por curso, docentes y pagos es lento y desorganizado, y requiere de tiempo para realizarlo en ocasiones de hasta una semana para tener todo en orden.

El proceso de inscripción se lleva de forma manual, en este proceso verifica la totalidad de los documentos, se recolectan los datos a través de una ficha denominada libro de vida con el fin de crear expedientes del estudiante y se la anexan los documentos exigidos (partida de nacimiento, certificado de ciclo básico, cédula de identidad, 2 fotografías), además el registro se encuentra abierto por cada curso que se promociona a lo largo del año.

Este tipo de proceso en Compulab ocasiona pérdida de recursos y materiales innecesarios, equivocaciones en la asignación de cursos y docentes, ya que muchas veces los estudiantes son mal informados sobre sus horarios y tutores.

## **OBJETIVOS**

### **1. Objetivo General**

- ✓ Desarrollar un Prototipo de Sistema de Registro Académico y pago de aranceles basado en las necesidades de Compulab-Estelí 2012.

### **2. Objetivos específicos**

- ✓ Realizar el análisis y diseño del prototipo del sistema utilizando la metodología hipermedia orientada a objetos (OOHDM).
- ✓ Modelar el sistema utilizando UML.
- ✓ Automatizar el proceso de matrícula y el registro de notas.
- ✓ Implementar el sistema desarrollado en Compulab-Estelí.

## **JUSTIFICACIÓN**

La información se ha vuelto clave dentro de las empresas, es un elemento no sólo necesario sino también estratégicamente definitorio. Se entiende que la información no es un activo más de la empresa sino un recurso estratégico. El volumen de información dentro de las empresas aumenta a nivel exponencial. Por ello, se hace necesario mejorar las herramientas existentes para procesar todo este cúmulo de datos.

El presente proyecto permitirá a Compulab Estelí llevar un control estricto del registro académico del estudiante, la información del pago de mensualidad y las notas actualizadas de cada estudiante por curso y los docentes que lo imparten.

Además simplificará y agilizará el trabajo, trayendo al personal nuevas formas de organización y trabajo.

Vale la pena aclarar que debido a la forma como se desarrolló el sistema será fácilmente extensible; se podrá acoplar módulos implementados en el futuro para ampliar la funcionalidad de este mismo.

Y como punto más relevante permitirá conocer a cabalidad la información académica del centro lo que ayudará a la toma de decisiones correctas.

# **CAPITULO I**

## **MARCO TEORICO**

## **MARCO TEORICO**

Los grandes avances en la informática han hecho posible que se desarrollen productos, sistemas y aplicaciones destinadas a la automatización de los procesos cotidianos en las empresas, instituciones, escuelas, etc.

Para el desarrollo de este sistema fue necesario contar con las teorías que nos permitieran fundamentar los procesos desarrollados. Además se abordan teorías sobre los sistemas de información y bases de datos que permitieron el desarrollo del sistema de información.

### **1. Sistemas de Información**

#### **Datos<sup>1</sup>**

Símbolos que describen condiciones, hechos, situaciones o valores y se caracterizan por no contener ninguna información. Un dato puede significar un número, una letra, un signo ortográfico o cualquier símbolo que represente una cantidad, una medida, una o palabra o una descripción.

#### **Información<sup>2</sup>**

Es un mensaje con un contenido determinado emitido por una persona hacia otra y, como tal, representa un papel primordial en el proceso de la comunicación, a la vez que posee una evidente función social. A diferencia de los datos, la información tiene significado para quien la recibe, por eso, los seres humanos siempre hemos tenido la necesidad de cambiar entre sí información que luego se transforman en acciones.

---

<sup>1</sup> Vanessa Ojeda Carrión, SISTEMAS DE INFORMACIÓN, Editorial de la Universidad Técnica Particular de Loja, 2006 consulta 1 Septiembre 2010.

<sup>2</sup> IDEM Pag. 15

## **Sistema<sup>3</sup>**

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia. Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software).

### **1.1 Sistema de información (SI)**

Es un conjunto de componentes interrelacionados entre sí, que reúne, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización.<sup>4</sup>

Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad, es decir, sistema, automatizado o manual, que engloba a personas, máquinas y/o métodos organizados para recopilar, procesar, transmitir datos que representan información<sup>5</sup>.

### **1.2 Tipos de sistemas de información<sup>6</sup>**

#### **1.2.1 Sistema de procesamiento de transacciones (TPS)**

Son los sistemas de negocios básicos que sirven a nivel operativo de la organización, estos sistemas computarizados efectúan y registran las transacciones diarias rutinarias, necesarias para la marcha del negocio, un

---

<sup>3</sup> Recuperado de <http://www.alegsa.com.ar/Diccionario/sistema.php>, consulta 18 Agosto 2010

<sup>4</sup> IDEM "Sistema de información Guía Didáctica" Pag. 16

<sup>5</sup> Raymond McLeod, Jr.: "Sistemas de información gerencial". Prentice Hall, 2000 [7ª edición] ISBN 970-17-0255-7. Consulta 18 Agosto 2010

<sup>6</sup> IDEM "Sistema de información GuíaDidactica" Pag. 26 a 29

ejemplo básico son los sistemas bancarios que permiten efectuar el negocio bancario como registro de depósitos, retiros, pago de cheques, etc.

*Características:*

- Nivel operacional, con tareas, recursos y metas preferidos y altamente estructurados
- Entradas: transacciones, eventos
- Procesamiento: clasificación, listado, actualización
- Salidas: informes detallados, listas, resúmenes
- Usuarios: personal de operaciones

### **1.2.2 Sistemas de trabajo de conocimientos (KWS)**

Son los que satisfacen las necesidades de información en el nivel de conocimientos de la organización. Los sistemas de trabajo de conocimientos ayudan a los trabajadores de conocimientos, mientras que los sistemas de automatización de oficinas ayudan sobre todo a los trabajadores de datos.

*Características:*

- Nivel de conocimiento
- Entradas: especificaciones de diseño, bases de conocimiento
- Procesamiento: modelado, simulaciones
- Salidas: modelos, gráficos, etc.
- Usuarios: staff técnico de la organización

### **1.2.3 Sistemas de automatización de oficinas (OAS)**

Son sistemas computarizados que utilizan herramientas como el procesador de palabra, el correo electrónico y sistemas de programación que han sido diseñados para incrementar la productividad de los empleados que manejan información en la oficina.

*Características:*

- Nivel de conocimiento
- Entradas: documentos, calendarios
- Procesamiento: gestión documental, calendarios y agendas, comunicación
- Salidas: documentos, planificaciones, correo
- Usuarios: administrativos, oficinistas, contables, operadores, etc.

#### **1.2.4 Sistemas de información gerencial (MIS)**

Sirven al nivel de administración de una organización, proporcionando a los administradores informes y, en algunos casos, acceso en línea a los registros de desempeño actuales e históricos de la organización, por ejemplo la “hoja de gerencia” en las organizaciones bancarias (que son las que resumen el avance periódico del banco comparando las captaciones, colocaciones, cartera vencida, abonos anticipados, etc.).

*Características:*

- Nivel de gestión
- Entradas: resumen de datos de transacciones, gran volumen de datos
- Procesamiento: generación de informes, modelos simples
- Salidas: informes sumarios, rutinarios o extraordinarios
- Usuarios: directivos intermedios
- Procesan datos obtenidos de los TPS: datos del pasado y del presente
- Orientación interna
- Decisiones: estructuradas y semi-estructurados
- Poco flexibles

### **1.2.5 Sistemas de soporte a decisiones (DSS).**

Sirven al nivel de administración de la organización. Los DSS ayudan a los administradores a tomar decisiones que son semi-estructuradas, únicas o que cambian rápidamente, y que no es fácil especificar por adelantado. Los DSS tienen que responder con la suficiente rapidez como para ejecutarse varias veces al día, cada vez que cambian las condiciones. Aunque los DSS utilizan información interna de los TPS y MIS, a menudo incluyen información de fuentes externas, como los precios actuales de ciertas acciones o precios de productos de competidores.

#### *Características:*

- Nivel de gestión
- Entradas: Bajo volumen de datos o grandes bases de datos optimizadas para análisis de datos, modelos analíticos y herramientas de análisis de datos
- Procesamiento: interactivos, simulaciones, análisis
- Salidas: informes especiales, análisis de decisiones, respuestas a preguntas
- Usuarios: profesionales, staff, etc.
- Ofrecen a los usuarios flexibilidad, adaptabilidad y rápida respuesta
- Operan con poca o ninguna asistencia de informáticos
- Dan soporte a decisiones y problemas cuyas soluciones no se pueden especificar de antemano.
- Utilizan sofisticadas técnicas estadísticas y modelos matemáticos para el análisis de datos.

### **1.2.6 Sistemas de información ejecutiva (EIS)**

Los administradores de nivel superior usan una categoría de sistemas de información llamados sistemas de apoyo a ejecutivos (ESS) para tomar decisiones. Los ESS sirven al nivel estratégico de la organización, apoyan decisiones no estructuradas y crean un entorno de computación y comunicaciones generalizado, en lugar de proporcionar una aplicación fija o una capacidad específica. Los ESS se diseñan de forma tal que puedan incorporar datos de sucesos externos, como nuevas leyes fiscales o nuevos competidores y también obtienen información resumida de los MIS y DSS internos. Estos sistemas filtran, comprimen y rastrean datos cruciales, haciendo hincapié en la reducción del tiempo y esfuerzo que se requieren para obtener información útil para los ejecutivos.

#### *Características:*

- Nivel estratégico
- Entradas: datos internos y externos
- Procesamiento: gráficos, simulaciones, interactivos
- Salidas: Proyecciones, pronósticos, respuestas a preguntas
- Usuarios: alta dirección
- Objetivo: acomodar las capacidades y funcionamiento de la empresa a los cambios del entorno presente y futuros.

### **1.3 Proceso del sistema de información<sup>7</sup>**

#### **1.3.1 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. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

#### **1.3.2 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. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

#### **1.3.3 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

---

<sup>7</sup> PARK 86 Parker, Donn, B. Rules of Ethics in Information Processing. Communication of ACM 11, 3 (1986) 198-201

que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

#### **1.3.4 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, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o módulo. En este caso, también existe una interface automática de salida.

### **1.4 Sistema Automático.<sup>8</sup>**

Efectúa y controla las secuencias de operaciones sin la ayuda de la actividad humana; dichos sistemas se encuentran dispersos en varios campos: industrias, producción, servicios públicos, electrodomésticos, etc.

Los sistemas automatizados para las oficinas dan soporte a los trabajadores de datos, por lo general, no crean nuevo conocimiento sino que usan esta información para analizarla y transforman datos, o para manejarla de alguna forma y luego compartirla y luego compartirla con toda la organización.

#### **1.4.1 Automatización**

Surge con el objetivo de utilizar la capacidad de las máquinas para llevar a cabo determinadas tareas que anteriormente eran realizadas por los seres humanos.<sup>9</sup>

---

<sup>8</sup> Kendall. 1997. Análisis y Diseño de Sistemas, Prentice Hall Consultado 13 Agosto 2010

<sup>9</sup> IDEM "Análisis y Diseño de Sistemas"

## 2. Registro académico

### Registro

Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos. Puede estar en la propia memoria central o en unidades de memoria de acceso rápido.<sup>10</sup>

### 2.1 Sistema de información de registro académico

El sistema de registro académico constituye un programa integrado de diseño y orientación académica que agiliza y promueve el mejoramiento de los procesos educativos mediante un sistema de información que integra el trabajo de planeación y docencia, con las actividades de aprendizaje y evaluación.<sup>11</sup>

Es un sistema de información de registro académico permite manejar la información de los estudiantes desde que se inscriben hasta que se gradúan, realizar los procesos de gestión de la programación académica, pre matrícula, matrícula, adiciones y cancelaciones, registro de calificaciones, procesos para grados, certificados, homologaciones y transferencias de los estudiantes y producir diversos informes estadísticos en forma de texto y gráficamente sobre los procesos propios de registro académico.

---

<sup>10</sup> <http://www.mastermagazine.info/termino/6510.php> Consultado 16 Agosto 2010

<sup>11</sup> ANDINASOFT - SOFTWARE & HARDWARE 2008 – 2010,  
[http://andinasoftcolombia.net/portal/index.php?option=com\\_content&task=view&id=2&Itemid=1](http://andinasoftcolombia.net/portal/index.php?option=com_content&task=view&id=2&Itemid=1)

El Registro Académico es una unidad operativa encargada de coordinar el desarrollo y la ejecución de los programas académicos relacionados con los procesos de inscripción, selección, admisión y matrícula de los aspirantes que ingresan por primera vez a una de las carreras ofrecidas; así como de cualquier estudiante regular, o de Educación Especial.<sup>12</sup>

## **2.2 Características del Registro Académico.**

Al no encontrar una definición concreta en internet o en otra fuente bibliografía hemos decidido crear una clasificación que se ajuste a la necesidad de nuestro marco teórico.

## **2.3 Registro académico y arancelario basado en el centro educativo.**

Sistema de registro académico en el que refleje los pagos hechos por estudiante, ya sean mensuales o trimestrales y de que curso o en qué nivel, además de brindar toda la información académica del estudiante.

## **2.4 Registro académico basado en la necesidad del docente.**

Los logros de cada estudiante por asignatura son indicados en forma clara y facilita determinar el progreso del estudiante en sus estudios a la vez que da indicaciones apropiadas a los profesores para el mejor desempeño docente encaminado a orientar a sus estudiantes.

---

<sup>12</sup> Universidad de Manizales (2009), [www.mngr@umanizales.edu.co](http://www.mngr@umanizales.edu.co), Obtenido de <http://www.umanizales.edu.co/services/resena.htm>

## 2.5 Funcionalidad del registro académico

- ✓ Automatizar los métodos y organización en el sistema de información.
- ✓ Establecer seguimientos de procedimientos organizativos.
- ✓ Independencia en el método y flujo de trabajo de los ejecutores.
- ✓ Optimizar tiempo de respuesta en solicitar la información.
- ✓ Cuantificar la información en forma estadística.<sup>13</sup>

## 2.6 Proceso de registro académico

Hemos decido explicar el proceso de registro académico según las necesidades de la institución y los pasos que se siguen en el registro manual, dilucidando la mejor opción para el sistema de información.

### 2.6.1 Matricula

Acción de inscribir a los estudiantes en listas para ingresar a los cursos. Se verifican sus datos personales y requisitos de educación media.

Formato donde se llenan datos generales y académicos para hacer constar que el estudiante está matriculado oficialmente en el curso correspondiente.

---

<sup>13</sup> [http://sekipedia.sek.net/sekipedia/images/1/1c/Brochure\\_U%2B.pdf](http://sekipedia.sek.net/sekipedia/images/1/1c/Brochure_U%2B.pdf). Consulta: 24-08-2010 Consultado 11 Agosto 2010

## **2.6.2 Actualización de expediente académico.**

Consiste en el llenado de notas, en los formatos que contienen el plan de estudio de cada curso.

## **2.6.3 Elaboración de informe de notas**

Este proceso se lleva a cabo con los resultados de las calificaciones obtenidas de los estudiantes según cada curso.

## **2.6.4 Elaboración de cuadros estadísticos**

Proceso para preparar y presentar en forma resumida los diferentes indicadores e información más importante que es necesaria para la toma de decisión.

# **3. Ingeniería Web**

Es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas *Web* de alta calidad. Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar qué tipo de proceso es más adecuado en función de las características del mismo<sup>14</sup>.

## **3.1 El proceso de la ingeniería Web**

Características como inmediatez, evolución y crecimiento continuos, nos llevan a un proceso incremental y evolutivo, que permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita.

---

<sup>14</sup> Murugesan et al., promotores iniciales del establecimiento de la *Ingeniería Web* como nueva disciplina, dan la siguiente definición: "*Web Engineering is the establishment and use of sound scientific, engineering and management principles and disciplined and systematic approaches to the successful development, deployment and maintenance of high quality Webbased systems and applications.*" (Murugesan, 3)

Según Pressman<sup>15</sup>, las actividades que formarían parte del marco de trabajo incluirían las tareas abajo enumeradas. Dichas tareas serían aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma.

Las actividades que forman parte del proceso son:

- ✓ Formulación
- ✓ Planificación análisis
- ✓ Modelización
- ✓ Generación de páginas
- ✓ Test y evaluación del cliente.

La **Formulación** identifica objetivos y establece el alcance de la primera entrega.

La **Planificación** genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.

El **Análisis** especifica los requerimientos e identifica el contenido.

La **Modelización** se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación.

La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.

---

<sup>15</sup> Pressman Roger, Ingeniería del software. Un enfoque práctico, Madrid, McGraw-Hill / interamericana de España, Abril 2012

En la **Generación de páginas** se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación<sup>16</sup>.

### 3.2 Control y garantía de la calidad

Una de las tareas colaterales que forman parte del proceso es el Control y Garantía de la Calidad (CGC). Todas las actividades CGC de la ingeniería software tradicional como son: establecimiento y supervisión de estándares, revisiones técnicas formales, análisis, seguimiento y registro de informes, etc., son igualmente aplicables a la *Ingeniería Web*.

Sin embargo, en la *Web* toman especial relevancia para valorar la calidad aspectos como: Usabilidad, Funcionabilidad, Fiabilidad, Seguridad, Eficiencia y Mantenibilidad<sup>17</sup>.

### 3.3 Control de la configuración

Establecer mecanismos adecuados de control de la configuración para la

*Ingeniería Web* es uno de los mayores desafíos a los que esta nueva disciplina se enfrenta. La *Web* tiene características únicas que demandan estrategias y herramientas nuevas. Hay cuatro aspectos importantes a tener en cuenta en el desarrollo de tácticas de control de la configuración para la *Web*<sup>18</sup>:

---

<sup>16</sup> Pressman Roger, *Ingeniería del software. Un enfoque práctico*, Madrid, McGraw-Hill / interamericana de España, página 813.

<sup>17</sup> L. Olsina, G. Lafuente, G. Rossi. "Specifying Quality Characteristics and Attributes for Websites." *Lecture Notes in Computer Science 2016* Springer, 2001.

<sup>18</sup> S. Dart, "Containing the Web Crisis Using Configuration Management," Proc. 1st ICSE Workshop on Web Engineering, ACM, Los Angeles, May 1999.

*Contenido:* Considerando la dinamicidad con la que el contenido se genera, es tarea compleja organizar racionalmente los objetos que forman la configuración y establecer mecanismos de control.

*Personal:* Cualquiera realiza cambios. Hay mucho personal no especializado que no reconoce la importancia que tiene el control del cambio.

*Escalabilidad:* Es común encontrar aplicaciones que de un día para otro crecen considerablemente. Sin embargo, las técnicas de control no escalan de forma adecuada.

*Política:* ¿Quién posee la información? ¿Quién asume la responsabilidad y coste de mantenerla?

### **3.4 La gestión del proceso**

En un proceso tan rápido como es el proceso de *Ingeniería Web*, donde los tiempos de desarrollo y los ciclos de vida de los productos son tan cortos, ¿merece la pena el esfuerzo requerido por la gestión? La respuesta es que dada su complejidad es imprescindible.

Entre los aspectos que añaden dificultad a la gestión destacamos:

- Alto porcentaje de contratación a terceros,
- El desarrollo incluye una gran variedad de personal técnico y no técnico trabajando en paralelo,
- El equipo de desarrollo debe dominar aspectos tan variados como, software basado en componentes, redes, diseño de arquitectura y navegación, diseño gráfico y de interfaces, lenguajes y estándares en Internet, test de aplicaciones *Web*, etc., lo que hace que el proceso de búsqueda y contratación de personal sea arduo.

### 3.5 Diferencia con la ingeniería de software

- ✓ A modo de breve resumen enumeramos las siguientes diferencias:
- ✓ *Confluencia de disciplinas:* Sistemas de Información, Ingeniería Software y Diseño
- ✓ Gráfico que requiere equipos multidisciplinares y polivalentes.
- ✓ *Ciclos de vida y tiempo de desarrollo muy cortos.*
- ✓ *Cambio continuo:* Necesidad de soluciones que permitan flexibilidad y adaptación conforme el proyecto cambia.
- ✓ *Requisitos fuertes de Seguridad, Rendimiento y Usabilidad.*

## 4. Arquitectura web

### 4.1 Aplicaciones multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores.

Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

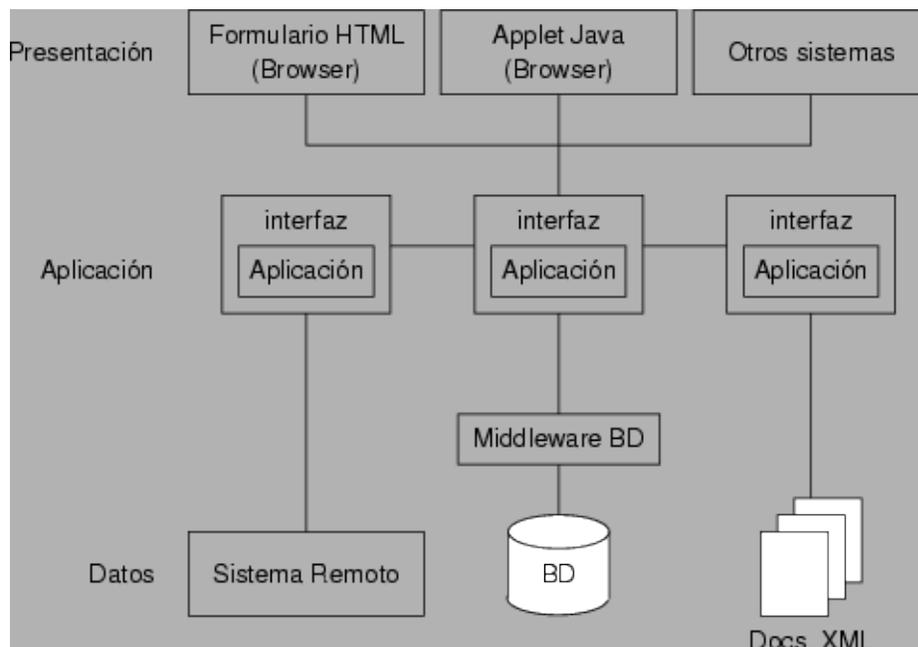
Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados.

La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

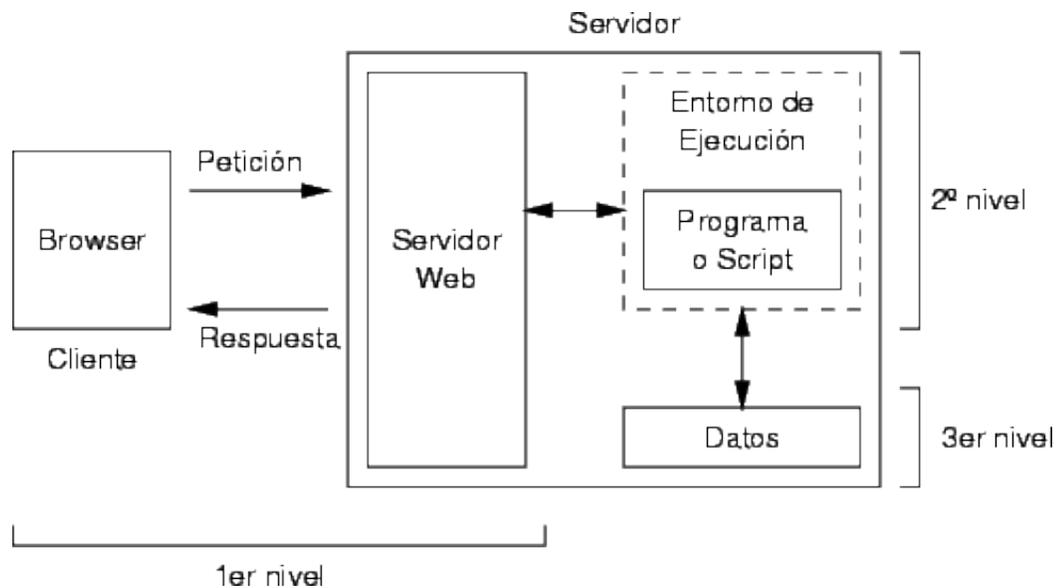
Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola.



**Ilustración 1: Arquitectura Multinivel.**

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles (véase la Figura 2.). El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



**Ilustración 2: Arquitectura Web de tres niveles.**

Lamentablemente, el uso de toda esta tecnología pasa por el dominio de técnicas de programación y de acceso a bases de datos, condición esta que no se puede presuponer en un curso de divulgación como éste. Así, nos vamos a restringir al uso de herramientas básicas a la hora de la construcción de un portal docente.

## 4.2 Herramientas de Programación

<b>Nivel</b>		<b>Lenguaje</b>
<b>Cliente</b>		<b>HTML, JAVASCRIPT</b>
<b>Servidor</b>	<b>Servidor</b>	<b>Apache</b>
	<b>Lenguajes de Scripts</b>	<b>PHP</b>
<b>Servidor de Datos</b>	<b>Base de datos</b>	<b>MySql</b>

**Tabla 2 Herramientas de Programación**

### 4.2.1 El lenguaje HTML<sup>19</sup>

Este lenguaje estructura documentos, la mayoría de los documentos tienen estructuras comunes (títulos, párrafos, listas...) que van a ser definidas por este lenguaje mediante tags. Cualquier cosa que no sea una tag es parte del documento mismo.

Este lenguaje no describe la apariencia del diseño de un documento sino que ofrece a cada plataforma que le dé formato según su capacidad y la de su navegador (tamaño de la pantalla, fuentes que tiene instaladas...). Por ello y para no frustrarnos, no debemos diseñar los documentos basándonos en como lucen en nuestro navegador sino que debemos centrarnos en proporcionar un contenido claro y bien estructurado que resulte fácil de leer y entender.

<sup>19</sup> <http://www.webestilo.com/html/cap2a.phtml>

## **Ventajas de HTML**

HTML tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación web y son:

- Su compatibilidad y su facilidad de aprendizaje debido al reducido número de tags que usa.

### **4.2.2 JavaScript<sup>20</sup>**

JavaScript es un lenguaje interpretado, al igual que VisualBasic, Perl, TCL... sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, usado principalmente en "clientes web", ya que son los navegadores que utilizamos para viajar por ella los que interpretan (y por tanto ejecutan) los programas escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que incorporan el código fuente de un programa, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

Los programas en JavaScript no son la primera forma que conoce la Web para transformar información, dado que el uso de CGIs está ampliamente difundido. La diferencia básica que existe entre un programa CGI y uno escrito en JavaScript es que el CGI se ejecuta en el servidor de páginas Web mientras que el programa en

Javascript se ejecuta en el cliente (es decir, en el navegador). Por regla general, el CGI necesita unos datos de entrada (que normalmente se proporcionan mediante un formulario), los procesa y emite un resultado en forma de documento HTML.

Esto implica tres transacciones en la red.

La primera carga la página del formulario, la segunda envía los datos al servidor, y la tercera recibe la nueva página que ha generado el CGI. Por el contrario, los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin

---

<sup>20</sup> [http://geneura.ugr.es/~victor/cursillos/curso\\_javascript\\_basico/jsb\\_intro.html](http://geneura.ugr.es/~victor/cursillos/curso_javascript_basico/jsb_intro.html)

necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados.

### ***Características de JavaScript***

Las dos principales características de JavaScript son:

- Es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones),
- JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas XWindows).

Esto implica que gran parte de la programación en JavaScript se centra en describir objetos (con sus variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.

### **4.2.3 WAMP<sup>21</sup>**

Es la abreviación de Windows, Apache, MySQL y PHP y es un proyecto desarrollado por franceses. Este paquete instala la versión 2.2 de Apache, PHP5,

MySQL 5.0.27, PhpMyAdmin y SQLitemanager en tu computadora.

Una de las ventajas de este paquete es la ofrecer la versión de PHP5 en el instalador y sus constantes actualizaciones. La instalación es bastante sencilla y destaca que los servicios de apache y de MySQL se instalarán en Windows bajo los nombres de wampapache y wampmysqld para no interferir con otras instalaciones previas de estos servidores.

---

<sup>21</sup> <http://www.en.wampserver.com/presentation.php>

#### 4.2.4 Apache Web Server

Es un servidor web libre, es decir, el encargado de construir y devolver las páginas web que solicitan los navegadores. Su nombre procede de "a patchy server", por ser una versión "parcheada" en 1995 de uno de los primeros servidores web, el NCSA HTTPD, y actualmente corre en muy diversas plataformas (Unices, Windows, etc.). Debido a su licencia libre pero no copyleft, existen también versiones propietarias de Apache, aunque es desarrollado y mantenido por la comunidad del software libre a través de la Fundación Apache.

#### 4.2.5 PHP<sup>22</sup>

(Pre-procesador Hipertexto) Es un lenguaje de programación implantado (enraizado) en HTML HTML-embed scripting lenguaje. La mayoría de su sintaxis está prestada de los lenguajes de programación C, Java y Perl, con la inclusión de algunos rasgos únicos de PHP. La meta del lenguaje es permitir a los que desarrollan sitios Web escribir rápidamente páginas generadas dinámicamente.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.... No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes. Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos.

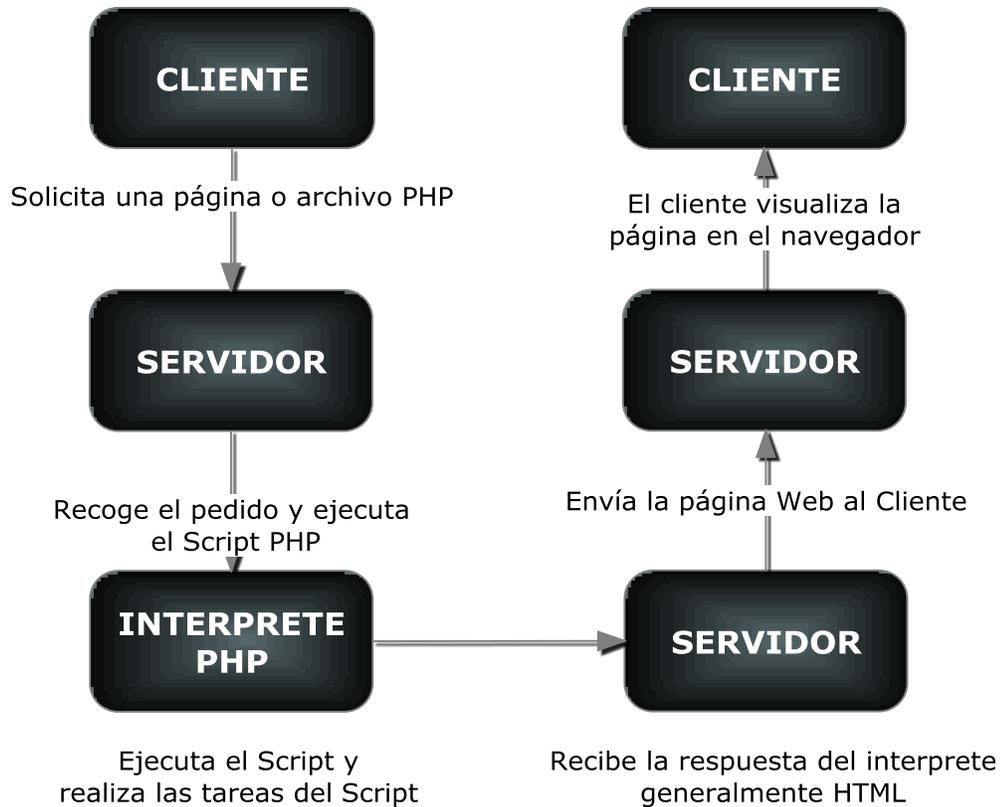
El programa PHP es ejecutado en el servidor y el resultado enviado al navegador.

El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

---

<sup>22</sup> <http://www.webestilo.com/php/php00.phtml>

## Funcionamiento de una página PHP



**Ilustración 3: Funcionamiento de PHP**

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

#### 4.2.6 PhpMyAdmin<sup>23</sup>

PhpMyAdmin es una utilidad que nos sirve para interactuar con una base de datos de forma muy sencilla y desde una interfaz web. Nos sirve por ejemplo para crear bases de datos, tablas, borrar o modificar datos, añadir registros, hacer copias de seguridad, etc. Es una aplicación tan útil que casi todos los hosting con MySQL disponen de ella, por ello se analizará su instalación. Además, vamos a usarlo para crear los usuarios MySQL para así poder utilizar las bases de datos de forma segura. Al ser una aplicación escrita en PHP, necesita de Apache y MySQL para poder funcionar.

#### 4.2.7 MySQL<sup>24</sup>

Es el servidor de bases de datos relacionales de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar. MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

*Las características principales de MySQL son:*

- **Es un gestor de base de datos.** Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- **Es una base de datos relacional.** Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

---

<sup>23</sup> <http://www.phpmyadmin.net>

<sup>24</sup> <http://www.webestilo.com/mysql/intro.phtml>

- **Es Open Source.** El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- **Es una base de datos muy rápida,** segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

#### **4.2.8 Macromedia Dreamweaver**

Es un editor WYSIWYG (What You See Is What You Get) de páginas web, creado por Macromedia. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Macromedia Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

#### **4.3 Estándares de diseño web<sup>25</sup>**

En un principio la construcción de un sitio web era una tarea más o menos simple.

Constaba de unas cuantas páginas enlazadas entre sí de forma sencilla, y cada una de ellas estaba formada por un código HTML básico, unas pocas imágenes y poco más.

Pero con el paso del tiempo las exigencias del mercado hicieron aparecer más y más lenguajes de programación basados en los protocolos TCP/IP, y especialmente en el HTTP, a la par que se introdujeron en el mundo web las tiendas virtuales y la banca electrónica, demandando sitios web capaces de poder operar con datos, con transacciones y con una interminable serie de nuevas aplicaciones concebidas para estos propósitos.

---

<sup>25</sup> <http://www.rodriogalindez.com/archivos/hacia-un-nuevo-modelo-para-construir-la-web/>

Esta circunstancia creó una situación inestable, en la que muchos sitios quedaban rápidamente obsoletos debido a las continuas innovaciones en navegadores y dispositivos.

Con los estándares Web podemos diseñar y generar sofisticados sitios de gran belleza garantizando que funcionen en el futuro. Por estándares nos referimos a lenguajes estructurales como XHTML y XML, lenguajes de presentación como CSS, modelos de objeto como el DOM del W3C y lenguajes de secuencia de comandos como ECMAScript.

#### **4.3.1 Desarrollar con estándares tiene muchas ventajas**

En general, la mayoría de los beneficios de usar estándares web parten de la premisa de separación de contenido, en XHTML, y presentación, en CSS. Entre algunas, de las muchas ventajas, podemos nombrar que los estándares:

- **Optimizan un sitio para motores de búsqueda (SEO).** Desarrollar con estándares produce código XHTML limpio y semántico. Puesto que la mayoría de los buscadores trabajan indexando el contenido de un sitio, éstos tienden a priorizar a aquellos sitios con menor cantidad de código basura o no estándar en el sitio.
- **Producen un sitio fácil de mantener.** El código resultante de un sitio con estándares es simple y se puede dividir en secciones, reduciendo la dependencia de un solo desarrollador y facilitando la comunicación entre varios grupos de trabajo en una empresa de desarrollo web (programación, maquetado, diseño, etc.).
- **Reducen el tamaño del sitio.** Debido a la eliminación de elementos HTML que formatean visualmente una página, agrupando toda la parte visual en CSS, el tamaño de un sitio se reduce drásticamente. Esto nos garantiza que el sitio será cargado rápidamente aún en conexiones lentas, como celulares.

- **Son un paso necesario para la accesibilidad.** A nivel dispositivo, los estándares aseguran que el contenido de nuestro sitio estará disponible para cualquier dispositivo con conexión a Internet. A nivel personas, implica tener en cuenta que nuestro sitio puede estar siendo accedido por personas con discapacidades, ya sean físicas (discapacidad visual, motriz, etc.) o de entorno o (sin mouse, con pantallas demasiado chicas, etc.).
  
- **Otorgan mayor flexibilidad al desarrollador web.** Esto permite a los desarrolladores, por un lado, ocuparse solamente de la parte estructural del sitio (XHTML) y por otro lado, a los diseñadores nos da la versatilidad suficiente como para cambiar cualquier aspecto del diseño de un sitio. Como ejemplo, Wired Magazine (<http://www.wired.com>) cambia el esquema de colores de su sitio todas las semanas, y para hacerlo modifica solamente una línea de código en su XHTML.

## 5. La Orientación a Objetos, OO<sup>26</sup>

Aunque UML puede emplearse en cualquier paradigma, como la programación estructurada o la lógica, está especialmente cerca del paradigma de la orientación a objetos. Por tanto, es precisa una familiarización con algunos detalles de este paradigma antes de continuar con UML.

### 5.1 ¿Qué es un Objeto?

De manera intuitiva, la tendencia general es asociar el término objeto con todo aquello a lo que se puede atribuir la propiedad física de masa, como una tostadora de pan, aunque es posible encontrar objetos de índole no tangible, como por ejemplo una dirección postal. En el ámbito de la informática, un objeto define una representación abstracta de las entidades del mundo, tangibles o no, con la intención de emularlas. Existe pues, una relación directa entre los objetos del mundo y los objetos informáticos, de modo que puede emplearse el término objeto de manera indistinta.

Los objetos tienen dos características, que son su estado y su comportamiento. El estado es una situación en la que se encuentra el objeto, tal que cumple con alguna condición o condiciones particulares, realiza alguna actividad o espera que suceda un acontecimiento. Una tostadora puede estar encendida y cargada de pan y, en cuanto a su comportamiento, lo normal en este estado es tostar pan.

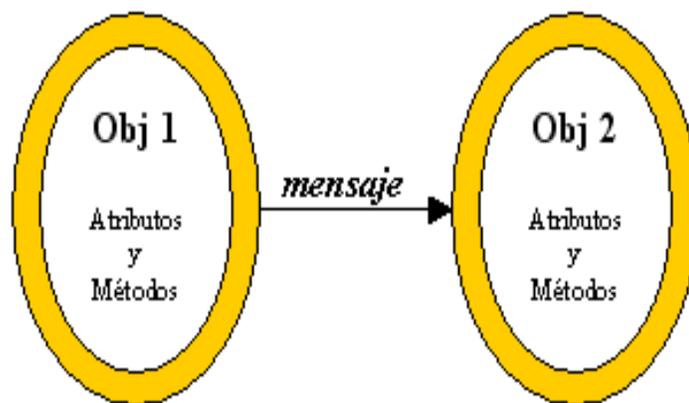
Los objetos mantienen su estado en uno o más atributos, que son simplemente datos identificados por un nombre, y exhiben su comportamiento a través de métodos, que son trozos de funcionalidad asociados al objeto. En este sentido, un objeto es realmente un conjunto de atributos y métodos. Pero un objeto sólo

---

<sup>26</sup> Grady Booch, James Rumbaugh, Ivar Jacobson, (1996) *El Lenguaje Unificado de Modelado*, Addison Wesley.

revela su verdadera utilidad cuando es integrado en un contexto de comunicación con otros objetos, a través del envío de mensajes, para componer un sistema mucho mayor y demostrar un comportamiento más complejo. Una tostadora en un armario resulta de poca utilidad, pero cuando interactúa con otro objeto, como un ser humano, se convierte en una herramienta útil para tostar pan. El humano intercambiaría con la tostadora el mensaje “tuesta el pan que tienes en la bandeja” a través de la pulsación del botón de tostar.

A partir del ejemplo anterior, es fácil deducir que el envío de mensajes es la forma en que se invocan los métodos de un objeto y que la invocación de métodos es el mecanismo a través del cual un objeto puede cambiar su estado o el de otro objeto. Los atributos y los métodos de un objeto pueden tener un menor o mayor grado de visibilidad, desde “privado” hasta “público”, lo que hace que aparezca un concepto nuevo, la encapsulación. La encapsulación oculta los detalles del funcionamiento interno del objeto, exponiendo sólo aquello que pueda ser de interés.



**Ilustración 4: Objetos comunicándose.**

## 5.2 ¿Qué es una Clase?

Los objetos no son entidades que existan de modo único. Hay muchos tipos de tostadoras e, igualmente, muchas tostadoras del mismo tipo. Se puede entender fácilmente el concepto de clase si nos permitimos emplear el término *tipo* como equivalente. Así, todos los objetos que son del mismo *tipo*, comparten el mismo juego de atributos y métodos (aunque cada objeto pueda tener un valor distinto asociado a cada atributo) y por tanto pertenecen a una misma clase. Las clases son como patrones que definen qué atributos y qué métodos son comunes a todos los objetos de un mismo tipo.

Cada objeto tiene sus atributos y su comportamiento, creados empleando una clase a modo de patrón. Una vez creado el objeto, pasa a ser una instancia particular de la clase a la que pertenece y sus atributos tienen unos valores concretos, que podrán variar de un objeto a otro (dos objetos distintos pertenecientes a la misma clase, pueden tener exactamente los mismos valores en todos sus atributos). A estos atributos, que pueden variar de un objeto a otro, se les conoce también como *variables de instancia*.

Hay atributos que, sin embargo, no varían de un objeto a otro, es decir todas las instancias de la clase a la que pertenecen, tienen el mismo valor para ese atributo. Todas las tostadoras del mismo tipo consumen los mismos Vatios y sus resistencias son de los mismos Ohmios. A estos atributos se les conoce como *variables de clase* y son compartidos por todas y cada una de las instancias de la clase. De manera análoga al caso de los atributos, encontramos *métodos de instancia* y *métodos de clase*.

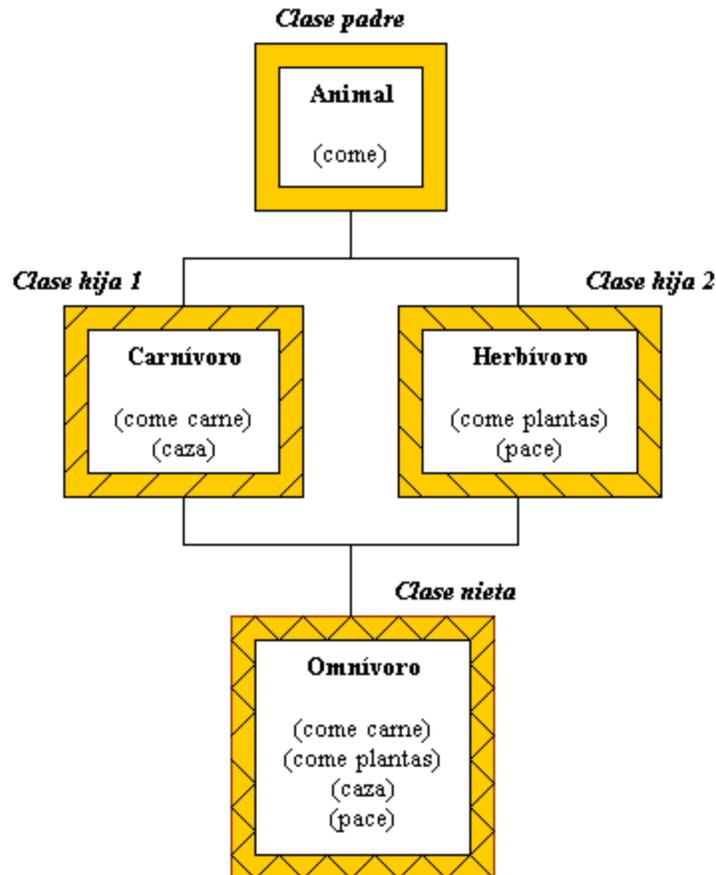
### 5.3 ¿Qué es la Herencia?

Los objetos se definen en función de clases, es decir, tomando una clase como patrón. Se puede saber mucho acerca de un objeto sabiendo la clase a la que pertenece. Por ejemplo, con decir que la “Russell Hobbs 10243 Kudos” es un tipo de tostadora, inmediatamente se sabe que se trata de una máquina para tostar pan, probablemente eléctrica y con por lo menos una ranura en la que insertar una rebanada de pan y un botón para activar su funcionamiento.

Las clases llegan un paso más lejos, permitiendo su definición en función de otras clases, de modo que es posible establecer una jerarquía de especialización. Una clase que se define en función de otra, hereda todos los atributos y métodos de aquella y permite el añadido de nuevos o la sobre escritura de los heredados. La clase patrón se conoce con el nombre de *superclase* o *clase padre*, mientras que la que hereda se conoce como *clase hija*. La herencia no está limitada simplemente a padre-hija(s), la jerarquía puede ser todo lo profunda que sea necesario, hablando en términos de nietas, biznietas, etc. De la misma manera, una clase puede heredar de varias clases a la vez.

En la siguiente figura se puede ver una jerarquía de especialización de dos niveles. La clase “Animal” es la raíz, la clase padre en la jerarquía. Especifica que los animales *comen*, como característica más significativa de éstos. En el primer nivel de especialización encontramos las clases “Carnívoro” y “Herbívoro”, ambas son sendos tipos de animal y por tanto *comen*, sólo que en el caso de los carnívoros se ha especializado el tipo de comida que comen para indicar que se trata de *carne*. Aparece una nueva característica de este tipo de animal, que es el hecho de que los carnívoros *cazan*. En el caso de los herbívoros, encontramos que *comen plantas* y *pacen*. En el segundo nivel de especialización, encontramos un animal que es a la vez “Herbívoro” y “Carnívoro” y, como cabe esperar, este nuevo tipo de animal puede hacer todo lo que pueden hacer sus

ancestros, comer carne, comer plantas, cazar y pacer, no encontrando ninguna característica adicional en los “Omnívoros”.



**Ilustración 5: La herencia.**

#### 5.4 ¿Qué es una Interfaz?

Una interfaz es un mecanismo que emplean dos objetos para interactuar. En nuestro ejemplo de la tostadora, el humano emplea el botón de tostar a modo de interfaz para pasar el mensaje “tuesta el pan que tienes en la bandeja”.

Las interfaces definen un conjunto de métodos para establecer el protocolo en base al cual interactúan dos objetos. En este sentido, existe una analogía entre

interfaces y protocolos. Para que el humano pueda tostar, debe seguir el protocolo establecido por la interfaz botón de tostar, consistente en pulsar dicho botón.

Las interfaces capturan las similitudes entre clases no relacionadas, sin necesidad de forzar una interrelación y son a su vez clases.

## 6. El Lenguaje Unificado de Modelado, UML

El UML es un lenguaje de modelado cuyo vocabulario y sintaxis están ideados para la representación conceptual y física de un sistema. Sus modelos son precisos, no ambiguos, completos y pueden ser trasladados directamente a una gran variedad de lenguajes de programación, como Java, C++ o Visual Basic, pero también a tablas de bases de datos relacionales y orientadas a objetos. Es posible generar código a partir de un modelo UML (ingeniería directa) y también puede construirse un modelo a partir de la implementación (ingeniería inversa), aunque en las dos situaciones debe intervenir un mayor o menor grado de supervisión por parte del programador, en función de lo buenas que sean las herramientas empleadas.

### 6.1 Bloques básicos de construcción de UML

Los bloques básicos de construcción de UML son tres, los elementos, las relaciones y los diagramas.

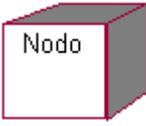
- **Los *elementos*** son abstracciones que actúan como unidades básicas de construcción. Hay cuatro tipos, los *estructurales*, los de *comportamiento*, los de *agrupación* y los de *notación*. En cuanto a los elementos estructurales son las partes estáticas de los modelos y representan aspectos conceptuales o materiales. Los elementos de comportamiento son las partes dinámicas de los modelos y representan comportamientos

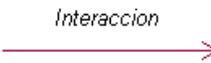
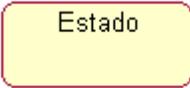
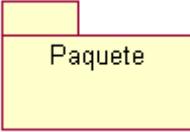
en el tiempo y en el espacio. Los elementos de agrupación son las partes organizativas de UML, establecen las divisiones en que se puede fraccionar un modelo. Sólo hay un elemento de agrupación, el paquete, que se emplea para organizar otros elementos en grupos. Los elementos de notación son las partes explicativas de UML, comentarios que pueden describir textualmente cualquier aspecto de un modelo. Sólo hay un elemento de notación principal, la nota.

- **Las relaciones** son abstracciones que actúan como unión entre los distintos *elementos*. Hay cuatro tipos, la *dependencia*, la *asociación*, la *generalización* y la *realización*.
- **Los diagramas** son la disposición de un conjunto de elementos, que representan el sistema modelado desde diferentes perspectivas. UML tiene nueve diagramas fundamentales, agrupados en dos grandes grupos, uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico. Los **diagramas estáticos son:** el de *clases*, de *objetos*, de *componentes* y de *despliegue*. Los **diagramas de comportamiento son:** el de *Casos de Uso*, de *secuencia*, de *colaboración*, de *estados* y de *actividades*.

### 6.1.1 Elementos

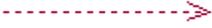
<b>E L E M</b>	Clase		Describe un conjunto de objetos que comparten los mismos atributos, métodos, relaciones y semántica. Las clases implementan una o más interfaces.
	Clase activa		Se trata de una clase, en la que existen procesos o hilos de ejecución concurrentes con otros elementos. Las líneas del contorno son más gruesas que en la clase "normal"
	Interfaz		Agrupación de métodos u operaciones que especifican un servicio de una clase o componente, describiendo su comportamiento, completo o parcial, externamente visible. UML permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.

E N T O S  E S T R U C T U R A L E S	Colaboración	 <p>Colaboracion</p>	Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor que la suma de los comportamientos de sus elementos.
	Caso de uso	 <p>Caso de Uso</p>	Describe un conjunto de secuencias de acciones que un sistema ejecuta, para producir un resultado observable de interés. Se emplea para estructurar los aspectos de comportamiento de un modelo.
	Componente	 <p>Comp</p>	Parte física y por tanto reemplazable de un modelo, que agrupa un conjunto de interfaces, archivos de código fuente, clases, colaboraciones y proporciona la implementación de dichos elementos.
	Nodo	 <p>Nodo</p>	Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con capacidad de procesar.

<b>Elementos de comportamiento</b>	Interacción		Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
	Máquinas de estados		Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.
<b>Elementos de agrupación</b>	Paquete		Se emplea para organizar otros elementos en grupos.
<b>Elementos de notación</b>	Nota		Partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo

**Tabla 3: Elementos UML**

### 6.1.2 Relaciones

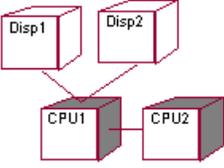
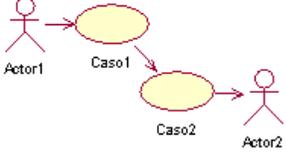
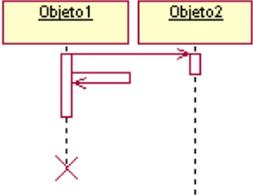
Dependencia		Es una relación entre dos elementos, tal que un cambio en uno puede afectar al otro.
Asociación		Es una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.
Generalización		Es una relación en la que el elemento generalizado puede ser substituido por cualquiera de los elementos hijos, ya que comparten su estructura y comportamiento.
Realización		Es una relación que implica que la parte realizante cumple con una serie de especificaciones propuestas por la clase realizada (interfaces).

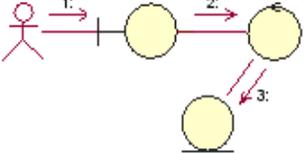
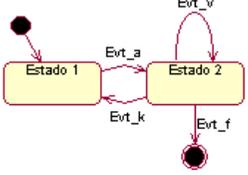
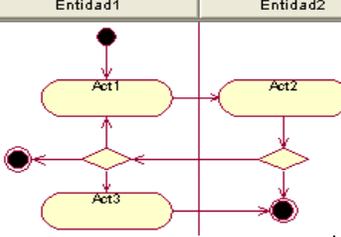
**Tabla 4: Relaciones UML**

### 6.1.3 Diagramas<sup>27</sup>

<b>M O D E L A N E S</b>	Clases		Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubriendo la vista de diseño estática del sistema.
	Objetos		Análogo al diagrama de clases, muestra un conjunto de objetos y sus relaciones, pero a modo de vista instantánea de instancias de una clase en el tiempo.
	Componentes		Muestra la organización y dependencias de un conjunto de componentes. Cubren la vista de implementación estática de un sistema. Un componente es un módulo de código, de modo que los diagramas de componentes son los análogos físicos a los diagramas de clases.

<sup>27</sup> Schneider G., Winters J.P., (2001) *Applying Use Cases: A Practical Guide*, Addison Wesley.

<p><b>T R U C T U R A</b></p>	<p>Despliegue</p>		<p>Muestra la configuración del hardware del sistema, los nodos de proceso y los componentes empleados por éstos. Cubren la vista de despliegue estática de una arquitectura.</p>
<p><b>M O</b></p>	<p>Casos de Uso</p>		<p>Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.</p>
	<p>Secuencia</p>		<p>Son diagramas de interacción, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos. Cubren la vista dinámica del sistema. El diagrama de secuencia resalta la ordenación temporal de los</p>

<p style="text-align: center;">D E L A N  C O M P O R T A M I E N T O</p>	<p style="text-align: center;">Colaboración</p>		<p>mensajes, mientras que el de colaboración resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos. En el diagrama de colaboración de la figura de la izquierda, se puede ver que los elementos gráficos no son cajas rectangulares, como cabría esperar, y en su lugar encontramos sus versiones adornadas. Estas versiones tienen como finalidad evidenciar un rol específico del objeto siendo modelado. En la figura encontramos de izquierda a derecha y de arriba abajo un Actor, una Interfaz, un Control (modela un comportamiento) y una Instancia (modela un objeto de dato).</p>
	<p style="text-align: center;">Estados</p>		<p>Muestra una máquina de estados, con sus estados, transiciones, eventos y actividades. Cubren la vista dinámica de un sistema. Modelan comportamientos reactivos en base a eventos.</p>
	<p style="text-align: center;">Actividades</p>		<p>Tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.</p>

**Tabla 5: Diagramas UML**

## 6.2 ¿Cómo utilizar UML?

UML es simplemente un lenguaje de modelado. Define un conjunto de elementos y relaciones entre ellos, que se emplean en la definición de modelos. UML es típicamente usado como parte de un proceso de desarrollo, con la ayuda de una herramienta CASE (Computer Aided Software Engineering), para definir requerimientos, interacciones y elementos del software que se está desarrollando. UML es independiente de cualquier proceso particular, no está ligado a ningún ciclo de vida de desarrollo del software concreto, no obstante se obtienen mayores beneficios si se selecciona un proceso que esté dirigido por Casos de Uso, se centre en la arquitectura y sea incremental.

La arquitectura de un sistema es el conjunto de decisiones significativas que se toma en torno a su organización, la selección de elementos estructurales, la definición de las interfaces entre estos elementos, su comportamiento, su división en subsistemas, qué elementos son estáticos y cuales dinámicos. La arquitectura también incluye el uso que se le va a dar al sistema, la funcionalidad, el rendimiento, la capacidad de adaptación, la reutilización, la capacidad de ser comprendido, las restricciones económicas, las temporales, los compromisos entre alternativas y los aspectos estéticos.

Un proceso incremental es aquél que consiste en sucesivas ampliaciones y mejoras de la arquitectura, a partir de una línea básica. Cada incremento resuelve los problemas encontrados en la versión anterior minimizando incrementalmente los riesgos más significativos para el éxito del proyecto.

Lo primero que se debe hacer para comenzar a desarrollar un proyecto con UML, es seleccionar una metodología de desarrollo que defina la naturaleza concreta del proceso a seguir. El modelo a definir en base al proceso elegido, se divide en realidad en varios tipos de modelo o vistas, cada una centrada en un aspecto o punto de vista del sistema. En general, independientemente del proceso que se emplee, se puede encontrar las siguientes vistas:

- **Vista de Casos de Uso:** Engloba los Casos de Uso que describen el comportamiento del sistema como lo verían los usuarios finales, los analistas y demás componentes del equipo de desarrollo. No especifica la organización del sistema. Con UML los **aspectos estáticos** de esta vista se pueden concretar con los diagramas de *Casos de Uso*; los **aspectos dinámicos** con los diagramas de iteración (*secuencia* y *colaboración*), diagramas de *estados* y de *actividades*.
- **Vista de Diseño:** Engloba las clases e interfaces que conforman el vocabulario del problema y su solución. Da soporte a los requisitos funcionales del sistema, es decir los servicios que proporciona a los usuarios finales. Con UML los **aspectos estáticos** de esta vista se pueden concretar con los diagramas de *clases* y de *objetos*; los **aspectos dinámicos** con los diagramas de iteración (*secuencia* y *colaboración*), diagramas de *estados* y de *actividades*.
- **Vista de Procesos:** Engloba los hilos y procesos que forman los mecanismos de sincronización y concurrencia del sistema. Da soporte al funcionamiento, capacidad de crecimiento y rendimiento del sistema. Con UML los **aspectos estáticos** de esta vista se pueden concretar con los diagramas de *clases*, de *clases activas* y de *objetos*; los **aspectos dinámicos** con los diagramas de iteración (*secuencia* y *colaboración*), diagramas de *estados* y de *actividades*.
- **Vista de Despliegue:** Engloba los nodos que forman la topología hardware sobre el que se ejecuta el sistema. Da soporte a la distribución, entrega e instalación de las partes que conforman el sistema físico. Con UML los **aspectos estáticos** de esta vista se pueden concretar con los diagramas *despliegue*; los **aspectos dinámicos** con los diagramas de

iteración (*secuencia y colaboración*), diagramas de *estados* y de *actividades*.

- **Vista de Implementación:** Engloba los componentes y archivos empleados para hacer posible el sistema físico. Da soporte a la gestión de configuraciones de las distintas versiones del sistema, a partir de componentes y archivos. Con UML los **aspectos estáticos** de esta vista se pueden concretar con los diagramas de *componentes*; los aspectos dinámicos con los diagramas de iteración (*secuencia y colaboración*), diagramas de *estados* y de *actividades*.

Un ejemplo de proceso para la construcción de un programa, podría ser similar al siguiente, teniendo en cuenta que el proceso descrito deja muchas cosas por ampliar y puede no adaptarse a las necesidades particulares de un grupo de trabajo determinado. Se proporciona meramente como un ejemplo de cómo se puede encajar UML como soporte para el desarrollo de un proyecto:

1. Iniciar y mantener *reuniones* con los usuarios finales del programa, para comprender sus necesidades, el contexto en que lo usarán y todos los detalles necesarios para comprender el ámbito del problema a resolver. Esta información será empleada para capturar las actividades y procesos involucrados y susceptibles de ser incorporados en el programa, a un nivel alto, y proporcionará la base para construir la vista de Casos de Uso.
2. Construir la *vista de Casos de Uso* definiendo exactamente la funcionalidad que se va a incorporar en el programa, desde el punto de vista de sus usuarios. El modelo resultante es realmente un mapeo de la información obtenida en el paso anterior, en el que cada nuevo Caso de Uso realiza un aspecto de la funcionalidad planteada. Refinar, en conjunto con los usuarios finales, todos los diagramas de Casos de Uso,

incluyendo requisitos y restricciones, para llegar a un acuerdo común en lo que el programa hará y no hará. En este punto puede ser conveniente diseñar escenarios de prueba que ayuden a verificar si el programa finalizado cumple con las expectativas del contrato.

3. Partiendo del modelo de Casos de Uso se comienza a estructurar los requisitos en una arquitectura llamada “línea base”. Se definen clases y relaciones entre ellas, los primeros diagramas de secuencia y colaboración, definiendo los comportamientos de cada clase, también las interfaces entre los diferentes elementos de la arquitectura. Se construye aquí la *vista de diseño* y la *vista de procesos*. Construir diagramas de clases más elaborados y refinar los comportamientos del sistema.
4. A medida que crece el modelo se puede fraccionar en componentes software y paquetes. Aparecen nuevos requisitos que deben ser integrados. Se define la *vista de despliegue*, que define la arquitectura física del sistema, y la *vista de implementación*.
5. Construir el sistema, repartiendo las tareas entre el equipo de programación.
6. Buscar errores de programación, o incluso de diseño, corregirlos e ir sacando sucesivas versiones del programa hasta llegar a una versión que cumpla con todos los requisitos especificados en el contrato con los usuarios.
7. Documentar y entregar el programa a los usuarios finales.

## 7. Aspectos metodológicos

### 7.1 Proceso de Iweb

Es claramente incremental y evolutivo.

Por la naturaleza intensiva, tendremos:

- Amplia y diversa población de usuarios (obtención y modelado de requisitos)
- Arquitectura altamente especializada (exigencias en el diseño)

### 7.2 Paradigma espiral orientado a la web (modelo del proceso Iweb)<sup>28</sup>

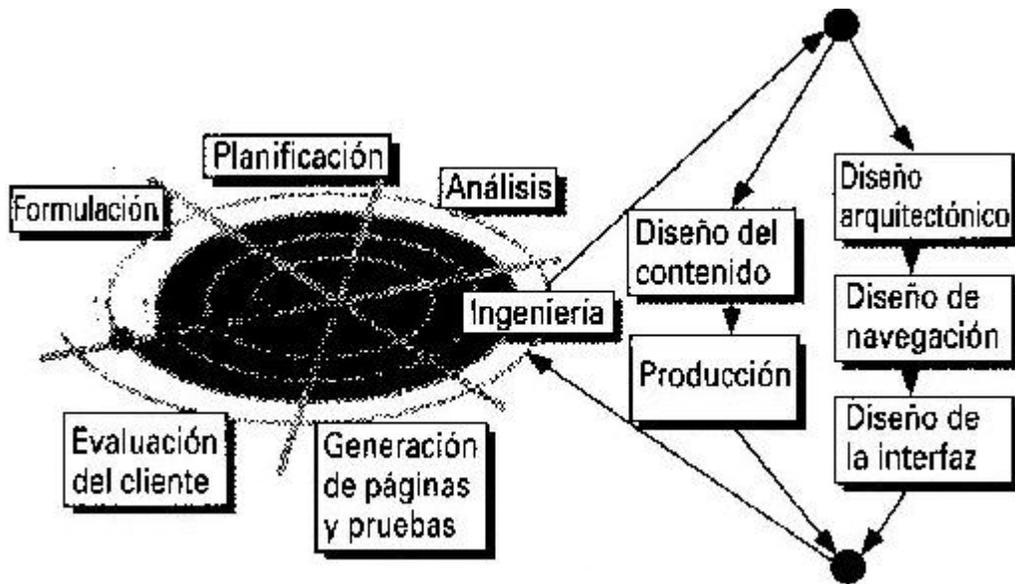


Ilustración 6: El modelo del proceso IWeb

<sup>28</sup> □ Pressman Roger, Ingeniería del software. Un enfoque práctico, Madrid, McGraw-Hill / Interamericana de España, 1997

A grandes rasgos:

- **Formulación:** Se identifican las metas y objetivos
- **Planificación:** Estimación del coste global del proyecto, riesgos, etapas y sub-etapas.
- **Análisis:** Establecimiento de los requisitos técnicos y de diseño (estéticos) e identificación de los elementos de contenido.
- **Ingeniería:** Dos tareas paralelas:
  - Diseño del contenido y producción: hechas por personal NO técnico. Recopilación de información, medios audiovisuales, a integrar en la App.
  - Diseño arquitectónico, de navegación y del interfaz: hecho por técnicos
- **Generación de páginas:** Se adecua al diseño arquitectónico, de navegación y de interfaz, el contenido provisto para sacar las páginas HTML, XML, etc. Es en esta fase donde se integra la WebApp con el software intermedio (CORBA, DCOM, JavaBeans).
- **Pruebas:** Se hace una navegación intensiva sobre la aplicación para descubrir errores, visualizarla en otros navegadores y ser consciente cuanto menos de las limitaciones y posibles “bugs”.
- **Evaluación del cliente:** No es la última fase. Es una fase a ejecutar cada vez que se termina alguna de las anteriores. Los cambios se hacen efectivos por el flujo incremental del proceso.

### 7.2.1 Formulación

Para hacer una correcta formulación, debemos preguntarnos, entre otras cosas:

- Por qué y para que hacer la WebApp?
- Por que es necesaria?
- Quien la va a usar?

Las respuestas serán muy generales, y no entraran en detalles. Podemos clasificar las metas específicas en:

- **Metas Informativas:** Definen los objetivos sobre el contenido e información que se dará al usuario.

- **Metas Aplicables:** Son los servicios o tareas que puede realizar la WebApp. Después de las metas, haremos el Perfil del Usuario, determinando las principales características de los potenciales navegadores y clientes.

Más adelante, se hace la Afirmación del Ámbito, con la que vemos la posible integración con sistemas ya existentes, como pueden ser bases de datos.

### 7.2.2 Análisis

Identifica los datos y requisitos funcionales y de comportamiento para la WebApp.

Durante la IWeb, se realizan 4 tipos de análisis:

- **Análisis del contenido:** Se puede utilizar el modelado de datos, y en esta etapa se identifica todo el contenido que se va a proporcionar. (Texto, gráficos, imágenes, video y sonido)

- **Análisis de la interacción:** Se realizan casos prácticos y sus casos de uso para la descripción detallada de la interacción usuario-WebApp.

- **Análisis funcional:** Se detallan las funciones y operaciones de procesamiento adicionales que se aplicaran en el contenido de la WebApp

- **Análisis de la configuración:** Se detalla y describe el lugar donde va a residir la App. (Intranet, Internet o Extranet). También se tiene que identificar la infraestructura de los componentes y el grado de utilización de la base de datos para generar el contenido.

En todo caso es recomendable hacer un documento que recoja la información de todo el proceso de análisis y que será revisado y modificado para hacer otro documento que pasarle a los diseñadores de la WebApp. En el caso de una App grande no es recomendable hacer un documento muy extenso, porque los requisitos estarán cambiando continuamente, y quedaría obsoleto antes de terminarlo.

### 7.2.3 Diseño

La característica de inmediatez obliga a que los diseños se hagan rápidamente y a que sean evolucionales. Muchas veces la rapidez o precipitación en el diseño nos cierra puertas a la evolución de la aplicación.

Los ingenieros Web, trabajan bajo los siguientes elementos técnicos:

- **Principios y métodos de diseño:** Facilitaran la adaptación, pruebas, mejoras y uso.

- Modularidad eficaz (cohesión alta y acoplamiento bajo)

- Elaboración pasó a paso

- Diseño orientado a objetos y diagramas UML

- **Reglas de oro:** que se han ido construyendo desde los inicios de Internet

- **Configuraciones de diseño:** Aplicables a los elementos funcionales y a los documentos, gráficos y estética general.

- **Plantillas:** Dotan de una estructura similar cada elemento, configuración de diseño, o documento a utilizar dentro de la WebApp. Se hace posible pasando como parámetros a esa plantilla, los datos relevantes, que darán cuerpo al esquema.

### 7.2.3.1 Diseño Arquitectónico

Se encarga de la definición de la estructura global hipermedia y en la aplicación de las configuraciones de diseño y plantillas. Dicha estructura depende de las metas establecidas, del contenido y de la filosofía de navegación. Típicamente hay:

- **Estructuras lineales:** cuando es predecible la sucesión de interacciones. Por ejemplo en la entrada, y validación de datos, hay una estructura lineal. También existen lineales con flujo opcional, y lineal con desviaciones.
- **Estructuras reticulares:** Solo si el contenido de la Web puede ser organizado en dos o más dimensiones. Para ellos el contenido debe ser muy regular. Por ejemplo, marcas de electrodomésticos y tipos de electrodomésticos.
- **Estructuras jerárquicas:** Son las más comunes. En las jerarquías de software tradicionales se fomentan el flujo de control solo a lo largo de las ramas verticales. En una WebApp se pueden enlazar por hipertexto ramas verticales de la misma estructura. Es el “Acoplamiento”.
- **Estructura en red (o de web pura):** Es como la arquitectura “en evolución” de los sistemas OO. Se enlaza todo con todo. Da mucha flexibilidad de navegación, aunque a veces es confusa para el usuario. Es común combinar varias de las estructuras, dando lugar a estructuras híbridas.

Los **patrones de diseño** pueden aplicarse en el nivel de componente (cuando se requiere la funcionalidad del proceso de datos), jerárquico, y de navegación (que tratan sobre como el usuario podrá moverse por el contenido de la aplicación)

Entre estos últimos, están:

- Ciclo: Se devuelve al usuario al nodo de contenido visitado anteriormente.
- Anillo de Web: Se enlazan páginas de un mismo tema.
- Contorno: Cuando varios ciclos inciden en otro
- Contrapunto: durante la narración se añaden comentarios de hipertexto.
- Mundo de espejo: Varias narraciones desde puntos de vista distintos
- Tamiz: Se presentan opciones que el usuario va eligiendo, hasta llegar a un punto que el mismo habrá provocado con sus decisiones.
- Vecindario: Marco de navegación uniforme por todas las páginas web

### 7.2.3.2 Diseño de navegación

Una vez establecida la arquitectura se define la ruta que permitirá acceder al contenido y a los servicios. Se deberá identificar una **semántica** para según qué usuarios y definir una **sintaxis** (mecánica) para la navegación.

Se tendrán, habitualmente, varios papeles: visitante, cliente, cliente registrado, cliente privilegiado, administrador, etc. La semántica para cada rol será distinta.

El diseñador crea una USN (Unidad Semántica de Navegación) para cada meta asociada a cada rol de usuario. Cada USN tiene unas “formas de navegación” (WoN) para que cada usuario llegue a cada meta que se proponga.

Entre las opciones de enlaces (texto, iconos, botones, interruptores, metáforas gráficas, etc.) deberemos elegir la que más se adecuen al interfaz de nuestra web.

Desde el punto de vista de los buscadores hoy por hoy es mejor un enlace texto con la palabra con la que nos gustaría dotar de importancia a la página web enlazada que cualquier otra cosa.

Sin embargo, desde nuestra visión de diseño, los botones, imágenes e iconos que usemos deberán tener un aspecto clickable. Los enlaces de texto deberán tener un color característico, diferenciador del resto del documento.

También se harán necesarias ayudas a la navegación por el sitio: una vista de esquema, un mapa web, tabla de contenidos, mecanismos de búsqueda y servicios dinámicos de ayuda.

### **7.2.3.3 Diseño de la interfaz**

Además de las consideraciones de diseño de interfaces de cualquier otro software, en WebApps es necesario considerar nuevos factores, todos ellos, bastante subjetivos.

Algunas sugerencias generalizadas son:

- Los errores de servidor deben ser mínimos. El usuario tiene poca paciencia, y generalmente muchos otros recursos en la Web.
- No se debe obligar a hacer leer grandes cantidades de texto, sobre todo si estamos en alguna de las secciones de Ayuda de nuestra App.
- Evitar poner “En construcción”. Crea expectativas decepcionantes.
- Evitar el scroll. Un usuario poco experto “no sabe que existe el scroll”. Todo lo que se le pueda dar en un “pantallazo” será mejor entendido por la mayoría.
- Los menús de navegación estarán disponibles en todas las páginas. Las funciones de navegación no deberán depender del navegador que se esté usando.
- La estética nunca deberá sustituir la funcionalidad.
- Las opciones de navegación y el resto de funcionalidades deberán ser obvias.

### 7.2.4 Pruebas

Son el proceso de ejercitar el software con el fin de encontrar y corregir los errores. En las WebApps, es un reto, debido a la variedad de navegadores, sistemas operativos, plataformas hardware y protocolos de comunicación.

Las estrategias y tácticas a seguir son:

1. El modelo de contenido es revisado para descubrir errores: similar a un corrector ortográfico.
2. El modelo de diseño es revisado para descubrir errores de navegación: Se revisan los posibles errores 404 de navegación, y vemos si cada enlace lleva a la correspondiente USN de la meta del rol de usuario a la que pertenece.
3. Se aplican pruebas de unidad a los componentes de proceso seleccionado y las páginas Web: en muchos casos la unidad comprobable más pequeña es la propia página web. Muchas veces no es posible o practico comprobar elementos más pequeños como formularios, objetos, mapas de imágenes, etc.
4. Se construye la arquitectura y se realizan las pruebas de integración: La estrategia para la prueba de integración depende de la arquitectura que se haya elegido. En estructuras jerárquicas lineales, reticulares o sencillas, es muy similar a como se integran los módulos del software convencional. En jerarquías mezcladas o arquitecturas de red, es similar a los sistemas OO.
5. La WebApp ensamblada se prueba para conseguir una funcionalidad global y un contenido: Se hace una prueba de acciones visibles y de salidas reconocibles para el usuario.
6. Se implementa la WebApp en una variedad de configuraciones diferentes de entornos y comprobar así la compatibilidad con cada configuración: Se lleva a cabo una matriz de referencias cruzadas con sistemas operativos, plataformas de hardware, navegadores y protocolos de comunicación. Se hacen pruebas

para cubrir los errores asociados con todas y cada una de las configuraciones posibles.

7. La WebApp se comprueba con una población de usuarios finales controlada y monitorizada: Se hacen grupos de usuarios según los posibles roles, se hace un uso intensivo y se evalúan los resultados, para ver errores de contenido y navegación, usabilidad, compatibilidad, fiabilidad y rendimiento.

## **8. Metodología: OOHDM (object oriented hypermedia design methodology)<sup>29</sup>**

### **8.1 Introducción OOHDM**

Object Oriented Hypermedia Design Methodology (OOHDM, Método de Diseño Hipermedia Orientado a Objetos), propuesto por Schwabe y Rossi (1998).

OOHDM tiene por objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia.

Las metodologías tradicionales de ingeniería de software, o las metodologías para sistemas de desarrollo de información, no contienen una buena abstracción capaz de facilitar la tarea de especificar aplicaciones hipermedia. El tamaño, la complejidad y el número de aplicaciones crecen en forma acelerada en la actualidad, por lo cual una metodología de diseño sistemática es necesaria para disminuir la complejidad y admitir evolución y reusabilidad.

Producir aplicaciones en las cuales el usuario pueda aprovechar el potencial del paradigma de la navegación de sitios web, mientras ejecuta transacciones sobre bases de información, es una tarea muy difícil de lograr. En primer lugar, la navegación posee algunos problemas. Una estructura de navegación robusta es una de las claves del éxito en las aplicaciones hipermedia. Si el usuario entiende

---

<sup>29</sup> Schwabe, D. y Rossi, G. (1998). Developing Hypermedia Applications using OOHDM

dónde puede ir y cómo llegar al lugar deseado, es una buena señal de que la aplicación ha sido bien diseñada.

Construir la interfaz de una aplicación web es también una tarea compleja; no sólo se necesita especificar cuáles son los objetos de la interfaz que deberían ser implementados, sino también la manera en la cual estos objetos interactuarán con el resto de la aplicación.

En hipermedia existen requerimientos que deben ser satisfechos en un entorno de desarrollo unificado (framework). Por un lado, la navegación y el comportamiento funcional de la aplicación deberían ser integrados. Por otro lado, durante el proceso de diseño se debería poder desacoplar las decisiones de diseño relacionadas con la estructura navegacional de la aplicación, de aquellas relacionadas con el modelo del dominio.

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

## **8.2 Fases de la metodología OOHDM<sup>30</sup>**

OOHDM como técnica de diseño de aplicaciones hipermedia, propone un conjunto de tareas que según Schwabe, Rossi y Simone (s. f.) pueden resultar costosas a corto plazo, pero a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar el coste de evoluciones y mantenimiento.

Esta metodología plantea el diseño de una aplicación de este tipo a través de cinco fases que se desarrollan de un modo iterativo.

Estas fases son:

- Determinación de Requerimientos.
- Diseño Conceptual.
- Diseño Navegacional.

---

<sup>30</sup> Schwabe, D., Rossi, G. y Simone, J. (s. f.). Systematic Hypermedia Application Design with OOHDM.

- Diseño de Interfaz Abstracta.
- Implementación.

### 8.2.1 Determinación de Requerimientos

La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema.

Según German (2003)<sup>31</sup> primero que todo es necesaria la recopilación de requerimientos. En este punto, se hace necesario identificar los actores y las tareas que ellos deben realizar. Luego, se determinan los escenarios para cada tarea y tipo de actor.

Los casos de uso que surgen a partir de aquí, serán luego representados mediante los Diagramas de Interacción de Usuario (UIDs), los cuales proveen de una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea. Con este tipo de diagramas se capturan los requisitos de la aplicación de manera independiente de la implementación.

Ésta es una de las fases más importantes, debido a que es aquí donde se realiza la recogida de datos, para ello se deben de proporcionar las respuestas a las siguientes interrogantes:

- ¿Cuáles son los tópicos principales que serán atendidos?
- ¿Cómo los tópicos están relacionados entre sí?
- ¿Qué categoría de usuarios serán atendidos?
- ¿Cuáles son las tareas principales que serán abordadas?
- ¿Qué tareas corresponden a qué categoría de usuarios?
- ¿Los recursos disponibles son competitivos con la información levantada?

---

<sup>31</sup>German, D. (2003). The Object Oriented Hypermedia Design Method. [Documento en línea]. Disponible: <http://www.telemidia.pucrio.br/oohdm/oohdm.html>.

Con las preguntas mencionadas anteriormente, se puede recaudar de cierta manera las bases necesarias para la construcción de una aplicación hipermedia exitosa, sin embargo mientras mayor sea el nivel de profundidad de la recolección de datos, mayor probabilidad de realizar una aplicación adecuada a las necesidades de los usuarios.

### **8.2.2 Diseño Conceptual**

Durante esta actividad se lleva a cabo, según Koch (2002)<sup>32</sup> un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos.

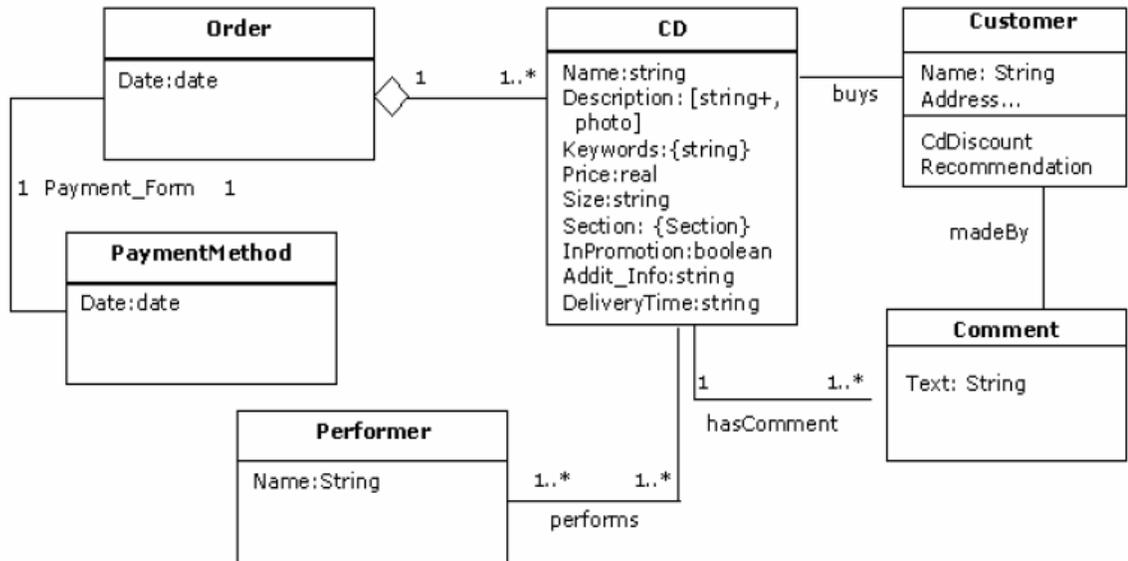
El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones que son usadas para construir enlaces.

Como es de costumbre en modelos orientados a objetos, las clases son descritas por un conjunto de atributos y métodos (implementando el comportamiento de las clases), siendo aún, organizadas en jerarquías (parte-de y es uno/a). En la Figura

5 se puede observar un ejemplo de cómo se representa un diagrama de clases.

---

<sup>32</sup> Koch, N. (2002). Ingeniería de Requisitos en Aplicaciones para la Web—Un estudio comparativo.



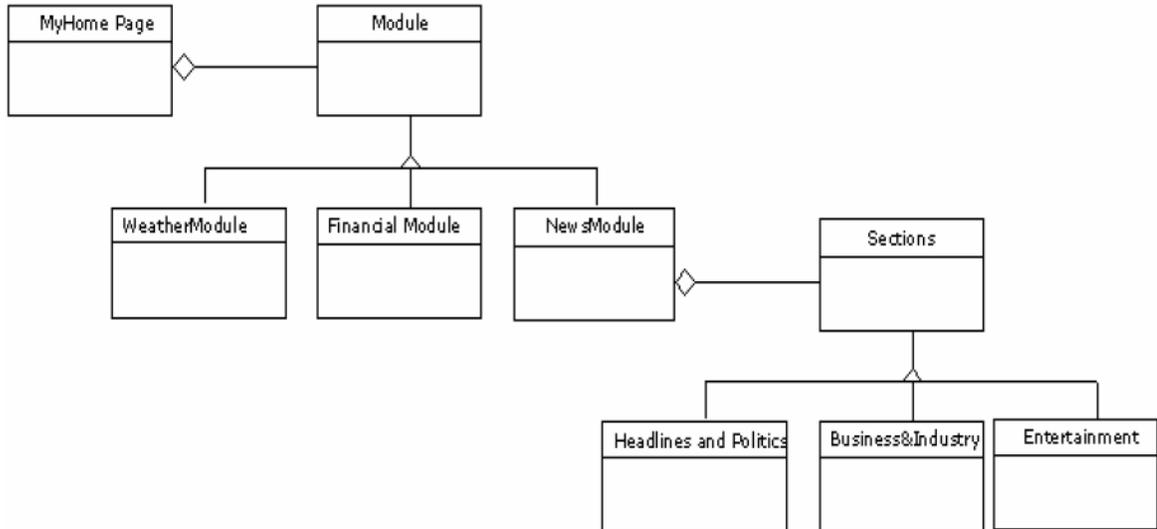
**Ilustración 7: Modelo Conceptual para una Tienda de CD's.**

### 8.2.3 Diseño Navegacional

La primera generación de aplicaciones web fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipertexto.

Según Koch (2002) El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales, los cuales se definen a continuación:

- **Esquema de Clases Navegacionales:** establece las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, llamadas navegacionales como son los nodos, los enlaces y otras clases que representan estructuras o formas alternativas de acceso a los nodos, como los índices y los recorridos guiados (Koch, ob. cit), ver Figura 6.



**Ilustración 8: Esquema básico navegacional para my.yahoo.com.**

• **Esquema de Contexto Navegacional:** es el que permite la estructuración del hiperespacio de navegación en sub-espacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se accede a un objeto (nodo) en un contexto determinado (Koch, ob. cit). Sánchez (s.f.) comenta con respecto a esta fase, “es la fase en que diseñamos la aplicación teniendo en cuenta los usuarios a los que va dirigida y los objetivos de la misma”, en pocas palabras, es la fase en que se plantea la manera de cómo será la navegación del usuario en el hiperdocumento.

Las tareas que se ejecutan son las siguientes:

- Se reorganiza la información representada en el modelo conceptual.
- Se estructura la vista de navegación sobre el modelo conceptual.

Una innovación de OOHDM es que los objetos sobre los cuales navega el usuario no son objetos conceptuales, sino otro tipo de objetos que se construyen a partir de uno o más objetos conceptuales, lo cual implica a su vez que el usuario navegue a través de enlaces, muchos de los cuales no se pueden derivarse directamente en relaciones conceptuales.

Este modelo implementa un conjunto de datos predefinidos, los cuales se describen a continuación:

- **Nodos:** son contenedores de información, éstos se definen como vistas orientadas a objetos de las clases conceptuales. Los nodos se pueden definir combinando atributos de clases relacionadas en el esquema conceptual (Sánchez, ob. cit).<sup>33</sup>
- **Enlaces:** son los que identifican las relaciones implementadas en el esquema conceptual. Las clases de los enlaces especifican sus atributos, comportamiento y los objetos fuentes del mismo. Estos representan las posibles formas de comenzar la navegación (Sánchez, ob. cit).
- **Estructuras de Acceso:** Las estructuras de acceso actúan como índices o diccionarios que permiten al usuario encontrar de forma rápida y eficiente la información deseada. Los menús, los índices o las guías de ruta son ejemplos de estas estructuras. Las estructuras de acceso también se modelan como clases, compuestas por un conjunto de referencias a objetos que son accesibles desde ella y una serie de criterios de clasificación de las mismas.
- **Contexto Navegacional:** Para diseñar bien una aplicación hipermedia, hay que prever los caminos que el usuario puede seguir, así es como únicamente se podrá evitar información redundante o que el usuario se pierda en la navegación. En OOHDM un contexto navegacional está compuesto por un conjunto de nodos, de enlaces de clases de contexto y de otros contextos navegacionales. Estos son introducidos desde clases de navegación (enlaces, nodos o estructuras de acceso), pudiendo ser definidas por extensión o de forma implícita.

---

<sup>33</sup> Sánchez, M. (s. f.). Interfaz de Usuario en el Desarrollo de un Simulador de Conducción

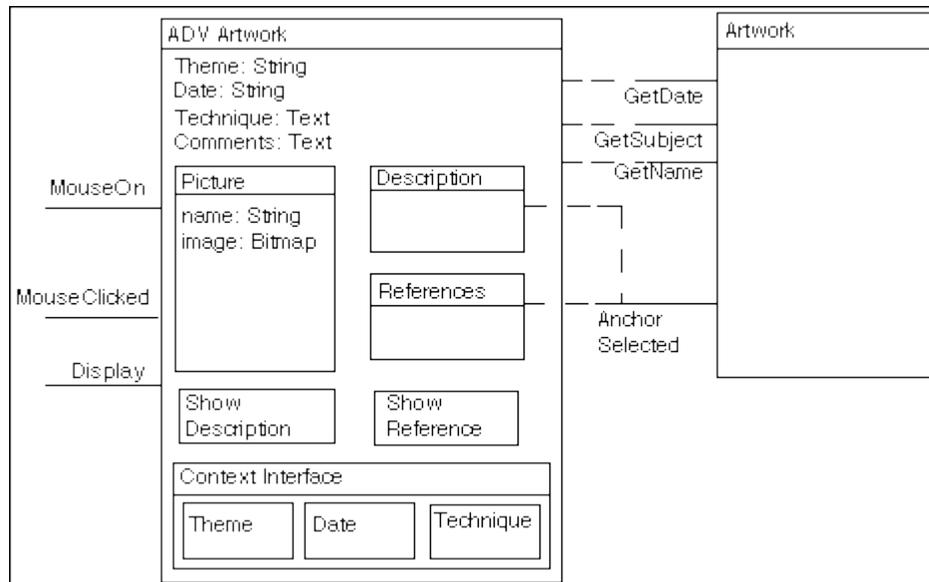
- **Clase de Contexto:** Es otra clase especial que sirve para complementar la definición de una clase de navegación. Por ejemplo, sirve para indicar qué información está accesible desde un enlace y desde dónde se puede llegar a él.

### 8.2.4 Diseño de Interfaz Abstracta

Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz. Según Schwabe, Rossi y Simone (s.f.) esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación.

El modelo de interfaz ADVs (Vista de Datos Abstractos), especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs (Vista de Datos Abstractos), en la pantalla real son hechas en la fase de implementación (Schwabe y otros, ob. cit.).

En la Figura 7 se puede observar un ejemplo de diagramas de configuración, y en la Figura 8 se pueden apreciar el diagrama de eventos que ocurre sobre el mismo.



**Ilustración 9: Diagrama de Configuración para los nodos ADV.**

### **8.2.5 Implementación**

En esta fase, el diseñador debe implementar el diseño. Según Schwabe y Rossi. (1998), hasta ahora todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación.

Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse.

A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada. Es de especial importancia el hacer notar que hoy en día, hay muchos y varios ambientes de implementación, con características distintas. Es claro, por ejemplo, que no se puede usar el mismo conjunto de líneas de acción en la traducción de un proyecto OOHDM para un documento HTML que para un programa en Macromedia Flash.

### **8.3 VENTAJAS DE OOHDM<sup>34</sup>**

De acuerdo con Silva y Mercerat (2001) OOHDM como metodología de desarrollo de aplicaciones de hipermedia, proporciona ventajas como:

- La recuperación de la información puede realizarse sin problemas.
- Se pueden crear enlaces entre nodos cualesquiera.
- La modularidad y la consistencia se potencian.
- Marco idóneo para la autoría en colaboración.
- Soporte a diferentes modos de acceso a la información.

En la actualidad, el desarrollo de software empleando patrones de proyecto, se encuentra en crecimiento según Gamma, Helm, Johnson y Vlissides (1995), sin embargo, su potencial se encuentra inexplorado en el campo de hipermedias,

---

<sup>34</sup> Silva, D. y Mercerat, B. (2001). Construyendo aplicaciones web con una metodología de diseño orientada a objetos.

especialmente a la hora de describir las arquitecturas para la navegación e interface en aplicaciones de hipermedia.

OOHDM según Schwabe y Rossi (1998) se torna diferente y superior a otras metodologías de desarrollo de aplicaciones de hipermedia al ofrecer la ventaja de patrones de proyecto poderosos como primitivas para la construcción del modelo navegacional de una aplicación hipermedia.

Puede decirse con base en todas las ventajas antes mencionadas que OOHDM toma en cuenta las crecientes necesidades de analistas y programadores de aplicaciones hipermedia, y se presenta como una técnica ideal de desarrollo para la producción de aplicaciones evolutivas de alta calidad.

Análisis de Factibilidad

# **CAPITULO II**

## **ANALISIS Y DISEÑO DEL SISTEMA**

## 1. Estudios de factibilidad.

### 1.1 Factibilidad técnica.

Compulab-Estelí cuenta con el equipo o hardware necesario para la implantación y sustento del sistema a desarrollar, cuenta con 2 laboratorios de 15 ordenadores, específicamente en el área administrativa, se cuenta con 3 computadores muy bien equipadas que servirían de host y una maquina nueva que fungiría como servidor durante la jornada laboral, en la siguiente tabla se puede observar las características de los equipos.

Equipo	Si	No	Marca	Modelo	Procesador	RAM	HDD	Cant.
Desktop-PC	X		Placa genérica Intel	DG43NB	Dual Core 3.0 GHZ	2 GB	250 GB	3
Impresora	X		Cannon	LBP3100	16 Ppm			1
Fotocopiador a- Impresora	X		Sharp	2040CS	20 Ppm			1
Servidor	X		Placa genérica Intel	DG41TY	Core 2 Duo	4 GB	500 GB	1
Patch Panel		X						
Switch		X						
Router		X						
Estabilizador	X		Tripp-lite	OMNI650LC D	8 Contactos			2
Batería	X		Forza	SL-761	6 Contactos			3
Cableado Estructurado		X						

**Tabla 6: Especificaciones de los equipos**

El sistema completo funcionará basado en un servidor local, por lo cual la institución educativa no tendrá que gastar en licencias de software, pues solamente necesitará un navegador web para acceder a la interfaz y MySQL como gestor de la base de datos.

Respecto al manejo de este sistema, no se necesitan extensivas capacitaciones especiales, pues el programa contará con una fácil navegación y un diseño sencillo de utilizar para cualquier persona, además de un sencillo manual FAQ para consultas.

Los administradores del sistema podrán también realizar modificaciones, introducción y eliminación de datos con mucha facilidad siempre desde el servidor local, utilizando un nombre de usuario y contraseña autorizados.

En cuanto al software de desarrollo, será suficiente con las siguientes herramientas o programas los cuales se detallan a continuación y son de poseen licencias libres excepto el manejador de contenido:

- ✓ WampServer 2.0
- ✓ PHP Version 5.3.0
- ✓ MySQL 5.1.36
- ✓ DreamWeaver Cs5

Se puede realizar una actualización de ese software cuando hay nuevas versiones disponibles de PHP y MySQL.

Los diseños de red que ya posee la institución y que se adecuan perfectamente a las necesidades, son como se describe a continuación:

## Diseño de red lógico

### Diseño Lógico de la Red

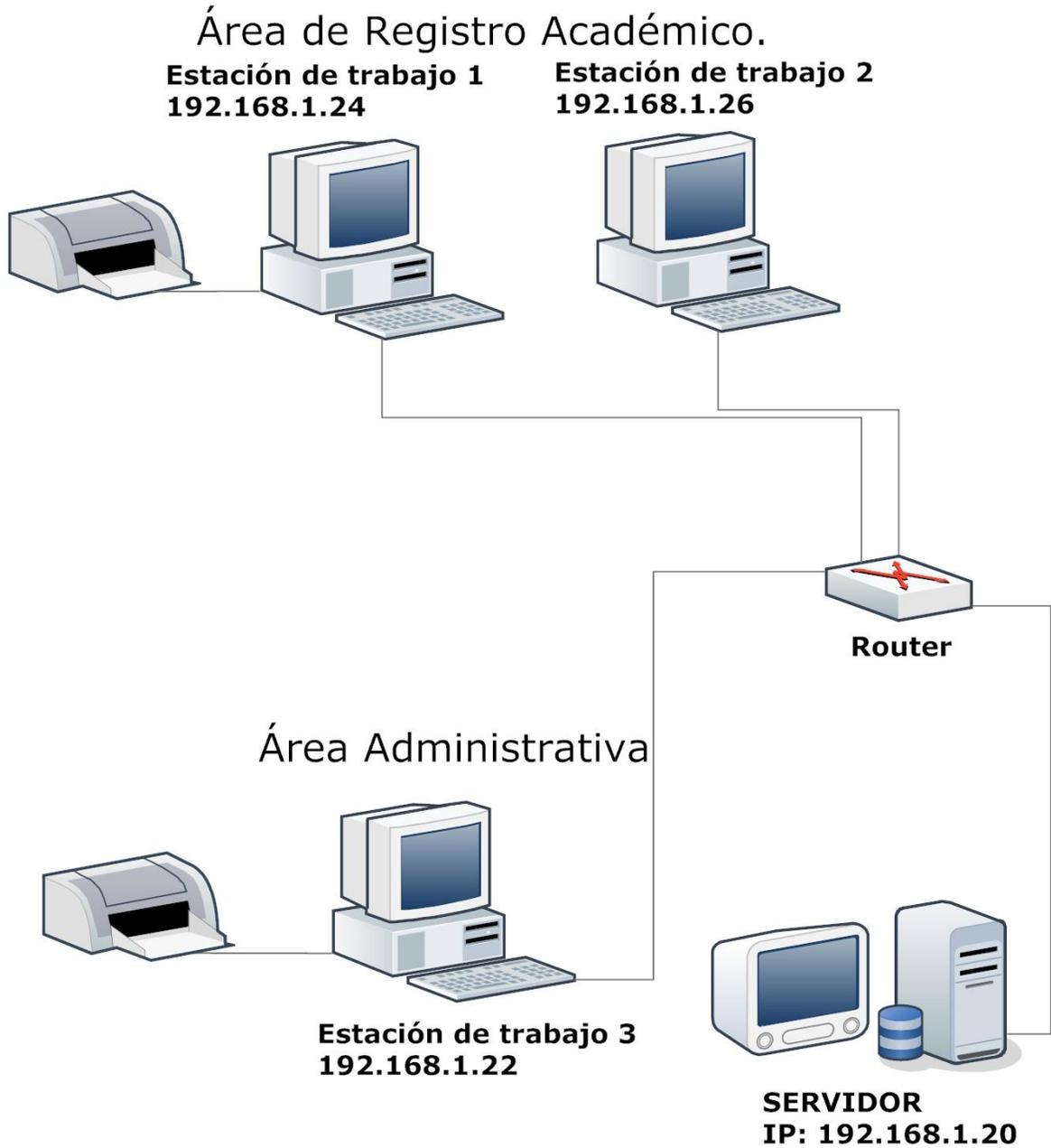
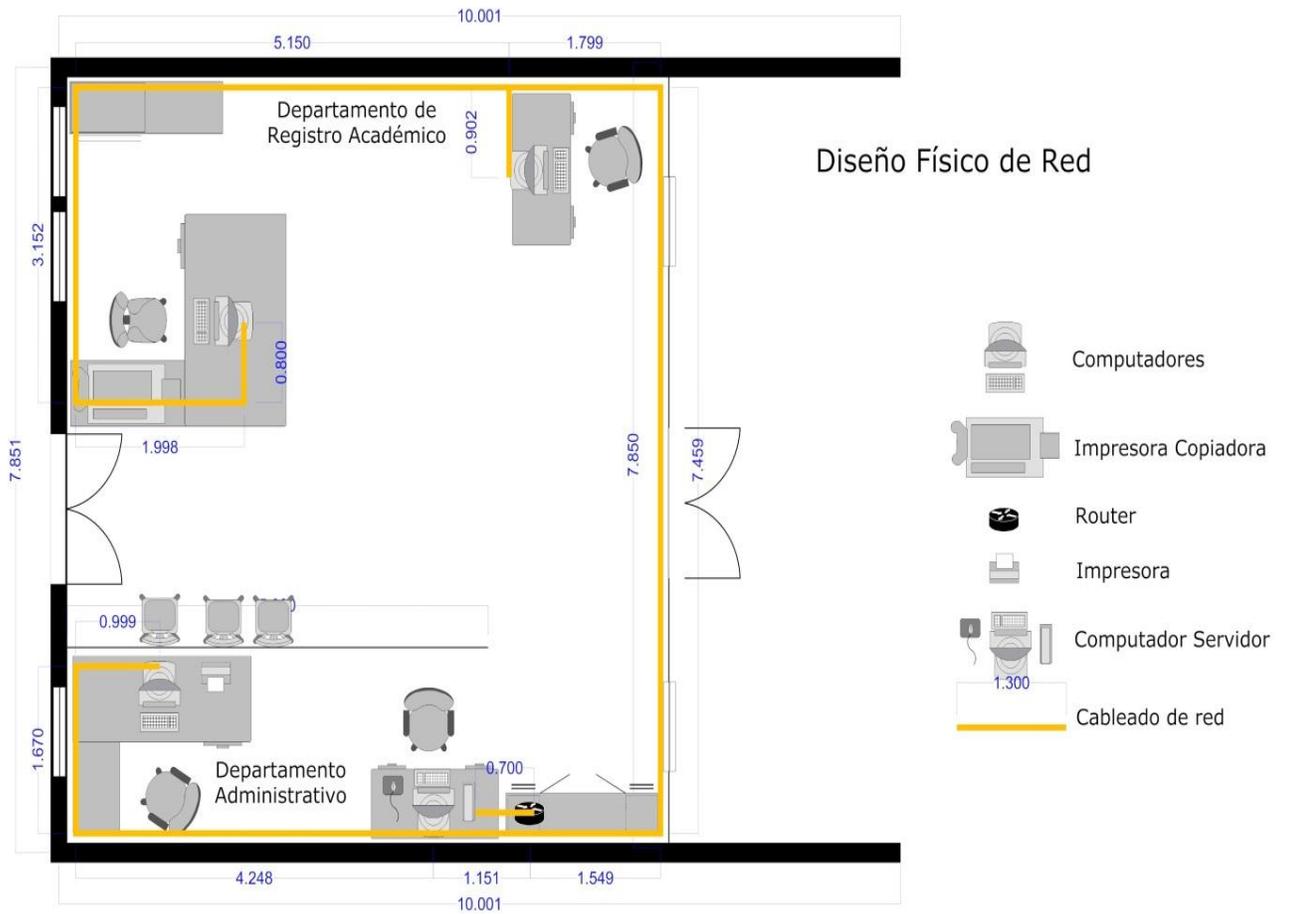


Ilustración 10: Diseño de red lógico

## Diseño de red físico.



**Ilustración 11: Diseño de red físico.**

### 1.2 Factibilidad operacional.

Es completamente posible que este nuevo sistema se use como se ha planeado, pues no representa procesos complejos ni difíciles de usar para los administradores o autoridades de la institución educativa. Además, lo que se pretende es sustituir un sistema obsoleto como es el registro manual, el cual tiene muchas desventajas.

El sistema ofrecerá grandes ventajas y seguridad, pues se tendrá toda la información en base de datos y disponible desde cualquier computadora con acceso al servidor local.

Por otro lado se logró identificar habilidades mínimas necesarias para el funcionamiento del sistema, los que se muestran con mayor detalle en la siguiente tabla, esta se obtuvo partir de la aplicación del el cuestionario por lo tanto el tiempo de adaptación del sistemas será relativamente corto.

Aspectos \ Usuarios	Administradora	Secretaria Académica
Estudios		
Primaria	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Secundaria	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Estudios superiores	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Curso caja	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Curso operador	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Habilidades		
Impresión de documentos	Alta: <u>X</u> Media: ___ Baja: ___	Alta: <u>X</u> Media: ___ Baja: ___
Agilidad en teclado	Alta: <u>X</u> Media: ___ Baja: ___	Alta: <u>X</u> Media: ___ Baja: ___
Uso de paquete Ofimático	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___

**Tabla 7: Habilidades mínimas para usuarios del sistema.**

Tomando en cuenta que la tendencia de todas las empresas modernas a manejar la información a través de la informática y que esto les genera un gran beneficio, este sistema tiene muy altas posibilidades de aceptación por parte de la institución entera, por lo que se aprecia que tiene una muy buena factibilidad operacional y que debería ser implementado.

### 1.3 Factibilidad económica.

Para determinar la factibilidad económica se debe tomar en cuenta el análisis de costos y beneficios asociados con este proyecto. El fin que se persigue es que los beneficios excedan a los costos.

Con este sistema se obtiene un excelente rendimiento y economía, pues no se hace necesario gastar en caros recursos de hardware ni en licencias caras de software debido a que todo el funcionamiento estará basado en servidor local y recursos ya existentes en la empresa. Los costos de implementación son muy económicos ya que la institución no necesitará gastar en complicadas capacitaciones debido a que el sistema será muy fácil de usar y la información estará segura y accesible las veinticuatro horas del día y los trescientos sesenta y cinco días del año.

Entre los costos cuantificables o tangibles se tienen el pago de la licencia de DreamWeaver para el desarrollo del sistema, así como también el pago que recibirán los programadores del sistema. Se tienen beneficios tangibles como lo es el evidente ahorro en equipo de hardware y en licencias de algún software.

A continuación se presenta un resumen detallado del análisis costo-beneficio para este sistema:

DETALLE	PRECIO	COMENTARIO
Hardware	\$500.00 de computadora	Este sistema requiere de uso de por lo menos 2 computadora (servidor/cliente) con los siguientes requerimientos mínimos:  PC Pentium IV 2.5 GHz  1 GB RAM  200 GB en disco duro  Impresora para reportes

		<p>Escritorio</p> <p>Resolución 800 x 600 pixeles</p> <p>Cualquier sistema operativo</p> <p>Navegador Internet Explorer, Mozilla o equivalente.</p> <p>Intranet o red local</p> <p>Como se observa, los requerimientos de hardware no son elevados y existen grandes beneficios.</p>
Red local	<p>\$15 Switch.</p> <p>\$15 de instalación.</p> <p>\$15 red cableada</p> <p>\$ 15 Modem</p>	<p>Se utilizará una PC (Servidor) para alojar la base de datos y es a donde se conectaran las otras PC disponibles.</p>
Software	\$30	<p>Solo se necesita comprar la licencia de DreamWeaver CS5, luego no se necesita comprar ninguna licencia de software, dado que las otras aplicaciones son Freeware y son PHP y MySql</p>
Costos de desarrollo e implementación	\$10000 Aprox.	<p>Este es el precio estimado del desarrollo e implementación del sistema de registro.</p>

**Tabla 8: Costos de Sistema**

Además de lo que ya se ha mencionado, se puede agregar también que este sistema tiene también otros beneficios como lo es la excelente portabilidad (pues no necesita de hardware especial sino que funciona desde cualquier computadora o dispositivo que cuente con un navegador web y acceso al servidor local).

Luego de este análisis, se considera que este sistema es factible porque tiene más ventajas que desventajas y resultaría útil para un centro escolar que quiera tener un eficiente control de matrículas como el que se propone en este documento.

### **1.3.1 Estimación de Costo del Software**

Para la estimación del costo del software se utilizó la metodología de COCOMO por la madurez que tiene en este tipo de cálculos. Primero se presentan conceptos claves de COCOMO y luego se realiza el cálculo del costo del software a desarrollar.

#### **1.3.1.1 Definición**

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. El autor de este modelo es el Dr. Barry Boehm y lo ha descrito en sus libros *Software Engineering Economics* (Prentice-Hall, 1981) and *Software Cost Estimation with COCOMO II* (Prentice Hall, 2000).

El modelo original COCOMO ha tenido mucho éxito pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. COCOMO II apunta hacia los proyectos software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años.

### 1.3.1.2 Metodología

En esta sección se aborda el cálculo más fundamental en el modelo de COCOMO - el uso de las Ecuaciones de Esfuerzo para estimar el número de Persona-meses exigido desarrollar un proyecto. La mayoría de los otros resultados de COCOMO, incluso las estimaciones para la duración, requisitos, y mantenimiento, se deriva de esta cantidad.

#### **Líneas de Código Fuente (SLOC) / *Source Lines of Code***

Los cálculos de COCOMO están basados en tu estimación del tamaño de un proyecto, medido en las Líneas de Código Fuente (SLOC).

El manual de definición del modelo COCOMO II define las Líneas de Código Fuente usando listas de control adaptadas del Instituto de Ingeniería de Software. Existen herramientas que examinan archivos fuentes y cuentan las Líneas de Código Fuente usando las definiciones de COCOMO II.

Una Línea de Código Fuente (SLOC) se define así:

“Sólo las líneas fuentes que son entregadas como parte del producto, están incluidas - manejadores de prueba y software de apoyo se excluye.

“Las líneas de la fuente son creadas por el personal del proyecto – código generado por los generadores de las aplicaciones se excluye.

“Una línea de código es una declaración lógica.

“Se cuentan las declaraciones como líneas de código

“No se cuentan los comentarios como líneas de código

El modelo original COCOMO 81 fue definido en términos de Instrucciones Fuentes Entregadas, el cual es muy similar a Líneas de Código Fuente. El modelo más reciente de COCOMO II está definido en términos de Líneas de Código Fuente. La principal diferencia entre IFE y LCF es que una sola Línea lógica de Código Fuente puede ser varias líneas físicas. Por ejemplo, una

declaración "si-entonces-resto" se contaría como una LCF, pero podría contarse como varias IFE.

Este proyecto contiene 12698 sloc y la base de datos pesa 14 Kb.

### 1.3.1.2.1 Factores de Escala (SF Scale Factors)

Los cinco Factores de Escala de COCOMO II determinan el exponente usado en la Ecuación de Esfuerzo:

No	Factores	Referencia
1	Precedencia (PREC)	Si un producto es similar a muchos proyectos que se han desarrollado previamente, entonces la precedencia es alta.
2	Flexibilidad de Desarrollo (FLEX)	El Software debe ajustarse a prerequisites, finalización anticipada, especificaciones de interfaz externa.
3	Arquitectura / Resolución de Riesgo (RESL)	Minuciosidad del diseño y Eliminación de riesgos por revisión de diseño preliminar
4	Cohesión del Equipo (TEAM)	Explica los recursos de turbulencia y entropía del proyecto debido a dificultades en la sincronización de los implicados en el proyecto, usuarios, clientes, desarrolladores, etc...
5	Madurez del Proceso (PMAT)	El procedimiento para determinar PMAT se obtiene a través del Modelo de Madurez de Capacidad del Instituto de Ingeniería del Software (CMM). El periodo de tiempo para medir la madurez del proceso es el momento en el que el proyecto comienza.

**Tabla 9: Factores de Escala**

Factores de escala	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra	Este proyecto
PREC	Completamente nuevo	En gran manera nuevo	Un poco nuevo	Generalmente familiar	En gran manera familiar	Completamente familiar	6.20
	6.20	4.96	3.72	2.48	1.24	0.00	
FLEX	Riguroso	Relax Ocasional	Algo de relax	Conformidad general	Algo de conformidad	Metas generales	2.03
	5.07	4.05	3.04	2.03	1.01	0.00	
RESL	Poco 20%	Algo 40%	A menudo 60%	Generalmente 75%	Muchas veces 90%	Casi Siempre 90%	5.65
	7.07	5.65	4.24	2.83	1.41	0.00	
TEAM	Interacciones muy difíciles	Interacción algo difícil	Interacciones básicamente cooperativas	Cooperativo en gran manera	Altamente cooperativo	Excelentes Interacciones	0.00
	5.48	4.38	3.29	2.19	1.10	0.00	
PMAT	Sw-cmm nivel 1 bajo	Sw-cmm nivel 1 alto	Sw-cmm nivel 2	Sw-cmm nivel 3	Sw-cmm nivel 4	Sw-cmm nivel 5	4.68
	7.80	6.24	4.68	3.12	1.56	0.00	
$\sum_{j=1}^5 SF_j$							18.56

**Tabla 10: Factores de Escala 2**

### 1.3.1.2.2 Factores de Costo

COCOMO II tiene 17 factores de costo, se tienen que evaluar el proyecto, el ambiente de desarrollo y el equipo para fijar el valor de cada factor de costo. Los factores de costos son factores multiplicativos que determinan el esfuerzo requerido para completar el proyecto de software.

COCOMO II define cada uno de los drivers de costos y el multiplicador de esfuerzo asociado con cada valuación.

No	Factor	Referencia
1	Fiabilidad Requerida de Software ( <b>RELY</b> )	Esta es la medida de hasta qué punto el software debe realizar su función esperada durante un periodo de tiempo. Si el efecto de un fracaso es sólo una molestia ligera entonces RELY es Bajo. Si un fallo arriesgase vidas humanas entonces RELY es Muy Alto.
2	Medida del Volumen de Datos ( <b>DATA</b> )	Esta medida intenta capturar como que afecta en el desarrollo del producto, grandes requerimientos de datos. La medida se determina calculando D/P. La razón por la que es importante considerar el tamaño de la Base de Datos es por el esfuerzo necesario para generar datos de prueba que se usarán para ejecutar el programa.  $\frac{P}{D} = \frac{\text{DBSize (Bytes)}}{\text{PrgSize (SLOC)}}$ DATA se valora como Bajo si D/P es menor que 10 y Muy Alto si es mayor que 1000
3	Complejidad del Producto ( <b>CPLX</b> )	La complejidad se decide en 5 áreas: Funcionamiento de control, Funcionamiento computacional, Funcionamiento de Dispositivos dependientes, Funcionamiento del sector de datos y Funcionamiento del Gestor de Interfaz de Usuario. Se seleccionará el área o combinación de áreas que caracterizan al producto o a un subsistema del producto.

		La medida de complejidad es la media subjetiva de estas áreas.
4	Reutilización Requerida ( <b>RUSE</b> )	Este driver de coste explica el esfuerzo adicional necesario para construir componentes pensados para ser reutilizados en proyectos presentes o futuros
5	Documentación Asociada a las Necesidades del Ciclo de Vida ( <b>DOCU</b> )	Nivel de adecuación de la documentación del proyecto a las necesidades de su ciclo de vida.
6	Restricción del Tiempo de Ejecución ( <b>TIME</b> )	Esta es una medida de la restricción del tiempo de ejecución impuesta en un sistema software. Las medidas se expresan en términos de porcentaje de tiempo de ejecución disponible que se espera que sea usado por el subsistema o sistema que consume el recurso de tiempo de ejecución.
7	Restricción de Almacenamiento Principal ( <b>STOR</b> )	Esta medida representa el grado de restricción de almacenamiento principal impuesto a un sistema o subsistema software. Dado el notable aumento en el tiempo de ejecución disponible del procesador y de almacenamiento principal, uno puede cuestionar si estas variables de restricción son todavía pertinentes.  Los valores van desde nominal, menos que el 50%, a Extra Alto, 95%
8	Volatilidad de la Plataforma ( <b>PVOL</b> )	“Plataforma” se usa aquí para significar la complejidad del Hardware y Software (OS, DBMS, etc.) que el producto software necesita para realizar sus tareas. Si el software a desarrollar es un sistema operativo, entonces la plataforma es el hardware del ordenador. Si se desarrolla un Gestor de Base de Datos, entonces la plataforma es el hardware y el sistema operativo. Si se desarrolla un browser de texto de red, entonces la plataforma es la red, el hardware del ordenador, el sistema operativo y los repositorios de información distribuidos. La plataforma incluye

		cualquier compilador o ensamblador que soporta el desarrollo del sistema software. Los valores van desde Bajo donde cada 12 meses hay un cambio importante, hasta Muy Alto, donde hay un cambio importante cada 2 semanas
9	Habilidad del Analista (ACAP)	Los analistas son personal que trabaja en los requisitos de diseño de alto nivel y en diseño detallado. Los atributos principales que deben considerarse en esta medida son la habilidad de análisis y diseño, la eficiencia y minuciosidad y la habilidad para comunicar y cooperar. La medida no debe considerar el nivel de experiencia del analista, eso se mide con AEXP.
10	Habilidad del Programador (PCAP)	La evaluación debe basarse en la capacidad de los programadores como un equipo, más que individualmente. La habilidad del programador no debe considerarse aquí, eso se mide con AEXP.
11	Continuidad del Personal (PCON)	La escala de valores de PCON se mide en términos del movimiento de personal del proyecto anualmente: desde 3%, Muy Alto, hasta el 48%, Muy Bajo
12	Experiencia en las Aplicaciones (AEXP)	Esta medida depende del nivel de experiencia en aplicaciones del equipo de proyecto al desarrollar sistemas o subsistemas software. Un valor muy bajo para experiencia en aplicaciones es menor que 2 meses. Un valor muy alto es por experiencia de 6 años o más.
13	Experiencia Personal (PEXP)	Este driver de coste de Diseño Anticipado combina los 3 drivers de coste de Post-Arquitectura siguientes: Experiencia (AEXP), Experiencia en la Plataforma (PEXP) y Experiencia en el Lenguaje y Herramientas (LTEX). Un valor extra bajo es menor igual que 3 meses y uno extra alto es igual a 6 años.
14	Experiencia en la Herramienta y en el Lenguaje (LTEX)	Esta es una medida del nivel de experiencia en el lenguaje de programación y en la herramienta software del equipo de proyecto que desarrolla el sistema o subsistema software. Además de la experiencia programando en un lenguaje específico, las herramientas que dan soporte también influyen en el tiempo de

		desarrollo. Tener una experiencia de menos de 2 meses da un valor Bajo. Si es de 6 ó más años el valor es Muy Alto
15	Uso de Herramientas Software ( <b>TOOL</b> )	Los valores para la herramienta van desde edición y código simple, Muy Bajo, hasta herramientas integradas de gestión del ciclo de vida, Muy Alto
16	Desarrollo Multisitio ( <b>SITE</b> )	Determinar la medida de este driver incluye el cálculo y la medida de 2 factores: Localización del lugar (desde totalmente localizado hasta distribución internacional) y soporte de comunicación (desde correo de superficie y algún acceso telefónico hasta multimedia totalmente interactivo)
17	Planificación Temporal ( <b>SCED</b> )	Es usado para tomar en cuenta que un proyecto desarrollado en un calendario acelerado requerirá más esfuerzo que un proyecto desarrollado en su calendario óptimo. Una compresión del calendario del 74% se evalúa como Muy Bajo. Un alargamiento del 160% se valora como Muy Alto

**Tabla 11: Factores de Costo**

	FACTOR	MUY BAJO	BAJO	NORMAL	ALTO	MUY ALTO	EXTRA	ESTE PROYECTO
PRODUCTO	RELY	Inconvenientes insignificantes, que afectan solamente a los desarrolladores.	Mínimas pérdidas al usuario, fácilmente recuperables	Pérdidas moderadas al usuario recuperables sin grandes inconvenientes	Pérdida financiera elevada o inconveniente humano masivo	Vida humana en riesgo		0.82
		0.82	0.92	1.00	1.10	1.26	-	
	DATA		DB bytes/Pgm SLOC <10	10<=D/P<100	100<=D/P<1000	D/P >0 1000		14336/12698=1.1289
		-	0.90	1.00	1.14	1.28	-	0.9

	CPLX	<p>Pocas estructuras sin anidamiento: DO, CASE, IF_THEN_ELSE.</p> <p>Composición modular simple por medio de llamadas a procedimientos o simples script</p>	Estructuras anidadas sencillas	<p>Uso mayoritario de anidamientos sencillos.</p> <p>Algunos controles entre módulos.</p> <p>Tablas de decisión.</p> <p>Pasaje de mensajes o llamadas a subrutinas.</p>	<p>Programación estructurada con alto grado de anidamiento con predicados compuestos.</p> <p>Control de cola y pila.</p> <p>Procesamiento distribuido.</p> <p>Control en tiempo real con un procesador</p>	<p>Codificación recursiva.</p> <p>Manejo de interrupciones con prioridad fija.</p> <p>Sincronización de tareas, complejas llamadas a subrutinas</p>	<p>Planificación múltiple de recursos con cambio dinámico de prioridades.</p> <p>Control al nivel de microcódigo o. Control en tiempo real distribuido</p>	1
		0.73		0.87				

	RUSE		Ningún componente reusable	Reusable dentro del mismo proyecto	Reusable dentro de un mismo programa	Reusable dentro de una misma línea de productos	Reusable dentro de múltiples líneas de producto	1.15
		-	0.95	1.00	1.07	1.15	1.24	
	DOCU	Muchas necesidades del ciclo de vida sin cubrir	Algunas necesidades del ciclo de vida sin cubrir	Necesidades del ciclo de vida cubiertas en su justa medida	Necesidades del ciclo de vida cubiertas ampliamente	Necesidades del ciclo de vida cubiertas excesivamente		0.91
		0.81	0.91	1.00	1.11	1.23	-	
PLATAFORMA	TIME			Uso de <= 50% del tiempo de ejecución	70%	85%	95%	1
		-	-	1.00	1.11	1.29	1.63	

	STOR			Uso de <= 50% del porcentaje total de almacenamiento	70%	85%	95%	1
		-	-	1.00	1.05	1.17	1.46	
	PVOL		Un cambio principal cada 12 meses. Un cambio menor todos los meses	Cambio principal cada 6 meses. Cambio menor cada 2 semanas	Cambio principal cada 2 meses. Cambio menor uno por semana	Cambio principal cada 2 semanas. Cambio menor cada 2 días		0.87
		-	0.87	1.00	1.15	1.30	-	
PERSONA L	ACAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil		1.19
		1.42	1.19	1.00	0.85	0.71	-	
	PCAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil		0.88
		1.34	1.15	1.00	0.88	0.76	-	

	PCON	48 % por año	24 % por año	12 % por año	6% por año	3 % por año		1.29
		1.29	1.12	1.00	0.90	0.81	-	
	AEXP	<= 2 meses	<= 6 meses	1 año	3 años	6 años		0.88
		1.22	1.10	1.00	0.88	0.81	-	
	PEXP	<= 2 meses	<= 6 meses	1 año	3 años	6 años		0.91
		1.19	1.09	1.00	0.91	0.85	-	
	LTEX	<= 2 meses	<= 6 meses	1 año	3 años	6 años		0.91
		1.20	1.09	1.00	0.91	0.84	-	
PROYECTO	TOOL	Herramientas que permiten editar, codificar, depurar	Herramientas simples con escasa integración al proceso de desarrollo	Herramientas básicas, integradas moderadamente	Herramientas robustas y maduras, integradas moderadamente	Herramientas altamente integradas a los procesos, métodos y reuso		1
		1.17	1.09	1.00	0.90	0.78	-	

	SITE Ubicación Espacial	Internacional	Multi-ciudad y multicompañía	Multi-ciudad o multicompañía	Misma ciudad o área metropolitana	Mismo Edificio o complejo	Completamente Centralizado	0.8
	SITE Comunicación	Algún teléfono, mail	Teléfonos individuales, FAX	Email de banda angosta	Comunicaciones electrónicas de banda ancha	Comunicaciones electrónicas.	Multimedia Interactiva	
		1.22	1.09	1.00	0.93	0.86	0.80	
	SCED	75% del nominal	85% del nominal	100% del nominal	130% del nominal	160% del nominal		1
		1.43	1.14	1.00	1.00	1.00	-	
Factor de ajuste del Esfuerzo= $EAF = \prod_{j=1}^{17} CF_j$								0.5292

Tabla 12: Factores de Costo 2

### **1.3.1.2.3 Modelo Post-Arquitectura**

El modelo post-arquitectura de cocomo II, basado en 17 drivers de costos, es la forma de uso más común de cocomo II. Si no sabes suficiente sobre tu proyecto, para fijar los 17 drivers de costo (EJ: es muy temprano en el proceso de desarrollo para conocer todos los detalles), puedes usar el modelo Diseño Anticipado en vez de este modelo. Este fue el modelo que se adoptó debido al avance significativo en el proyecto.

### **1.3.1.2.4 Modelo Diseño Anticipado**

COCOMO II incluye un modelo de diseño anticipado, que usa solamente 7 drivers de costos. Este modelo es conveniente para estimaciones anticipadas, antes de que todos los detalles del proyecto sean resueltos.

Cada uno de los driver de costo del Diseño Anticipado corresponde a varios drivers de costo del Modelo Post-Arquitectura. Utiliza las mismas ecuaciones y factores de escala así como el modelo más completo de post-arquitectura.

### **1.3.1.2.5 Ecuación del Esfuerzo**

El modelo de COCOMO II realiza su estimación de esfuerzo requerido (medido en Personas-Meses- PM) basado primeramente en tu estimación del tamaño del proyecto de software (medido en miles de SLOC, KSLOC):

$$\text{Esfuerzo} = 2.94 * \text{EAF} * (\text{KSLOC})^E$$

Donde

EAF Es el Factor de Ajuste del Esfuerzo, derivado de los drivers de Costo.

E Es un exponente derivado de los 5 drivers de Escala.

$$E = 0.91 + 0.01 \sum_{j=1}^5 SFj$$

### 1.3.1.2.6 Factor de Ajuste del Esfuerzo

El factor de ajuste del esfuerzo en la ecuación de esfuerzo es simplemente el producto de los multiplicadores de esfuerzo correspondientes a cada uno de los drivers de costos de tu proyecto.

$$EAF = \prod_{j=1}^{17} CFj$$

### 1.3.1.2.7 Ecuación De Horario

La ecuación de horario de COCOMO II, predice el número de meses requeridos para completar el proyecto de software. La duración de un proyecto está basada en el esfuerzo pronosticado por la ecuación de esfuerzo.

$$\text{Duración} = 3.67 * (\text{Esfuerzo})^{SE}$$

Donde

Esfuerzo : Es el resultado de la Ecuación de Esfuerzo.

$$SE = 0.28 + 0.002 * \sum_{j=1}^5 SFj$$

Promedio de Personas = Esfuerzo / Duración

### 1.3.1.3 Cálculo de Costos

En esta sección se desarrollará la estimación del costo de este proyecto aplicando la metodología de COCOMO II mencionada anteriormente.

#### Fórmulas

a) Esfuerzo = 2.94 \* EAF \* (KSLOC)<sup>E</sup> = *Personas-Meses*

a.1)  $EAF = \prod_{j=1}^{11} CF_j$

a.2) KSLOC Miles de Líneas de Código (Tamaño del Programa)

a.3)  $E = 0.91 + 0.01 \sum_{j=1}^5 SF_j$

b) Duración = 3.67 \* (Esfuerzo)<sup>SE</sup>

b.1)  $SE = 0.28 + 0.002 * \sum_{j=1}^5 SF_j$

c) Promedio de Personas = Esfuerzo / Duración

#### Desarrollo

a) Esfuerzo = 2.94 \* EAF \* (KSLOC)<sup>E</sup> = *Personas-Meses*

Esfuerzo = 2.94 \* 0.5292 \* (12.698)<sup>1.0956</sup>

Esfuerzo = 2.94 \* 0.5292 \* 16.1902

Esfuerzo = 25.1895 personas - meses

a1) EAF = 0.5292 Ver tabla de factores de costos

a2) KSLOC=12.698 El sistema tiene cerca de 13,000 líneas de código.

a3)  $E = 0.91 + 0.01 \sum_{j=1}^5 SF_j$

E = 0.91 + 0.01 \* (18.56)

$$E = 1.0956$$

- b) Duración =  $3.67 * (\text{Esfuerzo})^{\text{SE}}$   
Duración =  $3.67 * (25.1895)^{0.3173}$   
Duración =  $3.67 * (2.7836)$   
Duración = 10.2158 meses

b1) 
$$SE = 0.28 + 0.002 * \sum_{j=1}^5 SF_j$$

$$SE = 0.28 + 0.002 * 18.56$$

$$SE = 0.3171$$

- c) Promedio de Personas = Esfuerzo / Duración  
Promedio de Personas =  $25.1895 / 10.2158$   
Promedio de Personas = 2.4657  
Se necesitan 2 personas para el proyecto

- d) *Costo Total del Software = Promedio de personas \* Duración \* Salario*

*Salario: El salario correspondería al cargo de Analista Programador y en el MECD está valorado en C\$12,000.00 (última actualización 2006)*

$$CTP = 2 * 10 * 12000$$

$$CTP = \text{C\$ } 240,000$$

*El costo total del software será de doscientos cuarenta mil córdobas netos.*

## **2. Descripción de Procesos de Negocios**

Para conocer y describir el método de trabajo de la institución, se procedió a recopilar información a través de la técnica de la entrevista y observación directa.

Dentro de la recopilación, tratamiento y recepción de la información obtenida a través de la entrevista concluimos que el proceso detallado se describe de la siguiente forma.

### **2.1 Definición de la situación Actual.**

Actualmente Compulab-Estelí está ubicado de Petronic El Carmen 75 vrs al este, esta institución brinda servicios de educación técnica y cursos avanzados ligados a las tecnologías, y está adscrita al INATEC.

Esta institución realiza sus operaciones de inscripción, registro académico y arancelario de manera manual o en hojas de cálculo de Microsoft Excel, la información (recibos, hojas de matrículas, cuadros de notas y tarjetas de pago) es almacenada en expedientes impresos en los archivadores de la institución.

Existen varios procesos que son ejecutados en el centro en relación al servicio de sus estudiantes veámoslos a continuación:

- Matrícula
- Control de Notas
- Elaboración Certificados de Notas
- Creación de Grupos.
- Control de Solvencias.

Reportes:

- Lista de estudiantes y docentes.
- Lista de insolventes.
- Lista de cursos y horarios.
- Lista de grupos.

## **2.2 Descripción del entorno.**

La institución cuenta con tres laboratorios de computación bien equipados los que responden por completo a la necesidad del centro como por ejemplo scanner, proyectores, quemadores, y todos sus medios son relativamente nuevos.

Además se cuenta con un plantel de trabajadores:

- ✓ 3 Administrativos
- ✓ 12 Docentes
- ✓ 2 Personas encargadas de seguridad y limpieza.

### **2.2.1 Proceso de Matricula.**

El estudiante solicita información sobre los cursos disponibles al área de registro académico, que es el encargado de proveer dicha información.

Una vez el estudiante tiene la información de los cursos y desea matricularse, procede la secretaria académica a llenar la hoja de matrícula del estudiante con los siguientes datos:

- ✓ Información general del Estudiante.

Nombre completo, Cedula de identidad, Estado civil, Edad, Dirección, Ciudad, Teléfono, E-mail.

✓ Antecedentes académicos.

Nivel, Grado que cursa, Año aprobado, Título obtenido, Nombre del centro de estudio, Cursos que han llevado, Conocimientos de informática, Trabajo, Nombre del centro de trabajo.

✓ Información general del curso que se ha elegido.

Curso inscrito, Fecha de inscripción, Modalidad, fecha de inicio, fecha de finalización, horario, Valor de matrícula, Valor de la mensualidad.

Una vez llenada la hoja de matrícula, el estudiante cancela el valor de la matrícula del curso deseado en caja, desde donde se envía el comprobante de pago a registro académico.

Con la información recopilada se registra al estudiante en el curso, es decir, se levanta la lista de estudiantes que desean determinado curso.

La hoja original es almacenada de forma física, en los archivadores de la institución, mientras se le entrega una copia al estudiante.

## Diagramas de Actividad

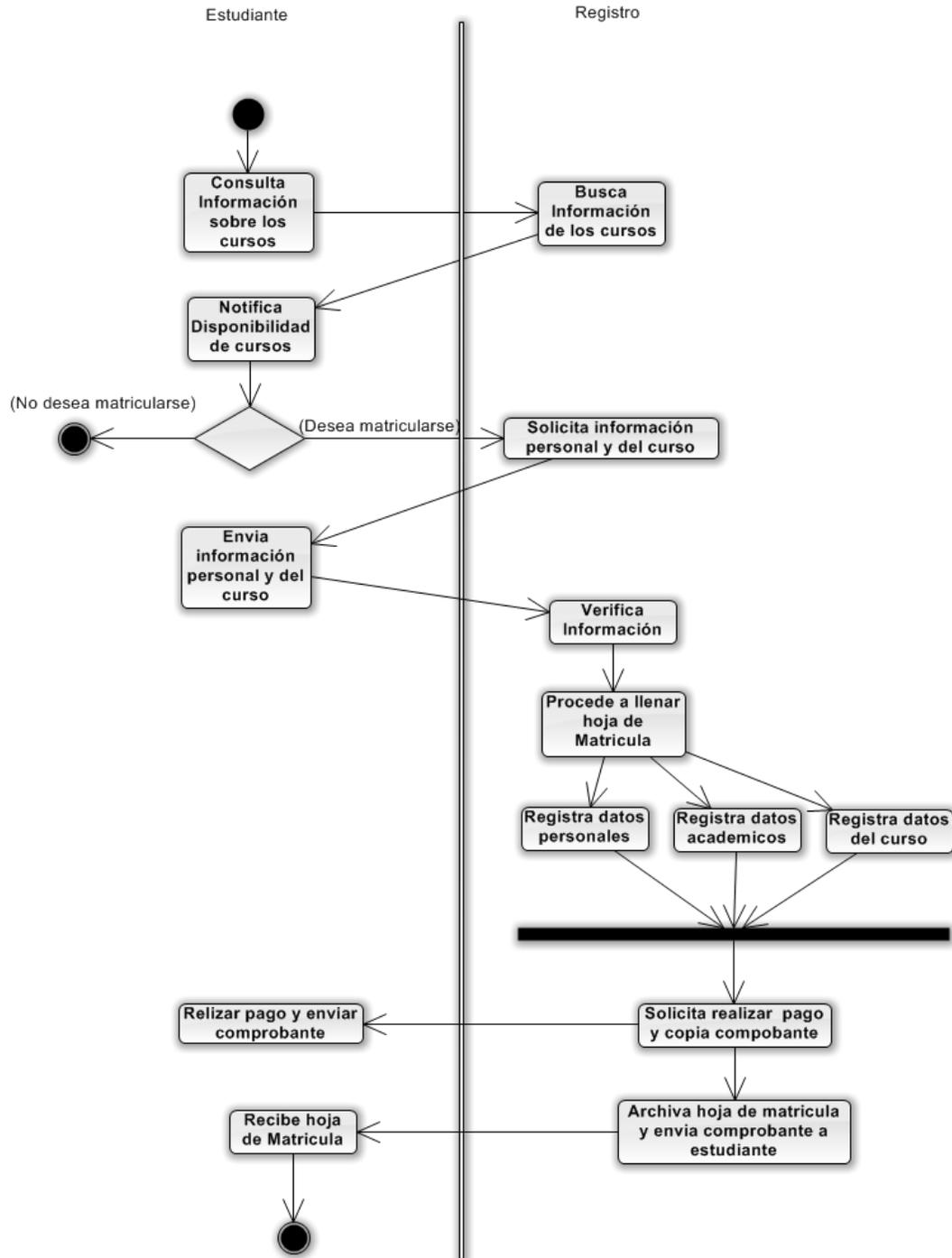


Ilustración 12: Diagrama de Actividad – Proceso de Matricula

### **2.2.2 Control de notas**

Los cursos se dividen en módulos, la cantidad de módulos dependen del curso, generalmente estos módulos son evaluados de forma práctica y teórica dependiendo del profesor.

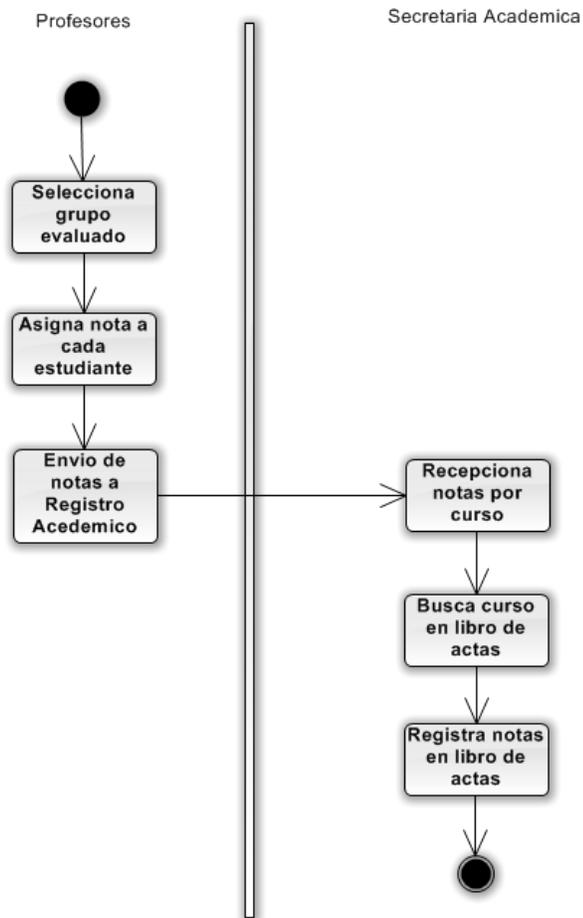
Una vez que el profesor realice las evaluaciones del módulo impartido. Procede a registrar la nota obtenida por cada estudiante, en una hoja denominada lista de notas. La lista de notas está compuesta por los siguientes campos:

- Nombre del profesor.
- Nombre del curso.
- Nombre del módulo.
- Modalidad y fecha.
- Listado con el nombre del estudiante y su correspondiente nota.

Luego esta lista es enviada a registro académico, para ser registrada en el libro de actas de notas según los cursos que se están impartiendo.

El libro de acta de notas está estructurado de la misma forma que la lista de notas que se le proporciona a cada uno de los profesores.

En caso de que un estudiante repruebe o falte a un módulo, se le programará una tutoría.



**Ilustración 13: Diagrama de Actividad – Control de Notas**

### 2.2.3 Elaboración de certificados de notas.

Una vez que finalicen los cursos, el departamento de registro académico debe emitir los certificados de notas a cada uno de los estudiantes o empresas que soliciten información de sus becados, de igual forma dichos certificados de notas son enviados al INATEC para que esta se encargue de elaborar sus respectivos acreditaciones y transmite de diplomas. Por otro lado, hay algunos estudiantes que también solicitan constancias para fines de carácter personal. Para todos los casos, el departamento de registro académico debe buscar en los libros de actas para poder tener a su disposición la información solicitada.

También los estudiantes pueden solicitar una constancia de estudios.  
 Estos certificados y constancias tienen un costo que debe ser cancelado en caja.

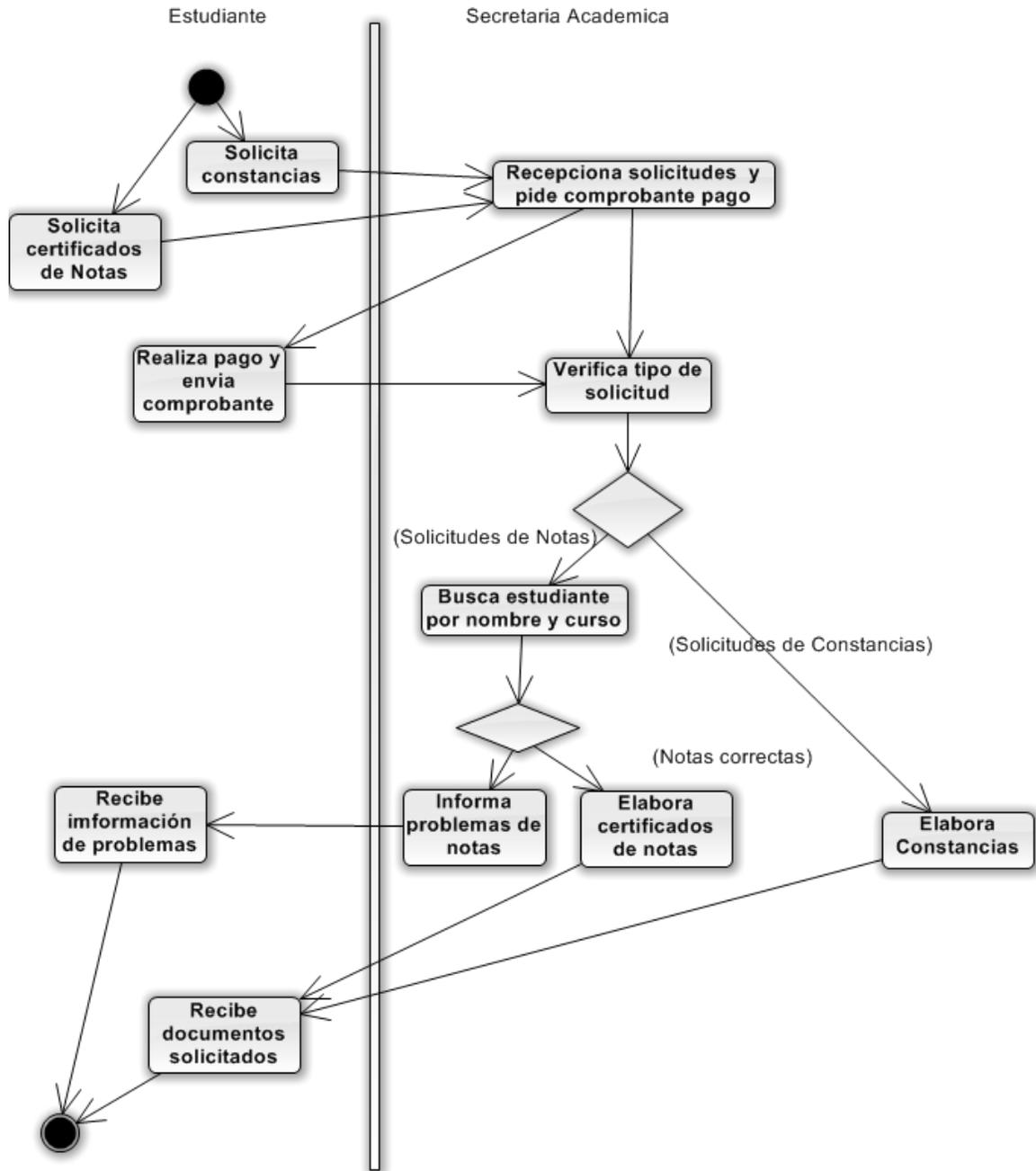


Ilustración 14: Diagrama de Actividad – Certificado de Notas

## 2.2.4 Creación de Grupos de Clases

El proceso de la creación de Grupos de clase se da cuando el administrador establece las condiciones de apertura de cada curso. Cabe destacar que en este proceso también interviene registro académico ya que se encarga de establecer las modalidades y el horario.

Los pasos que se siguen son los siguientes:

Se establece el nombre del curso y los módulos que contendrá, la modalidad, fecha de inicio y final del curso.

Una vez que el periodo de apertura de matrícula ha finalizado, registro académico procede asignar a cada profesor los correspondientes grupos para impartir clases.

El proceso de asignación se da de acuerdo a la especialidad y el grado de preparación de cada profesor. Cabe mencionar que un profesor puede impartir múltiples asignaturas.

La asignación es registrada en un documento de Word y es impresa e impartida a cada profesor.

Dicho documento contiene los siguientes datos:

- Nombre del Profesor
- Nombre de la Carrera
- Nombre de la Asignatura
- Grupo
- Horario
- Listado de estudiantes que pertenecen al grupo asignado

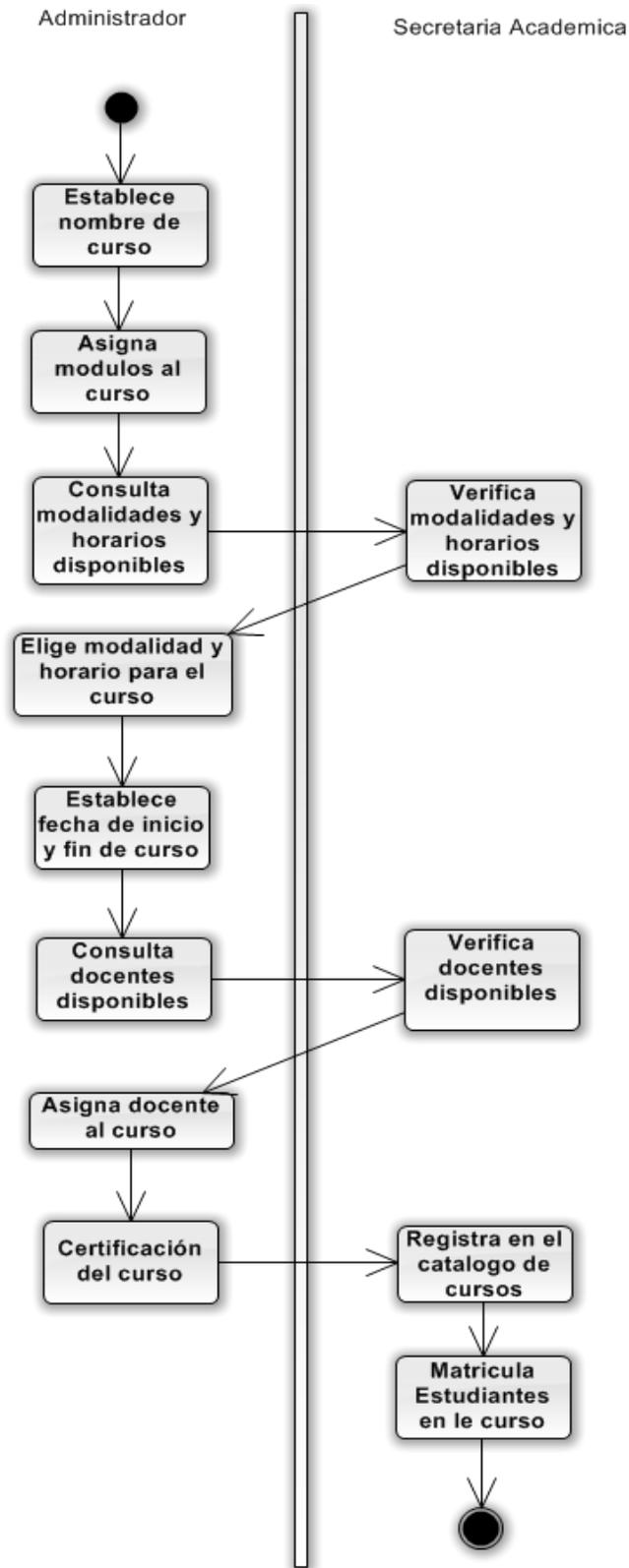


Ilustración 15: Diagrama de Actividades – Grupos de Clases

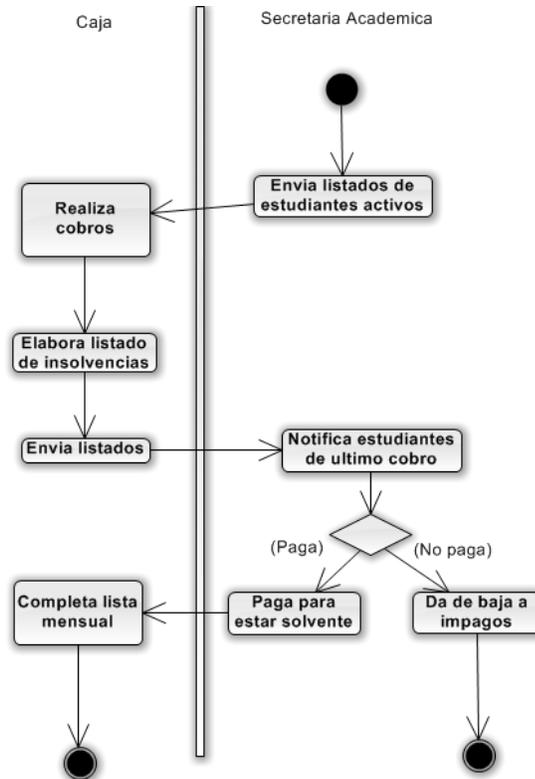
### 2.2.5 Control de Solvencias

Este control en particular tiene la participación de los departamentos de registro académico y caja.

Los pagos de los cursos son mensuales, con la norma de pago mensual por adelantado, es decir, que durante los primeros 10 días del mes se tiene que cancelar el arancel mensual.

Si el estudiante no logra cancelar el arancel al finalizar el mes, pasara a un estatus de insolvencia y no podrá seguir recibiendo el curso.

Para determinar los estudiantes insolventes registro académico envía los listados de los estudiantes activos, caja a su vez lleva un listado de los estudiantes que han abonado y los que no, y diariamente pasa el informe a secretaría académica, si estudiantes siguen pendientes se realiza un último cobro, si paga se completa la lista de solventes, sino se suspende del curso a los estudiantes impagos.



**Ilustración 16: Diagrama de Actividad – Solvencias**

### **3. Análisis y Diseño UML**

El sistema se modeló usando Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language).

Es un lenguaje gráfico que utilizamos para visualizar, construir y documentar nuestro sistema de información. De esta manera presentamos aspectos conceptuales tales como procesos del negocio, funciones del sistema, y aspectos concretos como esquemas de bases de datos.

De esta manera elaboramos un sistema de información con un buen diseño y funcional. Para el diseño del prototipo de sistemas de información para el registro académico en Compulab-Estelí se realizaron los diagramas de actividades,

Diagramas de caso de usos, diagrama de secuencia, diagrama de colaboración, y diagrama de despliegue los cuales son partes del sistema de modelación Utilizado en este documento.

#### **3.1 Flujos de Trabajos Persistentes.**

Nuestros flujos de trabajos encontrados a través de las entrevistas y la observación directa que se pueden automatizar se describen a continuación:

- Registro de Matricula.
- Registro grupos.
- Registro y administración de notas de estudiantes.
- Registro de módulos.
- Registro de modalidades y horarios.
- Control de Solvencias.
- Elaboración de Reportes

## **3.2 Análisis de los casos de usos y sus escenarios**

Por medio de los casos de usos demostramos como debe comportarse el sistema, el cual nos ayudó a encontrar requisitos desde el punto de vista del usuario.

A continuación definimos los actores los cuales se pueden describir como los roles que asume una persona, un sistema o entidad. Hay que tener en cuenta que un usuario puede acceder al sistema como distintos actores.

### **3.2.1 Actores encontrados en el Análisis**

- **Administrador:** Podrá acceder al sistema para consultar, modificar y registrar datos en el sistema, este usuario se encargara de administrar el sistema en su totalidad; realizando las actividades elementales.
- **Personal de registro académico:** Podrá ingresar y manipular (modificar y eliminar registros) los datos del sistema, es el encargado de matrícula y generar reportes vitales.

### 3.2.2 Diagramas de Casos de Uso.

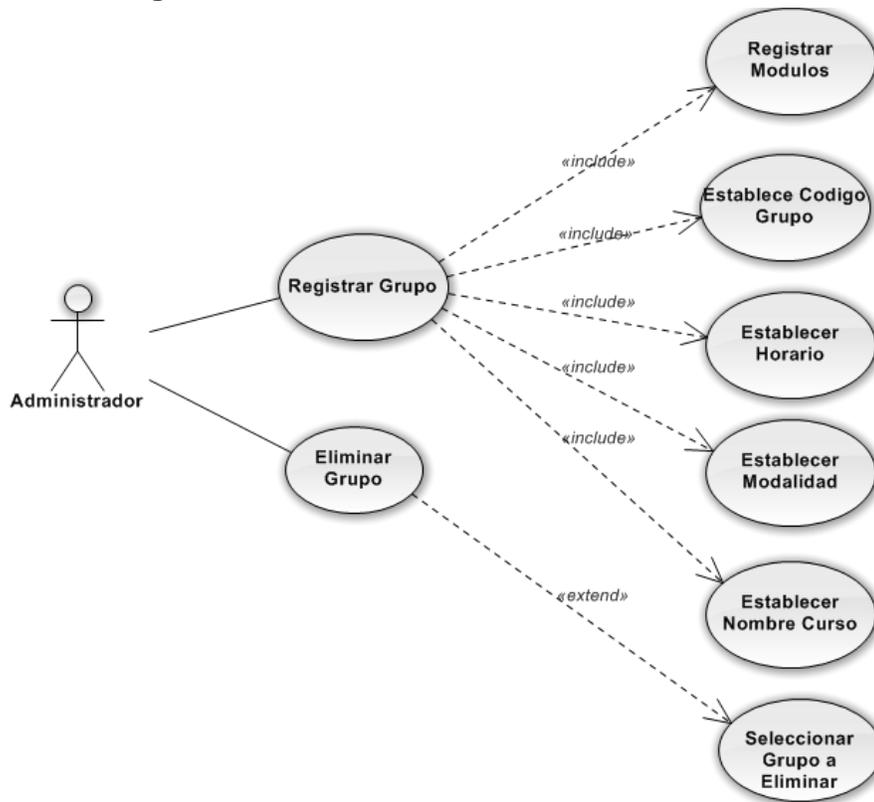


Ilustración 17: Diagrama de Casos de uso – Grupos.

Caso de Uso	Registrar Grupo			
Definición:	Se utiliza para crear los grupos con su correspondiente código, curso, modalidad, horarios y fechas de inicio y fin del curso			
Prioridad:	√ (1)Vital	(2)Importante	(3)Conveniente	
Urgencia:	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>				
Nombre	Definición			
Administrador	Persona encargada de realizar actividades esenciales del sistema			
<b>Escenario</b>				
Nombre:	Registrar Grupo			
Pre-condiciones:	Ninguna			
Iniciado por:	Administrador			

<b>Finalizador por:</b>	<b>Administrador</b>
<b>Post- condiciones:</b>	<b>Registrar Módulos Registrar Estudiantes</b>
<b>Operaciones:</b>	<ul style="list-style-type: none"> <li>1- Establecer Código Grupo.</li> <li>2- Establecer Horarios.</li> <li>3- Establecer Modalidad.</li> <li>4- Establecer Nombre Curso.</li> </ul>
<b>Excepciones:</b>	<b>Ninguna</b>

<b>Caso de Uso</b>	<b>Eliminar Grupo</b>			
<b>Definición:</b>	Se utiliza para eliminar los grupos que ya han terminado o que han expirado.			
<b>Prioridad:</b>	√ (1)Vital	(2)Importante	(3)Conveniente	
<b>Urgencia:</b>	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>				
<b>Nombre</b>	<b>Definición</b>			
<b>Administrador</b>	Persona encargada de realizar actividades esenciales del sistema			
<b>Escenario</b>				
<b>Nombre:</b>	<b>Eliminar Grupo</b>			
<b>Pre- condiciones:</b>	Tiene que existir al menos un grupo.			
<b>Iniciado por:</b>	<b>Administrador</b>			
<b>Finalizador por:</b>	<b>Administrador</b>			
<b>Post- condiciones:</b>	<b>Ninguna</b>			
<b>Operaciones:</b>	1- Seleccionar Grupo a eliminar.			
<b>Excepciones:</b>	<b>Ninguna</b>			

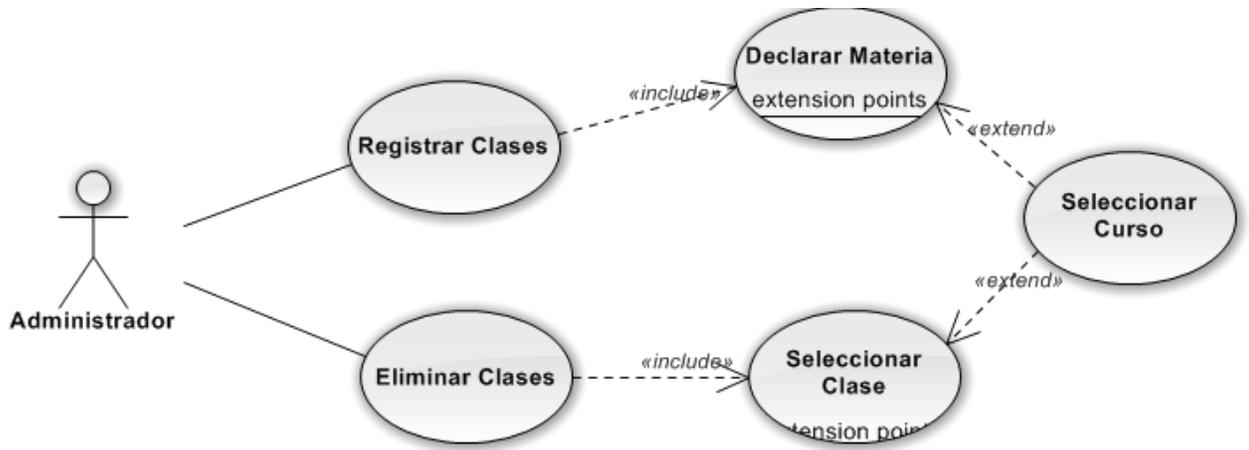


Ilustración 18: Diagrama de Caso de Uso - Clases

Caso de Uso	Registrar Clases			
Definición:	Se utiliza para registrar las Clases y las materias de los cursos.			
Prioridad:	√	(1)Vital	(2)Importante	(3)Conveniente
Urgencia:	√	(1)Inmediata	(2)Necesario	(3)Puede Esperar
<b>Actores</b>				
Nombre	Definición			
Administrador	Persona encargada de realizar actividades esenciales del sistema			
<b>Escenario</b>				
Nombre:	Registrar Clases			
Pre- condiciones:	Tiene que existir al menos un grupo.			
Iniciado por:	Administrador			
Finalizador por:	Administrador			
Post- condiciones:	Ninguna			
Operaciones:	1- Declarar Materia 2- Seleccionar Curso.			
Excepciones:	Ninguna			

<b>Caso de Uso</b>	<b>Eliminar Clases</b>			
<b>Definición:</b>	<b>Se utiliza para eliminar las clases y las materias de los cursos.</b>			
<b>Prioridad:</b>	√	<b>(1)Vital</b>	<b>(2)Importante</b>	<b>(3)Conveniente</b>
<b>Urgencia:</b>	√	<b>(1)Inmediata</b>	<b>(2)Necesario</b>	<b>(3)Puede Esperar</b>
<b>Actores</b>				
<b>Nombre</b>	<b>Definición</b>			
<b>Administrador</b>	<b>Persona encargada de realizar actividades esenciales del sistema</b>			
<b>Escenario</b>				
<b>Nombre:</b>	<b>Eliminar Clases</b>			
<b>Pre- condiciones:</b>	<b>Tiene que existir al menos una Clase Registrada.</b>			
<b>Iniciado por:</b>	<b>Administrador</b>			
<b>Finalizador por:</b>	<b>Administrador</b>			
<b>Post- condiciones:</b>	<b>Ninguna</b>			
<b>Operaciones:</b>	<b>1- Seleccionar Clase.</b>			
<b>Excepciones:</b>	<b>Ninguna</b>			

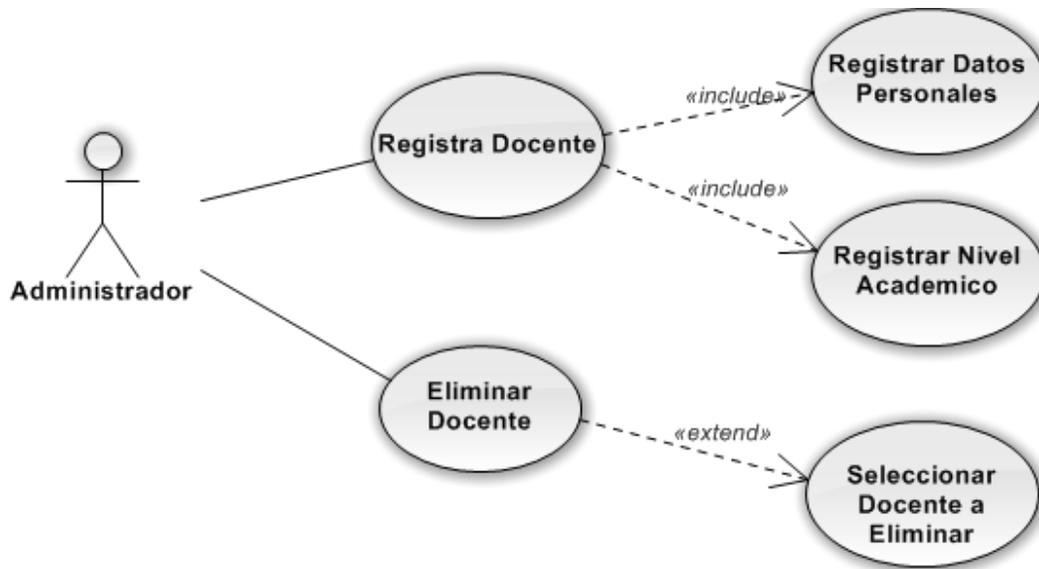
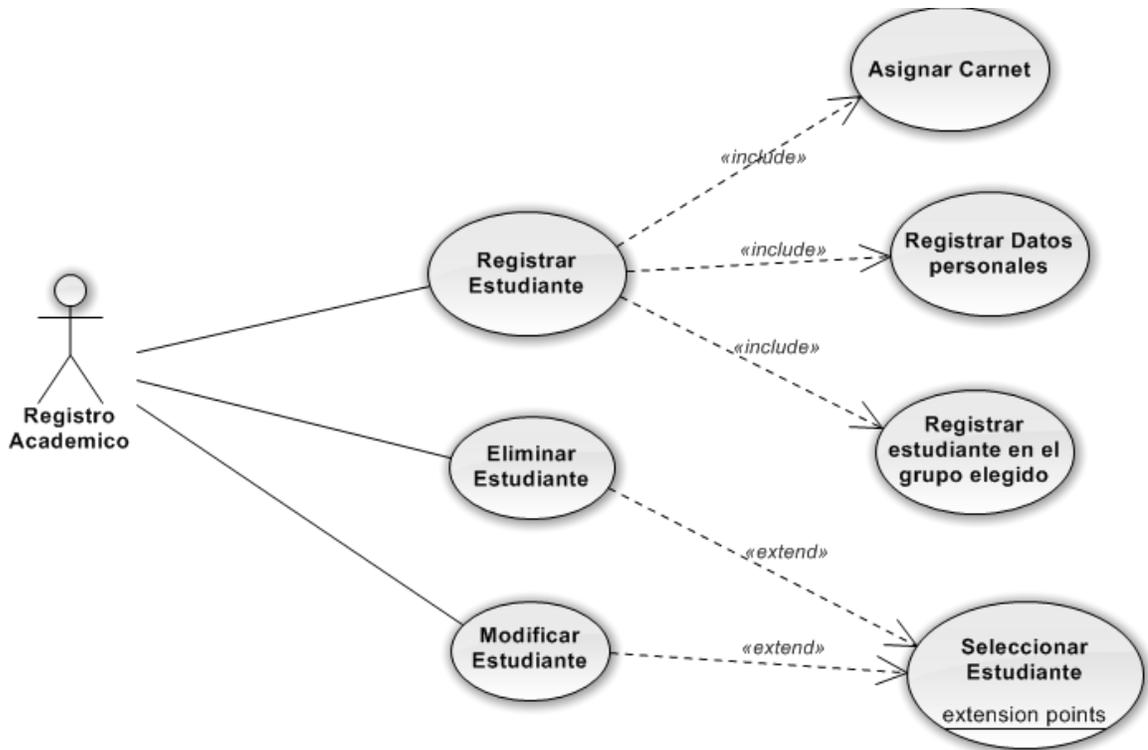


Ilustración 19: Diagrama de Casos de Uso - Docentes

Caso de Uso	Registrar Docentes		
Definición:	Se utiliza para registrar a los docentes que impartirán clases.		
Prioridad:	√ (1)Vital	(2)Importante	(3)Conveniente
Urgencia:	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar
Actores			
Nombre	Definición		
Administrador	Persona encargada de realizar actividades esenciales del sistema		
Escenario			
Nombre:	Registrar Docentes		
Pre- condiciones:	Ninguna		
Iniciado por:	Administrador		
Finalizador por:	Administrador		
Post- condiciones:	Catálogo de Docentes Horarios		
Operaciones:	1- Registrar datos personales. 2- Registrar nivel profesional.		
Excepciones:	Ninguna.		

<b>Caso de Uso</b>	<b>Eliminar Docentes</b>			
<b>Definición:</b>	<b>Se utiliza para eliminar a los docentes que han caducado.</b>			
<b>Prioridad:</b>	√ (1)Vital	(2)Importante	(3)Conveniente	
<b>Urgencia:</b>	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>				
<b>Nombre</b>	<b>Definición</b>			
<b>Administrador</b>	<b>Persona encargada de realizar actividades esenciales del sistema</b>			
<b>Escenario</b>				
<b>Nombre:</b>	<b>Eliminar Docentes.</b>			
<b>Pre- condiciones:</b>	<b>Tiene que existir al menos un docente.</b>			
<b>Iniciado por:</b>	<b>Administrador</b>			
<b>Finalizador por:</b>	<b>Administrador</b>			
<b>Post- condiciones:</b>	<b>Ninguno.</b>			
<b>Operaciones:</b>	<b>1- Seleccionar Docente.</b>			
<b>Excepciones:</b>	<b>Ninguna.</b>			



**Ilustración 20: Diagrama de Casos de Uso - Estudiantes**

Caso de Uso	Registrar Estudiante		
Definición:	Se utiliza para registrar los datos del estudiante y el curso elegido.		
Prioridad:	√ (1)Vital	(2)Importante	(3)Conveniente
Urgencia:	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar
<b>Actores</b>			
Nombre	Definición		
Secretaria Académica	Persona que es encargada de las actividades relacionadas con las matriculas, del control y mantenimientos de notas y listas de estudiantes		
<b>Escenario</b>			
Nombre:	Registrar Estudiante		
Pre-condiciones:	Registrar Grupo		
Iniciado por:	Secretaria Académica		
Finalizador por:	Secretaria Académica		
Post-	Control de notas.		

<b>condiciones:</b>	<b>Control de Solvencias.</b> <b>Reportes varios.</b>
<b>Operaciones:</b>	1- Asignar Carnet. 2- Registrar Datos personales. 3- Registrar en el grupo deseado.
<b>Excepciones:</b>	Ninguna

<b>Caso de Uso</b>		<b>Eliminar Estudiante</b>			
<b>Definición:</b>	Se utiliza para eliminar los datos del estudiante que han caducado.				
<b>Prioridad:</b>	√ (1)Vital	(2)Importante	(3)Conveniente		
<b>Urgencia:</b>	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar		
<b>Actores</b>					
<b>Nombre</b>	<b>Definición</b>				
Secretaria Académica	Persona que es encargada de las actividades relacionadas con las matriculas, del control y mantenimientos de notas y listas de estudiantes				
<b>Escenario</b>					
<b>Nombre:</b>	Eliminar Estudiante.				
<b>Pre- condiciones:</b>	Registrar Grupo. Registrar Estudiante.				
<b>Iniciado por:</b>	Secretaria Académica				
<b>Finalizador por:</b>	Secretaria Académica				
<b>Post- condiciones:</b>	Ninguna.				
<b>Operaciones:</b>	1- Seleccionar Estudiante.				
<b>Excepciones:</b>	Ninguna				

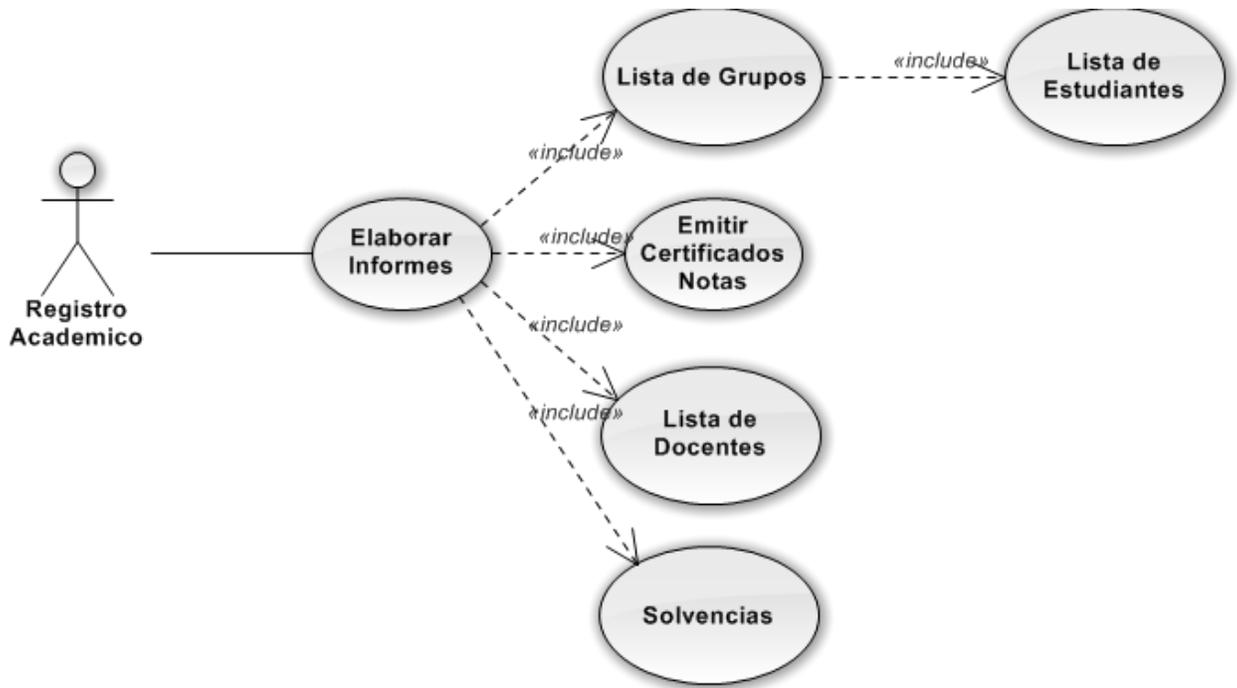
<b>Caso de Uso</b>	<b>Modificar Estudiante</b>			
<b>Definición:</b>	<b>Se utiliza para Modificar los datos del estudiante que se desea.</b>			
<b>Prioridad:</b>	√ (1)Vital	(2)Importante	(3)Conveniente	
<b>Urgencia:</b>	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>				
<b>Nombre</b>	<b>Definición</b>			
<b>Secretaria Académica</b>	<b>Persona que es encargada de las actividades relacionadas con las matriculas, del control y mantenimientos de notas y listas de estudiantes</b>			
<b>Escenario</b>				
<b>Nombre:</b>	<b>Modificar Estudiante.</b>			
<b>Pre- condiciones:</b>	<b>Registrar Grupo. Registrar Estudiante.</b>			
<b>Iniciado por:</b>	<b>Secretaria Académica</b>			
<b>Finalizador por:</b>	<b>Secretaria Académica</b>			
<b>Post- condiciones:</b>	<b>Ninguna.</b>			
<b>Operaciones:</b>	<b>1- Seleccionar Estudiante.</b>			
<b>Excepciones:</b>	<b>Ninguna</b>			



**Ilustración 21: Diagrama de Casos de Usos – Notas**

Caso de Uso		Registar notas			
Definición:	Se utiliza para registrar las notas del estudiante				
Prioridad:	√	(1)Vital	(2)Importante	(3)Conveniente	
Urgencia:	√	(1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>					
Nombre	Definición				
Secretaria Académica	Persona que es encargada de las actividades relacionadas con las matriculas, del control y mantenimientos de notas y listas de estudiantes				
<b>Escenario</b>					
Nombre:	Registar notas				
Pre-condiciones:	Registar Estudiante Registar Docente Registar Grupo				
Iniciado por:	Secretaria Académica				
Finalizador por:	Secretaria Académica				
Post-condiciones:	Elaboración de Informes				
Operaciones:	1- Seleccionar Grupo				

	<b>2- Seleccionar Modulo</b> <b>3- Ingresar notas a estudiante.</b>
<b>Excepciones:</b>	<b>Ninguna</b>



**Ilustración 22: Diagrama de Casos de Uso – Informes**

<b>Caso de Uso</b>	<b>Elaborar Informes</b>			
<b>Definición:</b>	<b>Se utiliza para crear informes.</b>			
<b>Prioridad:</b>	√ (1)Vital	(2)Importante	(3)Conveniente	
<b>Urgencia:</b>	√ (1)Inmediata	(2)Necesario	(3)Puede Esperar	
<b>Actores</b>				
<b>Nombre</b>	<b>Definición</b>			
<b>Secretaria Académica</b>	Persona que es encargada de las actividades relacionadas con las matriculas, del control y mantenimientos de notas y listas de estudiantes			
<b>Escenario</b>				
<b>Nombre:</b>	<b>Elaborar Informes</b>			
<b>Pre-condiciones:</b>	<b>Registrar Estudiantes</b> <b>Registrar Docentes</b> <b>Registrar Notas</b> <b>Registrar Grupo</b>			
<b>Iniciado por:</b>	<b>Secretaria Académica</b>			
<b>Finalizador por:</b>	<b>Secretaria Académica</b>			
<b>Post-condiciones:</b>	<b>Ninguna</b>			
<b>Operaciones:</b>	<b>1- Emitir listados de grupos</b> <b>2- Emitir notas</b> <b>3- Emitir listados de Docentes</b> <b>4- Emitir Solvencias</b>			
<b>Excepciones:</b>	<b>Ninguna</b>			

### 3.3 Diagramas de Secuencia

Un diagrama de secuencia es una vista gráfica de un escenario que muestra la interacción de objetos basados en una secuencia de tiempo, lo que sucede primero, lo que sucede después.

Los diagramas de secuencia establecen los roles de los objetos y ayudan a proveer la información esencial para determinar las responsabilidades de clases e interfaces.

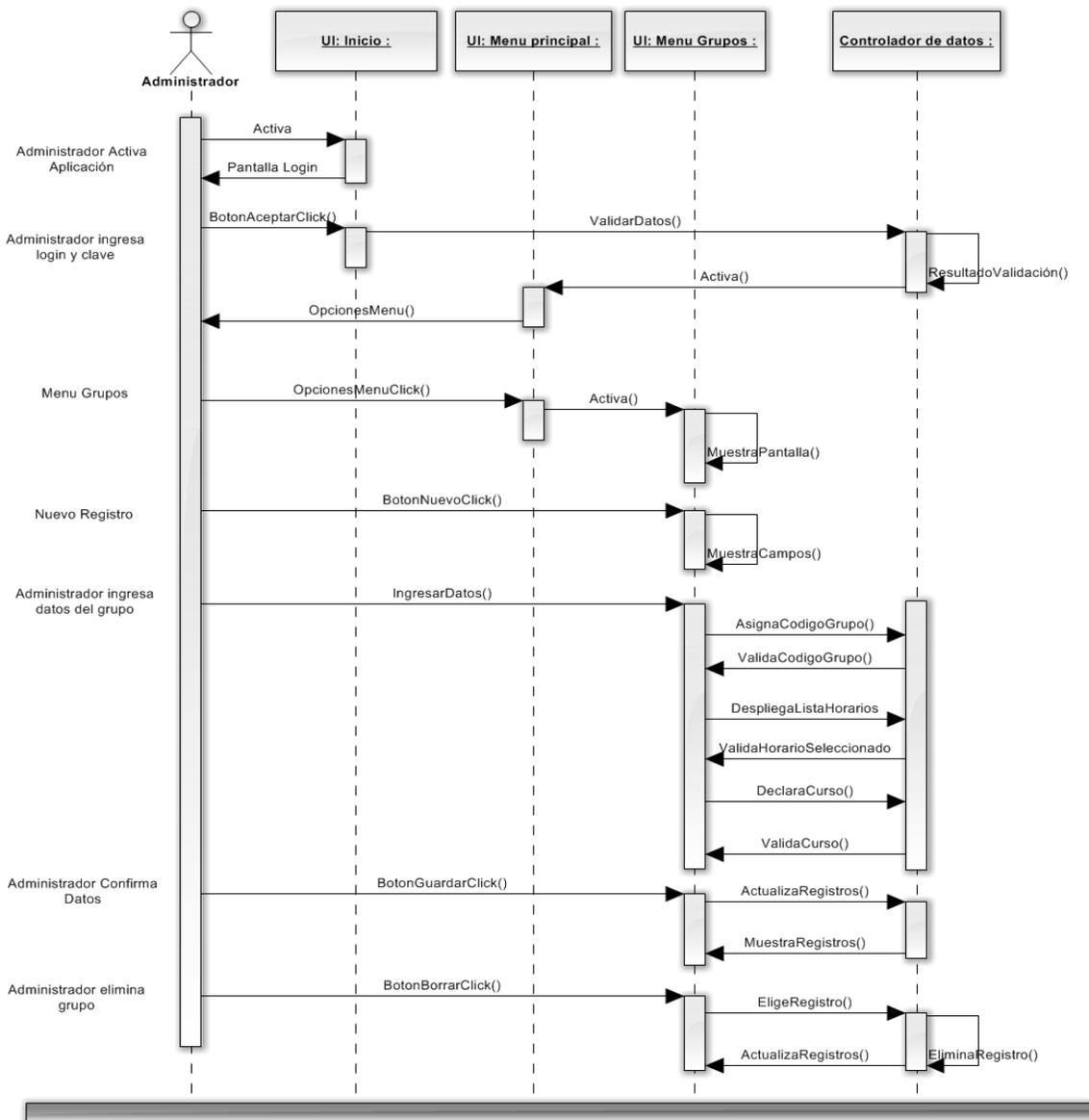


Ilustración 23: Diagrama de Secuencia – Casos de uso – Grupos.

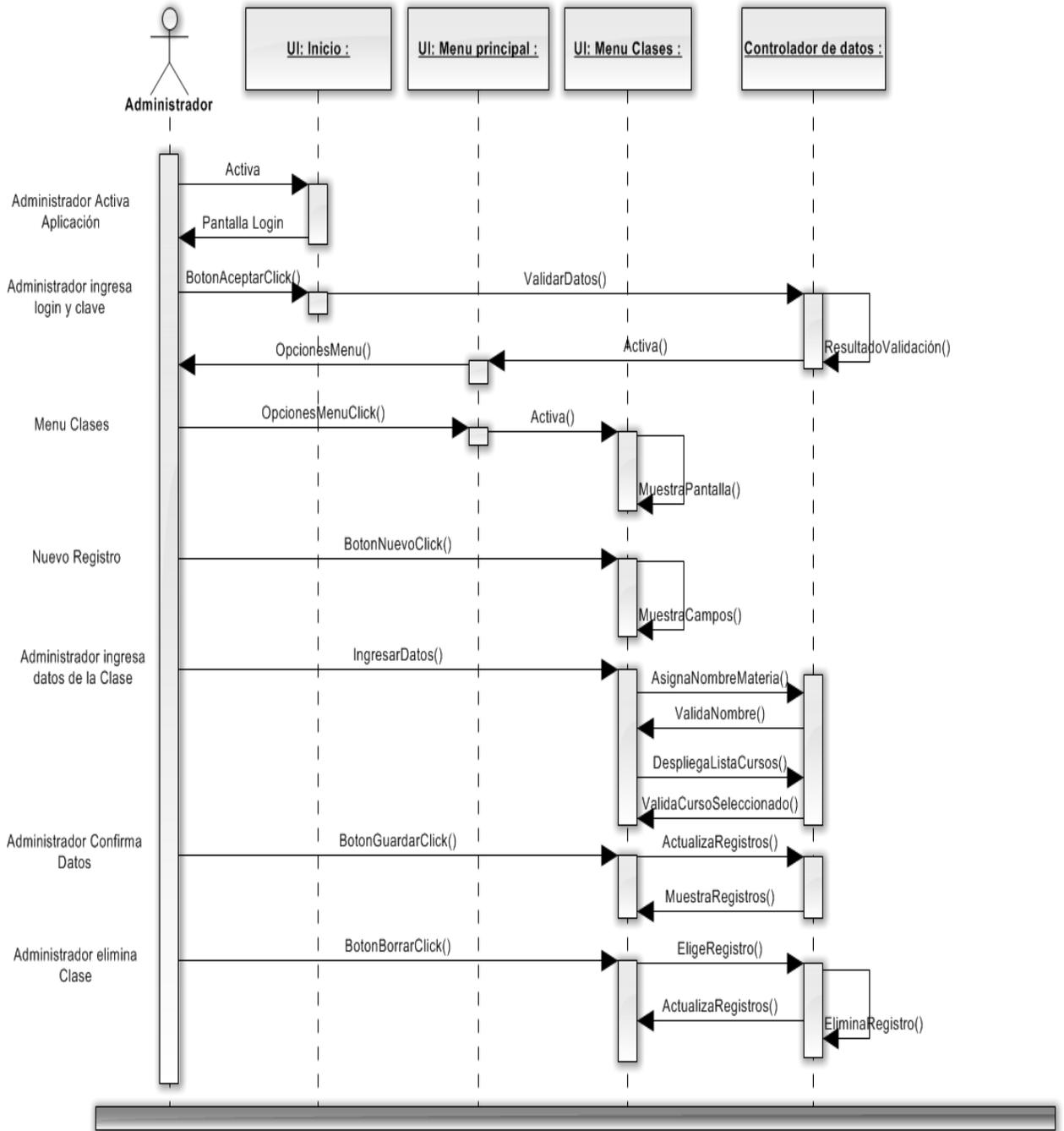


Ilustración 24: Diagrama de Secuencia – Caso de Uso - Clases

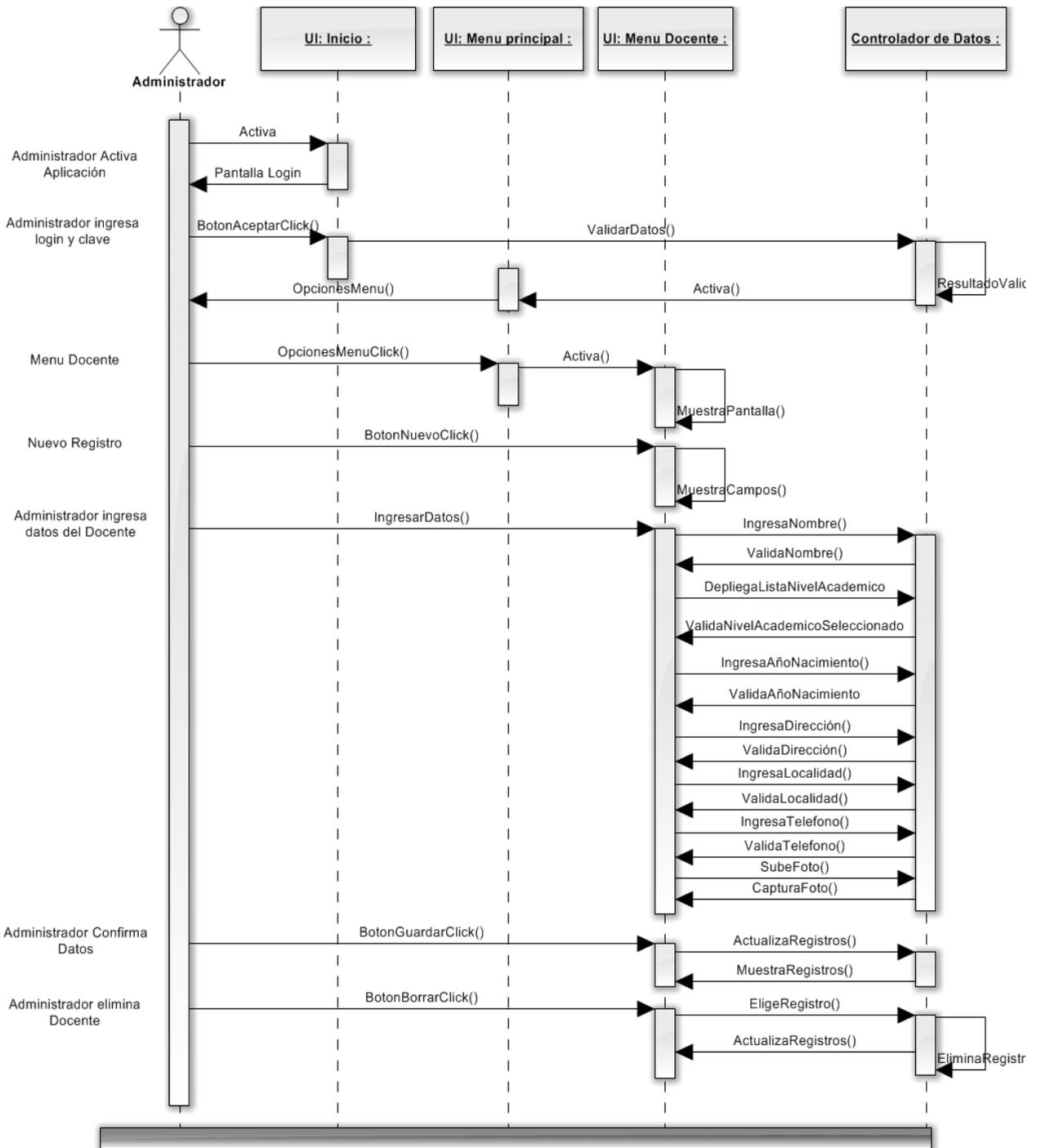


Ilustración 25: Diagrama de Secuencia 3 – Casos de Uso - Docentes

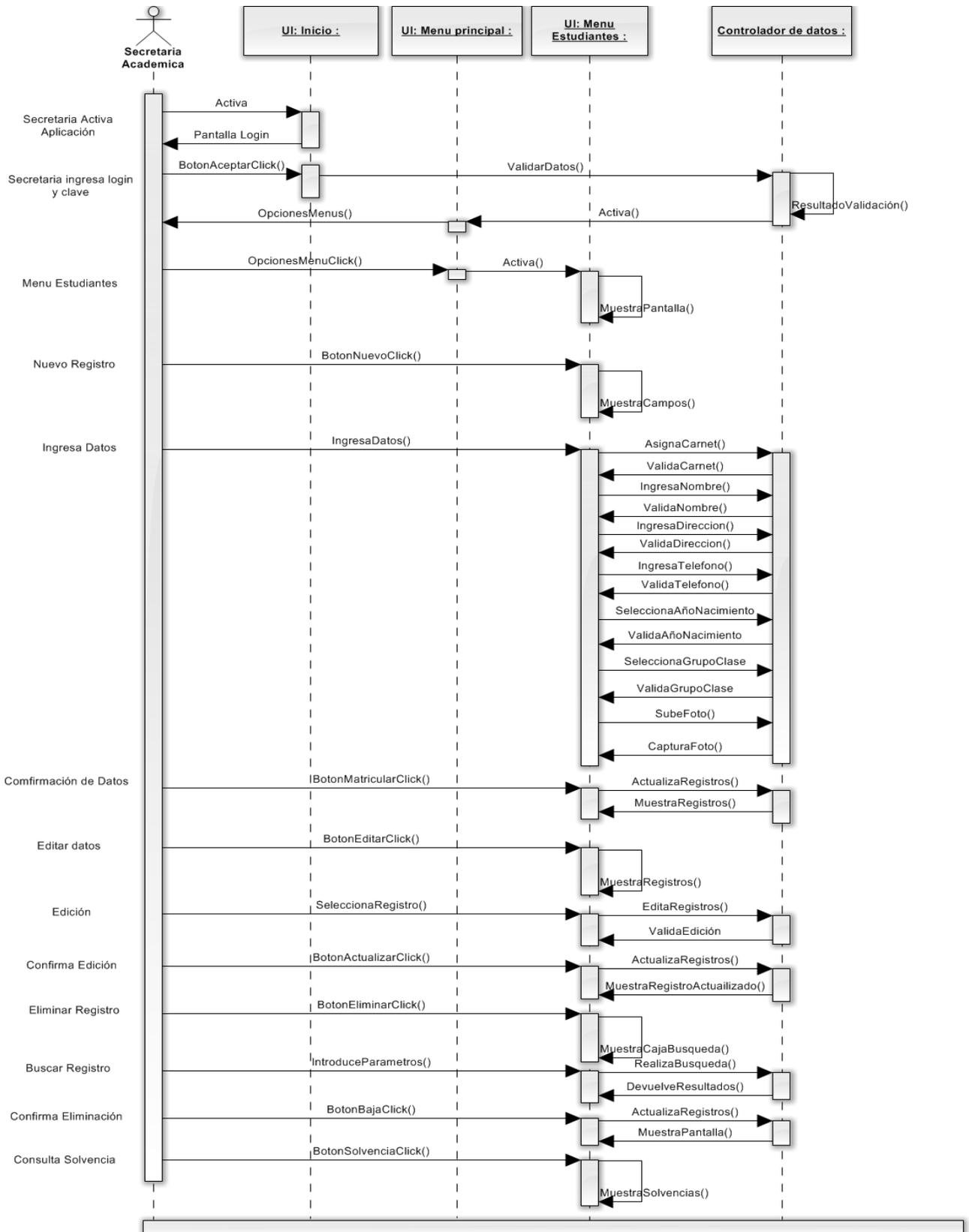


Ilustración 26: Diagrama de Secuencia – Casos de Uso - Estudiantes

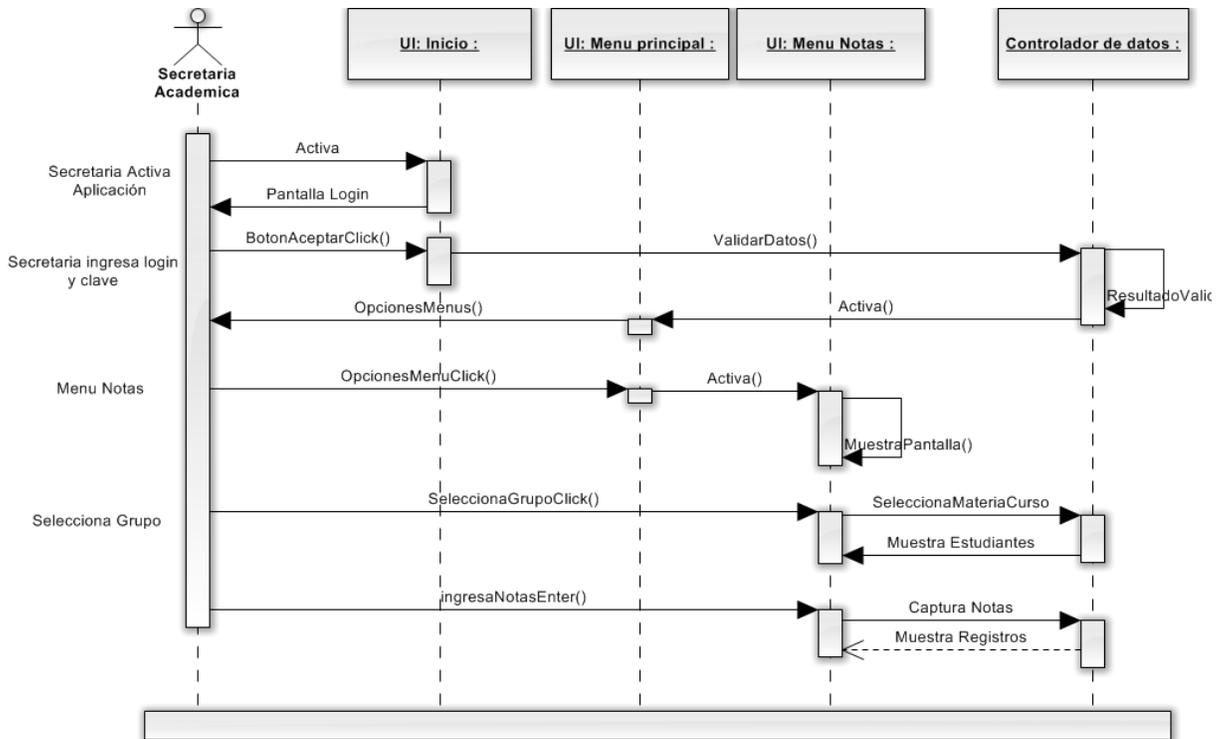


Ilustración 27: Diagrama de Secuencia – Casos de Uso – Notas

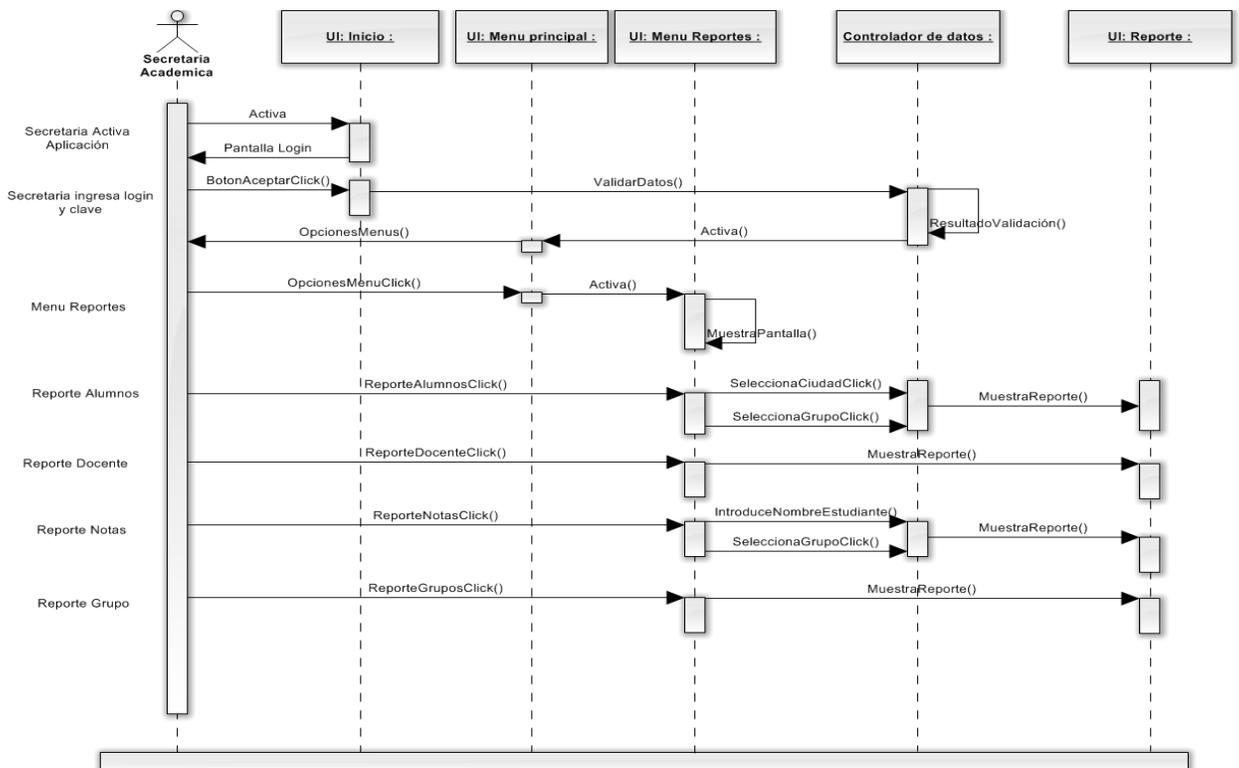
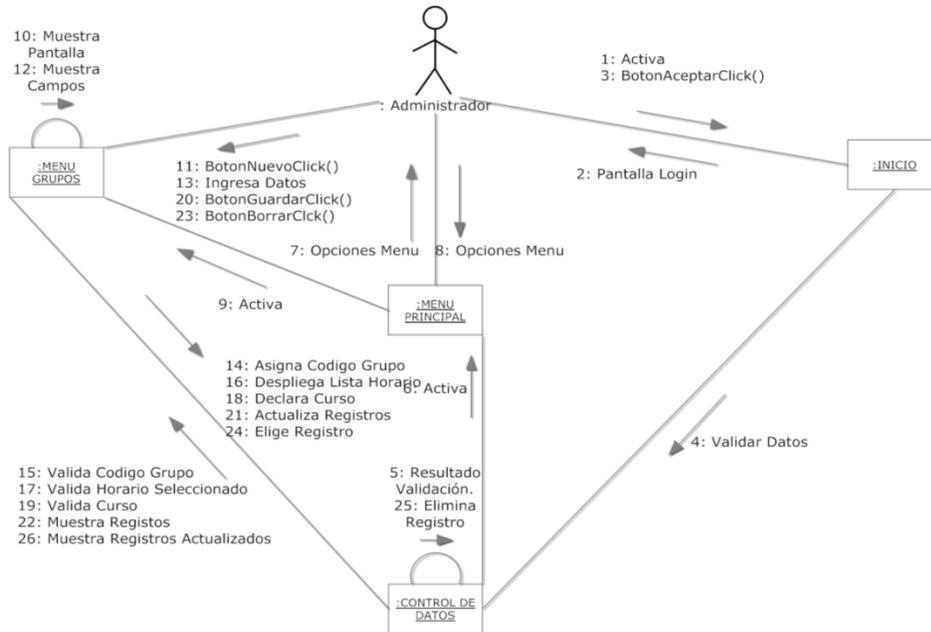


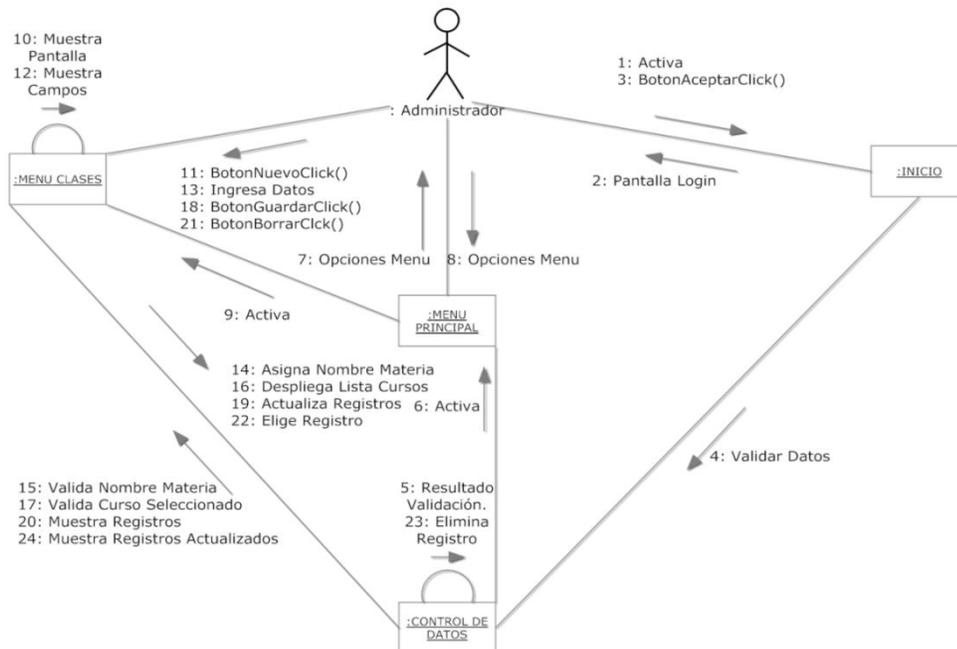
Ilustración 28: Diagrama de Secuencia - Caso de Uso - Informes

### 3.4 Diagramas de Colaboración.

Estos diagramas son obtenidos de los diagramas de secuencia. La diferencia es que no se presenta el orden cronológico de los mensajes sino que hace énfasis en el conjunto de los actores y objetos que se envían mensajes entre sí.



**Ilustración 29: Diagramas de Colaboración – Grupo**



**Ilustración 30: Diagrama de Colaboración – Clase**

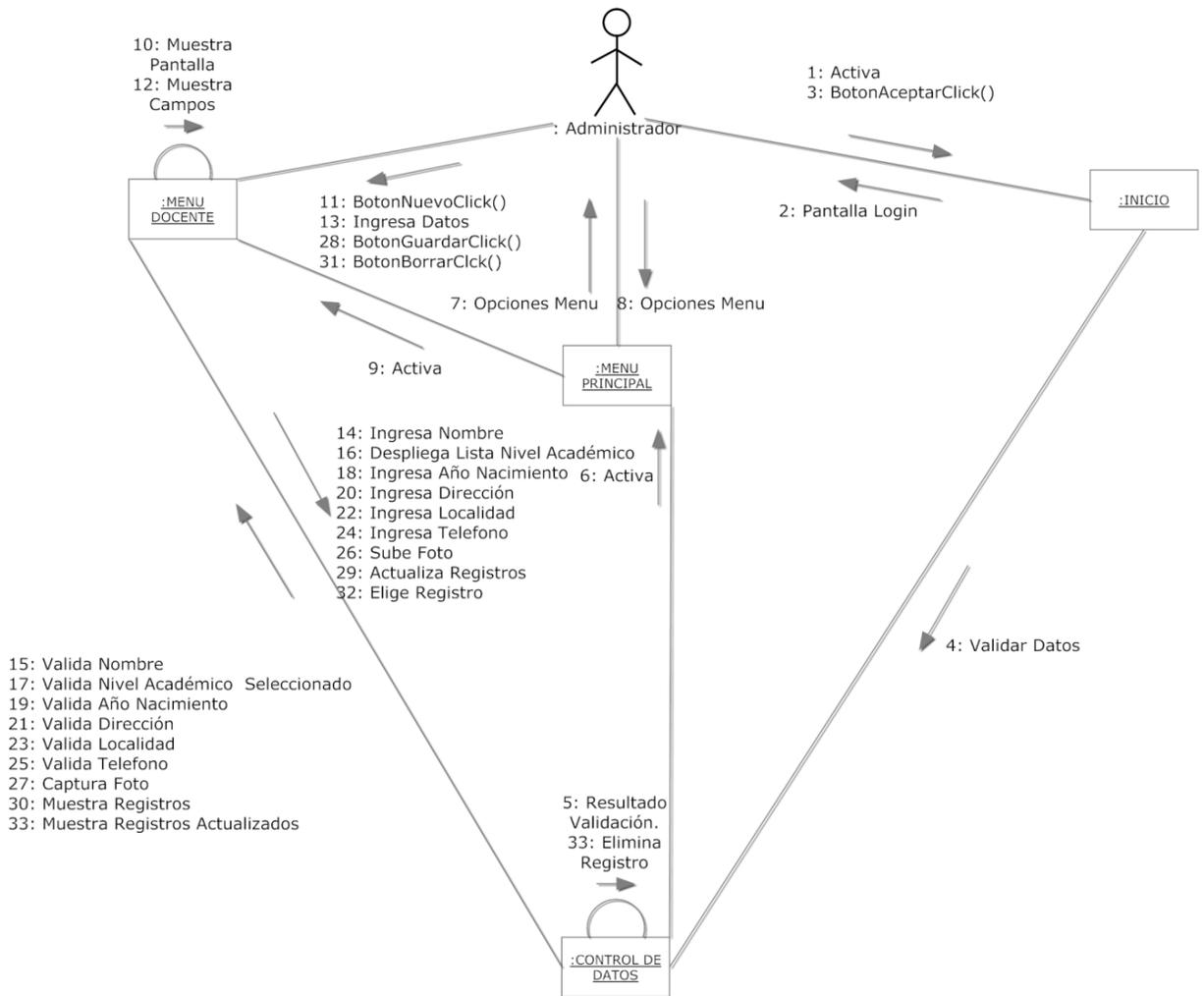
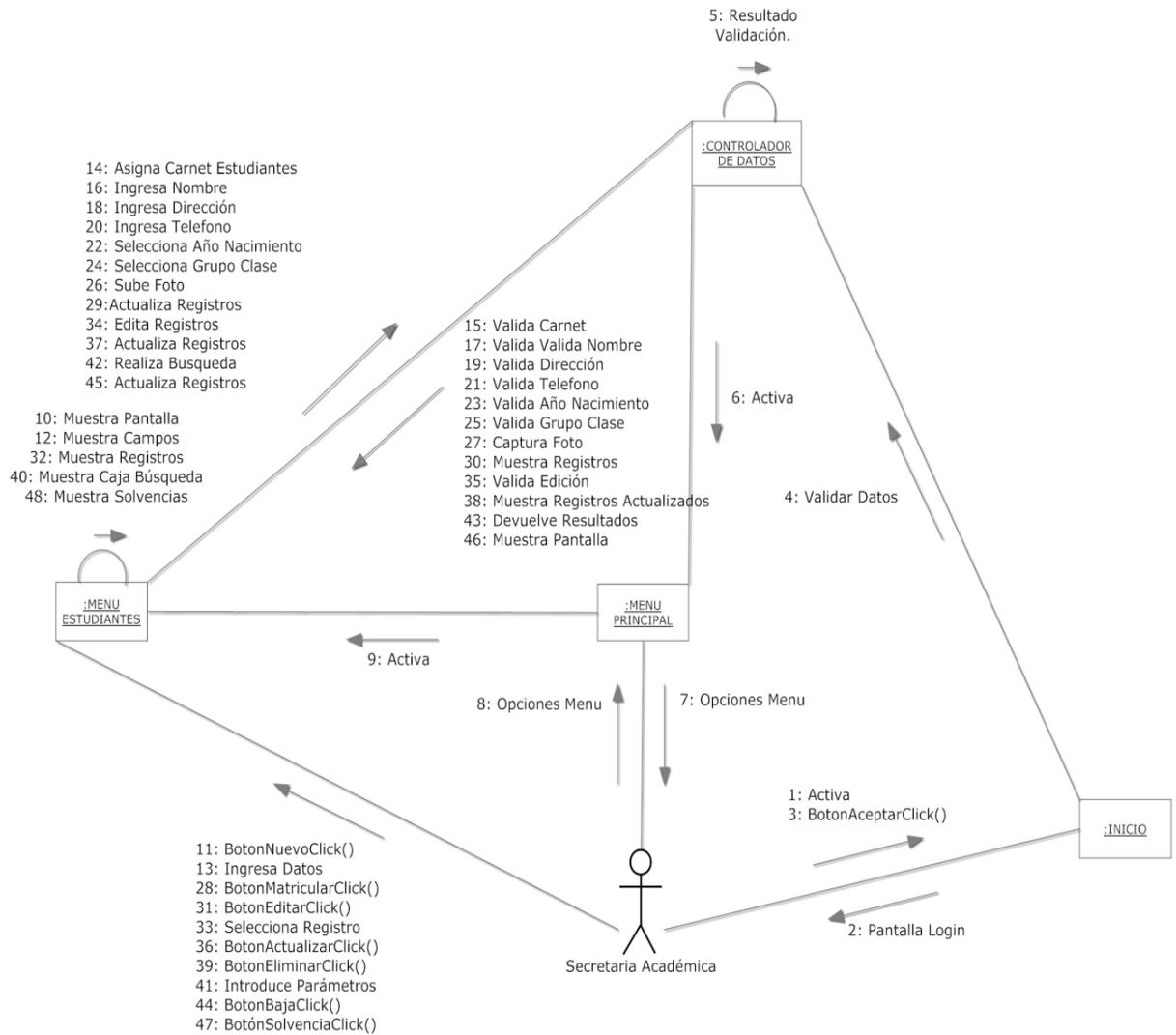
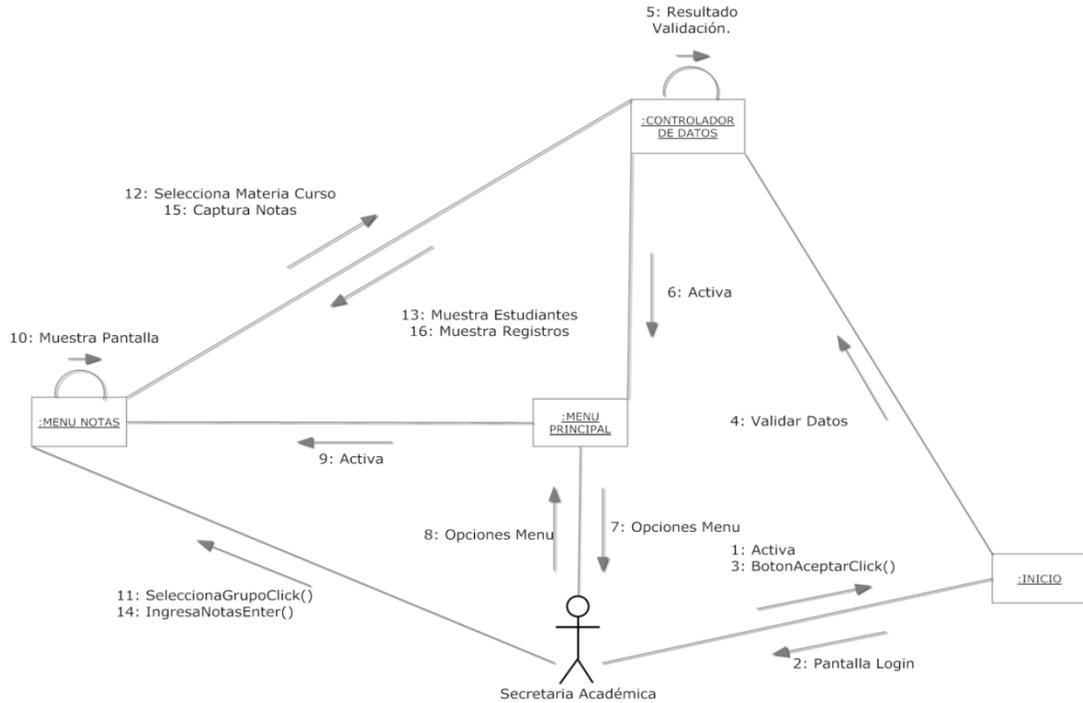


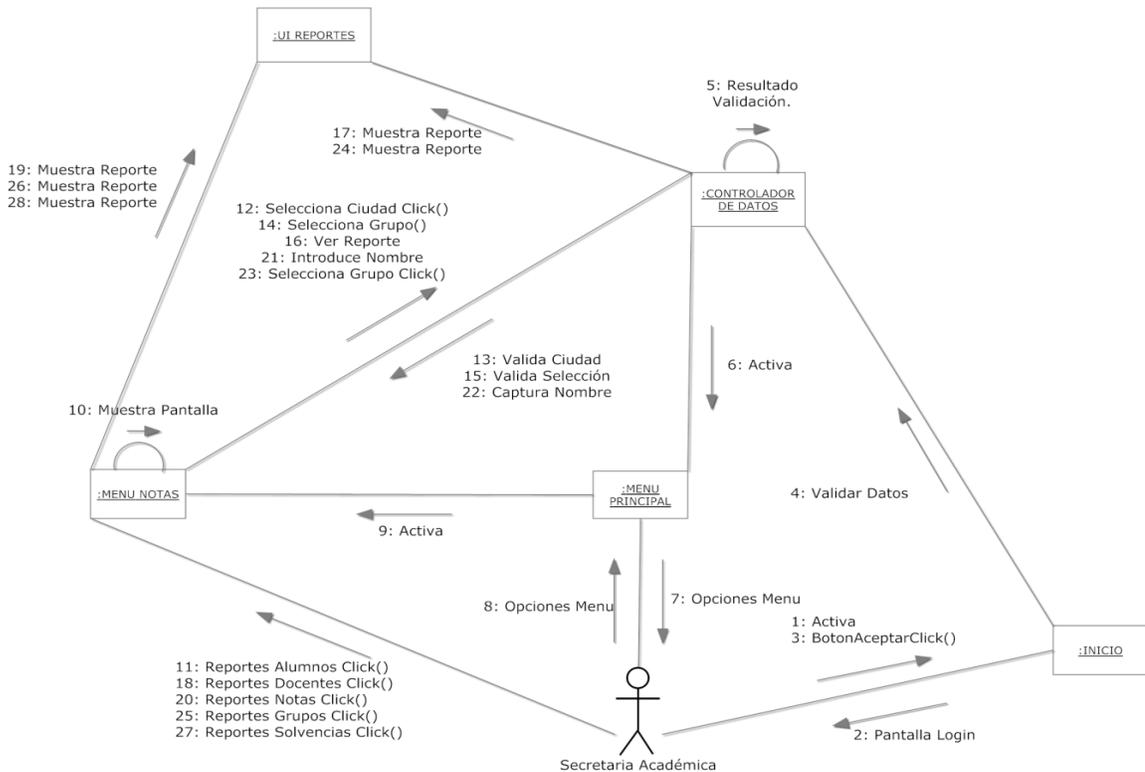
Ilustración 31: Diagrama de Colaboración – Docente



**Ilustración 32: Diagrama de Colaboración – Estudiante**



**Ilustración 33: Diagrama de Colaboración – Notas**



**Ilustración 34: Diagrama de Colaboración – Informes**

### 3.5 Diagrama de Clases

Muestran el sistema en sus diferentes partes lógicas (clases) y las relaciones de las mismas. Esto describe a través de una serie de diagramas el funcionamiento y el centro de la estructura del sistema.

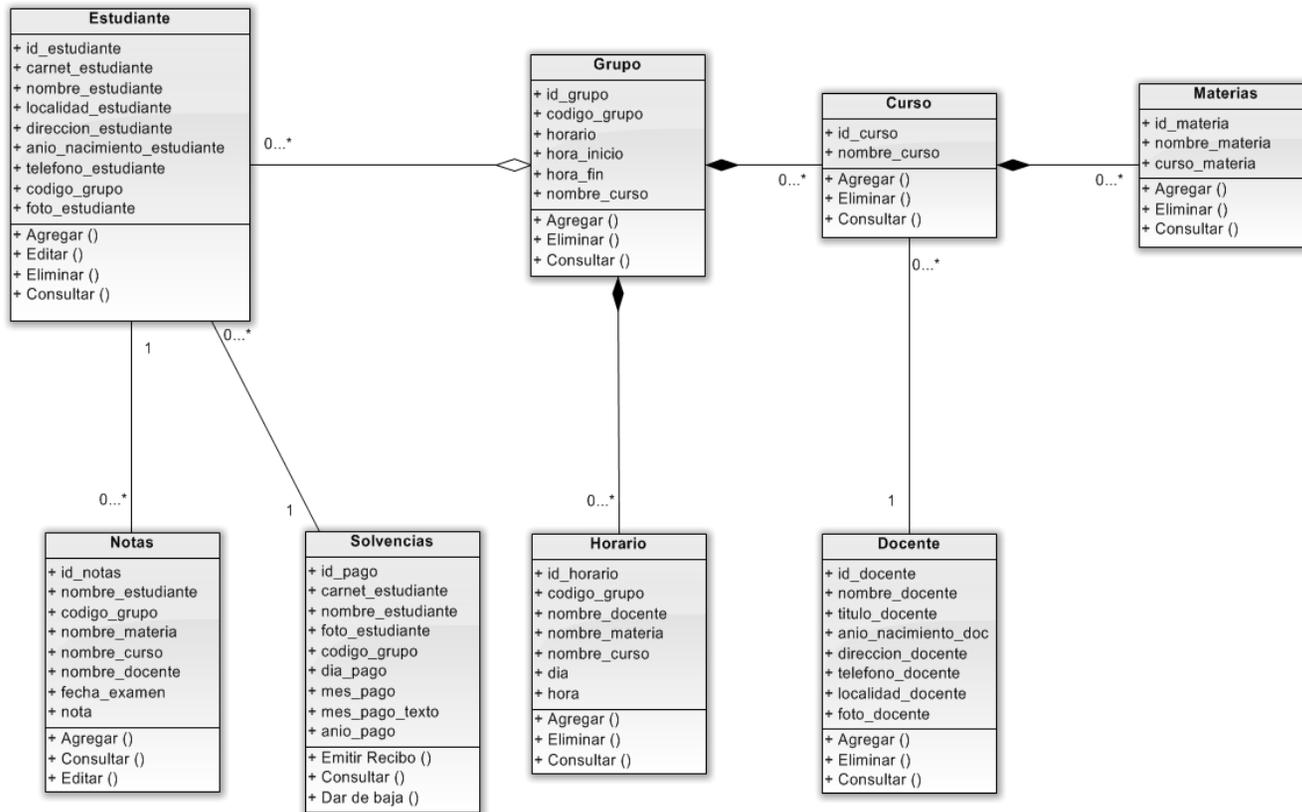
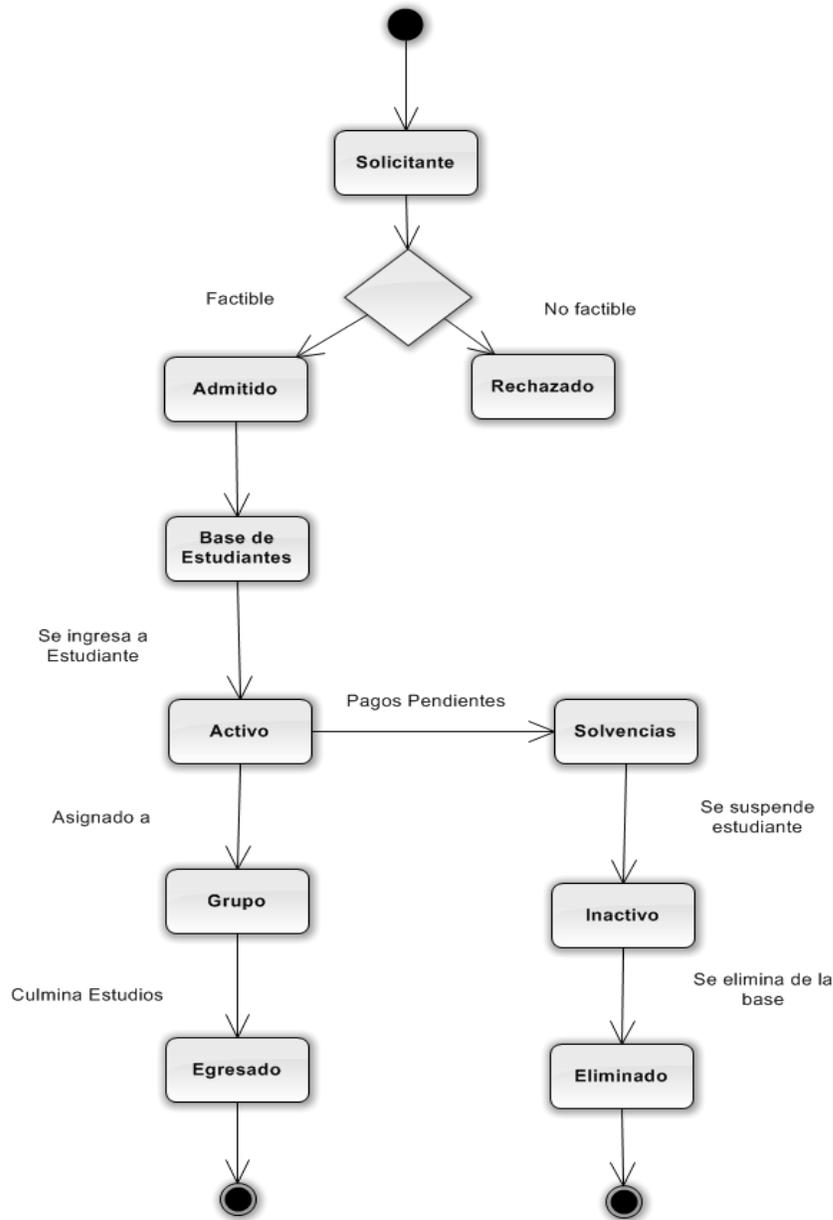


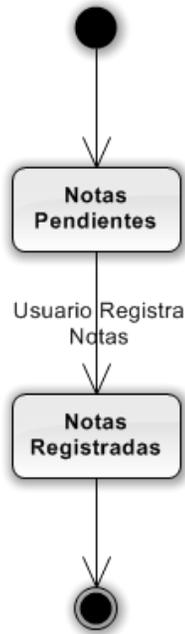
Ilustración 35: Diagrama de Clases

### 3.6 Diagramas de Estados

Los diagramas de estado modelan el comportamiento dinámico de clases individual o de cualquier tipo de objeto. Muestran las secuencias de estado por las cuales pasa un objeto, los eventos que causan la transición de un estado a otro y las acciones que resultan de un cambio de estado.



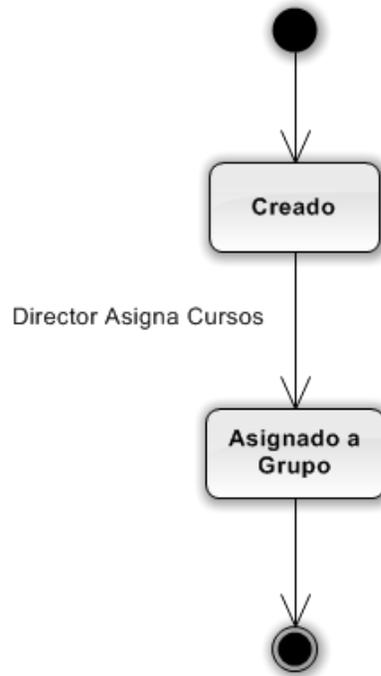
**Ilustración 36: Diagramas de Estado - Estudiante**



**Ilustración 37: Diagramas de Estado - Notas**



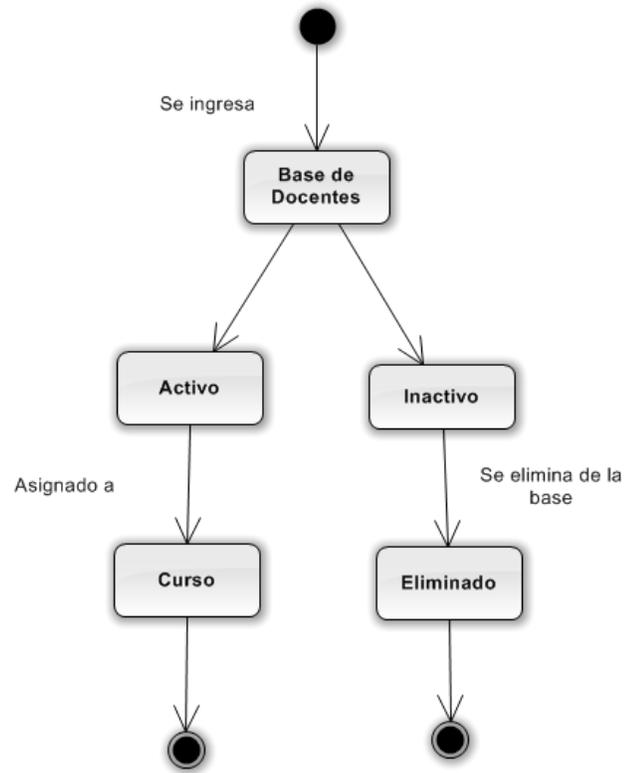
**Ilustración 38: Diagramas de Estado - Grupo**



**Ilustración 39: Diagramas de Estado - Cursos**



**Ilustración 40: Diagramas de Estado - Materia**



**Ilustración 41: Diagramas de Estado - Docente**

### 3.7 Diagrama de componentes

Presenta la relación de los componentes principales del sistema.

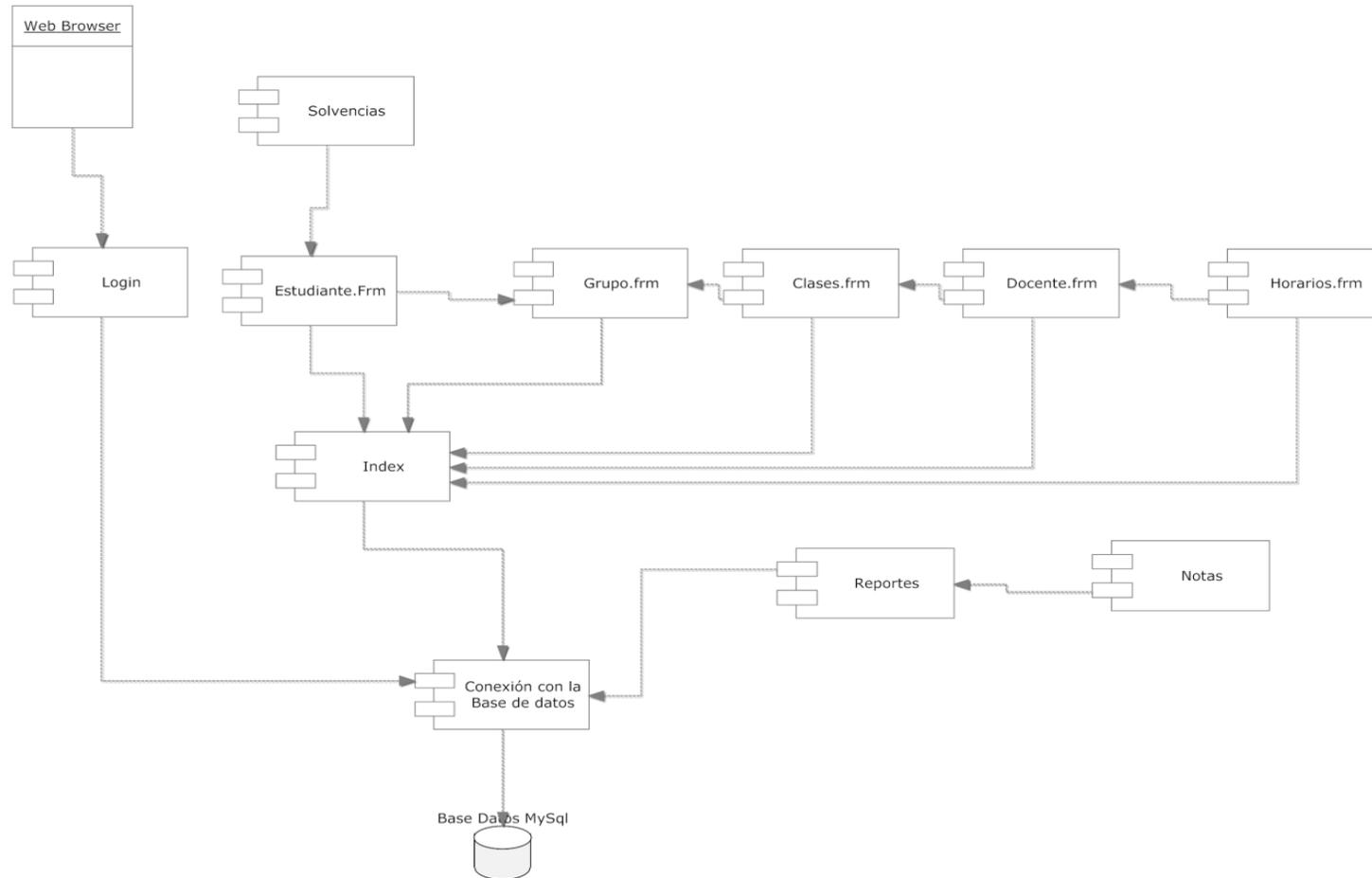
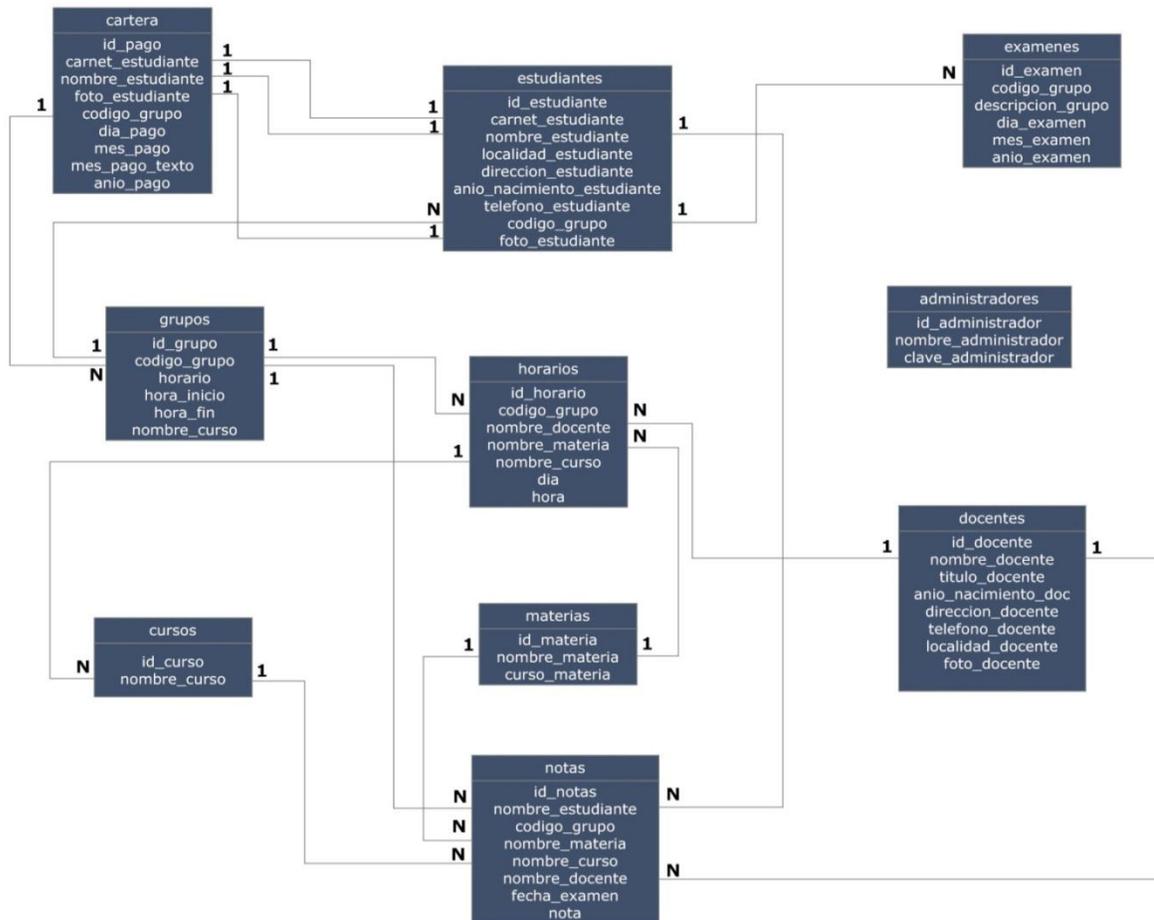


Ilustración 42: Diagrama de componentes

### 3.8 Modelo de datos

#### Diagrama Entidad – Relación

A continuación se muestra la estructura de relaciones de las tablas.



**Ilustración 43: Modelo Entidad – Relación UML**

### 3.8.1 Diccionario de datos.

Campo	Tipo de dato	Tamaño	Descripción
Id_estudiante	Int	255	
Carnet_estudiante	Varchar	50	
Nombre_estudiante	Varchar	50	
Localidad_estudiante	Varchar	50	
Dirección_estudiante	Varchar	100	
Anio_nacimiento_estudiante	Int	5	
Teléfono_estudiante	Int	8	
Código_grupo	Varchar	50	
Foto_estudiante	Varchar	100	

**Tabla 13: Tabla Estudiante**

Campo	Tipo de dato	Tamaño	Descripción
Id_examen	Int	255	
Código_grupo	Varchar	50	
Descripción_grupo	Varchar	200	
Dia_examen	Int	2	
Mes_examen	Int	2	
Anio_examen	Int	4	

**Tabla 14: Tabla Exámenes**

Campo	Tipo de dato	Tamaño	Descripción
Id-grupo	Int	255	
Codigo_grupo	Varchar	50	
Horario	Varchar	50	
Hora_inicio	Varchar	10	
Hora_fin	Varchar	10	
Nombre_curso	Varchar	50	

**Tabla 15: Tabla Grupos**

Campo	Tipo de dato	Tamaño	Descripción
Id_curso			
Nombre_curso			

**Tabla 16: Tabla Curso**

Campo	Tipo de dato	Tamaño	Descripción
Id_horario	Int	255	
Código_grupos	Varchar	50	
Nombre_docente	Varchar	50	
Nombre_materia	Varchar	50	
Nombre_curso	Varchar	50	
Día	Varchar	20	
Hora	Varchar	50	

**Tabla 17: Tabla Horarios**

Campo	Tipo de dato	Tamaño	Descripción
Id_materia	Int	255	
Nombre_materia	Varchar	50	
Curso_materia	Varchar	50	

**Tabla 18: Tabla Materias**

Campo	Tipo de dato	Tamaño	Descripción
Id_docente	Int	255	
Nombre_docente	Varchar	50	
Titulo_docente	Varchar	10	
Anio_nacimiento_docente	Int	5	
Dirección_docente	Varchar	100	
Teléfono_docente	Int	8	
Localidad_docente	Varchar	50	
Foto_docente	Varchar	100	

**Tabla 19: Tabla Docentes**

Campo	Tipo de dato	Tamaño	Descripción
Id_nota	Int	255	
Nombre_estudiante	Varchar	100	
Código_grupo	Varchar	50	
Nombre_materia	Varchar	50	
Nombre_curso	Varchar	50	
Nombre_docente	Varchar	50	
Fecha_examen	Varchar	10	
Nota	Int	3	

**Tabla 20: Tabla Notas**

Campo	Tipo de dato	Tamaño	Descripción
Id-pago	Int	255	
Carnet_estudiante	Varchar	50	
Nombre_estudiante	Varchar	50	
Foto_estudiante	Varchar	100	
Código_grupo	Varchar	50	
Día_pago	Int	2	
Mes_pago	Int	2	
Mes_pago_texto	Varchar	20	
Anio_pago	Int	5	

**Tabla 21: Tabla Cartera**

Campo	Tipo de dato	Tamaño	Descripción
Id_administrador	Int	255	
Nombre_administrador	Varchar	50	
Clave_administrador	Varchar	50	

**Tabla 22: Tabla Administradores**

### 3.8.2 Relaciones de la base de datos

Entidades participantes	Cardinalidad	Atributos o campo
Cartera – Estudiante	1:1	Carnet_estudiante
Cartera – Estudiante	1:1	Nombre_estudiante
Cartera – Estudiante	1:1	Foto_estudiante
Cartera – Grupos	1:N	Codigo_grupo
Estudiante – Grupos	N:1	Codigo_grupo
Estudiante – Notas	1:N	Nombre_estudiante
Estudiante – Exámenes	1:N	Codigo_grupo
Grupos – Horarios	1:N	Codigo_grupo
Grupos – Notas	1:N	Codigo_grupo
Horarios – Cursos	1:N	Nombre_curso
Horarios – Docentes	N:1	Nombre_docente
Horarios – Materias	N:1	Nombre_materia
Docentes – Notas	1:N	Nombre_docente
Materias – Notas	1:N	Nombre_materia
Cursos – Notas	1:N	Nombre_curso

**Tabla 23: Relaciones de la base de datos**

### 3.9 Diagrama Navegacional

Muestra el conjunto completo de todas las páginas de la aplicación web, y corresponde al mismo del usuario Administrador.

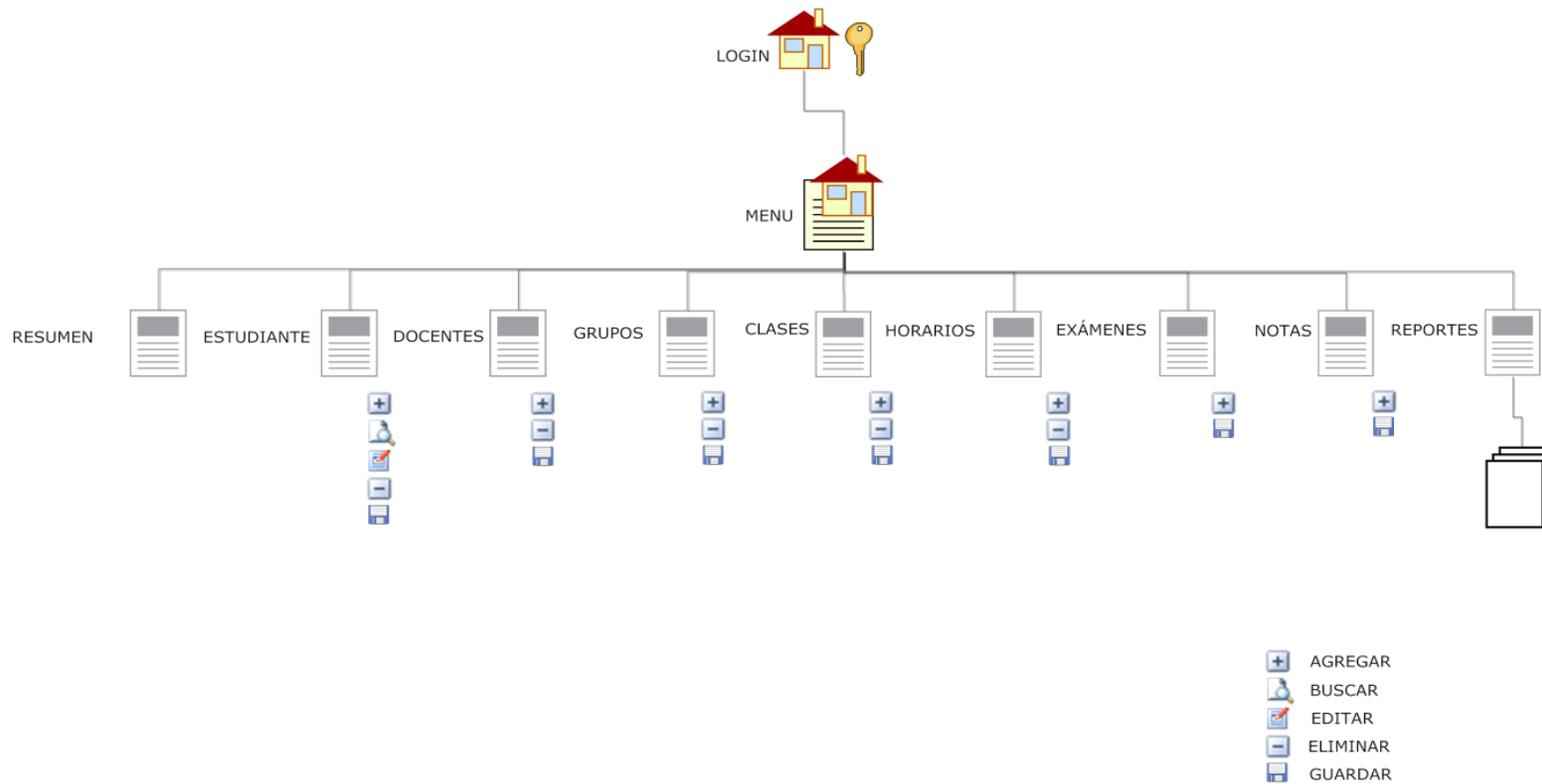
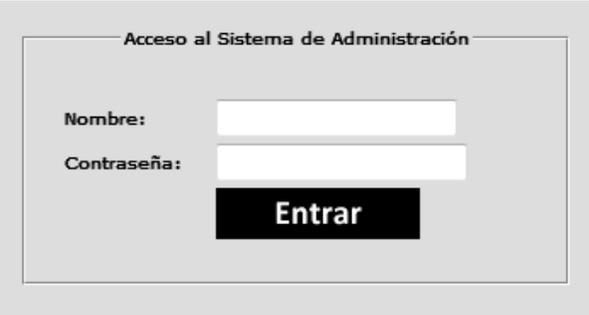


Ilustración 44: Diagrama Navegacional

### 3.10 Interfaces

#### 3.10.1 Login

Esta es la primera interfaz del sistema. Aquí se realiza la autenticación del usuario. Se verifica si la cuenta y el password del usuario se encuentra en la base de datos, si la cuenta está activa y en dependencia del tipo de usuario se comprueban algunas reglas de negocio. Si el usuario es válido, entonces se le permite entrar a la aplicación del Sistema de Registro Académico.



Acceso al Sistema de Administración

Nombre:

Contraseña:

**Entrar**

**Ilustración 45: Login**

#### 3.10.2 Menú Resumen

Este es el menú y aparece luego de que el usuario ha sido autenticado. El menú, presenta las opciones o links disponibles según el tipo de usuario para realizar los pertinentes procedimientos.



**Ilustración 46: Menú Resumen**

### 3.10.3 Menú Grupos

Al acceder al menú grupos podemos notar los grupos que ya han sido inscritos y también nos permite ver sus respectivos integrantes.



Ilustración 47: Menú – Grupos

#### 3.10.3.1 Menú Nuevo Grupo

Si pulsamos el botón “Nuevo Grupo”, se nos despliega los siguientes campos para llenar.

Al finalizar y verificar, procedemos a dar click en el botón “Guardar”, para finalizar el procedimiento.



Ilustración 48: Menú Nuevo Grupo

### 3.10.3.2 Menú Borrar Grupo

Si pulsamos el botón “Borrar Grupo”, se nos despliega todos los grupos ya registrados para elegir el grupo que se desea eliminar.



Ilustración 49: Menú – Borrar Grupo

### 3.10.4 Menú Estudiante

Al acceder al menú estudiantes podemos ver los estudiantes que ya han sido inscritos y sus datos principales.

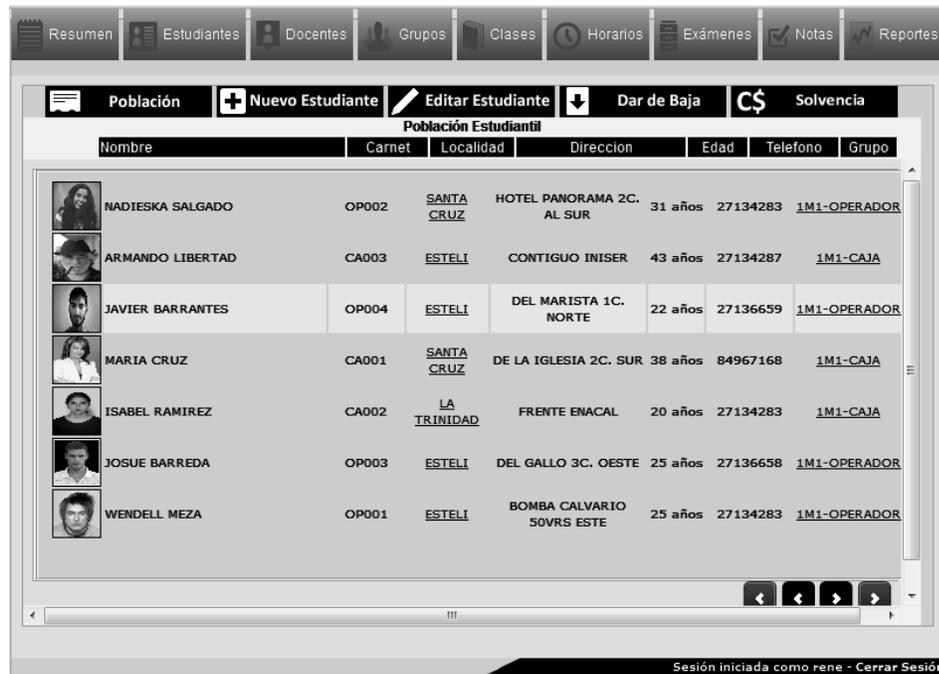


Ilustración 50: Menú Estudiante

### 3.10.4.1 Menú Nuevo Estudiante

Si pulsamos el botón “Nuevo Estudiante”, se nos despliega los siguientes campos para llenar.

Al finalizar y verificar, procedemos a dar click en el botón “Matricular”, para finalizar el procedimiento.

The screenshot shows a web application interface with a top navigation bar containing icons and labels for 'Resumen', 'Estudiantes', 'Docentes', 'Grupos', 'Clases', 'Horarios', 'Exámenes', 'Notas', and 'Reportes'. Below this is a secondary menu with buttons for 'Población', 'Nuevo Estudiante' (highlighted), 'Editar Estudiante', 'Dar de Baja', and 'Solvenca'. The main form area contains several input fields: 'Carnet', 'Nombre', 'Localidad', 'Dirección', 'Año de Nacimiento' (with a dropdown menu showing '2000'), 'Telefono', 'Grupo' (with a dropdown menu showing '00-0015'), and a 'Foto' field with a 'Subir Imagen' button. At the bottom right of the form are two buttons: 'Matricular' and 'Cancelar'.

Ilustración 51: Menú Nuevo Estudiante

### 3.10.4.2 Menú Editar Estudiante

Si pulsamos el botón “Editar Estudiante”, se nos despliega todos los campos para re-escribir el estudiante se desea editar.

Luego damos en el botón “Actualizar”.

The screenshot shows the same web application interface as above, but with the 'Editar Estudiante' button highlighted in the secondary menu. The form fields are identical to the previous screenshot, but the 'Matricular' button is replaced by an 'Actualizar' button at the bottom right.

Ilustración 52: Menú Editar Estudiante

### 3.10.4.3 Menú Eliminar Estudiante

Este procedimiento consta de 2 etapas.

La primera etapa es la búsqueda del sujeto a eliminar.



Ilustración 53: Menú Eliminar Estudiante

Una vez encontrado el estudiante se procede dar de baja.



Ilustración 54: Menú Eliminar Estudiante 2

### 3.10.5 Menú Docente

Al acceder al menú Docentes podemos ver los docentes que ya han sido inscritos y sus datos principales.



Ilustración 55: Menú Docente

#### 3.10.5.1 Menú Inscribir Docente

Si pulsamos el botón “Inscribir Docente”, se nos despliega los siguientes campos para llenar.

Al finalizar y verificar, procedemos a dar click en el botón “Guardar”, para finalizar el procedimiento.



Ilustración 56: Menú Inscribir Docente

### 3.10.5.2 Menú Eliminar Docente

Si pulsamos el botón “Eliminar Docente”, se nos despliega la lista de docentes, donde podemos elegir cual eliminar haciendo click sobre él.



Ilustración 57: Menú Eliminar Docente

### 3.10.6 Menú Clases

Al acceder al menú Clases observamos las materias y el curso que ya han sido registrados.



Ilustración 58: Menú Clases

### 3.10.6.1 Menú Nueva Clase

Se muestra el campo para declarar la materia y elegir el curso correspondiente.

Luego de verificar los datos, procedes a dar click en el “Botón Guardar”.



Ilustración 59: Menú Nueva Clase

### 3.10.6.2 Menú Borrar una Clase

Si pulsamos el botón “Borrar una Clase”, se nos despliega la lista, donde podemos elegir cual eliminar haciendo click sobre el botón “Borrar”



Ilustración 60: Menú Borrar una Clase

### 3.10.7 Menú Horarios

Al acceder al menú Horarios observamos los grupos, docentes, materias y curso con sus respectivos horarios.

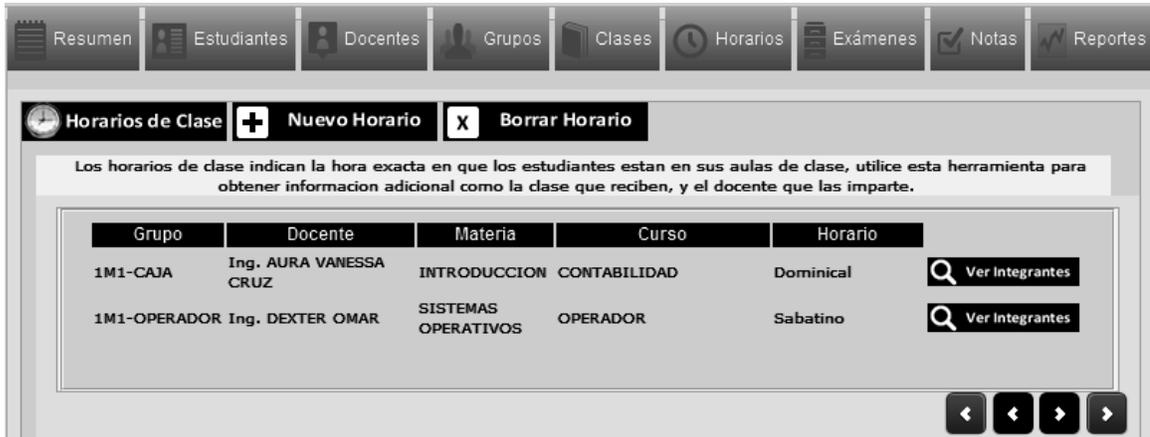


Ilustración 61: Menú Horarios

#### 3.10.7.1 Menú Nuevo Horario

Si pulsamos el botón “Nuevo Horario”, se nos despliega los siguientes campos para llenar y elegir.

Al finalizar y verificar, procedemos a dar click en el botón “Guardar”, para finalizar el procedimiento.

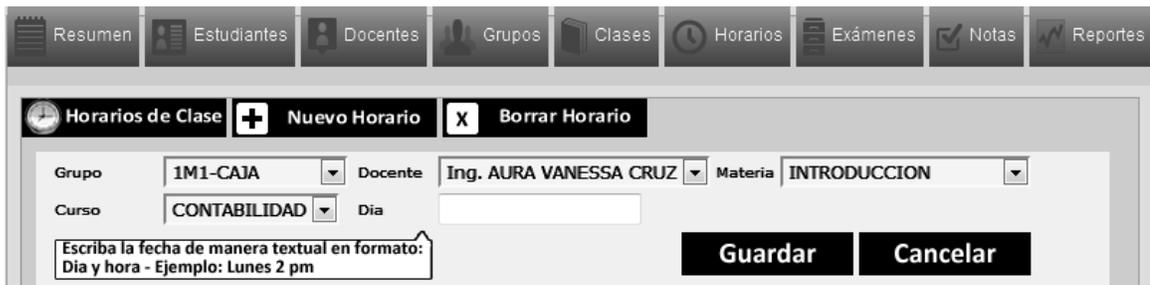


Ilustración 62: Menú Nuevo Horario

### 3.10.7.2 Menú Borrar Horario

Si pulsamos el botón “Borrar Horario”, se nos despliega la lista, donde podemos elegir cual horarios deseamos eliminar haciendo click sobre el botón “Borrar”



Ilustración 63: Menú Borrar Horario

### 3.10.8 Menú Notas

Este proceso consta de 2 etapas:

Primero, al acceder al menú notas, podemos elegir el grupo para asignar notas.

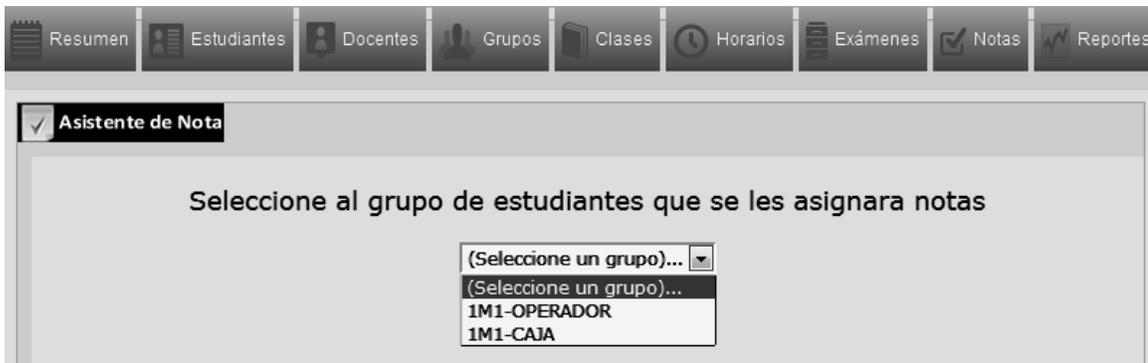


Ilustración 64: Menú Notas

### 3.10.9 Menú Reportes

En este menú se elige el reporte que se desea visualizar.

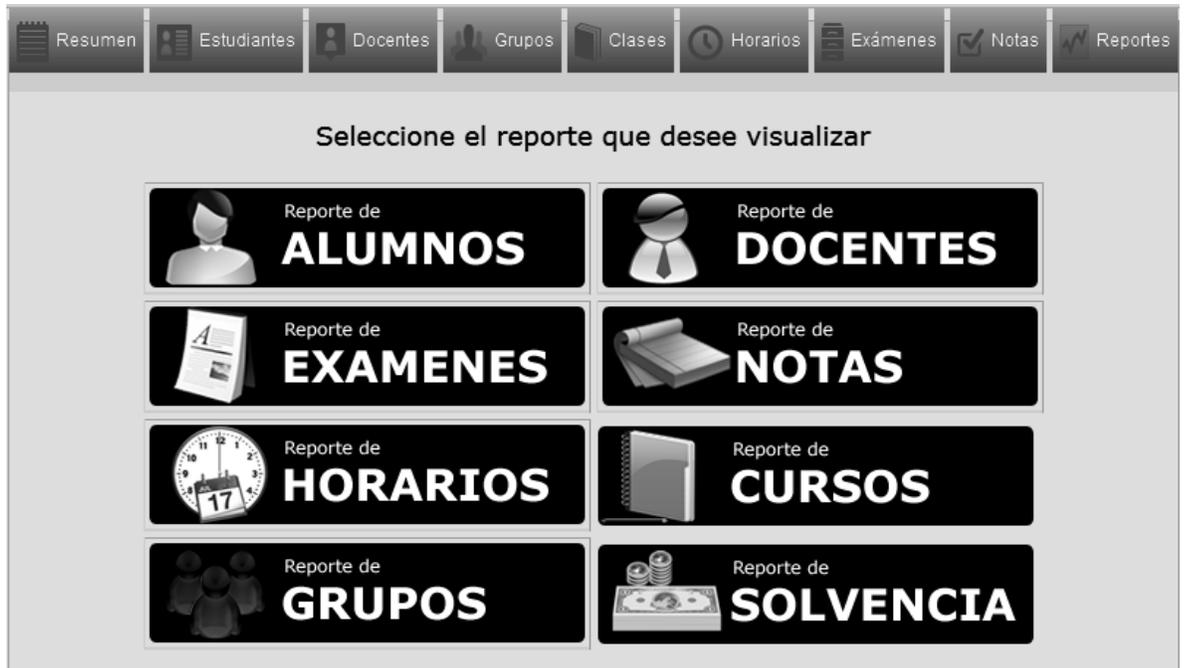


Ilustración 65: Menú Reportes

#### **4. Pruebas al sistema**

Se realizaron pruebas del sistema para ajustar algunos detalles de programación, estas se ejecutaron con 1 servidor y 4 clientes introduciendo información simultáneamente, para realizar esta prueba se hicieron una serie de configuraciones que se detallan a continuación:

Primero se configuró una de un red con 5 computadoras, un servidor el cual contenía la base de datos y 4 clientes que accedían a esta información en forma paralela.

Segundo se Configuró 4 computadoras clientes que contenían el sistema de inscripción para realizar las pruebas, estos al momento de su configuración no presentaron anomalías, resultando las pruebas de acuerdo a lo planeado.

## 5. Implantación del sistema

Se propone no abandonar de la noche a la mañana el sistema que ya tiene la empresa, sino que trabajar paralelamente al otro sistema los primeros días mientras la institución adquiere confianza con este nuevo sistema, y con gran prontitud desearán abandonar el sistema anterior al darse cuenta de la gran cantidad ventajas que se tienen con este modelo. Se recomienda fuertemente que se manejen copias de respaldo locales en un disco duro para casos de emergencia en los que por alguna razón la empresa no tenga acceso al servidor, para que se tenga acceso a los datos de manera local por cualquier emergencia, o si en el peor de los casos, los administradores del sistema perdieran la contraseña de acceso, que el sistema no se perdiera en su totalidad. Igualmente se recomienda pensar en un eficiente manejo de la contraseña de acceso de administrador del sistema, pues es peligroso que ésta esté únicamente en poder de una persona, ya que genera dependencia total en esa persona y el día que éste individuo falte, el sistema se perdería completamente.

Este sistema requiere de uso de por lo menos 2 computadoras (servidor/cliente) con los siguientes requerimientos mínimos:

- ✓ PC Pentium IV 2.5 GHz
- ✓ 1 GB RAM
- ✓ 200 GB en disco duro
- ✓ Impresora para reportes
- ✓ Escritorio
- ✓ Resolución 800 x 600 pixeles
- ✓ Cualquier sistema operativo
- ✓ Navegador Internet Explorer, Mozilla o equivalente.
- ✓ Intranet o red local (switch, modem, medios)

Como se observa, los requerimientos de hardware no son elevados y existen grandes beneficios.

En cuanto a la instalación del sistema en el servidor inmediatamente se suben todos los archivos del programa, incluyendo el código de la página web e imágenes, así como la creación de la base de datos. Esto instala todo en el servidor y quedará listo para su uso.

La creación de respaldo es muy recomendable. La base de datos también se podrá exportar al disco duro desde PhpMyAdmin.

El manual de usuario no requiere de explicaciones de grandes extensiones, pues el uso del sistema es muy amigable y no es complicado. Lo único que se incluye es la sección FAQ en la misma página web, con preguntas frecuentes como por ejemplo, cómo agregar un registro para una tabla, cómo modificarlo o eliminarlo, la manera de imprimir reportes.

En cuanto a la capacitación de personal, se ofrecerá una capacitación gratuita, en la que se explicará el funcionamiento del sistema y cómo se deberá manipular por parte de los encargados de realizar este proceso en registro académico y en la dirección del centro educativo. Se ofrece igualmente garantía y soporte técnico sobre el uso e implementación del sistema. Es evidente que las ventajas son muchas y la satisfacción y beneficios que se obtendrán con este sistema está garantizada.

## **CONCLUSIONES**

El uso de la metodología para el modelado del sistema, fue la mejor elección para dicho fin, debido a la riqueza de su simbología para representar los diferentes procesos que se utilizan.

La automatización de la matrícula, el registro académico y el control arancelario, constituye un paso imprescindible para garantizar el orden y la producción de estadísticas con un mayor nivel de confianza. Y al implementar el prototipo se observó un mejor flujo de trabajo en la institución

Se concluye afirmando que este prototipo es de gran utilidad y representa una solución eficiente, y aunque tiene un coste económico y de infraestructura organizacional, los beneficios superan a los gastos y por lo tanto es una excelente opción con grandes ventajas y soluciones atractivas para la institución.

## **RECOMENDACIONES**

Para la implementación del sistema de información del registro académico en Compulab - Estelí, se requiere tomar en consideración las siguientes recomendaciones:

- Capacitar al personal sobre el uso adecuado del sistema de información, Además; dar a conocer los beneficios que proporcionara al instituto la implementación del mismo.
- Habilitar las estaciones de trabajo donde se instalara el sistema de Registro Académico, con requerimientos mínimos establecidos para el óptimo funcionamiento de la aplicación en tiempo de ejecución.
- Implementar un modelo de red de adecuado que garantice la transferencia de la información, de las estaciones de trabajo clientes hacia el servidor.

## **BIBLIOGRAFÍA**

- ✓ Gutiérrez J. y Chanca 2008, Análisis, diseño e implantación de sistemas de información. Tesis de Grado publicada bajo CC.
- ✓ Ibarra y Mercado, 2004, Sistemas de inscripción de un sistema automatizado, Tesis de Grado no publicada Colegio Universitario de Administración y Mercadeo C.U.A.M. EXTENSIÓN CARACAS
- ✓ Henry F. Korth, Abraham Silberschartz, Fundamentos de Bases de Datos, Segunda Edición, 1993.
- ✓ Kendall y Kendall, Análisis y diseño de Sistemas, Sexta Edición 2006.
- ✓ Alejandro Alberca Manzaneque, Modelos Avanzados de Bases de Datos, UNIVERSIDAD DE CASTILLA-LA MANCHA
- ✓ Vanessa Ojeda Carrión, SISTEMAS DE INFORMACIÓN, Guía didáctica, Sistema de gestión de base de datos.
- ✓ Welling, Luke y Thomson Laura, Desarrollo web con php y MySql php 5 y MySql 4.1 y 5, Anaya Multimedia. Madrid, España. ES. 2005
- ✓ Pressman Roger, Ingeniería de Software, MC. Graw-Hill, 2002.
- ✓ Pressman Roger, Ingeniería del software. Un enfoque práctico, Madrid, McGraw-Hill / Interamericana de España, 1997

- ✓ S. Murugesan, Y. Deshpande, S. Hansen, A. Ginige. "Web Engineering A New Discipline for Development of Web - Based Systems." Lecture Notes in Computer Science" Springer, 2001.
- ✓ S. Dart, "Containing the Web Crisis Using Configuration Management," Proc. 1st ICSE Workshop on Web Engineering, ACM, Los Angeles, May 1999.
- ✓ L. Olsina, G. Lafuente, G. Rossi. "Specifying Quality Characteristics and Attributes for Websites." Lecture Notes in Computer Science 2016 Springer, 2001.
- ✓ Gamma, Y., Helm, R., Johnson R. y Vlissides, J. (1995). Design Patterns: Elements of reusable object-oriented software. USA: Addison Wesley.
- ✓ Grady Booch, James Rumbaugh, Ivar Jacobson, (1996) *El Lenguaje Unificado de Modelado*, Addison Wesley.
- ✓ Schneider G., Winters J.P., (2001) *Applying Use Cases: A Practical Guide*, Addison Wesley.

### **Fuentes de Internet**

German, D. (2003). The Object Oriented Hypermedia Design Method. [Documento en línea]. Disponible: <http://www.telemidia.pucrio.br/oohdm/oohdm.html>.

Koch, N. (2002). Ingeniería de Requisitos en Aplicaciones para la Web—Un estudio comparativo. [Documento en línea]. Disponible: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>.

Sánchez, M. (s. f.). Interfaz de Usuario en el Desarrollo de un Simulador de

Conducción [Documento en línea] Disponible:

[http://www.cc3.net/sanchez/Sanchez\\_M.htm](http://www.cc3.net/sanchez/Sanchez_M.htm).

Schwabe, D. y Rossi, G. (1998). Developing Hypermedia Applications using OOHDM [Documento en línea]. <http://www.oohdm.inf.pucrio.br:8668/space/pessoas+ligadas+ao+OOHDM/ExOOHDM.pdf>.

Schwabe, D., Rossi, G. y Simone, J. (s. f.). Systematic Hypermedia Application Design with OOHDM. [Documento en línea]. <http://wwwx.cs.unc.edu/~barman/HT96/P52/section1.html>.

Silva, D. y Mercerat, B. (2001). Construyendo aplicaciones web con una metodología de diseño orientada a objetos. [Documento en línea]. Disponible: [www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art_c.pdf).

### **Direcciones Varias**

<http://www.desarrolloweb.com/articulos/713.php?manual=27>

<http://www.guiaweb.gob.cl/guia/capitulos/tres/accesorapido.htm>

<http://www.hostito.com/es/faq/index.php>

<http://www.sindominio.net/ayuda/glosario/>

<http://es.wikipedia.org/wiki/Dreamweaver>

<http://www.buscandohost.com/Glosario.asp>

<http://www.16-bits.com.ar/archivos/hacia-un-nuevo-modelo-para-construir-laweb/>

<http://www.informandote.com/jornadasingweb/programa.asp>

<http://www.itlp.edu.mx/publica/tutoriales/analisis/24.htm>

<http://www.ati.es/gt/LATIGOOO/Op96/Ponen6/atio6p06.html>

[http://www.servidorti.uib.es/jbidi/jbidi2000/02\\_2000.pdf](http://www.servidorti.uib.es/jbidi/jbidi2000/02_2000.pdf)

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>

<http://www.informandote.com/jornadasIngWEB/articulos/jiw01.pdf>

<http://www.rspa.com/spi/webe.html>

<http://www.geocities.com/rcascos/>

<http://gidis.ing.unlpam.edu.ar/personas/olsinal/olsinal.html>

<http://www5.ulpgc.es/servidores/bib-inge/indices/647441.htm>

<http://www.creangel.com/uml/home.php>

<http://www.guiaweb.gob.cl/guia/capitulos/tres/accesorapido.htm>

[http://www.servidorti.uib.es/jbidi/jbidi2000/02\\_2000.pdf](http://www.servidorti.uib.es/jbidi/jbidi2000/02_2000.pdf)

<http://www.creangel.com/uml/diagramas.php>

[http://www.dcc.uchile.cl/~afierro/d\\_clases.html](http://www.dcc.uchile.cl/~afierro/d_clases.html)

**ANEXOS**

## **Anexos 1**

### **Entrevista y guía de observación**

Universidad Nacional de Ingeniería

UNI-Norte

*Sede Estelí*

Entrevista dirigida a personal administrativo Compulab-Estelí.

*Objetivo:* Conocer los procesos del área administrativa y determinar las necesidades de la automatización las inscripciones, registro académico y arancelario Compulab-Estelí.

Referencia Técnica:

Método: Entrevista

Técnica: No estructurada

Fecha: \_\_\_\_\_

Duración: \_\_\_\_\_

Lugar: \_\_\_\_\_

Sujeto: \_\_\_\_\_

Sexo: \_\_\_\_\_

#### I. Análisis de sistema actual

1. Como realiza usted el proceso de inscripción y que insumos necesita?
2. Como realiza usted el proceso de registro académico?
3. Como realiza usted el proceso de control arancelario?

4. Cuáles son las limitantes que enfrenta actualmente con el sistema actual de registro y control arancelario?

5. La institución cuenta con un equipo de cómputo que lleve la organización correcta de los procesos de inscripción, registro, y arancelarios?

Si  No

6. Qué opina usted del método actual con el cual se llevan a cabo los procesos antes mencionados?

7. Cuánto cuesta la matrícula para cada curso?

8. Cuanto es el arancel para cada curso?

9. Cuantos cursos hay?

10. Cuanto es la duración por clase y curso?

11. ¿Quién realiza el proceso de?

Matrícula

Cobro

Registro académico: notas, promedios, informes, control

12. Que documentos se extienden a cada estudiante en cada proceso?

II. Determinación de la necesidad del sistema

1. Cree que el desarrollo de un sistema automatizado facilitaría el trabajo y de qué forma?

2. Qué clase de sistema informático le gustaría y que necesita que haga el sistema?

3. Quienes usarían el sistema?

4. Le gustaría tener algún tipo de estadística o informe?

Si X No \_\_\_

5. Trabaja usted con computadoras?

Si X No \_\_\_

6. Cree necesario el desarrollo de un manual para el nuevo sistema o capacitación alguna?

Si X No \_\_\_

Algo más que desee agregar?

Tabla de habilidades mínimas.

Usuarios	Administradora	Secretaria Académica
Aspectos		
Estudios		
Primaria	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Secundaria	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Estudios superiores	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Curso caja	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Curso operador	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___
Habilidades		
Impresión de documentos	Alta: <u>X</u> Media: ___ Baja: ___	Alta: <u>X</u> Media: ___ Baja: ___
Agilidad en teclado	Alta: <u>X</u> Media: ___ Baja: ___	Alta: <u>X</u> Media: ___ Baja: ___
Uso de paquete Ofimático	Si: <u>X</u> No: ___	Si: <u>X</u> No: ___

## Anexos 2

### Guía de Observación.

Nombre del observador:

Método: Observación no Duración:  
participativa

Lugar: Fecha de aplicación:

Desempeño a Evaluar: Proceso de inscripción.

Instrucciones: Observa detenidamente las actividades que se realizan en el proceso de inscripción por estudiante y llenar cada cuadro con la información correspondiente, es importante anotar las observaciones pertinentes.

Actividad	Descripción	Actor	Duración	Observación

Desempeño a Evaluar: Proceso de registro académico.

Instrucciones: Observa detenidamente las actividades que se realizan en el proceso de registro académico y llenar cada cuadro con la información correspondiente, es importante anotar las observaciones pertinentes.

Actividad	Descripción	Actor	Duración	Observación

Desempeño a Evaluar: Proceso de registro arancelario.

Instrucciones: Observa detenidamente las actividades que se realizan en el proceso de registro arancelario y llenar cada cuadro con la información correspondiente, es importante anotar las observaciones pertinentes.

Actividad	Descripción	Actor	Duración	Observación

## **Anexos 3**

### **Glosario**

**Análisis de sistemas:** Estudio de una tarea o función para comprenderla y encontrar mejores maneras de realizarla.

**Aplicación:** Programa diseñado para una determinada función

**Base de Datos:** Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos

**Cliente:** Programa que demanda servicios de otra computadora llamada servidor, y se hace cargo de la interacción necesaria con el usuario.

**Depuración:** Detección, localización y eliminación de errores en un programa. También llamado debugging.

**Diagrama de Flujo:** Es la representación gráfica de una secuencia de instrucciones de un programa que ejecuta un computador para obtener un resultado determinado.

**Diagrama Entidad Relación:** Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de *entidades*, que son objetos que existen y que se distinguen de otros por sus características

**FAQ (Frequently Asked Questions):** Documentos informativos que recogen las respuestas a las preguntas formuladas más frecuentemente por los usuarios de un servicio determinado.

**Información:** Es el resultado del procesamiento de datos. Todo aquello que permite adquirir cualquier tipo de conocimientos.

**Informática:** Es la ciencia del tratamiento automático de la información mediante una computadora. La informática es un amplio campo que incluye los

fundamentos teóricos, el diseño, la programación y el uso de las computadoras (ordenadores).

Módem (MOdulador - DEModulador): Aparato que convierte las señales digitales en analógicas y viceversa. Permite la comunicación entre dos computadoras a través de la línea telefónica.

Red de Área Local (LAN): Red de área local restringida a una zona limitada, por lo general, el mismo edificio o la misma planta de un edificio.

Software libre: Es el que respeta la libertad del usuario, ateniéndose a las 4 libertades que plantea la Free Software Foundation: De usarlo para el fin que se quiera; De realizar copias; De modificarlo para ajustarlo a nuestro gusto; De distribuir las mejoras. Adicionalmente se suele decir que la única restricción es que cada uno que reciba ese software, debe heredar esas libertades.

Sistema de Información: Está constituido por la base de datos, todos los programas de ingreso, actualización, consulta e informes de datos y los procedimientos manuales y por máquina.