



# **UNIVERSIDAD NACIONAL DE LOJA**

## **ÁREA DE ENERGIA, INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES**

**TEMA:**

**“SISTEMA DE GESTIÓN ACADÉMICA E IMPLEMENTACIÓN DE LA RED  
PARA LA UNIDAD EDUCATIVA ANEXA A LA UNIVERSIDAD NACIONAL  
DE LOJA”**

**AUTORES:**

**VERÓNICA ISABEL LUDEÑA GONZÁLEZ  
EDISON LEONARDO CORONEL ROMERO**

**DIRECTOR:**

**ING. MILTON LABANDA**

**LOJA - ECUADOR  
2007**

Ing. Milton Labanda Jaramillo  
Director de Tesis

### **Certifico**

Que he revisado minuciosamente el trabajo de tesis denominado “SISTEMA DE GESTIÓN ACADÉMICA E IMPLEMENTACIÓN DE LA RED PARA LA UNIDAD EDUCATIVA ANEXA A LA UNIVERSIDAD NACIONAL DE LOJA” , desarrollado por los egresados Verónica Isabel Ludeña González y Edison Leonardo Coronel Romero, por lo cual autorizo su presentación y sustentación

Loja, Julio de 2007

---

Ing. Milton Labanda

**AUTORIA**

Las opiniones, criterios, análisis, diseño, conclusiones y recomendaciones vertidos en el presente informe, corresponden exclusivamente a sus autores

Edison I. Coronel Romero

Verónica I. Ludeña González

## **DEDICATORIA**

Dedico la presente tesis de Grado a mis Padres y a mis Hermanas ya que sin ellos no hubiera sido posible obtener tan deseada meta como lo es de obtener el Título de Ingeniería en Sistemas

Verónica I Ludeña González

Dedico el presente trabajo a mis Padres Telmo y Leneida, a mis Hermanas Ivonne, Angie y Aylin, a mi Tía Mabel, a mi Sobrino Gabriel y a todos mis Familiares por su amor, comprensión y apoyo durante el desarrollo de este proyecto.

Edison L Coronel Romero

## **AGRADECIMIENTO**

Agradezco a Dios por haberme regalado la vida, a mis Padres por todo su amor, cariño y comprensión durante toda mi vida estudiantil y a mis Hermanas por estar siempre junto a mí apoyándome para siempre conseguir mis metas

Verónica I Ludeña González

Agradezco a la Universidad Nacional de Loja, por permitir mi formación en sus aulas, al Ingeniero Milton Labanda por su tiempo y dedicación al asesoramiento del presente proyecto, a mis familiares que me apoyaron durante el desarrollo de una de mis metas en la vida

Edison L Coronel Romero

## RESUMEN

La nueva reforma curricular de la Educación Básica en el País, acarrea consigo por un lado la integración de antiguas estructuras desvinculadas entre si, como lo son Jardín, Escuela y Colegio, para proporcionar diez años de educación básica y tres años de bachillerato; y, por otro lado trae consigo el incremento de servicios y actividades institucionales tanto en el área académica como administrativa.

La Universidad Nacional de Loja como ente activo de la sociedad busca contribuir con el arduo trabajo que realizan los docentes y administrativos que laboran en su unidad educativa anexa solucionando de alguna forma parte de los problemas que tienen, como lo son el manejo del flujo de información, generación y obtención de reportes en forma eficiente y oportuna, así mismo mejorando ciertos procesos de gestión académica, y de esta forma cumplir con su compromiso con la juventud lojana.

La presente investigación tiene como objeto la implantación de un Sistema de Gestión Académico para la Unidad Educativa Anexa a la Universidad Nacional de Loja con el propósito de coadyuvar al desarrollo y al proceso de integración de la unidad educativa al proporcionar la implantación de la red como base de la misma y una herramienta informática de apoyo a la gestión académica que permita manejar el volumen de información, servicios y actividades que se generarán a partir de dicha integración.

Para la consecución de dicha meta se ha tomado como herramientas de trabajo en relación de costos y beneficios la siguiente estructura para el desarrollo del software: como base de datos MySQL, por las características que brinda como base de datos y su licencia al ser software libre, para el desarrollo de la interfaz con el usuario java por las características ampliamente conocidas de robustez, flexibilidad, pero a través del IDE NetBeans como programa editor del software; como metodología ICONIX, que es un híbrido entre la programación extrema XP y el muy conocido RUP. A todo lo antes mencionado se une todas las librerías y aplicaciones de código abierto y licencia de distribución libre en la Web como lo son: jgoodies, swinglab, jfreereport, javahelp e izpack para facilitar el trabajo del analista programador y obtener software de calidad a bajo costo y en el menor tiempo posible para satisfacer a los usuarios y la facilidad y adaptación del software ante posibles cambios y evolución según el desarrollo de la Unidad Educativa.

## 2 ÍNDICE

AUTORIA .....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
2 ÍNDICE.....	vii
2.1 Tablas.....	x
2.2 Ilustraciones .....	xi
3 INTRODUCCIÓN .....	1
4 METODOLOGÍA .....	2
4.1 BREVE DESCRIPCIÓN DE LAS METODOLOGÍAS ANALIZADAS.....	2
4.2 COMPARACIÓN DE METODOLOGÍAS.....	5
4.2.1 Características del Proyecto. ....	5
4.2.2 Recursos Humanos .....	5
4.2.3 Requerimientos de Recursos Humanos para la Adopción.....	6
4.2.4 Facilidad de Aprendizaje de Metodología. ....	6
4.2.5 Ciclo de Vida de Actividades Según la Metodología.....	7
4.2.6 Facilidad de Implementación. ....	8
4.2.7 Consideraciones Finales. Ventajas y Desventajas.....	8
4.3 HERRAMIENTAS DE RECOLECCIÓN DE DATOS.....	9
4.4 ICONIX .....	9
4.4.1 Introducción. ....	9
4.4.2 Las Capacidades de ICONIX.....	11
4.4.3 El Modelo del Dominio. ....	13
4.4.4 Modelo de Casos de Uso.....	15
4.4.5 El Análisis de Robustez. ....	17
4.4.6 Diagrama de Secuencia.....	21
5 FUNDAMENTACIÓN TEÓRICA.....	26
5.1 Capítulo 1. JAVA.....	26
5.1.1 Características de Java .....	26
5.2 Capítulo 2. Open Source.....	28
5.2.1 Hibérnate.....	28
5.2.1.1 Visión General.....	29
5.2.1.2 Estados de instancia.....	32
5.2.1.3 Ejemplo de Hibernate. ....	33
5.2.2 SwingLab.....	34
5.2.3 JGoodies.....	47
5.2.4 JavaHelp.....	52
5.2.4.1 Características Básicas.....	53
5.2.4.2 Herramientas. ....	53
5.2.4.3 Librerías.....	54
5.2.4.4 Creación de la Ayuda con JavaHelp.....	54
5.2.4.5 Incorporación de la ayuda en la Aplicación.....	56
5.2.4.5.1 Clases Básicas Implicadas .....	56
5.2.4.5.2 Pasos para incorporar la ayuda en la Aplicación.....	57
5.2.5 JfreeReport .....	59
5.2.5.1 Pasos Básicos. ....	60
5.2.5.2 Aplicación de Ejemplo.....	60
5.2.6 IzPack.....	64
5.3 Capítulo 5. Bases de Datos .....	66

5.3.1	Modelos de datos.....	66
5.3.1.1	Modelos lógicos basados en objetos.....	66
5.3.1.1.1	Modelo Entidad-Relación.....	66
5.3.1.1.1.1	Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.....	67
5.3.1.1.1.2	Relaciones de Generalización-Especialización.....	68
5.3.1.1.1.3	Transformación del Modelo a un Esquema Relacional.....	69
5.3.1.1.1.4	Transformación del Modelo a un Esquema Híbrido.....	72
5.3.2	Independencia de los Datos.....	73
5.3.3	Lenguaje de Definición de Datos.....	73
5.3.4	Lenguaje de Manipulación de Datos.....	74
5.3.5	Manejador de Bases de Datos.....	74
5.3.6	SQL.....	75
5.3.6.1	Introducción.....	75
5.3.6.2	Componentes del SQL.....	76
5.4	Redes.....	78
5.4.1	Posibles Tecnologías a Utilizar.....	78
5.4.1.1	Gigabit Ethernet Sobre Cobre.....	78
5.4.1.1.1	Cómo funciona Gigabit Ethernet.....	78
5.4.2	Cables de Fibra Óptica.....	80
5.4.3	Direcciones IP Distribución De Subredes Y Hosts.....	80
5.4.4	Direcciones IP Privadas.....	82
6	EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN.....	83
7	DESARROLLO DE LA PROPUESTA ALTERNATIVA.....	87
7.1	Introducción.....	87
7.2	Análisis.....	90
7.2.1	Identificar Casos de Uso.....	90
7.2.2	Diagramas de Paquetes.....	94
7.2.3	Requerimientos Funcionales de los Casos de Uso.....	95
7.2.4	Glosario de Términos.....	98
7.2.5	Modelo del Dominio.....	103
7.3	Diseño.....	106
7.3.1	Descripción de los Casos de Uso.....	106
7.3.1.1	CU01 - Caso de uso Ingresar al Sistema.....	106
7.3.1.2	CU02 – Caso de Uso Configurar Sistema.....	107
7.3.1.3	CU03 – Caso de Uso Mantener Unidad Educativa.....	109
7.3.1.4	CU04 – Caso de Uso Mantener Secciones.....	111
7.3.1.5	CU05 – Caso de Uso Mantener Niveles.....	113
7.3.1.6	CU06 – Caso de Uso Mantener Año Lectivo.....	115
7.3.1.7	CU07 – Caso de Uso Mantener Configuración.....	116
7.3.1.8	CU08 – Caso de Uso Mantener Periodos.....	117
7.3.1.9	CU09 – Caso de Uso Mantener Reportes.....	118
7.3.1.10	CU10 – Caso de Uso Mantener Dominios.....	119
7.3.1.11	CU11 – Caso de Uso Mantener Cursos.....	120
7.3.1.12	CU12 – Caso de Uso Mantener Recuperación.....	121
7.3.1.13	CU13 – Caso de Uso Mantener Supletorio.....	122
7.3.1.14	CU14 – Caso de Uso Mantener Inscripción.....	123
7.3.1.15	CU15 – Caso de Uso Mantener Matricula.....	125
7.3.1.16	CU16 – Caso de Uso Mantener Pensum.....	127
7.3.1.17	CU17 – Caso de Uso Mantener Especialidades.....	129



7.3.1.18	CU18 – Caso de Uso Mantener Materias .....	131
7.3.1.19	CU19 – Caso de Uso Asignar Materia a Paralelo .....	133
7.3.1.20	CU20 – Caso de Uso Mantener Profesor .....	135
7.3.1.21	CU21 – Caso de Uso Asignar Alumno a Paralelo.....	137
7.3.1.22	CU22 – Caso de Uso Asignar Paralelos a Especialidad .....	139
7.3.1.23	CU23 – Caso de Uso Asignar Materia a Profesor .....	141
7.3.1.24	CU24 – Caso de Uso Registrar Asistencia.....	143
7.3.1.25	CU25 – Caso de Uso Registrar Acreditación.....	145
7.3.1.26	CU26 – Caso de Uso Mantener Tipo de Usuario .....	147
7.3.1.27	CU27 – Caso de Uso Asignar Función.....	149
7.3.1.28	CU28 – Caso de Uso Mantener Usuarios .....	150
7.3.1.29	CU29 – Caso de Uso Mantener Cuenta.....	152
7.3.1.30	FR01 – Fragmento Buscar .....	153
7.3.1.31	FR02 – Fragmento imprimir Reporte .....	155
7.3.1.32	FR03 –Fragmento Administrar Ayuda .....	156
7.3.2	Diagramas de Secuencia. ....	158
7.3.2.1	DS01-DR01 Ingresar al Sistema .....	158
7.3.2.2	DS02- DR02 Configurar Sistema.....	159
7.3.2.3	DS03–DR03 Mantener Unidad Educativa.....	160
7.3.2.4	DS04–DR04 Mantener Secciones .....	161
7.3.2.5	DS05–DR05 Mantener Niveles.....	163
7.3.2.6	DS06–DR06 Mantener Año Lectivo .....	165
7.3.2.7	DS07–DR07 Mantener Configuración .....	166
7.3.2.8	DS08–DR08 Mantener Periodos .....	166
7.3.2.9	DS09–DR09 Mantener Reportes.....	167
7.3.2.10	DS10–DR10 Mantener Dominios .....	167
7.3.2.11	DS11–DR11 Mantener Cursos.....	168
7.3.2.12	DS12–DR12 Mantener Recuperación .....	169
7.3.2.13	DS13–DR13 Mantener Supletorio .....	169
7.3.2.14	DS14–DR14 Mantener Inscripción .....	170
7.3.2.15	DS15–DR15 Mantener Matricula.....	172
7.3.2.16	DS16–DR16 Mantener Pensum .....	174
7.3.2.17	DS17–DR17 Mantener Especialidades.....	176
7.3.2.18	DS18–DR18 Mantener Materias .....	177
7.3.2.19	DS19–DR19 Asignar Materia a Paralelo.....	178
7.3.2.20	DS20–DR20 Mantener Profesor .....	179
7.3.2.21	DS21–DR21 Asignar Alumno a Paralelo .....	181
7.3.2.22	DS22–DR22 Asignar Paralelos a Especialidad.....	183
7.3.2.23	DS23–DR23 Asignar Materia a Profesor .....	184
7.3.2.24	DS24–DR24 Registrar Asistencia.....	185
7.3.2.25	DS25–DR25 Registrar Acreditación .....	186
7.3.2.26	DS26–DR26 Mantener Tipos de Usuario .....	188
7.3.2.27	DS27–DR27 Asignar Función .....	190
7.3.2.28	DS28–DR28 Mantener Usuario. ....	191
7.3.2.29	DS29–DR29 Mantener Cuenta.....	193
7.3.3	Diagrama de Clases. ....	194
8	Implementación.....	201
8.1	Diagrama de Componentes. ....	201
8.2	Pruebas del Sistema. ....	202
8.2.1	Plan de pruebas.....	202

8.2.1.1	Introducción .....	202
8.2.1.2	Alcance. ....	203
8.2.1.3	Equipo de Trabajo .....	205
8.2.1.4	Referencias .....	205
8.2.1.5	Estrategia de pruebas .....	205
8.2.1.6	Tipos de pruebas.....	206
8.2.1.7	Metodología .....	206
8.2.1.8	Informe de pruebas .....	209
9	Diseño de Red Tentativo .....	220
9.1	Introducción .....	220
9.2	Estructura Física.....	221
9.3	Cableado Horizontal.....	224
9.4	Cable UTP.....	225
9.5	Jacks Y Tomas .....	225
9.6	Distribución de Cableado: .....	226
9.7	Racks .....	226
9.8	Especificaciones Ambientales. Temperatura y Humedad. ....	228
9.9	Topologías de la Red .....	228
9.10	Topología Física.....	229
9.10.1	18SWITCH GIGABIT ETHERNET D-LINK 10/100/1000 16 PUERTOS SOHO DGS-1216T .....	230
9.11	Topología Lógica .....	231
9.12	Selección de Cada Uno de los Dispositivos y sus Especificaciones .....	232
9.12.1	Dispositivos por Planta .....	232
9.12.1.1	Equipos para la Sección: Colegio.....	233
9.12.2	Equipos para la Sección: Escuela .....	233
9.12.3	Equipos para la Sección: Jardín.....	233
9.13	Asignación de Direcciones IP. ....	237
10	VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL .....	239
11	CONCLUSIONES .....	241
12	RECOMENDACIONES .....	243
13	BIBLIOGRAFÍA .....	245
15	ANEXOS.....	246
15.1	ANEXO 1. ANTEPROYECTO.....	247
15.2	ANEXO 2. MANUAL DE USUARIO.....	248
15.3	ANEXO 3. ENTREVISTAS.....	249
15.4	ANEXO 4. REGISTRO DE CASOS Y BITÁCORA DE PRUEBAS.....	250
15.5	ANEXO 5. ENCUESTAS. ....	251
15.6	ANEXO 6. CERTIFICACIONES.....	252

## 2.1 Tablas

Tabla 1.	Comandos DLL.....	76
Tabla 2.	Comandos DML.....	76
Tabla 3.	Cláusulas.....	77
Tabla 4.	Operadores Lógicos.....	77
Tabla 5.	Operadores de Comparación.....	78
Tabla 6.	Funciones.....	78

## 2.2 Ilustraciones

Gráfico 4-1. Metodología Según el Tamaño del Proyecto.....	5
Gráfico 4-2. Características del Equipo Trabajo. ....	5
Gráfico 4-3. Características del Equipo de Trabajo Según la Metodología.....	6
Gráfico 4-4. Facilidad de Aprendizaje de metodología. ....	7
Gráfico 4-5. Ciclo de Vida de Actividades Según la Metodología. ....	7
Gráfico 4-6. Facilidad de Implementación Según Metodología. ....	8
Gráfico 4-7. Ventajas y Desventajas Según Metodología. ....	8
Gráfico 4-8. Cuadro para manejar Casos de Uso en el Modelamiento de Objetos .....	10
Gráfico 5-1. Arquitectura de Hibernate .....	29
Gráfico 5-2 Arquitectura Hibernate Detallada .....	30
Gráfico 5-3 Arquitectura Detallada Compleja .....	30
Gráfico 5-4 Visor de JavaHelp .....	53
Gráfico 5-5. Asistente de Bibliotecas JavaHelp .....	58
Gráfico 5-6. Ejemplo Pantalla Principal Ayuda Generada con JavaHelp.....	59
Gráfico 5-7. Ejemplo Reporte .....	64
Gráfico 5-8. Arquitectura del Compilador IzPack.....	65
Gráfico 5-9 Relación de Generalización-Especialización.....	69
Gráfico 5-10 Esquema Relacional Resultante de la Transformación como Tabla Jerárquica. .	69
Gráfico 5-11 Esquema Relacional Resultante de la Transformación Usando Partición Vertical. .....	71
Gráfico 5-12 Esquema Relacional Resultante de la Transformación Usando Partición Horizontal. ....	72
Gráfico 5-13 Esquema Relacional Resultante de la Transformación Usando Partición Horizontal.....	73
Gráfico 9-1. Cableado Horizontal.....	224
Gráfico 9-2. Jack Rj-45.....	226
Gráfico 9-3. Racks .....	227

### 3 INTRODUCCIÓN

Los sistemas de Gestión Académica son la base del desarrollo académico administrativo de todas las instituciones que buscan ingresar a una nueva era de Educación Básica en el país y a la vez mantenerse dentro de los estándares que exigen nuestros impúberes para obtener de esta forma la excelencia académica que nuestro país se merece logrando así mejorar nuestra economía y el futuro de nuestra patria.

Debido a los cambios que exige y que conllevan dicha excelencia, quienes estamos inmersos de alguna forma en lo que es la ingeniería en sistemas estamos llamados a participar en el desarrollo y fortalecer este ente tan importante como lo es la educación aportando con pilares fundamentales en los cuales se desarrolla y se ampara como lo es el manejo adecuado, seguro y oportuno de la información que posee.

Y para obtener los mejores resultados en tiempo de desarrollo y ejecución nos decidimos por una excelente metodología de trabajo, ICONIX, la misma que nos llevará a obtener un resultado óptimo en la elaboración de nuestro software.

Para desarrollar nuestra interfaz con el usuario, tanto como la programación del software optamos por el mejor lenguaje de programación, JAVA, por su flexibilidad y portabilidad.

Para la base de datos utilizar MySQL, porque cumple con las características necesarias para un buen intercambio de datos entre el software y la base de datos, tomando en cuenta el volumen de información que gestiona la Unidad Educativa anexa a la Universidad Nacional de Loja.

Como sistema operativo base Linux.

## 4 METODOLOGÍA

Para la ejecución de este proyecto se ha tomado en cuenta algunos lineamientos a seguir como metodología de trabajo, ya que todo trabajo intelectual requiere del uso de un método y/o procedimiento que lo conduzca al conocimiento para llegar a cumplir un objetivo. Para lo cual realizaremos el análisis de algunas metodologías, las cuales se listan a continuación:

### 4.1 BREVE DESCRIPCIÓN DE LAS METODOLOGÍAS ANALIZADAS.

**“Extreme Programming (XP)**, es tal vez el más conocido de las metodologías Ágiles. Sus adeptos son básicamente desarrolladores que se cansaron de los modelos rígidos y ven en XP un modelo ágil, rápido, flexible y de calidad. Fue pensado para equipos de desarrollo pequeños (menos de 20 personas) aunque hay reportes de su uso en grupos más grandes.

XP presenta una metodología minimalista basado en un pequeño conjunto de valores, principios y prácticas. Valores: comunicación, simplicidad, retroalimentación y coraje. Estas prácticas deben tomarse como directrices y no como reglas:

Planificación, Pequeñas liberaciones de software, Metáfora, diseño simple, reconstrucción, pruebas, programación en parejas, integración continua, 40 horas a la semana, código estándar.

**Proceso Unificado de Desarrollo de Software (RUP)**. Provee una metodología disciplinada para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando el horario y el presupuesto).

Fue desarrollado y es mantenido por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte (personalizable). Maneja casos de uso, es de arquitectura céntrica y utiliza UML como lenguaje de notación. Captura muchas de las mejores prácticas del desarrollo de software moderno. RUP repite una serie de ciclos

compuestos por iteraciones a lo largo de la vida del sistema. Es incremental e iterativo. Cada ciclo consiste en 4 fases: inicio, elaboración, construcción y transición.

Cada fase se descompone en iteraciones. Cada fase termina en un hito, que consiste en un conjunto de artefactos. Dentro de cada fase, existen 9 áreas de trabajo (disciplinas).

- Modelado del Negocio.
- Requerimientos.
- Análisis y diseño.
- Implementación.
- Pruebas.
- Despliegue.
- Configuración Administrativa.
- Administración.
- Despliegue.

**ICONIX** es un modelo simple que puede ser instanciado como una mejora de RUP. La metodología ICONIX se encuentra ubicada entre la metodología RUP y la metodología XP. Maneja casos de uso como RUP, pero no posee todas sus características. Es relativamente pequeño como XP, pero no descarta las notaciones de análisis y diseño como lo hace XP.

La metodología ICONIX es minimalista y se focaliza en el área que queda entre los casos de uso y el código. "¿Cómo obtener desde los casos de uso el código?".

Los 3 elementos fundamentales de ICONIX son:

- Es iterativo e incremental.
- Capacidad de rastreo (en cada paso hay una referencia a los requerimientos.)
- Hace un uso estilizado de UML.

**Object-Oriented Software Process (OOSP).** Está formado por una colección de “modelos de procesos”. Un modelo de proceso es a su vez una colección de técnicas genéricas, acciones y/o tareas que solucionan un problema específico de procesos de software considerando las fuerzas y factores del problema.

Fase del Modelo de Proceso: determina las interacciones entre las etapa del modelo de proceso dentro de una fase del proceso.

Etapas del Modelo de Proceso: determina las tareas, usualmente llevadas a cabo en forma iterativa, dentro de una etapa del proceso.

Tareas del Modelo de Proceso: da soluciones probadas para problemas de bajo nivel dentro de cada etapa. Los modelo de proceso describen lo que se debe hacer pero no los detalles exactos de cómo hacerlo. Son bloques reusables a partir de los cuales se configura un modelo de proceso de software que se encuadre en las necesidades particulares de la organización.

**SCRUM.** Es un modelo agile que puede ser utilizado para manejar y controlar el desarrollo de software. Es una metodología que introduce las ideas de flexibilidad, adaptabilidad y productividad. No define ninguna técnica específica de desarrollo de software para la implementación. Fue desarrollado para manejar el proceso de desarrollo de sistemas, basado en manejar lo impredecible. SCRUM se concentra en cómo los miembros del equipo deben funcionar para producir un sistema flexible en un ambiente que cambia constantemente. Variables ambientales y técnicas impredecibles:

Requerimientos.

Recursos.

Tecnología.

Reuniones Continuas: diarias, de revisión, de diseño, de planificación, etc.

**Dynamic System Development Method (DSDM).** Nació como una formalización de las prácticas RAD, y es un modelo de proyecto para desarrollo de aplicaciones de negocios con períodos estrechos. El principal dominio de aplicación son los proyectos pequeños y/o medianos, con equipos de no más de 20 personas.

Filosofía: fijar tiempo y recursos, y luego ajustar consecuentemente la funcionalidad necesaria.

Existen 9 principios básicos que definen la ideología de DSDM:

- Participación Activa del Usuario: Administrador, Consejero y Visionario.
- Entrega frecuente de productos.

- Cambios reversibles durante el desarrollo.
- Requerimientos de fondo.”<sup>1</sup>

## 4.2 COMPARACIÓN DE METODOLOGÍAS.

### 4.2.1 Características del Proyecto.

- ¿Cuál es el tamaño esperado del proyecto?
- ¿Qué tamaño tiene el equipo de desarrollo?
- ¿Cuál es la complejidad del problema a resolver?

Process Models	Project Size	Team Size	Problem Complexity
RUP	Medium / Large	Medium/Large	Medium/High
ICONIX	Small / Medium	Small/Medium	Small/Medium
OOSP	Medium / Large	Medium/Large	Medium/High
XP	Small/Medium	Small	Medium/High
SCRUM	Small/Medium	Small	Medium/High
DS DM	Small/Medium	Small/Medium	Medium/High

Gráfico 4-1. Metodología Según el Tamaño del Proyecto.<sup>2</sup>

### 4.2.2 Recursos Humanos



Gráfico 4-2. Características del Equipo Trabajo.<sup>3</sup>

<sup>1</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) “Fanatismo” o desarrollo de software

<sup>2</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) “Fanatismo” o desarrollo de software



### 4.2.3 Requerimientos de Recursos Humanos para la Adopción

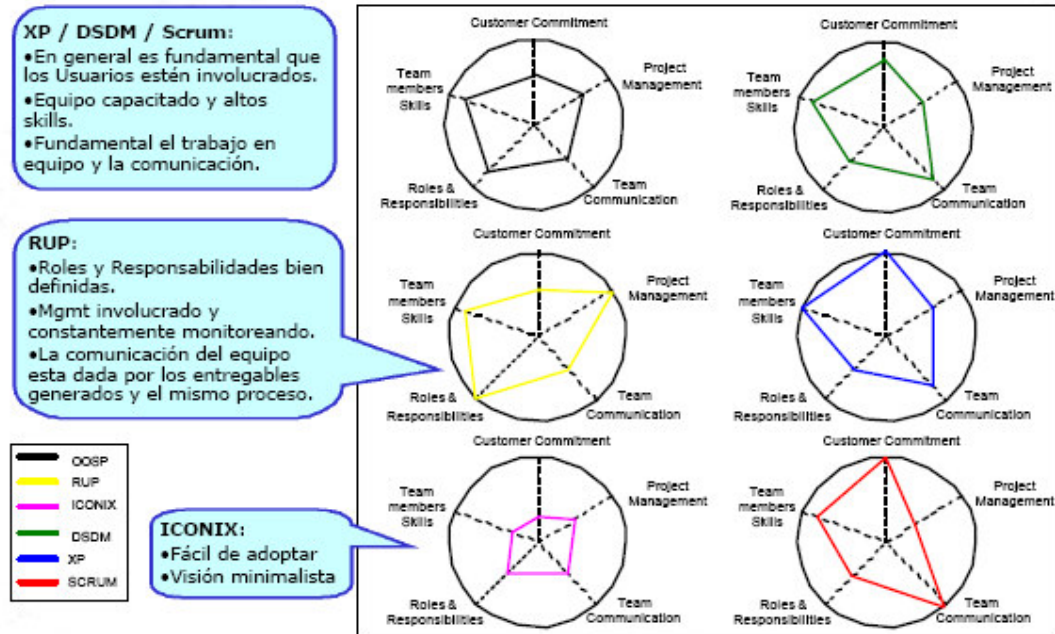


Gráfico 4-3. Características del Equipo de Trabajo Según la Metodología.<sup>4</sup>

### 4.2.4 Facilidad de Aprendizaje de Metodología.

- Curva de Aprendizaje: ¿Cuánto le toma a una organización incorporar el conocimiento necesario para aplicar correctamente el modelo?
- Herramientas e Integración de Herramientas: ¿Existen herramientas que den soporte a las actividades que hay que realizar?
- Soporte Externo: ¿Existe en el mercado soporte para adoptar el modelo?

<sup>3</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) “Fanatismo” o desarrollo de software

<sup>4</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) “Fanatismo” o desarrollo de software

Process Models	Learning curve	Tools and integration tools	External Support
RUP	Slow	Highly Supported	Highly Supported
ICONIX	Rapid	Some support available	Some support available
OOSP	Slow	Not mentioned	No support available
XP	Rapid	Not mentioned	Some support available
SCRUM	Rapid	Not mentioned	Some support available
DSDM	Moderate	Some support available	Highly Supported

Gráfico 4-4. Facilidad de Aprendizaje de metodología.<sup>5</sup>

#### 4.2.5 Ciclo de Vida de Actividades Según la Metodología.

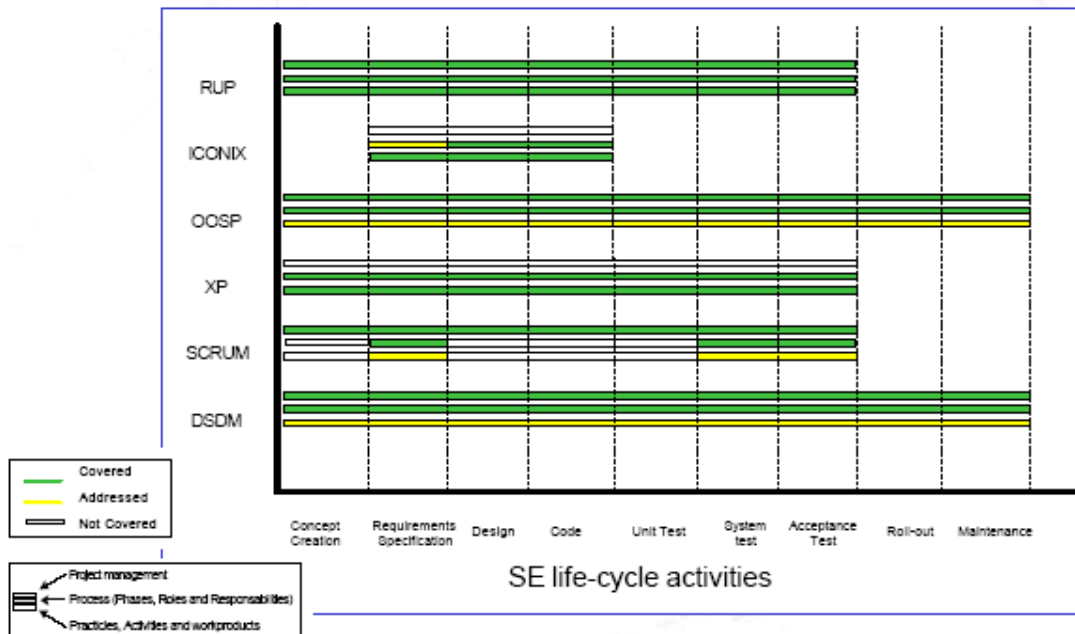


Gráfico 4-5. Ciclo de Vida de Actividades Según la Metodología.<sup>6</sup>

<sup>5</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) "Fanatismo" o desarrollo de software

<sup>6</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) "Fanatismo" o desarrollo de software

#### 4.2.6 Facilidad de Implementación.

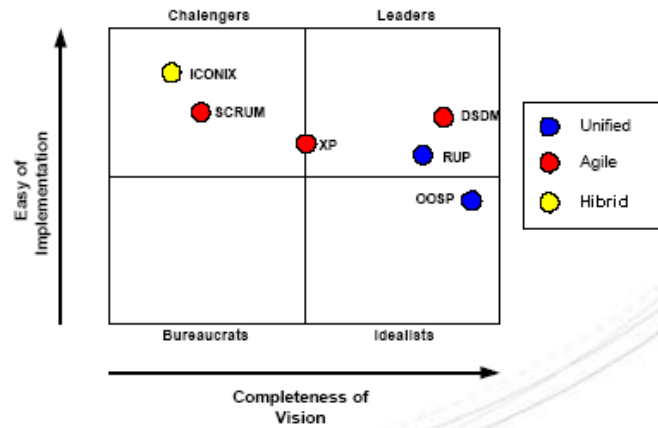


Gráfico 4-6. Facilidad de Implementación Según Metodología.<sup>7</sup>

#### 4.2.7 Consideraciones Finales. Ventajas y Desventajas.

	Ventajas	Consideraciones	Desventajas	Llaves para el (★) Éxito
RUP	Mucho soporte. Utilizado y difundido en la industria.	Casos de uso y arquitectura	Complejidad de customización / adaptación. No hay guías disponibles para esto.	Hay que ajustarlo a las necesidades de la organización.
ICONIX	Fácil de adoptar e implementar. Conjunto pequeño de prácticas.	No hay mucha información disponible.	No considera aspectos de Management.	Proyectos de negocio pequeños/medianos basado en objetos y/o componentes.
OOSP	Claro enfoque del proceso de desarrollo. Contempla todo el ciclo de vida.	Basado en CMM.	No hay mucho material disponible	Definir e instanciar los roles y procesos.
XP	Prácticas basadas en valores simples fáciles de transmitir. Buen ambiente de trabajo.	Se puede combinar con otros enfoques Agile.	Management sin mucho detalle. Problemas con proyectos grandes.	Guarda con implementaciones a medias.
SCRUM	Ideal para problemas complejos y de ambientes cambiantes.	Metodología liviana y fácil de entender.	Prácticas de desarrollo y testing no descritas.	Compromiso e involucramiento del usuario. Decisión de la organización para utilizarlo.
DSDM	Involucramiento y ownership del usuario. Soporte de un consorcio de empresas.	Identificar todos los tipos de usuarios.	Hay que pertenecer al consorcio.	Hay que aceptar la filosofía de trabajo. Estabilidad de las personas y los skills.

Gráfico 4-7. Ventajas y Desventajas Según Metodología.<sup>8</sup>

<sup>7</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) "Fanatismo" o desarrollo de software

Después de realizado el análisis de cada una de las posibles metodologías propuestas se determinó que ICONIX es la mejor opción para el desarrollo del presente proyecto, ya que sus características para desarrollo de proyectos medianos, utilización mínima de recursos humanos facilidad de adaptación, aprendizaje e implementación centrado en la especificación de requerimientos, diseño y código como también su visión minimalista. Permiten un total acoplamiento entre los desarrolladores del proyecto en mención con el alcance del mismo.

### **4.3 HERRAMIENTAS DE RECOLECCIÓN DE DATOS.**

Para la obtención de toda la información requerida se hará uso de las siguientes herramientas de recolección de datos:

Entrevistas, ya que da la posibilidad del contacto personal y aclarar cualquier duda que se presente en los datos obtenidos. Tanto en la fase de Especificación de Requerimientos, de recolección de datos y de seguimiento, las cuales cubren entre otras las siguientes necesidades: Llenar vacíos de información, testear nuevas ideas, testear conclusiones, testear propuestas.

La observación personal, mediante este método se determinará los requerimientos del sistema a implementar.

## **4.4 ICONIX**

### **4.4.1 Introducción.**

ICONIX es una metodología que maneja los casos de usos, con principios básicos tomados del RUP pero relativamente pequeño como XP, solo que este no descarta el análisis y el diseño. De aquí se derivan casos concretos, específicos, fácilmente comprensibles por el equipo de desarrollo.

---

<sup>8</sup> [www.pragmaconsultores.com](http://www.pragmaconsultores.com) “Fanatismo” o desarrollo de software

El ICONIX utiliza cuatro diagramas críticos, siguiendo un mismo enfoque y patrón en todos ellos. Este patrón será más o menos el siguiente:

ICONIX usa UML el cual genera un sistema mínimo de diagramas y algunas técnicas valiosas que llevaran de los casos del uso al código de forma rápida y eficiente. La idea de lo que el sistema debe hacer, es el inicio de algunos casos de uso con el objetivo de llegar al código fuente completo, probado y depurado y que haga lo que el caso de uso represente. Permittiéndonos de esta forma obtener de un rompecabezas, de algo confuso, lo que se quiere hacer por medio de descripciones no ambiguas, completas y rigurosas, llegando a obtener un código fuente bueno, de arquitectura sólida, diseño del software robusto.

La Figura siguiente muestra el cuadro del proceso. El diagrama retrata la esencia del enfoque aerodinámico al desarrollo del software, que incluye un juego mínimo de diagramas de UML y algunas valiosas técnicas que se toman de los casos del uso para codificar rápida y eficazmente. El enfoque es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos.

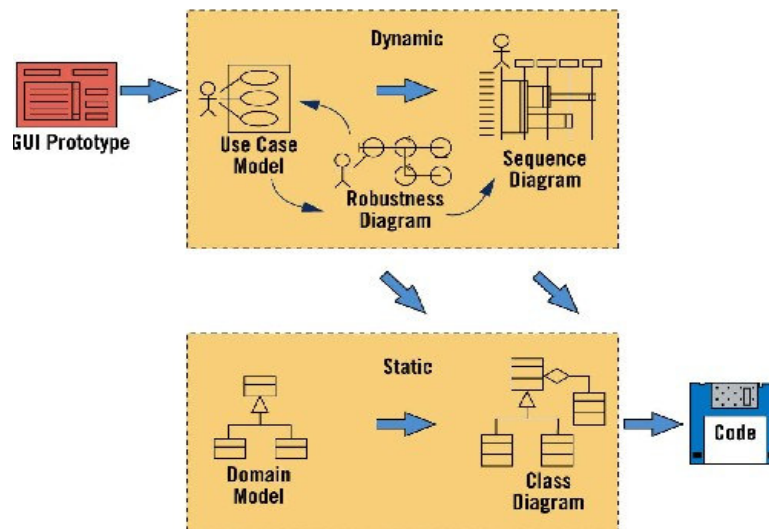


Gráfico 4-8. Cuadro para manejar Casos de Uso en el Modelamiento de Objetos<sup>9</sup>

<sup>9</sup> ROSENBERG, Doug, Applying Use Case Driven Object Modeling with UML, Copyright Addison-Wesley, 2001.

Primero, es iterativo e incremental. Las iteraciones múltiples ocurren entre el desarrollo del modelo del dominio e identificar y analizar los casos de uso. Otras iteraciones existen también, como los procesos del equipo a través del ciclo de vida. El modelo estático se refina incrementalmente durante las iteraciones sucesivas a través del modelo dinámico (compuesto de casos de uso, análisis de robustez y el diagrama de secuencia). Note sin embargo, que el acercamiento no requiere hitos formales y la teneduría de muchos libros; más bien, los esfuerzos de refinamiento producen los hitos naturales como el equipo del proyecto que gana conocimiento y experiencia.

Segundo, el enfoque ofrece un alto grado de seguimiento. Por el camino, a cada paso usted consultara de alguna manera los requisitos anteriores. Nunca hay un punto en que el proceso le permita desviarse lejos de las necesidades del usuario. Seguimiento se refiere también al hecho que usted puede seguir los objetos paso a paso como el análisis dentro del diseño.

Tercero, el enfoque ofrece uso aerodinámico del UML. Los pasos que nosotros describiremos en los siguientes temas representan un mínimo del acercamiento, ellos comprenden el juego mínimo de pasos que nosotros hemos encontrado para ser necesarios y suficiente en el desarrollo de un proyecto Orientado a Objetos exitoso. Enfocando en un subconjunto del grande y pesado UML, un equipo del proyecto también puede dirigirse fuera de "la parálisis del análisis".

#### **4.4.2 Las Capacidades de ICONIX.**

La solución de ICONIX incluye un ancho rango de ofrecimientos de servicios de negocios. Las soluciones de negocios de extremo a extremo se concentran en los servicios en tres áreas primarias, con la estrategia y planeación recubriendo cada área. La especialización equilibrada en las tres áreas (la experiencia del usuario, funcionalidad comercial, e infraestructura) contribuye al éxito de las soluciones que se entrega a los clientes.

En un sistema orientado a objetos, la estructura del código es definido por clases, antes de escribir el código debemos saber cuáles son las clases que lo conforman, para hacer esto nosotros necesitamos uno o más diagramas de clases que muestran las clases que

están dentro del sistema. Cada una de las clases debe tener un conjunto de atributos que son las características de los miembros que están contenidos en estas clases, las operaciones que pueden realizar. En otras palabras nosotros debemos tener todas las funciones de software identificadas. Además como están organizadas y como se relacionan unas con otras en un diagramas de clases. Una de las cosas más difíciles de hacer en el desarrollo de sistemas orientado a objetos es la localización de la conducta, lo que involucra tomar decisión de cada función del software que se va a construir. Para cada función se tiene que decidir qué clase en el diseño del software deberá contener esta función. Se necesita localizar todas las conductas del sistema, cada función del software necesita ser localizada dentro de un conjunto de clases que se están diseñando.

Los pasos básicos para comprender por completo el proceso de ICONIX es asociando los siguientes puntos:

- Análisis de Requerimientos.
  - Identificar el dominio de objetos del mundo real y la generalización y relaciones de agregación entre los objetos. Se inicia dibujando un diagrama de clases de alto nivel.
  - Si es viable hacer prototipos de lo que propone nuestro sistema. O reunir toda clase de información que se tiene de un sistema anterior al cual se lo va a mejorar (reingeniería).
  - Identificar nuestros casos de uso, usando los diagramas de casos de uso.
  - Organizar dentro de grupos los diagramas de casos de uso. Capturando esta organización dentro de diagramas de paquetes.
  - Localizar los requerimientos funcionales de los casos de uso y el dominio de los objetos en esta etapa.
- Análisis y Diseño Preliminar.
  - Escribir la descripción de los casos de uso; cursos básicos de las acciones que representan el flujo normal de eventos y los cursos alternativos que son utilizados con menor frecuencia y condiciones de error.
  - Desempeño en el análisis de robustez. Por cada caso de uso:

- Identificar el primer grupo de objetos que complementan el estado del escenario.
- Actualizar el modelo de dominio con nuevos objetos y atributos según como se vayan descubriendo.
- Finalmente se actualiza el diagrama de clases completando el análisis del proyecto.
- Diseño.
  - Identificar los mensajes que deben ser enviados entre objetos, además los objetos y asociaciones que deben ser invocados. Dibujar una secuencia de diagramas con casos textuales corriendo hacia abajo desde la izquierda hacia la derecha. Se continúa actualizando el diagrama de clases con atributos y operaciones encontradas en ellas.
  - Si se desea se muestra en el diagrama de colaboración las principales procesos entre objetos.
  - Si se desea se utilizan los diagramas de estado para mostrar en tiempo real la conducta de los objetos.
  - Finaliza el modelo estático pero agregando detalladamente información al diseño.
  - Verificar con el equipo si el diseño satisface todos los requerimientos que han sido identificados.
- Implementación.
  - Según se necesite se genera diagramas tales como diagramas de despliegue y de componentes, estos ayudan en la etapa de implementación.
  - Escribir y/o generar el código.
  - Pruebas de Unificación e integrar de desempeño.
  - Pruebas del desempeño del sistema y aceptación del usuario.

#### **4.4.3 El Modelo del Dominio.**



El modelo del dominio es una parte esencial del proceso de ICONIX. Construye la porción estática inicial de un modelo que es esencial al manejar su plan de la aplicación, antes de los casos del uso.

El enfoque de este tema es el modelo del dominio. El término "dominio del problema" se refiere al área que abarca cosas del mundo real y conceptos relacionados al problema que el sistema está diseñándose para resolver. El modelo del dominio es la tarea de descubrir " los objetos " (las clases) estos representan cosas y conceptos.

Dentro del proceso de ICONIX, el modelo de dominio activado involucra, fuera de los requisitos de los datos, construir un modelo estático del dominio del problema pertinente al sistema propuesto.

- Los Elementos importantes del modelo del Dominio

La primera cosa que se debe hacer cuando se construye un modelo estático de un sistema es el hallazgo de clases apropiadas que con precisión representan las abstracciones reales de los problemas que se presentan en el modelo del dominio. Si se ejecuta bien esta actividad, no sólo se tendrá una construcción sólida para construir el sistema, sino también las excelentes perspectivas para reutilización de sistemas que se diseñará y se construirá con el tiempo.

“Es probable que los mejores recursos de clases sean la declaración del problema de alto nivel, los niveles bajos de requisitos y conocimientos del experto sobre el espacio del problema. Para empezar, se debe poner todas las declaraciones pertinentes de estas áreas (e incluso otros) como se pueda encontrar, y luego se señala o resalta, todos los sustantivos de la frase. Se debe refinar las listas gradualmente, los sustantivos de las frases se volverán objetos y atributos, mientras los verbos se volverán funcionamientos y asociaciones. Los posesivos (" su," nuestro " y " suyo ") tienden a indicar que los sustantivos deben ser los atributos, en lugar de los objetos.

Luego, se debe seleccionar de la lista de clases candidatas y se debe eliminar los artículos innecesarios. Se debe buscar las clases que son redundantes, no pertinentes, incorrectas o vagas. Las clases no esenciales también pueden representar los conceptos

fuera del alcance del modelo, o representa las acciones aunque ellos se expresan como los nombres.

También se debe tomar algunas decisiones iniciales sobre la generalización (el " tipo de " o " es un " relación entre las clases) mientras se construye el diagrama de clases. Si se necesita, y es más cómodo para esta fase, se debe generalizar a más de un nivel de subclase. Se debe buscar tipos de declaraciones que son verdad en el mundo real. El modelamiento del dominio también es el área apropiada para las decisiones sobre las agregaciones.

Finalmente, tal como muchos diagramas de entidad - relación (ERD), su modelo del dominio, pone al día para mostrar las asociaciones (las relaciones estáticas entre los pares de clases) debe ser una verdadera declaración sobre el espacio del problema, independiente del tiempo (es decir, estática). Este modelo sirve como la construcción de su modelo de la clase estático.”<sup>10</sup>

#### **4.4.4 Modelo de Casos de Uso.**

Dentro del proceso de ICONIX, uno de los primeros pasos involucra la construcción del modelo de casos de uso. Este modelo se usa para capturar los requisitos del usuario de un nuevo sistema (si está desarrollándose desde el principio o basado en un sistema existente) detallando todos los guiones que los usuarios realizarán. Los casos del uso manejan al modelo dinámico y, por la extensión, el esfuerzo del desarrollo entero.

- Los Elementos Importantes.

La tarea de construir casos de uso para su nuevo sistema está basada en identificar inmediatamente tantos casos como se puede, y estableciendo una vuelta continua de escribir y refinar el texto que los describe entonces. En el camino, se descubrirá nuevos casos del uso, y también se factorizará los casos de uso que sean convenientes.

---

<sup>10</sup> ROSENBERG, Doug, Applying Use Case Driven Object Modeling with UML, Copyright Addison-Wesley, 2001. Pag. 3

Al identificar los casos del uso deben tener las correlaciones fuertes con material encontrado en el manual del usuario del sistema. La conexión entre cada caso del uso y una sección distinta de su guía del usuario debe ser obvia. Refuerza la noción fundamental que se está diseñando un sistema que conformará los puntos de vista de los usuarios. También proporciona un resumen conveniente de los medios del manejo de los caso de uso.

Una vez que tenga algún documento para un caso del uso, es tiempo de refinarlo asegurándose las frases estén claras y discreto, el formato básico de su texto es sustantivo-verbo- sustantivo, y los actores y los objetos del dominio potenciales son fáciles de identificar. También debe poner al día a su modelo del dominio como vaya descubriendo los nuevos objetos y extender la comprensión de los objetos que creo previamente. Y, es importante determinar todo los posibles cursos alternados de acción donde se requiera para cada caso de uso posible, una actividad que debe asumir la mayoría del tiempo.

Se puede usar varios mecanismos para factorizará fuera del uso común, tal como el manejo de errores, fijados en los casos de uso. Esto es normalmente eficaz, porque eliminándose el uso de los pequeños niveles aliviará el esfuerzo del análisis y no requiere de mucho tiempo al dibujar los diagramas de secuencia. Si usted usa la generalización de UML y las relaciones include y extends, o relaciones OML invokes y precedes, su meta debe ser fijar casos de uso pequeños, precisos, reusables.

Usted debe sentir el procedimiento ideal a las próximas fases del desarrollo que va a procesar cuando usted haya logrado las metas siguientes:

- Haber construido casos del uso que juntos respondan de toda la funcionalidad deseada del sistema.
- Haber producido las descripciones escritas claras y concisas del curso básico de la acción, junto con los cursos alternativos apropiados de la acción, para cada caso de uso.
- Haber factorizado fuera de los guiones en común a más de un caso de uso, mientras estructura lo que le sea más cómodo.

#### 4.4.5 El Análisis de Robustez.

Esta técnica es simple y útil se une el análisis al diseño asegurando que su texto de caso de uso es correcto. Se dirige caminos necesarios de acción y le permite continuar descubriendo los objetos.

Este tema enfoca el análisis de robustez que involucra análisis del texto de descripción de los casos del uso e identificando un conjunto de primeras suposiciones de los objetos que participarán en cada caso de uso, clasificando estos objetos en tres tipos:

- El objeto Límite que los actores usan para comunicarse con el sistema.
- El objeto Entidad que normalmente son los objetos del modelo del dominio.
- El objeto Control (qué nosotros normalmente llamamos controladores porque ellos no son a menudo los objetos reales), qué sirve como la " unión " entre el objeto Limite y el objeto entidad.



Gráfico 5-2. Los Iconos Visuales de los Tres Estereotipos para estos tipos de objetos<sup>11</sup>

Los Actores usan el objeto Límite para comunicarse con el sistema. Normalmente se derivan los objetos de Entidad de los modelos del dominio, y objetos de Control sirven como la enlace entre los objetos Límite y Entidad.

Esta técnica es simple pero muy útil porque sirve como un eslabón crucial entre el Análisis Modelo.

Usted refinará su texto del caso de uso y su modelo será sólido como resultado del análisis de robustez.

<sup>11</sup> <http://www.spinec.org/wp-content/ICONIX.pdf>. Pag. 10

Dentro del proceso de ICONIX, esta técnica simple pero muy útil sirve como un eslabón crucial entre el Modelo y el análisis, dónde el análisis de robustez reside dentro del proceso de ICONIX.

- Los Elementos Importantes.

El análisis de robustez juega varios papeles esenciales dentro del proceso de ICONIX. Se refinará su texto de caso de uso y su modelo estático diseñado como resultado del análisis de robustez.

El análisis de robustez proporciona un “control de sanidad” ayudándole a asegurar que su texto de caso de uso es correcto y que usted no ha especificado una conducta imposible para el sistema o el conjunto de objetos que se tiene no es razonable. Este refinamiento del texto de caso de uso cambia la naturaleza del texto de la perspectiva manual un de usuario a una descripción del uso en el contexto del modelamiento de objetos.

También proporciona una integridad y control de exactitud ayudándole a determinar si el caso uso toma la dirección de todos los caminos alternativos necesarios. El tiempo que se emplea en los dibujo de diagramas de robustez hasta aquí, y también hacia la producción del texto que adhiere a algunas pautas bien definidas, el tiempo que se ahorra es significativo para dibujar los diagramas secuencia.

El análisis de robustez habilita el descubrimiento continuo de objetos; un paso crucial porque ciertamente se obvió de algunos objetos durante el modelamiento del dominio. Usted también puede determinar diferencias de denominación de objetos y conflictos antes de que ellos causen serios problemas. Y, el análisis de robustez le ayuda a asegurar que usted ha identificado la mayoría de las clases del dominio antes de empezar los diagramas de secuencia.

Finalmente, el análisis de robustez llena el papel del Modelo preliminar, cerrando la brecha entre el análisis y el modelo detallado.

Los Objetos Limite son los objetos con que los actores (por ejemplo, los usuarios) estarán actuando recíprocamente en el nuevo sistema. Éstos frecuentemente incluyen ventanas, pantallas, diálogos y menús.

Los Objetos Entidad trazan a menudo las tablas de la base de datos y archivos que contienen la información que necesita sobrevivir a " la ejecución de caso de uso". Algunos de sus objetos entidad son " objetos transeúntes ", como los resultados de búsqueda.

Los objetos Control (controles) incluyen la lógica de la aplicación y sirve como el tejido que une entre los usuarios y los datos guardados. Esto es donde usted frecuentemente captura reglas de negocio cambiantes y políticas, y localiza los cambios a estos objetos sin romper su interfaz de usuario o su esquema de la base de datos. De vez en cuando (quizás 20 por ciento del tiempo), los controladores son " los objetos " reales en un modelo, pero los controladores normalmente sirven como el guías para asegurar que no se olvide de cualquier funcionalidad y la conducta del sistema requerido por sus casos de uso.

Usted realiza el análisis de robustez para un caso de uso utilizando el texto del caso de uso, una frase a la vez, y dibujando a los actores, el límite apropiado, el objeto entidad y el controlador, y las conexiones entre los varios elementos del diagrama. Usted debe poder encajar el camino básico y todos los caminos alternados en un diagrama. Cuatro reglas básicas se aplican:

- Los Actores sólo pueden interactuar con los objetos límite.
- Los objetos límite sólo pueden interactuar con controladores y actores.
- Los objetos entidad sólo pueden interactuar con controladores.
- Los controladores pueden interactuar con objetos limite y objetos entidad, y con otros controladores, pero no con actores

Los Objetos Límite y objetos Entidad son los sustantivos, y los controladores son los verbos. Los sustantivos no pueden interactuar con otros sustantivos, pero los verbos pueden interactuar con sustantivos o verbos.

Cualquiera que repase un diagrama de robustez debe poder leer un camino de acción en el texto del caso de uso, debe seguir a lo largo de las asociaciones en el diagrama, y debe ver un luz clara entre el texto y el cuadro. Se tendrá que volver a escribir el texto del caso de uso, para quitar la ambigüedad y poner explícitamente la referencia a los objetos límite y a los objetos entidad. La mayoría de las personas no escribe el texto de caso de uso perfecto en el primer proyecto.

Además de usar los resultados de análisis de robustez para ceñirse al texto de caso de uso, se debe refinar también continuamente al modelo estático. Los nuevos objetos que se descubre en el dibujo que los diagramas deben llevarse al diagrama de clase, y éste también es el tiempo correcto para agregar algunos atributos clave a sus clases más significantes.

#### 4.4.6 Diagrama de Secuencia.

La interacción diseñada le permite detallar la conducta de sus objetos y encuentre las clases apropiadas para los atributos y funcionamientos.

Este tema perfila los errores más comunes, y entonces explica cómo corregirlos. Este enfoque estará orientado en realizar la interacción del diseño usando UML y diagramas de secuencias.

Cuando usted termina con planeamiento de dominio y análisis de robustez, usted habrá encontrado la mayoría de los objetos en el problema y asignara algunos atributos a ellos. Se habrá definido las relaciones estáticas entre los objetos en su diagrama de la clase de alto nivel y unas relaciones dinámicas en sus diagramas de robustez.

Ahora es tiempo para diseñar como su software realmente trabajará (en otros términos, para definir la solución a su problema). El diseño de Interacción es la fase dónde usted construye a los hilos que unen sus objetos. Aquí, usted empezará a ver cómo su nuevo sistema realizará una conducta útil.

- Los Elementos Importantes.

Se quiere lograr tres metas primarias durante el diseño de interacción.

Primero, se debe asignar el comportamiento entre los objetos límite, entidad y de control. Durante el análisis de robustez, usted puede identificar un conjunto de objetos que pueden lograr la conducta deseada de sus casos del uso. Entonces se puede decidir qué objetos son responsables para cierta parte del comportamiento. Si no se tiene una clara idea de los objetos límite, entidad y control, es demasiado pronto para estar contemplando cómo se asignará el comportamiento. En ese caso, se necesitará regresar al análisis de robustez y realizarlo bien.

Segundo, se debe mostrar las interacciones detalladas que ocurren entre los objetos asociados con cada uno de los casos del uso. Los objetos actúan recíprocamente



enviando los mensajes a nosotros. Estos mensajes sirven como estímulos (es decir, un mensaje estimula a un objeto para realizar algunas acciones deseadas. Para cada unidad de comportamiento dentro de un caso de uso, se debe identificar los mensajes y métodos necesarios.

Tercero, se debe terminar la distribución de funcionamiento entre clases. Se debe apuntar para tener un 75 a 80 por ciento aproximadamente de sus atributos definidos dentro del modelo estático, cuando se haya terminado el análisis de robustez. Sin embargo, no empiece definiendo los funcionamientos durante el modelo del dominio y análisis de robustez. De hecho, se recomienda que no se asigne ningún método en este punto, porque no hay bastante información disponible.

Una vez que se ha conseguido el modelo de interacción, se debe tener bastante información. Entonces se puede poner el comportamiento detallado de sus objetos (en los diagramas de secuencia, en el contexto de su caso de uso) y se puede finalizar encontrando los lugares apropiadas para los atributos y funcionamientos. Mientras se hace este modelo dinámico, se estará actualizando y se extenderá su modelo estático, y esto solidificará su creciente conocimiento de cómo su nuevo sistema debe trabajar.

“El diagrama de secuencia del UML evolucionó de una combinación del diagrama de interacción de objetos de Jacobson y del diagrama de control de eventos del OMT. Dentro del enfoque de ICONIX, los diagramas de secuencia representan el producto de trabajo de un mayor modelo. Se dibuja un diagrama de secuencia que abarque el camino básico y todos los caminos alternativos dentro de cada uno de casos de uso. Los resultados forman el centro de su modelo dinámico (que es la conducta del tiempo de ejecución del sistema, incluyendo cómo se logrará esa conducta) que se define en gran detalle.”<sup>12</sup>

Hay cuatro tipos de elementos en un diagrama de secuencia: el texto para el camino de acción de los casos de uso, objetos, mensajes y métodos (funcionamientos).

---

<sup>12</sup> ROSENBERG, Doug, Applying Use Case Driven Object Modeling with UML, Copyright Addison-Wesley, 2001.

- El texto para el camino de acción de los casos de uso aparecen abajo en el lado izquierdo. Es una buena idea separar el texto con el espacio en blanco para que sea fácil ver qué frase(s) corresponde con cada uno de los elementos a la derecha.
- Los Objetos que usted trae directamente de sus diagramas de robustez, se representa con dos componentes: el nombre del objeto y la clase a que ese objeto pertenece. Éstos aparecen en una caja arriba de la página, de la forma `object::class`. Una línea punteada corre de esa caja hacia abajo en toda la longitud de la página. Usted puede mostrar los iconos de diagrama de robustez sobre las cajas del objeto.
- Los mensajes son las flechas entre los objetos. Una flecha del mensaje puede ir directamente entre dos líneas punteadas, entre una línea y un rectángulo del método, o entre dos rectángulos del método.
- Los métodos (funcionamientos) se muestran como rectángulos que quedan encima de las líneas punteadas que pertenecen a los objetos a que ellos se asignan. Usted puede usar las longitudes de estos rectángulos para reflejar el enfoque de mando dentro de la secuencia. Un método en particular parte del extremo del rectángulo.

Los cuatro pasos que se ejecutan cuando se realiza el dibujo del diagrama de secuencia a la manera de ICONIX son como sigue:

Paso 1. Copiar el texto del caso de uso obtenido para especificarlo. Péguelo en el margen izquierdo de la página. Esto se hace para permitir a ese texto servir como un recordatorio continuo de lo que usted necesita lograr. El resultado es que cuando usted está haciendo el diseño, el comportamiento del sistema requerido siempre está mirándolo fijamente a la cara. Pero si no se tiene todos los caminos de acción alternativos pertinentes escritos para cada uno de los casos de uso, no se debe proceder hasta que ellos estén en su lugar. Por otra parte, los diagramas no cubrirán todos los casos especiales, y usted no encontrará el comportamiento total del caso de uso. Esto significa que no se descubrirá todos los métodos necesarios para sus objetos.

Paso 2. Agregar los objetos entidad del diagrama de robustez. Cada uno de estos objetos es un tipo caso que aparece en el Diagrama de clase que representa el modelo estático. (Si se olvidó de actualizar el diagrama de clase estático en respuesta a los nuevos objetos que descubrió durante el análisis de robustez, hágalo ahora). Estos objetos deben tener la mayoría de sus atributos en sitio. Muchos de ellos estarán sirviendo de datos a otros objetos. Se puede esperar descubrir los atributos perdidos para trabajar el diagrama de secuencia. Sea meticuloso sobre agregarlos al modelo estático; es probable que esto sea su último paso antes del código.

Paso 3. Agregar los objetos límite del diagrama de robustez. En este punto se preguntara por qué no mencionamos la adición de los objetos límite al modelo del dominio. La razón es que estos objetos son parte de la solución. Respondiendo de los objetos límite de los diagramas de secuencia, usted empieza a integrar el modelo detallado.

Si se sigue el enfoque de ICONIX, los primeros tres pasos involucrados dibujan los diagramas de secuencia, su naturaleza es completamente mecánica.

Paso 4. Ponga los métodos en las clases. Esto involucra convertir los objetos control del diagrama de robustez en un conjunto de métodos y mensajes que incluyen el comportamiento deseado (De vez en cuando, usted puede dejar un control como un objeto control real). Use su diagrama de robustez como una lista de control, asegúrese que se tiene todo el comportamiento que el sistema requiere para los diagramas de secuencia. Entonces simplemente controle las respuestas de cada objeto control que se dibuja en los diagramas de secuencia.

“Hay dos estrategias básicas por convertir los controladores que aparecen en los diagramas de robustez: control en la pantalla y controlador de caso de uso. Si usted usara sólo una estrategia durante sus esfuerzos en la diagramación de secuencia, sería el patrón a seguir. La idea es que se debe establecer a los miembros del equipo, responsable para los diagramas, las normas del diseño que pueden usarse para todos los casos de uso.

Por otro lado, las diagramaciones son las interacciones entre varios objetos, puede decidir uno o más modelos del diseño, mejor establecidos, que encajaran. O quizás se podría desarrollar los nuevos modelos para establecer un enfoque regularizado para

diseñar resultados a los problemas que aparecen en los múltiples casos de uso. Esto es donde el desarrollo orientado a objetos tiene lugar.

A estas alturas, ya se ha verificado los diagramas de robustez con el texto del caso de uso. Verificando sus diagramas de secuencia con sus diagramas de robustez, se agrega una medida de convicción que se está diseñando en la contestación a lo que el usuario necesita (en otros términos, reuniendo requisitos).”<sup>13</sup>

---

<sup>13</sup> ROSENBERG, Doug, Applying Use Case Driven Object Modeling with UML, Copyright Addison-Wesley, 2001.

## **5 FUNDAMENTACIÓN TEÓRICA.**

### **5.1 Capítulo 1. JAVA.**

#### **5.1.1 Características de Java**

Es simple: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- Aritmética de punteros
- No existen referencias
- Registros (struct)
- Definición de tipos (typedef)
- Macros (#define)
- Necesidad de liberar memoria (free)

Es orientado a objetos: Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

Es distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite al programador acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Es Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

Java proporciona:

- Comprobación de punteros
- Comprobación de límites de arrays
- Excepciones
- Verificación de byte-codes

Arquitectura neutral: Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, Mac, Apple y probablemente haya grupos de desarrollo trabajando en el porting a otras plataformas.

Seguro: El código no produce desbordamiento de operandos en la pila. El tipo de los parámetros de todos los códigos de operación son conocidos y correctos. No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros.

El acceso a los campos de un objeto se sabe que es legal: public, private, protected No hay ningún intento de violar las reglas de acceso y seguridad establecidas

Es Portable: Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac

Interpretado:

Multithreaded: Al ser multithreaded (multihilvanado, en mala traducción), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads contruidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.

El beneficio de ser multithreaded consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

Dinámico: Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales (siempre que mantengan el API anterior).

## **5.2 Capítulo 2. Open Source.**

### **5.2.1 Hibérnate**

“Hibernate es una herramienta de mapeo objeto/relacional para entornos Java. El término mapeo objeto/relacional (MOR) hace referencia a la técnica de mapear una representación de datos desde un modelo de objetos a un modelo de datos relacional con un esquema basado en SQL.

Hibérnate no sólo se encarga de mapear clases Java a tablas de base de datos (y de tipos de datos de Java a tipos de datos SQL), sino que también provee facilidades de consulta y recuperación de datos y puede reducir significativamente el tiempo de desarrollo que de otra forma se gasta en el manejo de los datos en SQL y JDBC.

La meta de Hibérnate es relevar al desarrollador del 95 por ciento de las tareas comunes relacionadas a la programación de la persistencia de los datos. Hibérnate puede no ser la mejor solución para aplicaciones que usan solamente procedimientos almacenados para implementar la lógica de negocio en la base de datos, es más útil con modelos de dominio orientados a objetos y lógica de negocio basada en Java. Sin embargo, Hibérnate ciertamente puede ayudarte a quitar o encapsular código SQL específico y ayudará con la tarea común de traducción de resultados desde una representación tabular a un grafo de objetos.”<sup>14</sup>

#### 5.2.1.1 Visión General.

Una visión a (muy) alto nivel de la arquitectura de Hibérnate:

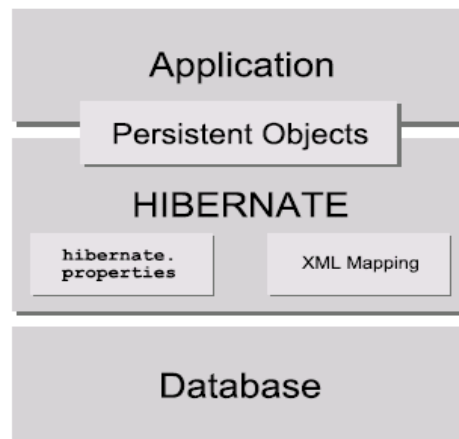


Gráfico 5-1. Arquitectura de Hibérnate<sup>15</sup>

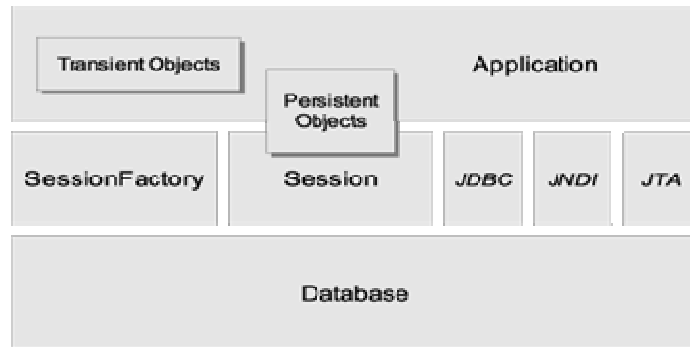
Este diagrama muestra a Hibérnate usando la base de datos y los datos de configuración para proveer servicios de persistencia (y objetos persistentes) a la aplicación.

<sup>14</sup> <http://www.programacion.net/java/tutorial/hibernate>

<sup>15</sup> HIBERNATE Documentación de referencia de Hibernate Versión 3.0.5

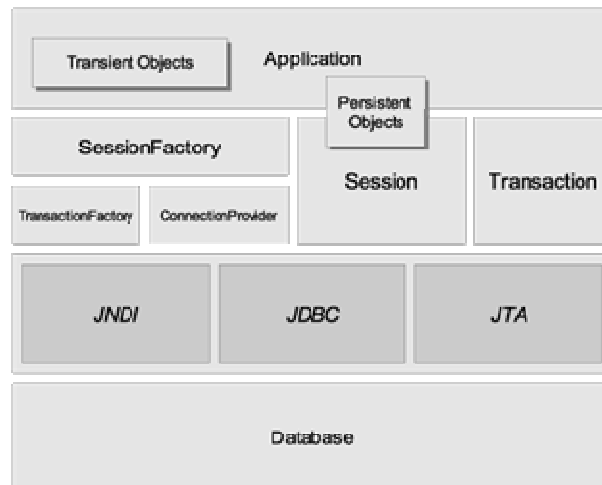


En la arquitectura "sencilla", es la aplicación la que provee sus propias conexiones JDBC y gestiona sus propias transacciones. Este enfoque usa un mínimo subconjunto de la API de Hibérnate:



**Gráfico 5-2 Arquitectura Hibérnate Detallada<sup>16</sup>**

La arquitectura "full cream" abstrae a la aplicación de las APIs de JDBC/JTA y deja que Hibérnate se encargue de los detalles.



**Gráfico 5-3 Arquitectura Detallada Compleja<sup>17</sup>**

Algunas definiciones de los objetos en los diagramas:

SessionFactory (org.hibernate.SessionFactory)

Caché threadsafe (inmutable) de mapeos compilados para una sola base de datos. Es una fábrica de Sesión y un cliente de ConnectionProvider. Opcionalmente, puede

<sup>16</sup> HIBERNATE Documentación de referencia de Hibernate Versión 3.0.5

<sup>17</sup> HIBERNATE Documentación de referencia de Hibernate Versión 3.0.5

mantener una caché (de segundo nivel) de datos reusables entre transacciones, a un nivel de proceso o de clúster.

Session (org.hibernate.Session)

Objeto mono-hebra, de corta vida que representa una conversación entre la aplicación y el almacenamiento persistente. Envuelve una conexión JDBC. Es una fábrica de Transaction. Mantiene una caché requerida (de primer nivel) de objetos persistentes, usada mientras se navega el grafo de objetos o se recuperen objetos por identificador.

Objetos y colecciones persistentes

Objetos de corta vida, mono-hebra conteniendo estado persistente y funcionalidad de negocio. Estos pueden ser JavaBeans/POJOs (Plain Old Java Objects, o sea, cualquier objeto Java), la única cosa especial en ellos es que están asociados actualmente con una (y sólo una) Session. Tan pronto como la Session sea cerrada, serán separados y estarán libres para ser usados en cualquier capa de aplicación. (por ejemplo, directamente como objetos de transferencia de datos hacia y desde la capa de presentación).

Objetos y colecciones transitorios y separados, Instancias de clases persistentes que no están actualmente asociadas con una Session. Pueden haber sido instanciadas por la aplicación y (aún) no haber sido hecho persistente, o pueden haber sido instanciadas por una Session cerrada.

Transaction (org.hibernate.Transaction), (Opcional) Un objeto de corta vida, mono-hebra, usado por la aplicación para especificar unidades atómicas de trabajo. Abstrae a la aplicación de las subyacentes transacciones JDBC, JTA o CORBA. En algunos casos, una Session puede extenderse sobre varias Transactions. Sin embargo, la demarcación de la transacción, ya sea usando la API subyacente o Transaction, nunca es opcional.

ConnectionProvider (org.hibernate.connection.ConnectionProvider), (Opcional) una fábrica (y pool) de conexiones JDBC. Abstrae a la aplicación del DataSource o DriverManager subyacente. No se expone a la aplicación, pero puede ser extendido/implementado por el desarrollador.

TransactionFactory (org.hibernate.TransactionFactory), (Opcional) una fábrica de instancias de Transaction. No se expone a la aplicación, pero puede ser extendido/implementado por el desarrollador.

#### Interfaces de Extensión

Hibernate ofrece muchas interfaces de extensión opcional que puedes implementar para modificar a medida el comportamiento de tu capa de persistencia. Para más detalles, mira la documentación de la API.

Dada una arquitectura "sencilla", la aplicación pasa por alto las APIs de Transaction/TransactionFactory y/o ConnectionProvider, para hablar directamente a JTA o JDBC.

#### **5.2.1.2 Estados de instancia**

Una instancia de una clase persistente puede estar en uno de tres estados diferentes, definidos respecto de su contexto de persistencia. El objeto Session de Hibernate es el contexto de persistencia:

Transitorio, la instancia no está y nunca estuvo asociada con un contexto de persistencia. No tiene identidad persistente (valor de clave primaria).

Persistente, la instancia está actualmente asociada con un contexto de persistencia. Tiene una identidad persistente (valor de clave primaria) y, quizás, una fila correspondiente en la base de datos. Para un contexto de persistencia en particular, Hibernate garantiza que la identidad persistente es equivalente a la identidad Java (localización en memoria del objeto).

Separado, la instancia estuvo una vez asociada con un contexto de persistencia, pero ese contexto fue cerrado, o la instancia fue serializada a otro proceso. Tiene una identidad

persistente y, quizás, una fila correspondiente en la base de datos. Para las instancias separadas, Hibernate no establece ninguna garantía sobre la relación entre identidad persistente e identidad Java.

### 5.2.1.3 Ejemplo de Hibernate.

- **Configuración básica para Conexión con la Base de Datos.**

```

1 <?xml version='1.0' encoding='utf-8'?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://hibernate.sourceforge.net/hibernate-configuration-
3.0.dtd">
5 <hibernate-configuration>
6     <session-factory>
7         <!-- Database connection settings -->
8         <property
name="connection.driver_class">com.mysql.jdbc.Driver</property>
9         <property
name="connection.url">jdbc:mysql://localhost/sga</property>
10        <property name="connection.username"></property>
11        <property name="connection.password"></property>
12        <!-- JDBC connection pool (use the built-in) -->
13        <property name="connection.pool_size">1</property>
14        <!-- SQL dialect -->
15        <property
name="dialect">org.hibernate.dialect.MySQLMyISAMDialect</property>
16        <!-- Echo all executed SQL to stdout -->
17        <property name="show_sql">>true</property>
18        <!-- Drop and re-create the database schema on startup -->
19        <property name="hbm2ddl.auto">update</property>
20
21
22
23        <mapping resource="com/sga/negocio/mapeo/Acceso.hbm.xml"/>
24        <mapping
resource="com/sga/negocio/mapeo/Acreditacion.hbm.xml"/>
25        <mapping
resource="com/sga/negocio/mapeo/Asistencia.hbm.xml"/>
26        <mapping
resource="com/sga/negocio/mapeo/Configuracion.hbm.xml"/>
27        <mapping resource="com/sga/negocio/mapeo/Curso.hbm.xml"/>
28        <mapping resource="com/sga/negocio/mapeo/Dominio.hbm.xml"/>
29        <mapping
resource="com/sga/negocio/mapeo/Especialidad.hbm.xml"/>
30        <mapping resource="com/sga/negocio/mapeo/Funcion.hbm.xml"/>
31        <mapping
resource="com/sga/negocio/mapeo/Inscripcion.hbm.xml"/>
32        <mapping resource="com/sga/negocio/mapeo/Materia.hbm.xml"/>
33        <mapping
resource="com/sga/negocio/mapeo/Matricula.hbm.xml"/>
34        <mapping resource="com/sga/negocio/mapeo/Nivel.hbm.xml"/>
35        <mapping
resource="com/sga/negocio/mapeo/Paralelo.hbm.xml"/>

```

```

36     <mapping
resource="com/sga/negocio/mapeo/Parametriza.hbm.xml"/>
37     <mapping resource="com/sga/negocio/mapeo/Pensum.hbm.xml"/>
38     <mapping resource="com/sga/negocio/mapeo/Periodo.hbm.xml"/>
39     <mapping resource="com/sga/negocio/mapeo/Persona.hbm.xml"/>
40     <mapping resource="com/sga/negocio/mapeo/Reporte.hbm.xml"/>
41     <mapping resource="com/sga/negocio/mapeo/Seccion.hbm.xml"/>
42     <mapping resource="com/sga/negocio/mapeo/Status.hbm.xml"/>
43     <mapping
resource="com/sga/negocio/mapeo/TipoUsuario.hbm.xml"/>
44     <mapping
resource="com/sga/negocio/mapeo/UnidadEducativa.hbm.xml"/>
45     <mapping
resource="com/sga/negocio/mapeo/YearLectivo.hbm.xml"/>
46
47
48
49
50     <!--
51     <property
name="cglib.use_reflection_optimizer">true</property>
52
53
54
55     <mapping resource="com/sga/negocio/mapeo/Alumno.hbm.xml"/>
56
57     <mapping
resource="com/sga/negocio/mapeo/Asistencia.hbm.xml"/>
58
59     <mapping
resource="com/sga/negocio/mapeo/Configuracion.hbm.xml"/>
60
61
62
63     <mapping
resource="com/sga/negocio/mapeo/Profesor.hbm.xml"/>
64
65     <mapping
resource="com/sga/negocio/mapeo/Representante.hbm.xml"/>
66
67     <mapping resource="com/sga/negocio/mapeo/Usuario.hbm.xml"/>
68
69
70
71
72     -->
73     </session-factory>
74 </hibernate-configuration>

```

### 5.2.2 SwingLab.

“SwingLabs es un laboratorio de Open Source para explorar nuevas maneras de hacer aplicaciones Swings más fáciles de escribir, con iteración mejorada y una apelación visual mayor. Es un proyecto sombrilla para varias iniciativas de Fuente Abierta patrocinadas por Sun Microsystems y es parte de la comunidad de java.net. Pueden

emigrarse código exitoso y conceptos a las versiones futuras de la plataforma de Java™.

El SwingX de SwingLab mantiene una colección poderosa de componentes del desktop para todos sus proyectos Swings. Disponible bajo la licencia de LGPL, el proyecto de SwingX es el inicio para sus proyectos personales y profesionales.”<sup>18</sup>

Por ejemplo “Java Calendar es un componente Swing que permite visualizar y obtener fechas desde una aplicación. La suite completa cuenta con un componente calendario, un combobox para adquirir fechas, un cuadro de diálogo modal con el mismo propósito, un editor de celdas de tabla que permite introducir una fecha en una celda de una tabla, y un Applet listo para ser empleado.”<sup>19</sup>

A continuación se lista un ejemplo de la utilización de los diferentes componentes de SwingLab que facilitan y mejora la interfaz al usuario en el desarrollo de un software:

```

1 package com.sga.gui.panel.util;
2
3 import com.sga.gui.PantallaPrincipal;
4
5 public class PanTask extends javax.swing.JPanel {
6
7     private javax.swing.JDesktopPane escritorio;
8     private PantallaPrincipal pantalla;
9
10    public PanTask() {
11        initComponents();
12        ocultaComponentes();
13        activaComponentes();
14    }
15
16    public void setActivo(boolean b){
17        this.setVisible(b);
18    }
19
20    private void activaComponentes(){
21        getHylAccesos().setEnabled(true);
22        getHylAcreditacion().setEnabled(true);
23        getHylAsignarAParalelos().setEnabled(true);
24        getHylAsistencia().setEnabled(true);
25        getHylCuenta().setEnabled(true);
26        getHylEspecialidad().setEnabled(true);
27        getHylInscripcion().setEnabled(true);
28        getHylMateria().setEnabled(true);
29        getHylMateriaAParalelo().setEnabled(true);

```

<sup>18</sup> <http://blog.taragana.com/index.php/archive/high-quality-open-source-java-swing-software-components/es>

<sup>19</sup> <http://blog.taragana.com/index.php/archive/high-quality-open-source-java-swing-software-components/es/javacalendar>.

```

30         getHylMateriaAProfesor().setEnabled(true);
31         getHylMatricula().setEnabled(true);
32         getHylParaleloAEspecialidad().setEnabled(true);
33         getHylPensum().setEnabled(true);
34         getHylProfesor().setEnabled(true);
35         getHylTiposDeUsuario().setEnabled(true);
36         getHylUsuarios().setEnabled(true);
37     }
38
39     private void ocultaComponentes(){
40         getHylAccesos().setVisible(false);
41         getHylAcreditacion().setVisible(false);
42         getHylAsignarAParalelos().setVisible(false);
43         getHylAsistencia().setVisible(false);
44         getHylCuenta().setVisible(false);
45         getHylEspecialidad().setVisible(false);
46         getHylInscripcion().setVisible(false);
47         getHylMateria().setVisible(false);
48         getHylMateriaAParalelo().setVisible(false);
49         getHylMateriaAProfesor().setVisible(false);
50         getHylMatricula().setVisible(false);
51         getHylParaleloAEspecialidad().setVisible(false);
52         getHylPensum().setVisible(false);
53         getHylProfesor().setVisible(false);
54         getHylTiposDeUsuario().setVisible(false);
55         getHylUsuarios().setVisible(false);
56     }
57
58     /**
59     *
60     * @return
61     */
62     public org.jdesktop.swingx.JXHyperlink getHylAccesos() {
63         return hylAccesos;
64     }
65
66     public void setHylAccesos(org.jdesktop.swingx.JXHyperlink
67 hylAccesos) {
68         this.hylAccesos = hylAccesos;
69     }
70     public org.jdesktop.swingx.JXHyperlink getHylAcreditacion() {
71         return hylAcreditacion;
72     }
73
74     public void setHylAcreditacion(org.jdesktop.swingx.JXHyperlink
75 hylAcreditacion) {
76         this.hylAcreditacion = hylAcreditacion;
77     }
78     public org.jdesktop.swingx.JXHyperlink
79 getHylAsignarAParalelos() {
80         return hylAsignarAParalelos;
81     }
82     public void
83 setHylAsignarAParalelos(org.jdesktop.swingx.JXHyperlink
84 hylAsignarAParalelos) {
85         this.hylAsignarAParalelos = hylAsignarAParalelos;
86     }

```

```
86     public org.jdesktop.swingx.JXHyperlink getHylAsistencia() {
87         return hylAsistencia;
88     }
89
90     public void setHylAsistencia(org.jdesktop.swingx.JXHyperlink
hylAsistencia) {
91         this.hylAsistencia = hylAsistencia;
92     }
93
94     public org.jdesktop.swingx.JXHyperlink getHylCuenta() {
95         return hylCuenta;
96     }
97
98     public void setHylCuenta(org.jdesktop.swingx.JXHyperlink
hylCuenta) {
99         this.hylCuenta = hylCuenta;
100    }
101
102    public org.jdesktop.swingx.JXHyperlink getHylEspecialidad() {
103        return hylEspecialidad;
104    }
105
106    public void setHylEspecialidad(org.jdesktop.swingx.JXHyperlink
hylEspecialidad) {
107        this.hylEspecialidad = hylEspecialidad;
108    }
109
110    public org.jdesktop.swingx.JXHyperlink getHylInscripcion() {
111        return hylInscripcion;
112    }
113
114    public void setHylInscripcion(org.jdesktop.swingx.JXHyperlink
hylInscripcion) {
115        this.hylInscripcion = hylInscripcion;
116    }
117
118    public org.jdesktop.swingx.JXHyperlink getHylMateria() {
119        return hylMateria;
120    }
121
122    public void setHylMateria(org.jdesktop.swingx.JXHyperlink
hylMateria) {
123        this.hylMateria = hylMateria;
124    }
125
126    public org.jdesktop.swingx.JXHyperlink
getHylMateriaAParalelo() {
127        return hylMateriaAParalelo;
128    }
129
130    public void
setHylMateriaAParalelo(org.jdesktop.swingx.JXHyperlink
hylMateriaAParalelo) {
131        this.hylMateriaAParalelo = hylMateriaAParalelo;
132    }
133
134    public org.jdesktop.swingx.JXHyperlink
getHylMateriaAProfesor() {
135        return hylMateriaAProfesor;
136    }
137
```



```
138     public void
setHylMateriaAProfesor(org.jdesktop.swingx.JXHyperlink
hylMateriaAProfesor) {
139         this.hylMateriaAProfesor = hylMateriaAProfesor;
140     }
141
142     public org.jdesktop.swingx.JXHyperlink getHylMatricula() {
143         return hylMatricula;
144     }
145
146     public void setHylMatricula(org.jdesktop.swingx.JXHyperlink
hylMatricula) {
147         this.hylMatricula = hylMatricula;
148     }
149
150     public org.jdesktop.swingx.JXHyperlink
getHylParaleloAEspecialidad() {
151         return hylParaleloAEspecialidad;
152     }
153
154     public void
setHylParaleloAEspecialidad(org.jdesktop.swingx.JXHyperlink
hylParaleloAEspecialidad) {
155         this.hylParaleloAEspecialidad = hylParaleloAEspecialidad;
156     }
157
158     public org.jdesktop.swingx.JXHyperlink getHylPensum() {
159         return hylPensum;
160     }
161
162     public void setHylPensum(org.jdesktop.swingx.JXHyperlink
hylPensum) {
163         this.hylPensum = hylPensum;
164     }
165
166     public org.jdesktop.swingx.JXHyperlink getHylProfesor() {
167         return hylProfesor;
168     }
169
170     public void setHylProfesor(org.jdesktop.swingx.JXHyperlink
hylProfesor) {
171         this.hylProfesor = hylProfesor;
172     }
173
174     public org.jdesktop.swingx.JXHyperlink getHylTiposDeUsuario()
{
175         return hylTiposDeUsuario;
176     }
177
178     public void
setHylTiposDeUsuario(org.jdesktop.swingx.JXHyperlink
hylTiposDeUsuario) {
179         this.hylTiposDeUsuario = hylTiposDeUsuario;
180     }
181
182     public org.jdesktop.swingx.JXHyperlink getHylUsuario() {
183         return getHylUsuarios();
184     }
185
186     public void setHylUsuarios(org.jdesktop.swingx.JXHyperlink
hylUsuarios) {
```

```
187         this.hylUsuarios = hylUsuarios;
188     }
189
190     public org.jdesktop.swingx.JXTaskPaneContainer
getJXTaskPaneContainer1() {
191         return jXTaskPaneContainer1;
192     }
193
194     public void
setJXTaskPaneContainer1(org.jdesktop.swingx.JXTaskPaneContainer
jXTaskPaneContainer1) {
195         this.jXTaskPaneContainer1 = jXTaskPaneContainer1;
196     }
197
198     public org.jdesktop.swingx.JXTaskPane getTskAlumnos() {
199         return tskAlumnos;
200     }
201
202     public void setTskAlumnos(org.jdesktop.swingx.JXTaskPane
tskAlumnos) {
203         this.tskAlumnos = tskAlumnos;
204     }
205
206     public org.jdesktop.swingx.JXTaskPane getTskMateria() {
207         return tskMateria;
208     }
209
210     public void setTskMateria(org.jdesktop.swingx.JXTaskPane
tskMateria) {
211         this.tskMateria = tskMateria;
212     }
213
214     public org.jdesktop.swingx.JXTaskPane getTskMatricula() {
215         return tskMatricula;
216     }
217
218     public void setTskMatricula(org.jdesktop.swingx.JXTaskPane
tskMatricula) {
219         this.tskMatricula = tskMatricula;
220     }
221
222     public org.jdesktop.swingx.JXTaskPane getTskProfesor() {
223         return tskProfesor;
224     }
225
226     public void setTskProfesor(org.jdesktop.swingx.JXTaskPane
tskProfesor) {
227         this.tskProfesor = tskProfesor;
228     }
229
230     public org.jdesktop.swingx.JXTaskPane getTskUsuarios() {
231         return tskUsuarios;
232     }
233
234     public void setTskUsuarios(org.jdesktop.swingx.JXTaskPane
tskUsuarios) {
235         this.tskUsuarios = tskUsuarios;
236     }
237
238     public javax.swing.JDesktopPane getEscritorio() {
239         return escritorio;
```

```
240     }
241
242     public void setEscritorio(javax.swing.JDesktopPane escritorio)
243     {
244         this.escritorio = escritorio;
245     }
246     public PantallaPrincipal getPantalla() {
247         return pantalla;
248     }
249
250     public void setPantalla(PantallaPrincipal pantalla) {
251         this.pantalla = pantalla;
252     }
253
254     public org.jdesktop.swingx.JXHyperlink getHylUsuarios() {
255         return hylUsuarios;
256     }
257
258     public org.jdesktop.swingx.calendar.JXMonthView
getMntCalendario() {
259         return mntCalendario;
260     }
261
262     public void
setMntCalendario(org.jdesktop.swingx.calendar.JXMonthView
mntCalendario) {
263         this.mntCalendario = mntCalendario;
264     }
265
266     // <editor-fold defaultstate="collapsed" desc=" Generated Code
">
267     private void initComponents() {
268         jXTaskPaneContainer1 = new
org.jdesktop.swingx.JXTaskPaneContainer();
269         tskMatricula = new org.jdesktop.swingx.JXTaskPane();
270         hylInscripcion = new org.jdesktop.swingx.JXHyperlink();
271         hylMatricula = new org.jdesktop.swingx.JXHyperlink();
272         tskAlumnos = new org.jdesktop.swingx.JXTaskPane();
273         hylAsignarAParalelos = new
org.jdesktop.swingx.JXHyperlink();
274         hylAcreditacion = new org.jdesktop.swingx.JXHyperlink();
275         hylAsistencia = new org.jdesktop.swingx.JXHyperlink();
276         tskProfesor = new org.jdesktop.swingx.JXTaskPane();
277         hylProfesor = new org.jdesktop.swingx.JXHyperlink();
278         hylMateriaAProfesor = new
org.jdesktop.swingx.JXHyperlink();
279         tskMateria = new org.jdesktop.swingx.JXTaskPane();
280         hylPensum = new org.jdesktop.swingx.JXHyperlink();
281         hylEspecialidad = new org.jdesktop.swingx.JXHyperlink();
282         hylParaleloAEspecialidad = new
org.jdesktop.swingx.JXHyperlink();
283         hylMateria = new org.jdesktop.swingx.JXHyperlink();
284         hylMateriaAParalelo = new
org.jdesktop.swingx.JXHyperlink();
285         tskUsuarios = new org.jdesktop.swingx.JXTaskPane();
286         hylTiposDeUsuario = new org.jdesktop.swingx.JXHyperlink();
287         hylUsuarios = new org.jdesktop.swingx.JXHyperlink();
288         hylAccesos = new org.jdesktop.swingx.JXHyperlink();
289         hylCuenta = new org.jdesktop.swingx.JXHyperlink();
```

```
290         mntCalendario = new
org.jdesktop.swingx.calendar.JXMonthView();
291
292         setLayout(new java.awt.BorderLayout());
293
294         jXTaskPaneContainer1.setToolTipText("Barra de Tareas");
295         jXTaskPaneContainer1.setAutoscrolls(true);
296         tskMatricula.setExpanded(false);
297         tskMatricula.setScrollOnExpand(true);
298         tskMatricula.setSpecial(true);
299         tskMatricula.setTitle("Matricula");
300         tskMatricula.setToolTipText("Matricula");
301         hylInscripcion.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Inscripcion.png")));
302         hylInscripcion.setText("Inscripcion");
303         hylInscripcion.addActionListener(new
java.awt.event.ActionListener() {
304             public void actionPerformed(java.awt.event.ActionEvent
evt) {
305                 hylInscripcionActionPerformed(evt);
306             }
307         });
308
309         tskMatricula.getContentPane().add(hylInscripcion);
310
311         hylMatricula.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Matricula.png")));
312         hylMatricula.setText("Matricula");
313         hylMatricula.addActionListener(new
java.awt.event.ActionListener() {
314             public void actionPerformed(java.awt.event.ActionEvent
evt) {
315                 hylMatriculaActionPerformed(evt);
316             }
317         });
318
319         tskMatricula.getContentPane().add(hylMatricula);
320
321         jXTaskPaneContainer1.add(tskMatricula);
322
323         tskAlumnos.getContentPane().setLayout(new
org.jdesktop.swingx.VerticalLayout());
324
325         tskAlumnos.setExpanded(false);
326         tskAlumnos.setScrollOnExpand(true);
327         tskAlumnos.setSpecial(true);
328         tskAlumnos.setTitle("Alumnos");
329         tskAlumnos.setToolTipText("Alumnos");
330         hylAsignarAParalelos.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/AsignaraParalelo.png
")));
331         hylAsignarAParalelos.setText("Asignar a Paralelos");
332         hylAsignarAParalelos.addActionListener(new
java.awt.event.ActionListener() {
333             public void actionPerformed(java.awt.event.ActionEvent
evt) {
334                 hylAsignarAParalelosActionPerformed(evt);
335             }
336         });
337
338         tskAlumnos.getContentPane().add(hylAsignarAParalelos);
```

```

339
340         hylAcreditacion.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Acreditacion.png")))
;
341         hylAcreditacion.setText("Acreditaci\u00f3n");
342         hylAcreditacion.addActionListener(new
java.awt.event.ActionListener() {
343             public void actionPerformed(java.awt.event.ActionEvent
evt) {
344                 hylAcreditacionActionPerformed(evt);
345             }
346         });
347
348         tskAlumnos.getContentPane().add(hylAcreditacion);
349
350         hylAsistencia.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Asistencia.png")));
351         hylAsistencia.setText("Asistencias");
352         hylAsistencia.addActionListener(new
java.awt.event.ActionListener() {
353             public void actionPerformed(java.awt.event.ActionEvent
evt) {
354                 hylAsistenciaActionPerformed(evt);
355             }
356         });
357
358         tskAlumnos.getContentPane().add(hylAsistencia);
359
360         jXTaskPaneContainer1.add(tskAlumnos);
361
362         tskProfesor.getContentPane().setLayout(new
org.jdesktop.swingx.VerticalLayout());
363
364         tskProfesor.setExpanded(false);
365         tskProfesor.setScrollOnExpand(true);
366         tskProfesor.setSpecial(true);
367         tskProfesor.setTitle("Profesor");
368         tskProfesor.setToolTipText("Profesor");
369         hylProfesor.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Profesor.png")));
370         hylProfesor.setText("Profesor");
371         hylProfesor.addActionListener(new
java.awt.event.ActionListener() {
372             public void actionPerformed(java.awt.event.ActionEvent
evt) {
373                 hylProfesorActionPerformed(evt);
374             }
375         });
376
377         tskProfesor.getContentPane().add(hylProfesor);
378
379         hylMateriaAProfesor.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/AsignarmateriaaProfe
sor.png")));
380         hylMateriaAProfesor.setText("Asignar Materia a Profesor");
381         hylMateriaAProfesor.addActionListener(new
java.awt.event.ActionListener() {
382             public void actionPerformed(java.awt.event.ActionEvent
evt) {
383                 hylMateriaAProfesorActionPerformed(evt);
384             }

```

```

385         });
386
387         tskProfesor.getContentPane().add(hylMateriaAProfesor);
388
389         jXTaskPaneContainer1.add(tskProfesor);
390
391         tskMateria.getContentPane().setLayout(new
org.jdesktop.swingx.VerticalLayout());
392
393         tskMateria.setExpanded(false);
394         tskMateria.setScrollOnExpand(true);
395         tskMateria.setSpecial(true);
396         tskMateria.setTitle("Materia");
397         tskMateria.setToolTipText("Materia");
398         hylPensum.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Pensum.png")));
399         hylPensum.setText("Pensum");
400         hylPensum.addActionListener(new
java.awt.event.ActionListener() {
401             public void actionPerformed(java.awt.event.ActionEvent
evt) {
402                 hylPensumActionPerformed(evt);
403             }
404         });
405
406         tskMateria.getContentPane().add(hylPensum);
407
408         hylEspecialidad.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Especialidad.png")))
;
409         hylEspecialidad.setText("Especialidad");
410         hylEspecialidad.addActionListener(new
java.awt.event.ActionListener() {
411             public void actionPerformed(java.awt.event.ActionEvent
evt) {
412                 hylEspecialidadActionPerformed(evt);
413             }
414         });
415
416         tskMateria.getContentPane().add(hylEspecialidad);
417
418         hylParaleloAEspecialidad.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/ParaleloaEspecialida
d.png")));
419         hylParaleloAEspecialidad.setText("Paralelo a
Especialidad");
420         hylParaleloAEspecialidad.addActionListener(new
java.awt.event.ActionListener() {
421             public void actionPerformed(java.awt.event.ActionEvent
evt) {
422                 hylParaleloAEspecialidadActionPerformed(evt);
423             }
424         });
425
426         tskMateria.getContentPane().add(hylParaleloAEspecialidad);
427
428         hylMateria.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Materia.png")));
429         hylMateria.setText("Materia");
430         hylMateria.addActionListener(new
java.awt.event.ActionListener() {

```

```

431         public void actionPerformed(java.awt.event.ActionEvent
432     evt) {
433         hylMateriaActionPerformed(evt);
434     }
435     });
436     tskMateria.getContentPane().add(hylMateria);
437
438     hylMateriaAParalelo.setIcon(new
439     javax.swing.ImageIcon(getClass().getResource("/16/MateriaaParalelo.png
440     ")));
441     hylMateriaAParalelo.setText("Materia a Paralelo");
442     hylMateriaAParalelo.addActionListener(new
443     java.awt.event.ActionListener() {
444         public void actionPerformed(java.awt.event.ActionEvent
445     evt) {
446         hylMateriaAParaleloActionPerformed(evt);
447     }
448     });
449     tskMateria.getContentPane().add(hylMateriaAParalelo);
450     jXTaskPaneContainer1.add(tskMateria);
451
452     tskUsuarios.getContentPane().setLayout(new
453     org.jdesktop.swingx.VerticalLayout());
454
455     tskUsuarios.setExpanded(false);
456     tskUsuarios.setScrollOnExpand(true);
457     tskUsuarios.setSpecial(true);
458     tskUsuarios.setTitle("Usuarios");
459     tskUsuarios.setToolTipText("Usuario");
460     hylTiposDeUsuario.setIcon(new
461     javax.swing.ImageIcon(getClass().getResource("/16/TiposdeUsuario.png")
462     ));
463     hylTiposDeUsuario.setText("Tipos de Usuario");
464     hylTiposDeUsuario.addActionListener(new
465     java.awt.event.ActionListener() {
466         public void actionPerformed(java.awt.event.ActionEvent
467     evt) {
468         hylTiposDeUsuarioActionPerformed(evt);
469     }
470     });
471     tskUsuarios.getContentPane().add(hylTiposDeUsuario);
472
473     hylUsuarios.setIcon(new
474     javax.swing.ImageIcon(getClass().getResource("/16/Usuarios.png")));
475     hylUsuarios.setText("Usuarios");
476     hylUsuarios.addActionListener(new
477     java.awt.event.ActionListener() {
478         public void actionPerformed(java.awt.event.ActionEvent
479     evt) {
480         hylUsuariosActionPerformed(evt);
481     }
482     });
483     tskUsuarios.getContentPane().add(hylUsuarios);
484
485     hylAccesos.setIcon(new
486     javax.swing.ImageIcon(getClass().getResource("/16/Accesos.png")));

```

```

478         hylAccesos.setText("Accesos");
479         hylAccesos.addActionListener(new
java.awt.event.ActionListener() {
480             public void actionPerformed(java.awt.event.ActionEvent
evt) {
481                 hylAccesosActionPerformed(evt);
482             }
483         });
484
485         tsuUsuarios.getContentPane().add(hylAccesos);
486
487         hylCuenta.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/16/Cuenta.png")));
488         hylCuenta.setText("Cuenta");
489         hylCuenta.addActionListener(new
java.awt.event.ActionListener() {
490             public void actionPerformed(java.awt.event.ActionEvent
evt) {
491                 hylCuentaActionPerformed(evt);
492             }
493         });
494
495         tsuUsuarios.getContentPane().add(hylCuenta);
496
497         jxTaskPaneContainer1.add(tsuUsuarios);
498
499         mntCalendario.setToolTipText("Calendario");
500         mntCalendario.setTraversable(true);
501         org.jdesktop.layout.GroupLayout mntCalendarioLayout = new
org.jdesktop.layout.GroupLayout(mntCalendario);
502         mntCalendario.setLayout(mntCalendarioLayout);
503         mntCalendarioLayout.setHorizontalGroup(
504
mntCalendarioLayout.createParallelGroup(org.jdesktop.layout.GroupLayou
t.LEADING)
505             .add(0, 213, Short.MAX_VALUE)
506         );
507         mntCalendarioLayout.setVerticalGroup(
508
mntCalendarioLayout.createParallelGroup(org.jdesktop.layout.GroupLayou
t.LEADING)
509             .add(0, 172, Short.MAX_VALUE)
510         );
511         jxTaskPaneContainer1.add(mntCalendario);
512
513         add(jxTaskPaneContainer1, java.awt.BorderLayout.CENTER);
514
515     } // </editor-fold>
516
517     private void
hylInscripcionActionPerformed(java.awt.event.ActionEvent evt) {
518         getPantalla().getAcciones().accionInscripcion();
519     }
520
521     private void
hylMatriculaActionPerformed(java.awt.event.ActionEvent evt) {
522         getPantalla().getAcciones().accionMatricula();
523     }
524
525     private void
hylAsignarAParalelosActionPerformed(java.awt.event.ActionEvent evt) {

```



```
526         getPantalla().getAcciones().accionAsignaAParalelos();
527     }
528
529     private void
530     hylAcreditacionActionPerformed(java.awt.event.ActionEvent evt) {
531         getPantalla().getAcciones().accionAcreditacion();
532     }
533
534     private void
535     hylAsistenciaActionPerformed(java.awt.event.ActionEvent evt) {
536         getPantalla().getAcciones().accionAsistencias();
537     }
538
539     private void
540     hylCuentaActionPerformed(java.awt.event.ActionEvent evt) {
541         getPantalla().getAcciones().accionCuenta();
542     }
543
544     private void
545     hylAccesosActionPerformed(java.awt.event.ActionEvent evt) {
546         getPantalla().getAcciones().accionAcceso();
547     }
548
549     private void
550     hylUsuariosActionPerformed(java.awt.event.ActionEvent evt) {
551         getPantalla().getAcciones().accionUsuario();
552     }
553
554     private void
555     hylTiposDeUsuarioActionPerformed(java.awt.event.ActionEvent evt) {
556         getPantalla().getAcciones().accionTipoUsuario();
557     }
558
559     private void
560     hylMateriaAParaleloActionPerformed(java.awt.event.ActionEvent evt) {
561         getPantalla().getAcciones().accionMateriaAParalelo();
562     }
563
564     private void
565     hylMateriaActionPerformed(java.awt.event.ActionEvent evt) {
566         getPantalla().getAcciones().accionMateria();
567     }
568
569     private void
570     hylParaleloAEspecialidadActionPerformed(java.awt.event.ActionEvent
571     evt) {
572         getPantalla().getAcciones().accionParaleloAEspecialidad();
573     }
574
575     private void
576     hylEspecialidadActionPerformed(java.awt.event.ActionEvent evt) {
577         getPantalla().getAcciones().accionEspecialidad();
578     }
579
580     private void
581     hylPensumActionPerformed(java.awt.event.ActionEvent evt) {
582         getPantalla().getAcciones().accionPensum();
583     }
584
585     private void
586     hylMateriaAProfesorActionPerformed(java.awt.event.ActionEvent evt) {
```

```

574
getPantalla().getAcciones().accionAsignarMateriaAProfesor();
575     }
576
577     private void
hylProfesorActionPerformed(java.awt.event.ActionEvent evt) {
578         getPantalla().getAcciones().accionProfesor();
579     }
580
581
582     // Variables declaration - do not modify
583     private org.jdesktop.swingx.JXHyperlink hylAccesos;
584     private org.jdesktop.swingx.JXHyperlink hylAcreditacion;
585     private org.jdesktop.swingx.JXHyperlink hylAsignarAParalelos;
586     private org.jdesktop.swingx.JXHyperlink hylAsistencia;
587     private org.jdesktop.swingx.JXHyperlink hylCuenta;
588     private org.jdesktop.swingx.JXHyperlink hylEspecialidad;
589     private org.jdesktop.swingx.JXHyperlink hylInscripcion;
590     private org.jdesktop.swingx.JXHyperlink hylMateria;
591     private org.jdesktop.swingx.JXHyperlink hylMateriaAParalelo;
592     private org.jdesktop.swingx.JXHyperlink hylMateriaAProfesor;
593     private org.jdesktop.swingx.JXHyperlink hylMatricula;
594     private org.jdesktop.swingx.JXHyperlink
hylParaleloAEspecialidad;
595     private org.jdesktop.swingx.JXHyperlink hylPensum;
596     private org.jdesktop.swingx.JXHyperlink hylProfesor;
597     private org.jdesktop.swingx.JXHyperlink hylTiposDeUsuario;
598     private org.jdesktop.swingx.JXHyperlink hylUsuarios;
599     private org.jdesktop.swingx.JXTaskPaneContainer
jXTaskPaneContainer1;
600     private org.jdesktop.swingx.calendar.JXMonthView
mntCalendario;
601     private org.jdesktop.swingx.JXTaskPane tskAlumnos;
602     private org.jdesktop.swingx.JXTaskPane tskMateria;
603     private org.jdesktop.swingx.JXTaskPane tskMatricula;
604     private org.jdesktop.swingx.JXTaskPane tskProfesor;
605     private org.jdesktop.swingx.JXTaskPane tskUsuarios;
606     // End of variables declaration
607
608 }
609

```

### 5.2.3 JGoodies.

El proyecto de código abierto JGoodies está compuesto de varias librerías que ayudan a conseguir aplicaciones Swing de mejor aspecto en una fracción del tiempo previamente requerido.

Está compuesto entre otros, los siguientes subproyectos:

- Looks, Dos l&f (Windows y Plastic) de aspecto muy delicado y elegante entre otros..

- forms, sirve para construir paneles de modo eficiente y elegante.
- Validation, para validar los formularios Swing y presentar los resultados de modo consistente.
- Animations, en tiempo real utilizando java.

El proyecto de código abierto JGoodies permite mejorar de un programa en muy poco tiempo. Ayuda a construir paneles elegantes y constantes de manera eficiente

Un ejemplo claro de cómo se puede utilizar los recursos de esta librería se muestra en el ejemplo adjunto:

```

1 package com.sga.util;
2
3 import com.jgoodies.looks.LookUtils;
4 import com.jgoodies.looks.plastic.Plastic3DLookAndFeel;
5 import com.jgoodies.looks.plastic.PlasticLookAndFeel;
6 import com.jgoodies.looks.plastic.PlasticTheme;
7 import com.jgoodies.looks.plastic.PlasticXPLookAndFeel;
8 import com.jgoodies.looks.plastic.theme.BrownSugar;
9 import com.jgoodies.looks.plastic.theme.DarkStar;
10 import com.jgoodies.looks.plastic.theme.DesertBlue;
11 import com.jgoodies.looks.plastic.theme.DesertBluer;
12 import com.jgoodies.looks.plastic.theme.DesertGreen;
13 import com.jgoodies.looks.plastic.theme.DesertRed;
14 import com.jgoodies.looks.plastic.theme.DesertYellow;
15 import com.jgoodies.looks.plastic.theme.ExperienceBlue;
16 import com.jgoodies.looks.plastic.theme.ExperienceGreen;
17 import com.jgoodies.looks.plastic.theme.ExperienceRoyale;
18 import com.jgoodies.looks.plastic.theme.InvertedColorTheme;
19 import com.jgoodies.looks.plastic.theme.LightGray;
20 import com.jgoodies.looks.plastic.theme.Silver;
21 import com.jgoodies.looks.plastic.theme.SkyBlue;
22 import com.jgoodies.looks.plastic.theme.SkyBluer;
23 import com.jgoodies.looks.plastic.theme.SkyGreen;
24 import com.jgoodies.looks.plastic.theme.SkyKrupp;
25 import com.jgoodies.looks.plastic.theme.SkyPink;
26 import com.jgoodies.looks.plastic.theme.SkyRed;
27 import com.jgoodies.looks.plastic.theme.SkyYellow;
28 import com.sga.gui.PantallaPrincipal;
29 import javax.swing.JFrame;
30 import javax.swing.LookAndFeel;
31 import javax.swing.SwingUtilities;
32 import javax.swing.UIManager;
33 import javax.swing.plaf.metal.DefaultMetalTheme;
34 import javax.swing.plaf.metal.MetalLookAndFeel;
35 import javax.swing.plaf.metal.OceanTheme;
36
37 public class Look {
38
39     public static final int BROWN_SUGAR = 1;

```

```

40     public static final int DARK_STAR =2;
41     public static final int DESERT_BLUE=3;
42     public static final int DESERT_BLUER=4;
43     public static final int DESERT_GREEN=5;
44     public static final int DESERT_RED=6;
45     public static final int DESERT_YELLOW=7;
46     public static final int EXPERIENCE_BLUE=8;
47     public static final int EXPERIENCE_GREEN=9;
48     public static final int EXPERIENCE_ROYALE=10;
49     public static final int INVERTED_COLOR_THEME=11;
50     public static final int LIGHT_GRAY=12;
51     public static final int SILVER=13;
52     public static final int SKY_BLUE=14;
53     public static final int SKY_BLUER=15;
54     public static final int SKY_GREEN=16;
55     public static final int SKY_KRUPP=17;
56     public static final int SKY_PINK=18;
57     public static final int SKY_RED=19;
58     public static final int SKY_YELLOW=20;
59     public static final int DEFAULT=21;
60     public static final int WINDOWS=22;
61     public static final int MOFIL=23;
62
63     public static PlasticLookAndFeel pt=new PlasticLookAndFeel();
64     public static Plastic3DLookAndFeel pt3 = new
Plastic3DLookAndFeel();
65     public static PlasticXPLookAndFeel ptx = new
PlasticXPLookAndFeel();
66     public static LookAndFeel lf = UIManager.getLookAndFeel();
67
68     public Look() {
69         PlasticLookAndFeel pt=new PlasticLookAndFeel();
70     }
71
72     public static void updateLookAndFeel(String nombre, JFrame
frame){
73         try{
74             UIManager.setLookAndFeel(nombre);
75             SwingUtilities.updateComponentTreeUI(frame);
76         }catch (Exception ex){}
77     }
78
79     public static void setLookPlastic(int tipo, PantallaPrincipal
p){
80         Inicializa.getUsuario().setLook(tipo);
81         if(tipo<21){
82             setTemaPlastic(tipo);
83             pt.setHighContrastFocusColorsEnabled(true);
84             try {
85                 UIManager.setLookAndFeel(pt);
86
87             UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
88             SwingUtilities.updateComponentTreeUI(p);
89             }catch (Exception e) {
90                 e.printStackTrace();
91             }
92             }else if(tipo==21)
93                 setDefault(p);
94             else if(tipo==22)

```

```

94
updateLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel"
,p);
95     else if(tipo==23)
96
updateLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel",p);
97     }
98
99     public static void setDefault(PantallaPrincipal p){
100         try {
101             OceanTheme ot = new OceanTheme();
102             MetalLookAndFeel.setCurrentTheme(ot);
103
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
104             SwingUtilities.updateComponentTreeUI(p);
105         }catch (Exception e) {
106             e.printStackTrace();
107         }
108     }
109
110     public static void setLook3DPlastic(int tipo,
PantallaPrincipal p){
111         setTemaPlastic(tipo);
112         pt.setHighContrastFocusColorsEnabled(false);
113         try {
114             UIManager.setLookAndFeel(pt);
115
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
116             SwingUtilities.updateComponentTreeUI(p);
117         }catch (Exception e) {
118             }
119     }
120
121     public static void setLookXPPlastic(int tipo,
PantallaPrincipal p){
122         setTemaPlastic(tipo);
123         pt.setHighContrastFocusColorsEnabled(false);
124         try {
125             UIManager.setLookAndFeel(pt);
126
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
127             SwingUtilities.updateComponentTreeUI(p);
128         }catch (Exception e) {
129             }
130     }
131
132
133     public static void updateLookAndFeel(String nombre,
PantallaPrincipal p){
134         try{
135             UIManager.setLookAndFeel(nombre);
136             SwingUtilities.updateComponentTreeUI(p);
137         }catch (Exception ex){}
138     }
139
140
141     private static void setTemaPlastic(int tipo){
142         switch (tipo){
143             case BROWN_SUGAR:
144                 pt.setPlasticTheme(new BrownSugar());
145                 break;

```

```
146         case DARK_STAR:
147             pt.setPlasticTheme(new DarkStar());
148             break;
149         case DESERT_BLUE:
150             pt.setPlasticTheme(new DesertBlue());
151             break;
152         case DESERT_BLUER:
153             pt.setPlasticTheme(new DesertBluer());
154             break;
155         case DESERT_GREEN:
156             pt.setPlasticTheme(new DesertGreen());
157             break;
158         case DESERT_RED:
159             pt.setPlasticTheme(new DesertRed());
160             break;
161         case DESERT_YELLOW:
162             pt.setPlasticTheme(new DesertYellow());
163             break;
164         case EXPERIENCE_BLUE:
165             pt.setPlasticTheme(new ExperienceBlue());
166             break;
167         case EXPERIENCE_GREEN:
168             pt.setPlasticTheme(new ExperienceGreen());
169             break;
170         case EXPERIENCE_ROYALE:
171             pt.setPlasticTheme(new ExperienceRoyale());
172             break;
173         case INVERTED_COLOR_THEME:
174             pt.setPlasticTheme(new InvertedColorTheme() {
175             });
176             break;
177         case LIGHT_GRAY:
178             pt.setPlasticTheme(new LightGray());
179             break;
180         case SILVER:
181             pt.setPlasticTheme(new Silver());
182             break;
183         case SKY_BLUE:
184             pt.setPlasticTheme(new SkyBlue());
185             break;
186         case SKY_BLUER:
187             pt.setPlasticTheme(new SkyBluer());
188             break;
189         case SKY_GREEN:
190             pt.setPlasticTheme(new SkyGreen());
191             break;
192         case SKY_KRUPP:
193             pt.setPlasticTheme(new SkyKrupp());
194             break;
195         case SKY_PINK:
196             pt.setPlasticTheme(new SkyPink());
197             break;
198         case SKY_RED:
199             pt.setPlasticTheme(new SkyRed());
200             break;
201         case SKY_YELLOW:
202             pt.setPlasticTheme(new SkyYellow());
203             break;
204     }
205 }
206 }
```

```

207
208 private static DefaultMetalTheme getTemaPlastic(int tipo){
209     switch (tipo){
210         case BROWN_SUGAR:
211             return new BrownSugar();
212         case DARK_STAR:
213             return new DarkStar();
214         case DESERT_BLUE:
215             return new DesertBlue();
216         case DESERT_BLUER:
217             return new DesertBluer();
218         case DESERT_GREEN:
219             return new DesertGreen();
220         case DESERT_RED:
221             return new DesertRed();
222         case DESERT_YELLOW:
223             return new DesertYellow();
224         case EXPERIENCE_BLUE:
225             return new ExperienceBlue();
226         case SKY_BLUER:
227             return new SkyBluer();
228         case SKY_GREEN:
229             return new SkyGreen();
230         case SKY_KRUPP:
231             return new SkyKrupp();
232         case SKY_PINK:
233             return new SkyPink();
234         case EXPERIENCE_GREEN:
235             return new ExperienceGreen();
236         case EXPERIENCE_ROYALE:
237             return new ExperienceRoyale();
238         case INVERTED_COLOR_THEME:
239             return new InvertedColorTheme() {
240                 };
241         case LIGHT_GRAY:
242             return new LightGray();
243         case SILVER:
244             return new Silver();
245         case SKY_BLUE:
246             return new SkyBlue();
247         case SKY_RED:
248             return new SkyRed();
249         case SKY_YELLOW:
250             return new SkyYellow();
251     }
252     return new DefaultMetalTheme();
253
254 }
255
256
257 }

```

#### 5.2.4 JavaHelp.

Sistema que facilita el desarrollo de ayuda on-line, es independiente de la plataforma (implementado en Java) y evita que cada usuario tenga que crear su propio sistema de ayuda propietario.

### 5.2.4.1 Características Básicas.

El visor de Ayuda consta de los siguientes puntos:

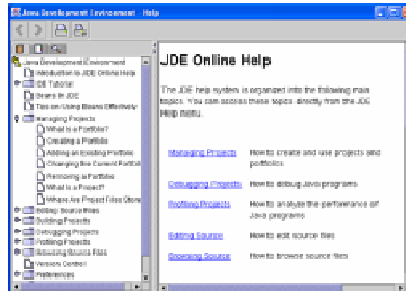


Gráfico 5-4 Visor de JavaHelp<sup>20</sup>

- Panel de contenidos
- Panel de navegación
- Tabla de contenidos
- Índice
- Búsqueda
- Distintas vistas de la información

### 5.2.4.2 Herramientas.

JavaHelp utiliza las siguientes herramientas para generar la ayuda de un sistema:

- **Hsvviewer**, Permite visualizar ficheros HelpSet. Puede invocarse desde línea de comandos o desde Windows.
- **Jhindexer**, programa de línea de comandos que crea la base de datos de búsqueda.
- **Jhsearch**, programa de línea de comandos que busca en la base de datos creada con jhindexer.

<sup>20</sup> [java.sun.com/products/javahelp](http://java.sun.com/products/javahelp)



### 5.2.4.3 Librerías.

JavaHelp utiliza las siguientes librerías para como soporte para la estructuración y el funcionamiento de la ayuda del software:

- **jh.jar**, librería estándar que incluye todo lo necesario para el visor de ayuda y los navegadores.
- **jhbasic.jar**, subconjunto de la librería jh.jar que no soporta la búsqueda.
- **jhtools.jar**, Incluye las herramientas para construir y consultar la base de datos.
- **jhall.jar**, contiene todas las clases de javahelp.

### 5.2.4.4 Creación de la Ayuda con JavaHelp.

- Creación de los temas de ayuda para el usuario (tópicos HTML). Para crear tópicos se puede emplear cualquier herramienta (o bien directamente) que genere HTML. Conviene agrupar tópicos relacionados y disponerlos jerárquicamente. Si no, al menos, es altamente recomendable crear un subdirectorio donde se incluyan todos los tópicos HTML.
- Creación de los ficheros de metadatos que JavaHelp emplea para presentar la información:
  - Crear Fichero HelpSet, es el único fichero del sistema de ayuda que es referenciado explícitamente por la aplicación. Es el primero que se carga cuando se activa el sistema de ayuda. Contiene toda la información necesaria para ejecutar el sistema de ayuda, tiene la extensión .hs y su sintaxis basada en XML. Contiene, al menos, la siguiente información:

- Fichero map, empleado para asociar identificadores a los nombre de las páginas HTML que constituyen la ayuda.
  - Vistas. Información que describe los navegadores que van a emplearse en el HelpSet (tabla de contenidos, índice, búsqueda).
  - Título del HelpSet.
  - <Home ID> Identificador de la página a ser mostrada cuando se invoca el visor de ayuda sin especificar ningún identificador de página concreto.
- 
- Crear Fichero Map, asocia identificadores a los ficheros HTML que contienen los tópicos, tiene extensión .jhm y la sintaxis basada en XML.
  - Crear Fichero Tabla de contenidos, este fichero describe el contenido y la distribución de la tabla de contenidos además su sintaxis basada en XML.
  - Crear Fichero Índice, este fichero describe la distribución y contenido del índice y su sintaxis basada en XML.
  - Crear base de datos de búsqueda, es necesario emplear el jhindexer para generar la base de datos de búsqueda, la base de datos está formada por seis ficheros localizados en la carpeta JavaHelpSearch. Para crear la base de datos de búsqueda se debe hacer lo siguiente:
    - jhindexer fuentes, se supone que fuentes es el directorio donde están almacenados todos los topics HTML;
    - jhindexer fuentes1 fuentes2, si tenemos más de un directorio donde estén almacenados los topics HTML. En este caso serían dos: fuentes1 y fuentes2.
    - Funcionamiento, jhindexer desciende recursivamente por la jerarquía de directorios indexando todos los ficheros que

encuentra y crea la carpeta JavaHelpSearch dónde almacena la base de datos de búsqueda.

- Comprimir y encapsular ficheros de ayuda en ficheros JAR (Este paso no será necesario). Una vez creada la ayuda se pueden encapsular los ficheros en un único fichero para entregar a los usuarios, el formato empleado por JavaHelp para encapsular y comprimir es JAR (Java ARchive); para ello se emplea el comando jar.

- Para crear un ficheros jar se debe ejecutar los siguientes comandos:

```
C:\ejercicios\ayudas> jar -cvf miayuda.jar *
```

- Para ver los ficheros incluidos en un jar

```
jar -tvf miayuda.jar
```

- Para extraer los ficheros del jar

```
jar -xvf miayuda.jar
```

#### **5.2.4.5 Incorporación de la ayuda en la Aplicación.**

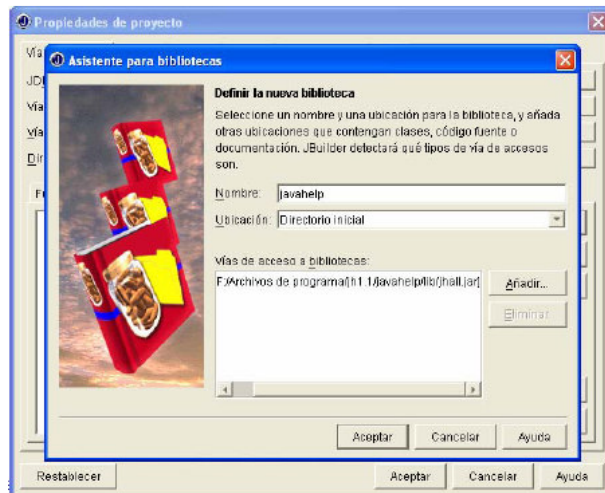
##### **5.2.4.5.1 Clases Básicas Implicadas**

- Clase URL
- Paquete a importar: import java.net.\*;
- Esta clase se utiliza para representar una URL (Uniform Resource Locator), es decir, para identificar un recurso.
- Un recurso puede ser algo tan simple como un fichero o un directorio, o bien puede ser un objeto más complejo.
  
- Clase HelpSet.

- Paquete a importar: `import javax.help.*;`
- Clase que permite tratar desde una aplicación Java el conjunto de datos que constituyen el sistema de ayuda (fichero helpset, fichero map, tabla de contenidos, índice y topics).
- Principales Métodos.
  - `findHelpSet` // método estático que localiza el fichero helpset y devuelve su URL.
  - `createHelpBroker` // crea un objeto HelpBroker para este HelpSet
- Clase HelpBroker
  - Paquete a importar: `import javax.help.*;`
  - Esta clase es la encargada de gestionar la visualización del contenido de la ayuda en la aplicación.
  - Proporciona los métodos apropiados para incorporar ayuda a la aplicación de una forma sencilla.
- Principales Métodos
  - `enableHelpKey` // Habilita la tecla de ayuda sobre un componente.
  - `enableHelpOnButton` // Hace que se despliegue la ayuda cuando se presiona un determinado botón o elemento de menú.
  - `enableHelp` // Indica el topic a mostrar cuando se pulse la tecla de ayuda sobre un determinado componente.

#### **5.2.4.5.2 Pasos para incorporar la ayuda en la Aplicación.**

- Incorporar la librería `jhall.jar` a la aplicación.  
Proyecto->Propiedades->Vías de acceso->Bibliotecas necesarias.



**Gráfico 5-5. Asistente de Bibliotecas JavaHelp<sup>21</sup>**

- Importar los paquetes que son necesarios en la aplicación
 

```
import javax.help.*;
import java.net.*; // para la clase URL
```
- Encontrar el fichero HelpSet y crear un objeto HelpSet
 

```
Try{
hsURL = HelpSet.findHelpSet(null,"ayuda/ayuda.hs"); //
localiza el fichero helpset
hs= new HelpSet(null,hsURL); // crea un objeto HelpSet
} catch (Exception ee){
OptionPane.showMessageDialog(this,"Fichero HelpSet no
encontrado");
return;
}
```
- Crear un objeto HelpBroker que facilita la visualización de la ayuda en la aplicación.
 

```
hb = hs.createHelpBroker ();
```
- Asociar la ayuda a un botón o elemento de menú.

<sup>21</sup> [java.sun.com/products/javahelp](http://java.sun.com/products/javahelp)

```
hb.enableHelpOnButton(jButton1,"introduccion",hs)
```

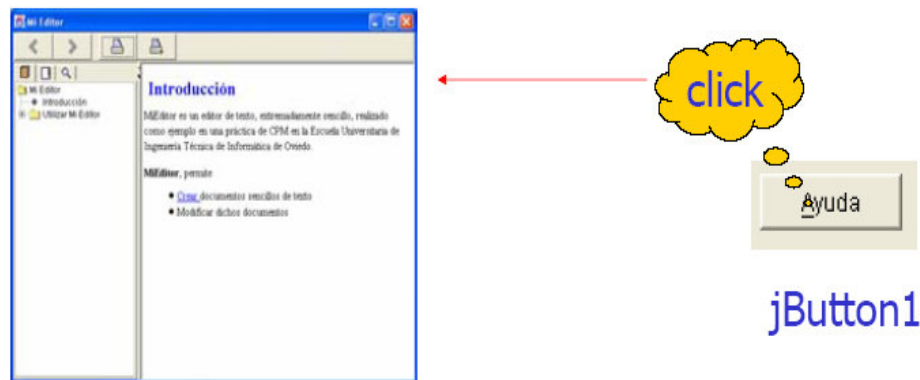


Gráfico 5-6. Ejemplo Pantalla Principal Ayuda Generada con JavaHelp<sup>22</sup>

- Asociar ayuda sensible al contexto a un componente. Para ello:
  - Hacer que un componente responda a la pulsación de la tecla de ayuda (F1)
 

```
hb.enableHelpkey(getRootPane(),"introduccion",hs);
```

// Responden a la tecla F1 todos los componentes situados en la ventana.
  - Asociar a cada componente una página que será la que se muestre cuando se pulse la tecla de ayuda (F1) sobre él.
 

```
hb.enableHelp(jButton1,"guardar",hs);
```

```
hb.enableHelp(jButton2,"abrir",hs);
```

### 5.2.5 JfreeReport

Jfreereport es una librería de clases de libre de java para generar reportes. Los datos fuente de jfreereport son a través de modelos de tablas de la interface generada por swing y formateada acorde se haya definido en archivos basados en XML. Los reportes pueden ser presentados previamente en la pantalla para luego ser impresos o guardados en diferentes formatos pudiendo ser: pdf, HTML, Excel csv y archivos planos txt.

<sup>22</sup> [java.sun.com/products/javahelp](http://java.sun.com/products/javahelp)

### 5.2.5.1 Pasos Básicos.

Existen tres pasos básicos para generar reportes con jfreereport:

- Un arreglo de datos que puede ser accesado vía TableModel interface (este modelo es implementado en Swing en la clase JTable).
- Crear un objeto jfreereport que va a controlar el formato de la generación del reporte. Toda la información que se formatea en los reportes es grabada por una instancia de la clase de jfreereport. Existen dos opciones para crear esta instancia: escribir un reporte con definición en formato XML, obtener la definición de jfreereport y crear su correspondiente objeto jfreereport. O se puede crear una instancia en código de jfreereport y generar el reporte títulos y elementos usando el jfreereport API.

Poner la definición del reporte dentro de un archivo XML es una buena idea, porque este nos permite cambiar el formato del reporte sin recompilar la aplicación. Crear reportes en código puede permitir una flexibilidad adicional si se quiere variar el formato en tiempo de ejecución, pero tiene la desventaja de que el formato de los reportes este dentro de la aplicación.

- Un link de datos (tableModel) con una instancia de jfreereport y pasar el reporte a vista previa como presentación al usuario. Previsualizar el reporte es en lo que más se tiene que tener cuidado en la librería de jfreereport. La pantalla de impresión previa se va a mostrara en la pantalla brindando el control de las páginas, pudiendo movilizarse a través de las mismas, aumento del tamaño de la visualización como también su disminución, imprimir el reporte o exportarlo a diferentes tipos de formatos como pdf, HTML, csv, Excel y XML.

### 5.2.5.2 Aplicación de Ejemplo

En el ejemplo helloworld.java nos permite hacer una básica revisión de los pasos para crear y visualizar los reportes con jfreereport.

- Los Datos. En este ejemplo los datos se ubican en una tabla como se muestra a continuación:

Column1;	Column2:
Hello	World

Esto se crea usando el siguiente código:

```
private TableModel createData()
{
    Object[] columnNames = new String[] {
        "Column1", "Column2" };
    DefaultTableModel
    result = new DefaultTableModel(columnNames,
    1);
    result.setValueAt("Hello", 0, 0);
    result.setValueAt("World!", 0, 1);
    return result;
}
```

La columna nombre es importante desde allí se va a referenciar los elementos dentro de la definición del reporte.

- El Reporte. La instancia de jfreereport usada controla el formato del reporte de “Hello World” creado en el código, esto solo demuestra que es su propio contenedor.

```
private JFreeReport createReportDefinition()
{
    JFreeReport report = new JFreeReport();
```



```

report.setName("A Very Simple Report");

TextElement t1 = ItemFactory.createStringElement(
    "T1",
    new Rectangle2D.Double(0.0,
        0.0, 150.0, 20.0), Color.black,
    ElementAlignment.LEFT.getOldAli
        gnment(),
    ElementAlignment.MIDDLE.getOl
        dAlignment(), null, // font
    "-", // null string
    "Column1"
);

report.getItemBand().addElement(t1);

TextElement t2 = ItemFactory.createStringElement(
    "T2",
    new Rectangle2D.Double(200.0,
        0.0, 150.0, 20.0), Color.black,
    ElementAlignment.LEFT.getOldAli
        gnment(),
    ElementAlignment.MIDDLE.getOld
        Alignment(),
    null, // letra
    "-", // cadena nula o vacia
    "Column2"
);
report.getItemBand().add
    Element(t2);    return
report;
}

```

Para entender que es lo que se hizo aquí, se necesita saber diseñar el reporte y controlar los elementos que son agregados dentro del reporte. El reporte consiste en el número de posibles bloques vacíos incluyendo:

- La cabecera del Reporte, imprime el inicio del reporte.
- El pie del Reporte, imprime los pies al final del reporte
- La cabecera de la Página, imprime la parte superior de cada página.
- El Pie de la Página, imprime el botón de cada página.
- El ítem del Bloque, imprime cada uno de los registros de datos que se encuentran dentro del TableModel.
- Un grupo de Cabecera adicional y/o grupo de pie de cada grupo definido en el reporte.

En la aplicación HelloWorld.java, solo el ítem del bloque es generado, los otros bloques están vacíos por defecto. Un elemento de texto es agregado por cada columna de datos dentro del TableModel. La clase que fabrica el ítem es usada para crear elementos individuales. Cada elemento lleva su propia posición, alineación, configuración de letra y otros atributos de presentación. Es importante también que cada elemento en el ejemplo sea diseccionado de la TableModel por la referencia al nombre de la columna en la tabla.

- Previsualizando el Reporte. Para previsualizar el reporte los datos se conectan con el reporte usando setData (método) y el reporte es pasado a la pantalla de pre visualización.

```
TableModel data = createData();
JFreeReport report = createReportDefinition();

report.setData(data);
try
{
    PreviewFrame preview = new PreviewFrame(report);
    preview.pack();
    preview.setVisible(true);
}
catch (ReportProcessingException e)
```

```

{
    Log.error("Failed to generate report ", e);
}

```

Corriendo el demo, se debería mostrar la siguiente pantalla de previsualización:

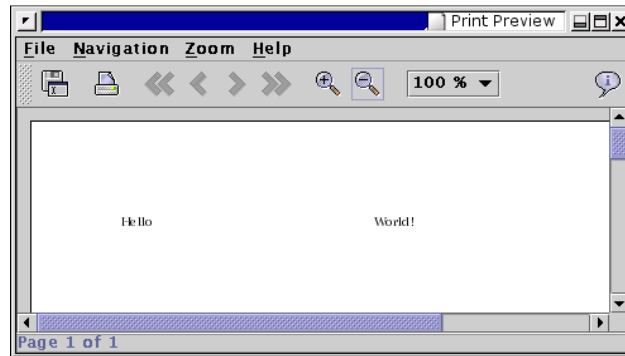


Gráfico 5-7. Ejemplo Reporte<sup>23</sup>

### 5.2.6 IzPack

“IzPack es una solución multiplataforma para generar instaladores basado en tecnología java, permitiendo la instalación del software creado en varias de las plataformas conocidas como Windows, Linux, MAC OS, BSD. A través de la generación de un simple archivo ejecutable jar que coloca en cada plataforma mientras tiene cuidado de las particularidades de cada sistema operativo. IzPack es open source (GNU GPL liscence) y muy personalizable.

Para empezar con IzPack, se debe saber cómo IzPack es organizado si se lo quiere utilizar. Primero debemos ir al directorio donde ha sido instalado IzPack en su máquina. Hay tres archivos de texto y se los debe ubicar en los directorios. Los más importantes por el momento son: bin, doc y simple. Doc contiene la documentación, en bin se puede ver que existen los scripts que compilan (unix y Windows) y son leídos al ir al archivo de instalación XML de la línea de comandos o de una herramienta externa.

<sup>23</sup> [java.sun.com/products/javahelp](http://java.sun.com/products/javahelp)

Compilando un sistema, la compilación de un sistema se realiza en forma silenciosa y modular. Indudablemente se puede usar dos formas para compilar: a través de la línea de comandos y Jakarta Ant.”<sup>24</sup>

La compilación toma como primera entrada un archivo de instalación XML que describe la instalación. (Relativamente en alto nivel). Este archivo contiene información detallada tal como el nombre de la aplicación, los autores, los archivos a instalar, el panel de uso, cuales archivos fuentes se van a cargar y mucho más. El compilador puede generar diferentes clases de compiladores, pero esta información no está localizada dentro del archivo XML. Por lo que son parámetros de la compilación.

Las opciones de compilación para la línea de comandos son los siguientes:

- -?: Muestra la lista de opciones disponibles.
- -b: Especifica el path base, si el path está especificado en el archive XML la cadena debe ser -b””.
- -k: Especifica la clase de instalación.
- -o: Especifica el nombre del archivo Jar instalador resultante.

### ARQUITECTURA DEL COMPILADOR

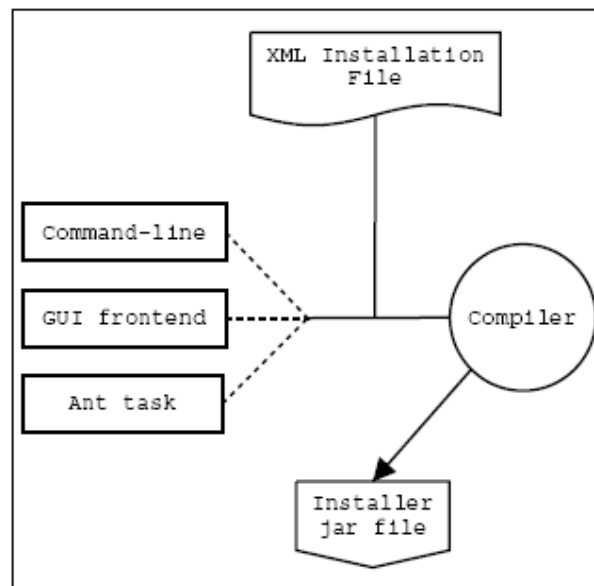


Gráfico 5-8. Arquitectura del Compilador IzPack

<sup>24</sup> [www.izforge.com/izpack](http://www.izforge.com/izpack)

## **5.3 Capítulo 5. Bases de Datos**

### **5.3.1 Modelos de datos.**

Para introducirnos en este tema, empezaremos definiendo que es un modelo.

Modelo: es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación la elaboramos de forma gráfica.

¿Qué es modelo de datos? Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

#### **5.3.1.1 Modelos lógicos basados en objetos.**

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

##### **5.3.1.1.1 Modelo Entidad-Relación.**

El modelo E-R se basa en una percepción del mundo real, la cual está formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos.

Una entidad es un objeto que existe y se distingue de otros objetos de acuerdo a sus características llamadas atributos. Las entidades pueden ser concretas como una persona o abstractas como una fecha. Por ejemplo: un alumno se distingue de otro por sus características particulares como lo es el nombre, o el número de control asignado al entrar a una institución educativa, así mismo, un empleado, una materia, etc.

Las entidades pueden ser de dos tipos:

- Tangibles, son todos aquellos objetos físicos que podemos ver, tocar o sentir.
- Intangibles, todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Las características de las entidades en base de datos se llaman atributos, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno; Clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de relaciones.

La relación entre ambas entidades la podemos establecer como Venta.

Una entidad se caracteriza y distingue de otra por los atributos, en ocasiones llamadas propiedades, que representan las características de una entidad. Los atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como dominio del atributo. Así cada entidad se describe por medio de un conjunto de parejas formadas por el atributo y el valor de dato. Habrá una pareja para cada atributo del conjunto de entidades.

#### **5.3.1.1.1 Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.**

El estándar SQL:1999 es el resultado de años de esfuerzos para plasmar conceptos Orientados a Objetos en la tecnología de las Bases de Datos Relacionales, dando nacimiento a los Sistemas de Base de Datos Objeto-Relacionales (ORDBMS). La principal motivación para el desarrollo del estándar fue juntar lo mejor de dos mundos tecnológicos: por un lado la capacidad de la orientación a objetos para desarrollar aplicaciones que requieren de modelos complejos, que permita la construcción de

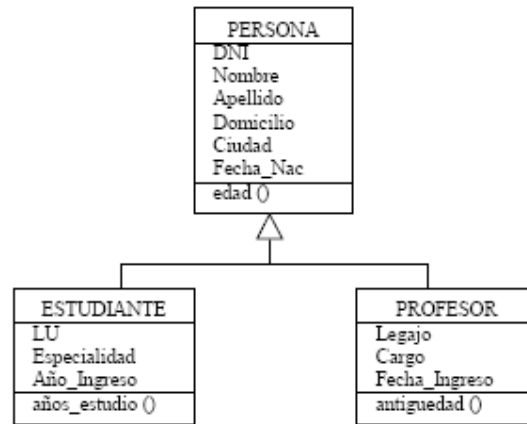
grandes sistemas empleando componentes preconfigurados, fáciles de mantener, rehusar y extender, y por el otro una tecnología de aceptación universal, con una gran capacidad para administrar persistencia, independencia de datos, concurrencia, distintos niveles de seguridad, con un lenguaje simple y muy difundido como SQL.

Los ORDBMS tienen la capacidad de realizar transformaciones basadas en la tecnología relacional, empleando mapeos bien conocidos y determinados del modelo de datos en tablas. Las tablas generadas deben estar normalizadas para eliminar redundancia y anomalías en los datos. Además, los ORDBMS tienen la capacidad de emplear relaciones complejas orientadas a objetos, en donde no existe una manera clara de cómo transformar un modelo OO en un ORDBMS de manera eficiente, ni se cuenta con una base teórica como la normalización en el caso relacional. Entre estos dos extremos, relacional y objeto-relacional, se pueden proponer transformaciones híbridas del modelo de datos orientado a objetos empleando elementos de ambos conjuntos.

Partiendo de un modelo de datos desarrollado en UML se proponen estrategias de diseño para Bases de Datos Objeto-Relacionales, sobre algunos aspectos involucrados en la transformación de estos modelos, tales como flexibilidad, dificultad para la implementación, incorporación de caminos de acceso y restricciones, navegabilidad.

#### **5.3.1.1.2 Relaciones de Generalización-Especialización.**

En una relación de generalización-especialización existe una jerarquía de tipos en la que se definen sucesivos niveles de subtipos que se especializan de manera incremental, heredando los atributos y el comportamiento de un ancestro común denominado supertipo, extendiendo su definición agregando nuevos atributos y métodos, o redefiniendo los métodos heredados de sus ancestros. Esta jerarquía de tipos provee un alto nivel de complejidad para un modelo determinado.



**Gráfico 5-9 Relación de Generalización-Especialización.**<sup>25</sup>

En el ejemplo del gráfico 6-14 se asume que los conjuntos de Estudiante y Profesor son disjuntos, y que en el universo de información que se quiere representar existen objetos de los tres tipos: personas, estudiantes y profesores, para contrastarlo con la posibilidad de que sólo existan profesores y estudiantes.

### 5.3.1.1.3 Transformación del Modelo a un Esquema Relacional.

Existen tres transformaciones posibles para la jerarquía de generalización-especialización propuesta:

- Modelo Plano.
- Partición Vertical.
- Partición Horizontal.

El modelo plano contempla la definición de una sola tabla donde se encontrarán todos los atributos de los distintos tipos. Aquellos atributos que no correspondan al tipo almacenado en una fila tendrán valor nulo.

DNI	Nombre	Apellido	Domicilio	Ciudad	Fecha_Nac	LU	Espec	Año_Ingreso	Legajo	Cargo	Fecha_Ingreso
1234	Juan	Garcia	abc 12	XYZ	11/11/74	212	ISI	1999	null	null	null
6789	José	Perez	lmn 56	LPJ	9/4/60	null	null	null	994	JTP	1/4/1990
4567	Luis	Lopez	zjk 34	KLO	24/10/65	null	null	null	null	null	null

**Gráfico 5-10 Esquema Relacional Resultante de la Transformación como Tabla Jerárquica.**<sup>26</sup>

<sup>25</sup> Implementación de Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.pdf

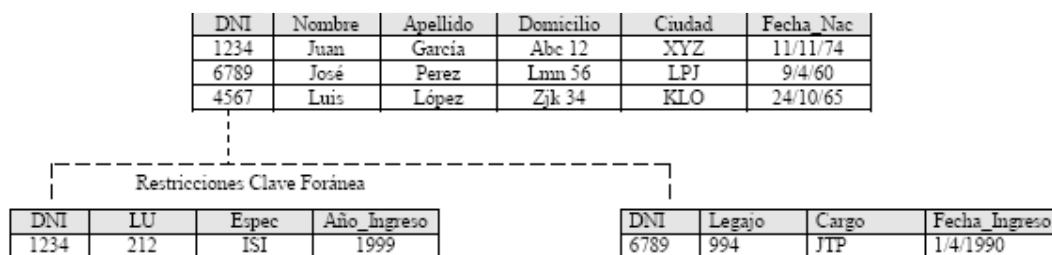
<sup>26</sup> Implementación de Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.pdf



La implementación plana tiene la dificultad del manejo de valores nulos, pero que los ORDBMS administran hoy con mucha eficacia. La implementación relacional, en la que se definen explícitamente los atributos de todos los tipos de la jerarquía, permite la definición de índices y/o restricciones adicionales para cada uno de ellos. Esto no es posible en la implementación objeto-relacional, ya que no se pueden definir restricciones en Persona\_table para los atributos que no pertenecen al tipo para el cual se define la tabla.

Para hacer persistentes objetos de tipos nuevos en Persona\_table basta con definir nuevos tipos en la jerarquía. Esto da mucha flexibilidad para cambiar el modelo y la jerarquía de Generalización-Especialización y la inclusión de nuevas relaciones en el universo que se modela. Para obtener el mismo efecto en una implementación relacional, se debe recurrir a un modo menos natural y flexible como cláusulas ALTER TABLE...ADD, y el agregado de restricciones y relaciones de claves foráneas con otras tablas. Una gran ventaja de la implementación objeto-relacional es la posibilidad de realizar operaciones y cálculos con los objetos de la jerarquía, por medio de sus funciones miembros. Por ejemplo, es muy fácil calcular la edad de los objetos de Persona\_table, ya que todos los subtipos heredan la función edad() definida en Persona\_typ, que calcula la edad de las personas a partir de la fecha de nacimiento. Además, es posible definir métodos que se pueden redefinir en los subtipos para dar lugar a una de las propiedades de la orientación a objetos: el polimorfismo. Esto no es posible siquiera plantearlo en el modelo relacional.

La partición vertical implica que se generará una tabla para cada uno de los tipos que componen la jerarquía, cada una con sus atributos. La clave del supertipo se replica como clave foránea en cada uno de los subtipos.



**Gráfico 5-11 Esquema Relacional Resultante de la Transformación Usando Partición Vertical.**<sup>27</sup>

En este modelo es posible analizar tres tipos de implementaciones: relacional, híbrida y objeto-relacional. La implementación relacional tiene la dificultad de tener que realizar operaciones de join cuando se quiere recuperar la información completa de un estudiante y/o profesor. Esto, dependiendo del tamaño de las tablas y la distribución de las filas en el almacenamiento, puede ser muy costoso. Además tiene definido el mecanismo de integridad referencial para administrar la integridad de las referencias manejadas por medio de las claves foráneas. El esquema híbrido tiene una gran flexibilidad al momento de navegar por los datos. Definido de modo bi-direccional, es posible, por ejemplo acceder a un estudiante por medio de su documento de identidad, o su número de libreta, y siguiendo las respectivas referencias recuperar el resto de la información accediendo a la tabla de estudiante y luego a sus datos personales. En este esquema no existe un mecanismo que mantenga la integridad de las referencias como en el caso de las claves foráneas del modelo relacional. Esto implica que si el objeto apuntado por una variable REF se elimina, no hay nada definido por el estándar que diga qué hacer ante este caso. Esta implementación permite el manejo de funciones miembros, aunque no permite la posibilidad de herencia de atributos ni métodos, ni de polimorfismo, que si los tiene la relación de generalización especialización definida por el estándar SQL: 1999.

La implementación objeto-relacional es la que presenta mayores ventajas porque puede cambiar flexiblemente las relaciones, hereda atributos y métodos, puede redefinir los métodos en los subtipos, puede definir índices para varios puntos de acceso a los datos, tiene la capacidad de definir restricciones sobre todos los atributos.

La partición horizontal implementa sólo las tablas correspondientes a los subtipos, trasladando a las mismas todos los atributos del supertipo. El inconveniente principal de

<sup>27</sup> Implementación de Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.pdf

esta transformación es que sólo se pueden representar dos tipos de objetos: estudiantes ó profesores. Los objetos persona no pueden ser representados en este esquema.

DNI	Nombre	Apellido	Domicilio	Ciudad	Fecha_Nac	LU	Espec	Año_Ingreso
1234	Juan	Garcia	abc 12	XYZ	11/11/74	212	ISI	1999

DNI	Nombre	Apellido	Domicilio	Ciudad	Fecha_Nac	Legajo	Cargo	Fecha_Ingreso
6789	José	Perez	lmn 56	LPJ	9/4/60	994	JTP	1/4/1990

**Gráfico 5-12 Esquema Relacional Resultante de la Transformación Usando Partición Horizontal.**<sup>28</sup>

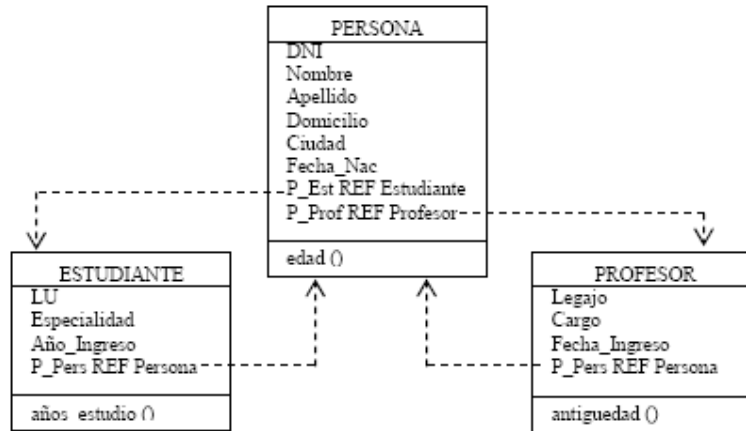
Esta partición es conveniente cuando en el universo a representar sólo existen objetos que pertenecen a los subtipos, por lo tanto no existen objetos a nivel de supertipos y no es necesario mantener una tabla para ellos. Comparando la implementación relacional con la objeto-relacional, podemos decir que la mayor diferencia entre ambas es el manejo de comportamiento entre una y otra. La definición de las operaciones sobre los tipos que no es posible hacer en una implementación relacional, sí se pueden hacer en una objeto-relacional, con todas las ventajas que esto propone. Esta implementación aventaja a la relacional en las características de flexibilidad, herencia, polimorfismo.

Son equivalentes en aspectos relacionados con el acceso a los datos y restricciones de integridad para los atributos.

#### **5.3.1.1.1.4 Transformación del Modelo a un Esquema Híbrido.**

Este esquema está más volcado a la tecnología Objeto-Relacional, porque aprovecha elementos incorporados por el estándar SQL: 1999, la definición de UDT's y la capacidad de definir referencias (punteros) entre objetos. El esquema de esta representación se muestra a continuación:

<sup>28</sup> Implementación de Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.pdf



**Gráfico 5-13 Esquema Relacional Resultante de la Transformación Usando Partición Horizontal<sup>29</sup>**

En el gráfico 6-18 las flechas con líneas de punto significa que se han definido referencias (punteros) entre los objetos. Estas referencias emplean los Identificadores de Objetos Únicos (OID) para el manejo de estos punteros.

### 5.3.2 Independencia de los Datos

Se refiere a la protección contra los programas de aplicación que puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos. Existen 2 niveles de independencia de datos:

- Independencia física de datos: es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
- Independencia lógica de datos: capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

### 5.3.3 Lenguaje de Definición de Datos

“El lenguaje de definición de datos, denominado por sus siglas como: DDL (Data definition Language). Permite definir un esquema de base de datos por medio de una

<sup>29</sup> Implementación de Modelos de Generalización-Especialización en Bases de Datos Objeto-Relacionales.pdf

serie de definiciones que se expresan en un lenguaje especial, el resultado de estas definiciones se almacena en un archivo especial llamado diccionario de datos.”<sup>30</sup>

#### **5.3.4 Lenguaje de Manipulación de Datos**

“La manipulación de datos se refiere a las operaciones de insertar, recuperar, eliminar o modificar datos; dichas operaciones son realizadas a través del lenguaje de manipulación de datos (DML, Data Manipulation Language), que es quién permite el acceso de los usuarios a los datos.”<sup>31</sup>

Existen básicamente 2 tipos de lenguajes de manipulación de datos:

- Procedimentales: los LMD requieren que el usuario especifique que datos se necesitan y cómo obtenerlos.
- No procedimentales: los LMD requieren que el usuario especifique que datos se necesitan y sin especificar cómo obtenerlos.

#### **5.3.5 Manejador de Bases de Datos.**

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos.

---

<sup>30</sup> [www.abcdatos.com/tutoriales/programacion/basesdedatos](http://www.abcdatos.com/tutoriales/programacion/basesdedatos)

<sup>31</sup> [www.abcdatos.com/tutoriales/programacion/basesdedatos](http://www.abcdatos.com/tutoriales/programacion/basesdedatos)

Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.

Respaldo y recuperación, consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.

Control de concurrencia, consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.

Seguridad e integridad, consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

El DBMS es conocido también como Gestor de Base de datos.

El DBMS es la interface entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal. En sí, un sistema manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

## **5.3.6 SQL**

### **5.3.6.1 Introducción.**

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos. Pero como sucede con cualquier sistema de normalización hay excepciones para casi todo, de hecho, cada motor de bases de datos tiene sus peculiaridades y lo hace diferente de otro motor, por lo tanto, el lenguaje SQL normalizado (ANSI) no nos servirá para resolver

todos los problemas, aunque si se puede asegurar que cualquier sentencia escrita en ANSI será interpretable por cualquier motor de datos.

### 5.3.6.2 Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Comandos. Existen dos tipos de comandos SQL:

- DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DDL.

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

**Tabla 1. Comandos DDL**

Comandos DML

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

**Tabla 2. Comandos DML**

### Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

**Tabla 3. Cláusulas**

### Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

**Tabla 4. Operadores Lógicos**

### Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual que
>=	Mayor o igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.



LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

**Tabla 5. Operadores de Comparación**

#### Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

**Tabla 6. Funciones**

## 5.4 Redes.

### 5.4.1 Posibles Tecnologías a Utilizar.

#### 5.4.1.1 Gigabit Ethernet Sobre Cobre.

Red de alta velocidad, las aplicaciones de negocios funcionan con mayor rapidez. Información e imágenes se comparten más rápidamente y el acceso a información se mejora considerablemente.

El Gigabit Ethernet sobre cobre es una tecnología de redes que conecta a computadoras con conexiones de alta velocidad, que funcionan sobre el mismo tipo de cables comunes (de cobre) que las redes de Fast Ethernet; cables de Categoría UTP-5, o superior. El Gigabit Ethernet está también disponible sobre cables de fibra, pero es más caro que el Gigabit Ethernet sobre cobre.

##### 5.4.1.1.1 Cómo funciona Gigabit Ethernet.

El Gigabit Ethernet sobre cobre funciona sobre cables de cobre. La razón por la cual esto es significativo es que los cables de cobre representan el tipo de cables más

comunes en la mayoría de las compañías. De otra forma, se requeriría que los clientes instalaran cables de fibra para poder lograr la velocidad gigabit (1000 Mbps). La aparición del Gigabit Ethernet sobre cobre ha eliminado las barreras de costos y ha cambiado los requerimientos de cableado, es nuestro deber reducir costos de implantación y por tanto utilizar el cableado de cobre UPT-5E que ya se halla extendido.

Una solución Gigabit confiable está compuesta de un switch y tarjetas interfaz para la red (NICs) Gigabit. Muchas veces, los servidores crean cuellos de botella en la red porque sólo se usan tarjetas de 10/100 Mbps en los servidores. En este caso, aunque haya un switch Gigabit en el centro, el límite de 10/100 Mbps tiene el efecto de no permitir que los usuarios gocen del máximo de beneficios de la red troncal Gigabit, por tanto como ya se mencionó se tendrá cuidado de instalar en los servidores NICs tipo 10/100/1000 Mbps

No obstante, la instalación de tarjetas 1000BASE-T en los servidores existentes impulsaría el rendimiento de la red, ahorrándole a la compañía el costo de tener que comprar nuevos servidores. A largo plazo, las inversiones en Gigabit Ethernet sobre cobre son extremadamente positivas para los clientes.

El Gigabit Ethernet puede extenderse a las computadoras de escritorio, permitiendo que los usuarios con mayores necesidades de poder obtengan el ancho de banda necesario para soportar las aplicaciones de negocios más exigentes, tales como las de soporte a transacciones de gestión académica.

Se optará por el tipo de tecnología de red, **GIGABIT ETHERNET SOBRE COBRE**, dado que los dispositivos e interfaces para su funcionamiento poseen las siguientes características.

- Simples de comprar,
- Simples de instalar,
- Simples de usar,
- Simples de administrar o gestionar,
- Simples de actualizar, y
- Simples de integrar en redes existentes y con aplicaciones existentes, con proyección futurista, permitiendo así posterior desarrollo.

#### **5.4.2 Cables de Fibra Óptica.**

No es viable en nuestro caso en particular ya que la construcción de una red troncal de fibra óptica representa una inversión grande requiriendo además de switches y cables especiales que por ahora y en el corto plazo no se consideran indispensables, dado el nivel de tráfico soportado y el número de usuarios con los cuales se cuenta.

#### **5.4.3 Direcciones IP Distribución De Subredes Y Hosts.**

Una dirección IP se base en el Protocolo Internet. Cada LAN debe tener su propia dirección IP exclusiva, ya que la dirección IP es fundamental para que se produzca la Internetworking en las WAN.

En un entorno de red IP, las estaciones terminales se comunican con otros servidores u otras estaciones terminales. Esto sucede porque cada nodo tiene una dirección IP, que es una dirección lógica única de 32 bits. Las direcciones IP existen en la capa 3, la capa de red, del modelo de referencia OSI. Estas direcciones son normalmente jerárquicas. Cada red de empresa tiene una dirección y los hosts que residen en la red comparten la misma dirección de red, pero cada host se identifica mediante una dirección única en la red.

Como ya se mencionó, la dirección IP es un valor de 32 bits escrito en forma de cuatro octetos. Esto significa que existen cuatro grupos, cada uno de los cuales contiene ocho números binarios compuestos por unos y ceros. Cada dirección IP consta de dos partes: el número de red y el número de host. El número de red identifica la red de la que forma parte el dispositivo. El número de host identifica la conexión del dispositivo a esa red.

IANA, Instituto encargado de asignación de direcciones IP, reserva las direcciones de clase A, para entidades gubernamentales cuyos rangos permitidos van desde 1 a 126, las direcciones IP de clase B para empresas medianas cuyos rangos permitidos van desde 128 a 191 y las direcciones IP de clase C para todos los demás cuyos rangos permitidos van desde 192 a 223. Clase D, reservada para multicast cuyos rangos permitidos van

desde 224 a 239 y clase E, para investigación cuyos rangos permitidos van desde 240 a 254.

Son direcciones IP reservadas, además cualquier dirección IP que tenga sólo números cero o sólo números uno en la parte de la dirección que se refiere al host. Las subredes son una extensión del número de red. Los administradores de la red deciden el tamaño de las subredes basándose en las necesidades de la organización y su crecimiento.

Una dirección de subred incluye un número de red, un número de subred dentro de la red y un número de host dentro de la subred. Al proporcionar este tercer nivel de direccionamiento, las subredes ofrecen flexibilidad adicional al administrador de la red. Las subredes se ocultan de las redes exteriores mediante el uso de máscaras denominadas máscaras de subred. Una máscara de subred tiene como propósito indicar a los dispositivos la parte de una dirección que corresponde al número de red (incluyendo la subred), y la parte que corresponde al host. Estas máscaras utilizan el mismo formato que el direccionamiento IP. 32 Bits divididos en cuatro octetos.

A su vez, las direcciones IP pueden ser:

- Direcciones IP estáticas (fijas). Un host que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de Internet con objeto de que estén siempre localizables por los usuarios de Internet. Estas direcciones hay que contratarlas.
- Direcciones IP dinámicas. Un host que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a Internet mediante un módem. Los proveedores de Internet utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez).

Y se clasifican en:

- Direcciones IP públicas. Son visibles en todo Internet. Un ordenador con una IP pública es accesible (visible) desde cualquier otro ordenador conectado a Internet. Para conectarse a Internet es necesario tener una dirección IP pública.
- Direcciones IP privadas (reservadas).

#### **5.4.4 Direcciones IP Privadas.**

Las redes privadas de organizaciones que no están directamente conectadas a Internet; esto es, las redes que se conectan por medio de un Proxy o un router a una única línea con una sola dirección IP dada por un proveedor de servicios, tienen asignado unos rangos de direcciones IP para su funcionamiento interno. Estos son: clase A cuyos rangos son 10.0.0.0 hasta 10.255.255.255; clase B cuyos rangos son 172.16.0.0 hasta 172.31.0.0 y clase C cuyos rangos son 192.168.0.0 hasta 192.168.255.0

Estas direcciones IP no son utilizadas por los routers para su comunicación con Internet, y se utilizan solo dentro de la organización. Estas redes (Intranet) tienen la ventaja de ser mucho menos accesibles a ataques desde el exterior.

Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utilizan en las empresas para los puestos de trabajo. Los ordenadores con direcciones IP privadas pueden salir a Internet por medio de un router (o Proxy) que tenga una IP pública. Sin embargo, desde Internet no se puede acceder a ordenadores con direcciones IP privadas.

## 6 EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN

En el transcurso de la historia las empresas o instituciones han ido desarrollándose acorde a los avances de la tecnología y el aumento del flujo de información, investigando la mejor forma de mantenerse en el mercado, por tal razón el desarrollo de proyectos de sistemas computacionales es una parte fundamental para toda empresa o institución que desea obtener éxito en las áreas que manejan gran cantidad de información.

La nueva reforma curricular de la Educación Básica en el País, acarrea consigo por un lado la integración de antiguas estructuras desvinculadas entre sí, como lo son Jardín, Escuela y Colegio, para proporcionar diez años de educación básica; y, por otro lado trae consigo el incremento de servicios y actividades institucionales tanto en el área académica como administrativa.

La Unidad Educativa anexa a la Universidad Nacional de Loja está integrada por los siguientes estamentos:

Jardín de Infantes	José Alejo Palacios.
Escuela	Pedro Víctor Falconí.
Colegio	Manuel Cabrera Lozano.

Por disposición del Honorable Consejo Universitario, a los 21 días del mes de octubre de 1978, se decide la creación y puesta en marcha del Jardín de Infantes José Alejo Palacios de la Universidad Nacional de Loja anexa a la Escuela de Psicología de la Facultad de Ciencias de la Educación, como respuesta a la necesidad pedagógica de experimentación para los alumnos en la especialidad de psicología infantil y educación parvularia y a su vez para atender a niños de lugares cercanos e incluso a hijos de maestros y empleados que laboran en la Universidad.

El Jardín de Infantes Pedro Alejo Palacios inició sus actividades en las instalaciones de la Asociación Obreros “Primero de Mayo”. Contando con una sola aula y cincuenta niños, además se crea una extensión en el barrio “Motupe”, en las instalaciones del Colegio “Manuel Cabrera Lozano” en el año de 1980, contando con tres aulas y tres

maestras. Actualmente sus actividades se realizan en la Ciudad Universitaria contando con 9 profesores de aula y tres especiales. Además de la extensión en “Motupe”.

El Colegio Experimental Universitario “Manuel Cabrera Lozano”, se creó mediante decreto Ministerial N° 95 del 29 de Febrero de 1972, posteriormente en el año 1977, se crea la extensión de “Motupe”. Su estructura actual cuenta con 64 profesores, 7 administrativos y 476 estudiantes distribuidos en el Nivel Básico; y, el Bachillerato. Ofrece bachilleratos en Ciencias, con las especializaciones de Físico-Matemáticas, Químico-Biológicas y Sociales; además el Bachillerato Técnico en Comercio y Administración, especialización Contabilidad. Para su funcionamiento dispone de la infraestructura universitaria como, ambientes administrativos, aulas, laboratorios, talleres, biblioteca, centros de cómputo, entre otros.

El colegio al ser un organismo anexo a la Universidad Nacional de Loja, cumple dos funciones fundamentales: primero servir de laboratorio y centro de práctica para los futuros docentes que egresan del Área de Educación, Arte y Comunicación; y, segundo brindar servicios educativos a la juventud lojana que proceden de hogares con bajos recursos económicos.

Es evidente que la actual estructura Académico-Administrativa de la Unidad Educativa anexa a la Universidad Nacional de Loja, no se encuentra integrada en sus tres estamentos que la conforman, rompiéndose el nexo que debe existir entre los mismos; el cual, según la nueva reforma curricular de la educación básica en el país, contempla desde el primer año (Jardín de Infantes), hasta el décimo año de educación básica (Tercer Curso de Colegio). Y el nivel de Bachillerato conformado por primero (cuarto curso) hasta tercero de bachillerato (sexto curso).

Cada uno de los estamentos que conforman la Unidad Educativa Anexa a la Universidad nacional de Loja realizan todos los procesos de gestión académica como lo son: inscripciones, matrículas, asistencias, acreditaciones, reportes, certificaciones, etc. En forma manual, la única forma de almacenamiento es un cuaderno donde almacenan las inscripciones y las actas que entregan los profesores cada determinado tiempo de la información final de acreditaciones y asistencias. Para verificar que cada una de las actas entregadas por profesores esté correcta con relación a cálculos de promedios de

acreditaciones y asistencias, es realizando nuevamente cada uno de dichos promedios lo que retrasa totalmente el proceso de entrega de libretas, certificaciones de asistencias, entre otros.

En el Jardín José Alejo Palacios se les ha asignado una computadora la cual es utilizada únicamente para realizar oficios.

En la Escuela Pedro Víctor Falconí se les ha asignado una computadora la cual es utilizada únicamente para realizar oficios, certificaciones en Microsoft Word y para generar las listas de alumnos por paralelo y cuando es período de matrículas tienen una plantilla en Microsoft Excel donde realizan la impresión de la matrícula, cabe indicar que nunca almacenan la información de las matrículas en forma digital y cuando necesitan información de la misma deben recurrir a las carpetas de cada uno de los estudiantes en busca de información.

En el Colegio Manuel Cabrera Lozano poseen tres computadoras, dos en secretarías y una en el departamento de orientación vocacional. De igual forma son usadas para generar listas de alumnos en Microsoft Excel y oficios, certificaciones en Microsoft Word.

En las diferentes secretarías manifestaron que los principales problemas de realizar sus procesos en forma manual son:

- ◆ La dificultad de encontrar la información de un determinado estudiante además de obtenerla en forma oportuna,
- ◆ El retraso en la elaboración de toda clase de reportes como: libretas de calificación, asistencias, certificados.
- ◆ Retraso en la realización de inscripciones y matrículas.
- ◆ Retraso en la entrega de listados de asistencia, libretas de calificación.
- ◆ El manejo del volumen de la información.
- ◆ La poca seguridad que posee la información de estudiantes como los datos de evaluación de los mismos, durante cada uno de los periodos y diferentes años lectivos.



Además cabe señalar que no existe ningún tipo de red dentro de cada uno de los estamentos y tampoco existe ningún tipo de red entre los mismo.

La Universidad Nacional de Loja como ente activo de la sociedad busca contribuir con el arduo trabajo que realizan los docentes y administrativos que laboran en su unidad educativa anexa solucionando de alguna forma parte de los problemas que tienen, como lo son el manejo del flujo de información, generación y obtención de reportes en forma eficiente y oportuna, así mismo mejorando ciertos procesos de gestión académica, y de esta forma cumplir con su compromiso con la juventud lojana.

El presente proyecto tiene como finalidad coadyuvar al proceso de integración de la unidad educativa al proporcionar una herramienta informática de apoyo a la gestión académica que permita manejar el volumen de información, servicios y actividades que se generarán a partir de dicha integración.

## **7 DESARROLLO DE LA PROPUESTA ALTERNATIVA**

### **7.1 Introducción.**

En el mundo de globalización en que nos encontramos todos inmersos es claro que todos las instituciones existentes buscan integrarse y desarrollarse acorde a los avances de la tecnología y el aumento del flujo de información, investigando la mejor forma de mantenerse siempre a la vanguardia del medio que los rodea, por tal razón el desarrollo de proyectos de sistemas computacionales es una parte fundamental para toda institución que desea obtener éxito y realizar todos sus procesos de la forma más rápida y efectiva.

Para llevar a cabo el desarrollo del proyecto denominado “Sistema de Gestión Académica e Implementación de la Red para la Unidad Educativa anexa a la Universidad Nacional de Loja”, es preciso tomar en cuenta un conjunto de hechos presentes en el espacio de nuestra realidad educativa, los sistemas de información no necesitan estar basados en las computadoras, pero con frecuencia lo están. El factor determinante es si un sistema puede ser mejorado incluyendo en él la capacidad del procesamiento por computadora. Si un sistema del tipo manual puede ejecutar su trabajo en forma eficiente y sin error, habrá pocos motivos para utilizar computadoras. Sin embargo, cuando crece el volumen de trabajo los procedimientos aumentan en complejidad, o las actividades llegan a estar más interrelacionadas, lográndose mejoras importantes al introducir la ayuda de un sistema informático.

El alcance de un sistema de información es justamente la organización misma, ya que esta representa su campo de acción. En toda organización se generan flujos de información que se mueven por toda la organización. Estos flujos de datos son agrupados bajo una serie de esquemas y forman los denominados sistemas de información.

Una vez determinado lo que es un sistema se diría que los sistemas de gestión académica se refieren a la organización de las diferentes titulaciones (planes de estudios, docente para cada curso académico, horarios de las asignaturas), de la matrícula, y de la gestión del expediente académico de los alumnos (calificaciones y actas, certificaciones y títulos, etc.).

El Sistema de Gestión Académica es un eslabón más en todo el proceso de automatización de los elementos clave del entorno de docencia, como son materiales didácticos, herramientas de comunicación, etc. El incremento del número de estudiantes y de los órganos que interactúan con el alumno, conduce a la automatización de todo el proceso de gestión de la atención al estudiante y servicios o prestaciones que se le ofrecen.

El proceso de interacción con el estudiante se realiza desde la primera solicitud de información hasta la entrega final del título. Ello se realiza mediante una estructura de módulos creados a partir de la secuencia lógica del proceso: acceso, matrícula, exámenes, evaluación de estudios previos, gestiones varias y títulos. Los planes de estudio actúan como elemento gestor y regulador de todo el proceso. Consiste en un sistema automático de validación que garantiza la correcta ejecución de cada uno de los elementos que integran la gestión académica.

Las principales partes de los Sistemas de gestión académico son:

El proceso de matriculación es una recopilación de datos donde se origina el proceso de gestión académica y mediante el cual el estudiante se incorpora a la Unidad Educativa quedando adscrito a sus programas y adquiere el compromiso de cumplir los estatutos, reglamentos y disposiciones de la misma.

Reportes, que es un recurso a disposición del estudiante, del tutor y de los responsables de la gestión académica. Se trata de una acción básicamente informativa sobre el progreso académico que describe con detalle todas las gestiones iniciadas, mostrando el estado en que se encuentra cada una de ellas (descripción de las asignaturas matriculadas, relación de las gestiones realizadas, estado de las solicitudes iniciadas, descripción de los años académicos, información sobre la titulación progresiva, etc.)

El sistema permite definir criterios de acceso a la información. De éste modo, el administrador conoce el estado de cada una de las gestiones iniciadas, pero el estudiante solamente accede a los datos oficiales.

La Secretaría, es junto con los reportes académicos, la segunda acción clave para la utilización del sistema. Éste módulo permite la gestión transversal del historial de cada estudiante con el detalle de cada uno de los cursos, sin necesidad de revisarlos uno por

uno. A su vez, se trata de un recurso al servicio del administrador para la gestión masiva de peticiones de los estudiantes que permite ordenar, priorizar y resolver cuestiones aplicando diversos criterios (por paralelos, carga horaria o número de casos).

La introducción de calificaciones (evaluación continua y nota de examen), así como la recuperación de la información y presentación de las asignaturas de las que el estudiante debe examinarse.

Las ventajas de la automatización de los Sistemas de Gestión Académica son:

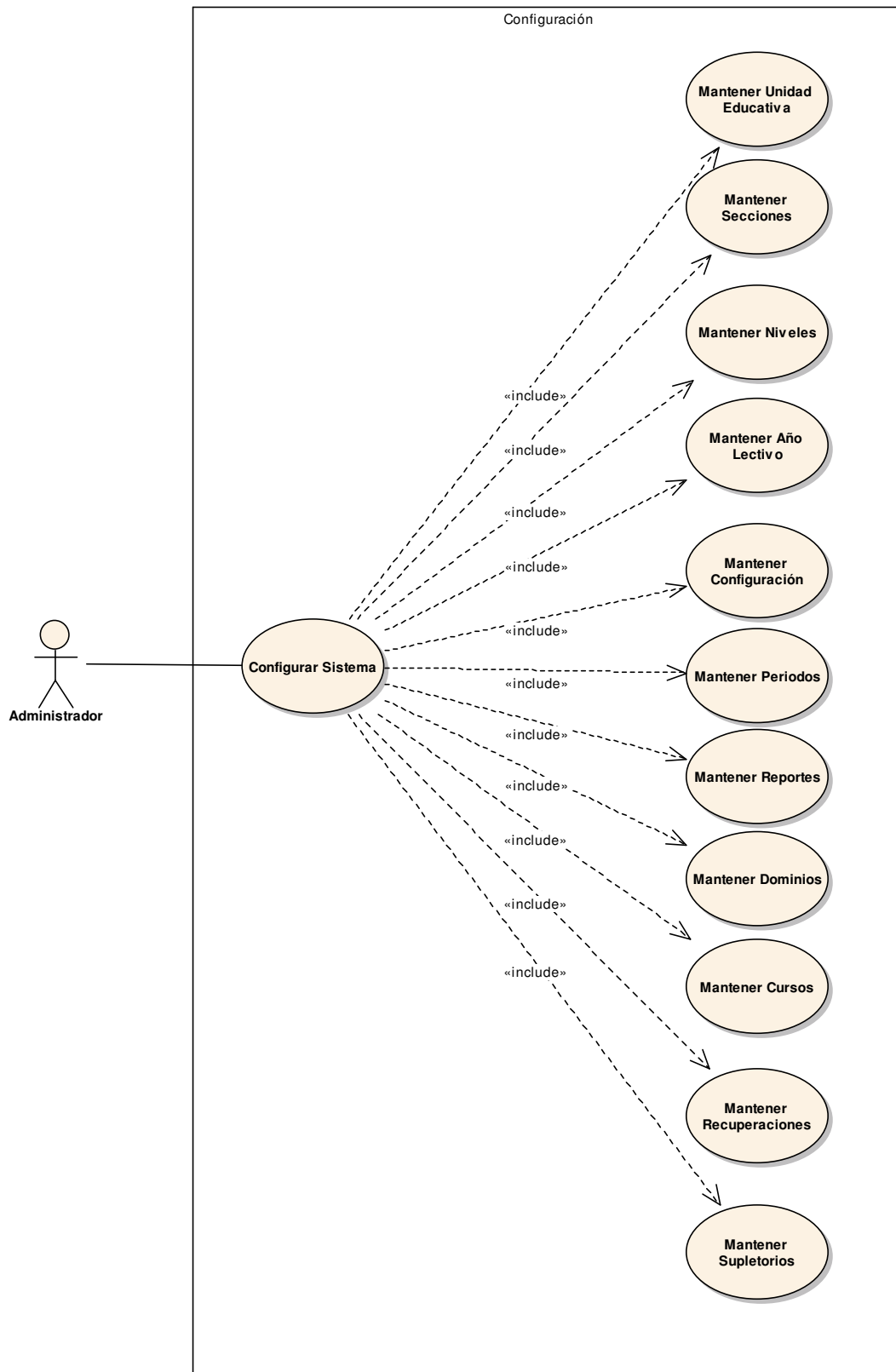
- Permite el desarrollo de nuevas necesidades, reduciendo el coste de la solución final y los plazos de implantación.
- Total adaptación y personalización a los planes de estudio.

Agiliza las tareas del personal administrativo y las del alumno, para cualquier acción relacionada con la gestión académica.

Por las razones antes expuestas se detalla a continuación el desarrollo de la propuesta alternativa según los pasos que indica la metodología descrita en el punto 4.4 (pág. 9)

## 7.2 Análisis.

### 7.2.1 Identificar Casos de Uso.



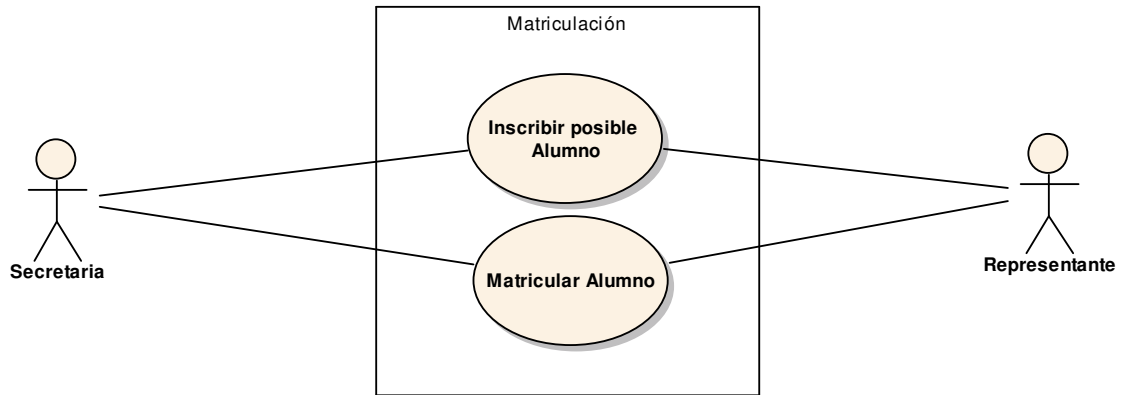
**Diagrama de Casos de Uso 1. DCU01-DCAN01 Configuración****Diagrama de Casos de Uso 2. DCU02-DCAN02 Matriculación**



Diagrama de Casos de Uso 3. DCU03-DCAN03 Estructural

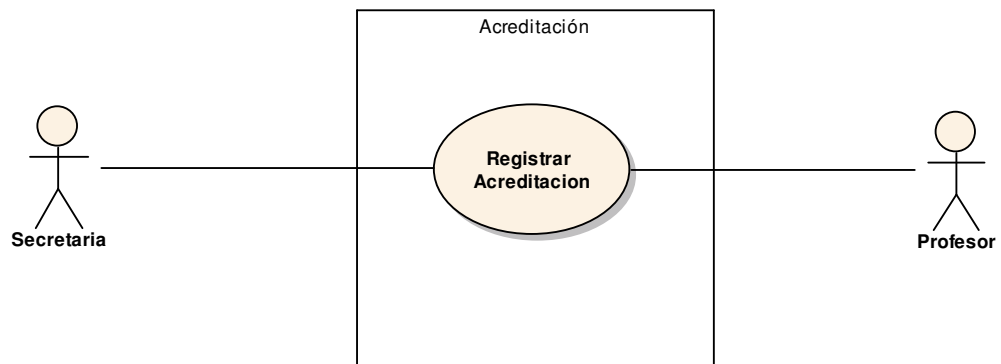


Diagrama de Casos de Uso 4. DCU04-DCAN04 Acreditación

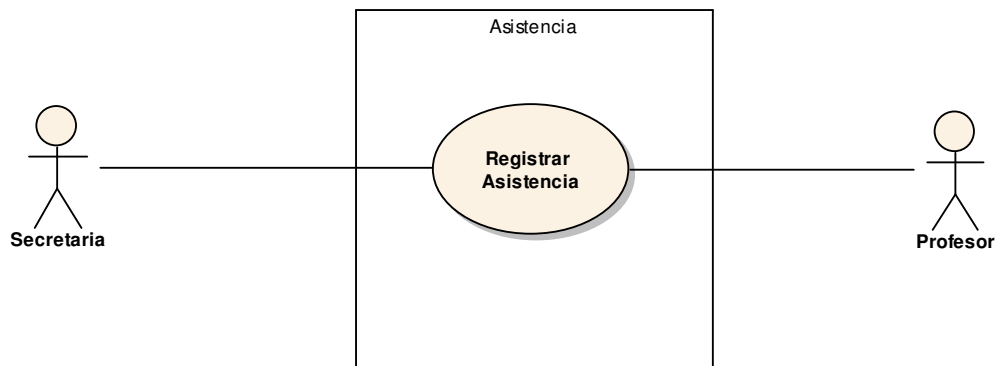


Diagrama de Casos de Uso 5. DCU05-DCAN05 Asistencias

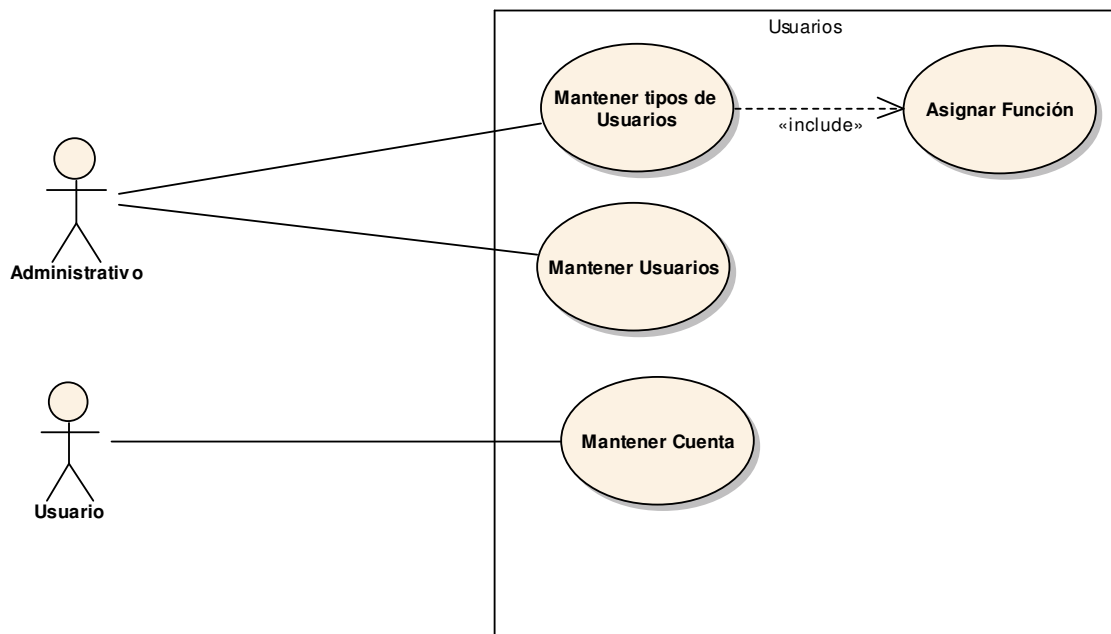
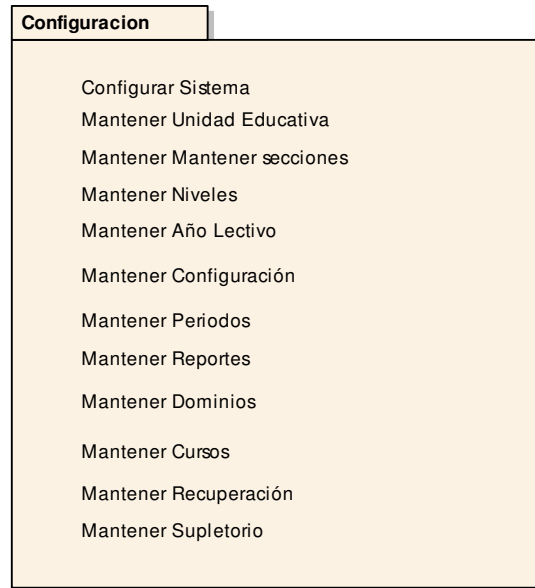


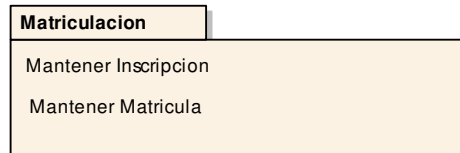
Diagrama de Casos de Uso 6. DCU06-DCAN06 Usuarios



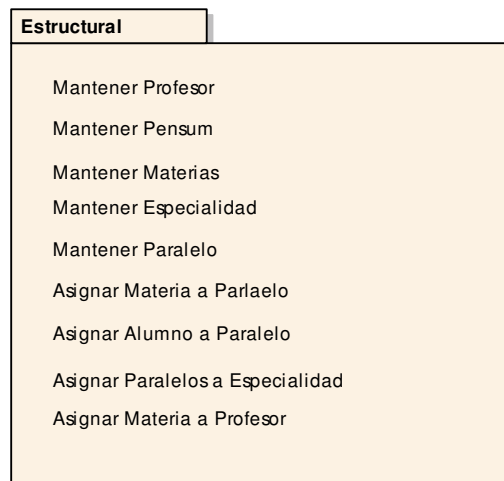
## 7.2.2 Diagramas de Paquetes.



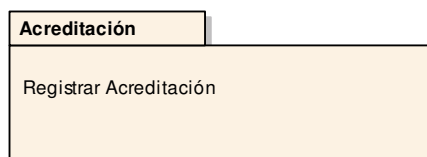
**Diagrama de Paquetes 1. DP01 Configuración del Sistema**



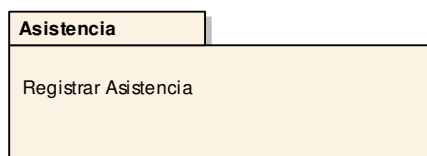
**Diagrama de Paquetes 2. DP02 Matriculación**



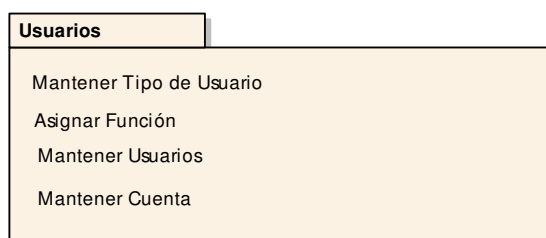
**Diagrama de Paquetes 3. DP03 Estructural**



**Diagrama de Paquetes 4. DP04 Acreditación**



**Diagrama de Paquetes 5. DP05 Asistencia**



**Diagrama de Paquetes 6. DP06 Usuarios**

### 7.2.3 Requerimientos Funcionales de los Casos de Uso.

Se desarrollará un sistema que permita el manejo de forma ordenada de los datos que se generan en una institución educativa, el cual deberá cubrir los siguientes ítems:

<b>CÓDIGO</b>	<b>REQUERIMIENTO</b>
<b>RQF01</b>	Permitir la inscripción de alumnos.
<b>RQF02</b>	Permitir la matriculación de los alumnos en determinado curso
<b>RQF03</b>	Registrar la acreditación de cada uno de los alumnos
<b>RQF04</b>	Registrar la asistencia de cada uno de los alumnos
<b>RQF05</b>	Crear de paralelos
<b>RQF06</b>	Crear cursos
<b>RQF07</b>	Asignar alumnos a un curso y cambio de paralelos
<b>RQF08</b>	Registrar y mantener los datos de cada uno de los profesores
<b>RQF09</b>	Asignar las materias a los profesores
<b>RQF10</b>	Crear niveles de estudios
<b>RQF11</b>	Crear el pensum de estudios para cada uno de los niveles
<b>RQF12</b>	Crear especialidades
<b>RQF13</b>	Asignar materias a cada una de las especialidades
<b>RQF14</b>	Crear materias para cada uno de los cursos en base al pensum existente de cada nivel
<b>RQF15</b>	Crear cursos en base a los niveles existentes
<b>RQF16</b>	Asignar Materias a cada uno de los paralelos

<b>RQF17</b>	Consultar e imprimir reportes de los alumnos
<b>RQF18</b>	Consultar e imprimir reportes de los profesores
<b>RQF19</b>	Crear años lectivos
<b>RQF20</b>	Configurar del sistema de acuerdo a la forma de calificar de cada uno de los niveles
<b>RQF21</b>	Crear periodos
<b>RQF22</b>	Fijar el porcentaje de asistencias con cual se aprueba un año lectivo
<b>RQF23</b>	Definir el número de periodos de cada año lectivo
<b>RQF24</b>	Determinar la calificación máxima por cada año lectivo
<b>RQF25</b>	Determinar la calificación mínima para aprobar un año lectivo
<b>RQF26</b>	Determinar la calificación mínima para aprobar un año lectivo
<b>RQF27</b>	Determinar la calificación de cada uno de los periodos
<b>RQF28</b>	Determinar el número de reportes de cada uno de los periodos
<b>RQF29</b>	Determinar la forma de calificación de los reportes
<b>RQF30</b>	Determinar la calificación de cada uno de los reportes
<b>RQF31</b>	Fijar el nombre de cada uno de los Dominios
<b>RQF32</b>	Determinar el nombre de los periodos
<b>RQF33</b>	Determinar el número de dominios de cada uno de los reportes (parámetros)
<b>RQF34</b>	Configurar supletorios nota mínima, nota a obtener
<b>RQF35</b>	Crear tipos de usuarios, usuarios propiamente, los cuales van a tener determinado acceso a ciertas funciones
<b>RQF36</b>	Registrar y Revisar accesos de los usuarios
<b>RQF37</b>	Mantener los datos de la cuenta de cada uno de los usuarios
<b>RQF38</b>	Crear Secciones
<b>RQF39</b>	Crear Unidades Educativas

Además también se identificaron los requerimientos no funcionales los cuales se listan a continuación:

<b>CÓDIGO</b>	<b>REQUERIMIENTOS NO FUNCIONALES</b>
<b>RQNF01</b>	Ser Consistente.
<b>RQNF02</b>	Ofrecer respuestas específicas.
<b>RQNF03</b>	Pedir confirmación de cualquier acción.
<b>RQNF04</b>	No permitir el ingreso de datos erróneos que no corresponden al tipo definido.
<b>RQNF05</b>	Clasificar las actividades por función y organizar los menús de pantallas de acuerdo a esto.
<b>RQNF06</b>	Proporcionar la ayuda de cualquier pantalla que se encuentre.
<b>RQNF07</b>	Usar frases verbales cortas para nombrar órdenes.
<b>RQNF08</b>	Permitir configuraciones individuales de interfaz de usuario.
<b>RQNF09</b>	Mostrar información de ayuda para el usuario.
<b>RQNF10</b>	Mostrar bitácora del sistema(accesos realizados)
<b>RQNF11</b>	Permitir atención concurrente a consultas.
<b>RQNF12</b>	Brindar tiempos de respuesta aceptables por parte del sistema.
<b>RQNF13</b>	Mostrar etiquetas claras para campos, botones, barras, menús.

### 7.2.4 Glosario de Términos.

<b>TERMINO</b>	<b>CATEGORÍA</b>	<b>COMENTARIO</b>
Asignar Funciones	Caso de Uso	Descripción del Proceso en el cual el administrador asigna funciones a determinado Clase de Usuario
Asignar Materia a Paralelo	Caso de Uso	Proceso por el cual la Secretaria asigna y distribuye las materias entre los diferentes paralelos del curso
Asignar Materia a Profesor	Caso de Uso	Descripción del proceso en el cual La secretaria asigna materias a los diferentes profesores
Asignar Paralelo a Especialidad	Caso de Uso	Descripción del proceso en el cual la Secretaria podrá asignar paralelos a cada una de las Especialidad
Buscar	Caso de Uso	Descripción del proceso en el cual un Usuario busca determinado elemento
Configurar Sistema	Caso de Uso	Descripción del proceso en el cual el Administrador del Sistema lo configura al SGA
Consultar Expediente Estudiantes	Caso de Uso	Descripción del proceso en el cual la Secretaria consulta el expediente del estudiante
Consultar Expediente Profesor	Caso de Uso	Descripción del proceso en el cual la Secretaria consulta el expediente del profesor
Consultar Expediente Curso	Caso de Uso	Descripción del proceso en el cual la Secretaria consulta el expediente del Curso
Controlar Accesos	Caso de Uso	Descripción del Proceso en el cual el administrador controla y consultar el acceso de cada uno de los Usuarios
Imprimir Reporte	Caso de Uso	Descripción del proceso en el cual la Secretaria podrá imprimir determinado reporte que es generado por el Sistema de Gestión Académica
Mantener Año Lectivo	Casos de Uso	Descripción del proceso en el cual se puede crear y configurar el periodo académico del Año Lectivo
Mantener Configuración	Caso de Uso	Descripción del proceso en el cual se registra cada uno de los parámetros de promoción de los alumnos
Mantener Cuenta	Caso de Uso	Descripción del proceso en el cual un Usuario podrá actualizar la cuenta
Mantener Cursos	Casos de uso	Descripción del Proceso en el cual el Administrador puede crear cursos basados en los Niveles de Educación existentes

Mantener Dominios	Caso de Uso	Descripción del proceso en el cual se registran los parámetros del dominio
Mantener Especialidad	Caso de Uso	Descripción del proceso en el cual la Secretaria podrá crear Especialidades para la Unidad Educativa
Mantener Inscripción	Caso de Uso	Descripción del proceso en el cual la Secretaria registra Inscripciones para posibles alumnos
Mantener Materias	Caso de Uso	Proceso en el cual la Secretaria seleccionara las materias para el curso, lo cual lo hace en base al pensum del Nivel al que pertenece el Curso
Mantener Matricula	Caso de Uso	Descripción del proceso en el cual la Secretaria registra Matriculas para Alumnos
Mantener Niveles	Caso de Uso	Descripción del proceso en el cual el Administrador del Sistema podrá crear los diferentes Niveles de Estudio para cada una e las secciones de la Unidad Educativa
Mantener Pensum	Caso de Uso	Descripción del proceso en el cual la Secretaria podrá crear el Pensum de estudio para cada uno de los Niveles de la Unidad Educativa
Mantener Periodos	Caso de Uso	Descripción del proceso en el cual se registran los parámetros para cada uno de los periodos
Mantener Profesor	Caso de Uso	Descripción del proceso en el cual la Secretaria registrara los datos de los Profesores
Mantener Recuperación	Caso de Uso	Descripción del proceso en el cual el administrador configura la recuperación para los periodos
Mantener Reporte	Caso de uso	Descripción del proceso en el cual se registran los parámetros del Reporte
Mantener Secciones	Caso de Uso	Descripción del proceso en el cual el Administrador del Sistema podrá crear las diferentes Secciones con las que cuenta la Unidad Educativa
Mantener Supletorio	Caso de Uso	Descripción del proceso en el cual el administrador configura los supletorios para el periodo lectivo
Mantener Clases de Usuarios	Caso de Uso	Descripción del proceso en el cual el Administrador crea Clases de Usuarios
Mantener Unidad Educativa	Caso de Uso	Descripción del proceso en el cual el Administrador del Sistema crea y edita los datos de la Unidad Educativa
Mantener usuarios	Caso de Uso	Descripción del proceso en el cual el administrador creara Usuarios en base a los Clases de Usuarios

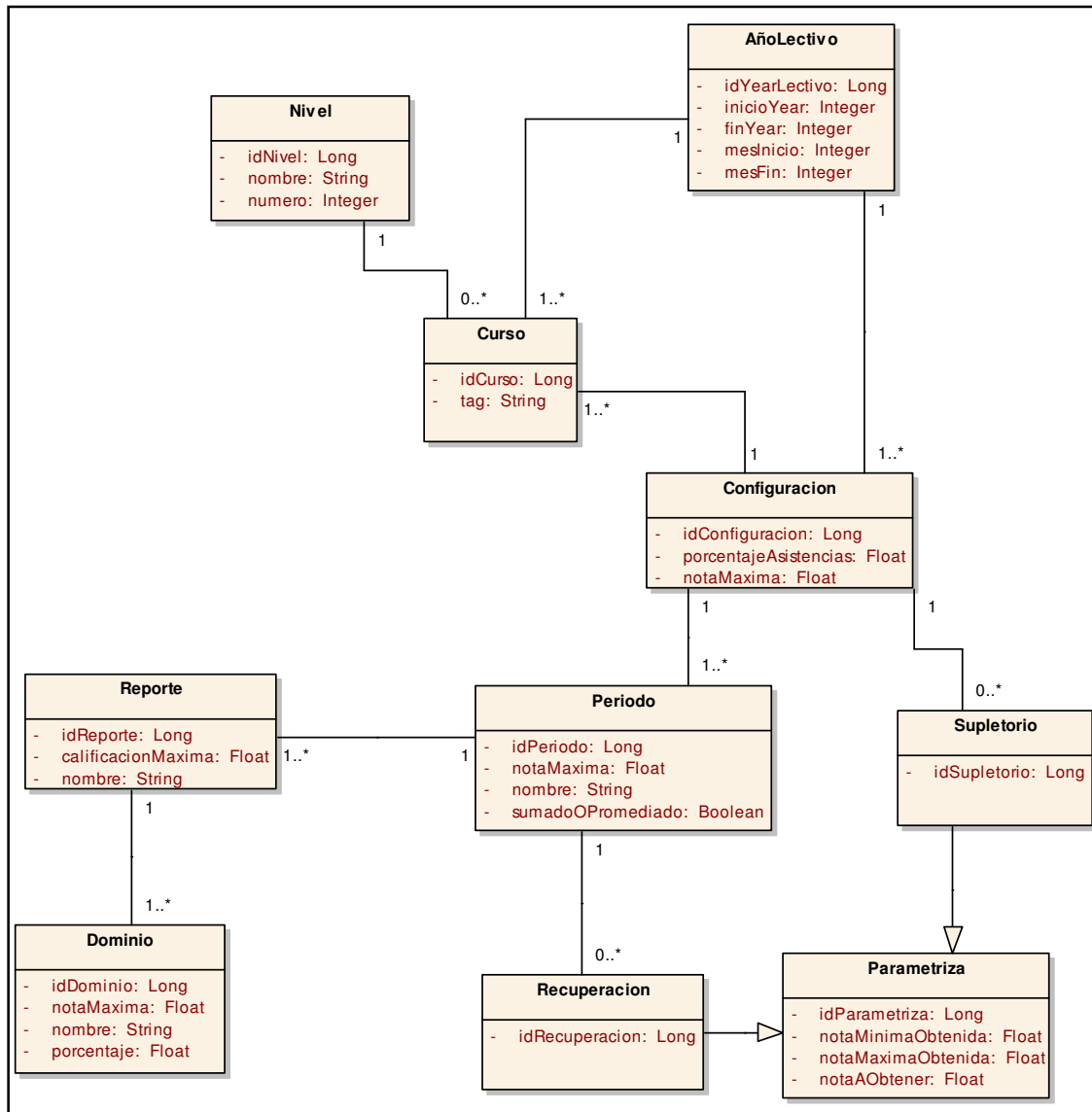
Registrar Acreditación	Caso de Uso	Descripción en el cual la Secretaria registra la acreditación de los alumnos
Registrar Asistencias	Caso de Uso	Descripción del proceso en el cual la Secretaria registra la asistencia de los alumnos
Inscripción	Clase	Proceso mediante el cual un alumno se registra en la Unidad educativa como posible alumno de la misma
Matrícula	Clase	Proceso mediante el cual un alumno se registra en la Unidad educativa como alumno de la misma
Acreditación	Clase	Proceso mediante el cual se asignan las notas obtenidas durante un año lectivo
Promedio	Clase	Proceso mediante el cual se obtiene el promedio según las acreditaciones obtenidas por un alumno
Asistencia	Clase	Proceso mediante el cual se registra la asistencia de un alumno en cada una de las materias recibidas durante un año lectivo
Persona	Clase	Individuo de la especie humana, hombre o mujer
Dirección	Clase	Ubicación actual en la cual reside una persona
Año Lectivo	Clase	Periodo de tiempo durante el cual se realizan las actividades académicas en una determinada Unidad educativa
Unidad educativa	Clase	Persona Jurídica que hace la vez de entidad que brinda enseñanza a la juventud
Nivel	Clase	Escala determinada que va adquiriendo un alumno según su avance académico dentro de una unidad educativa
Sección	Clase	Cada una de las partes en que se divide la Unidad Educativa
Configuración	Clase	Determinación de la forma del comportamiento con el que trabajará el software para cada uno de los diferentes niveles en cada año lectivo
Supletorio	Clase	Calificación adicional cuando un estudiante no ha cumplido con el puntaje mínimo para aprobar un nivel en una determinada materia
Nivel	Clase	Categoría o rango en el que se encuentra un estudiante
Curso	Clase	Nivel en el que encuentra un estudiante en un determinado año lectivo
Paralelo	Clase	Agrupación de diferentes alumnos de un determinado curso

Situación Familiar	Clase	Conjunto de factores del estado en que se encuentra la familia del estudiante
Posible Alumno	Clase	Aspirante a pertenecer a la Unidad Educativa
Status	Clase	Información final a cerca de cada uno de los alumnos en un año lectivo
Alumno	Clase	Estudiante que pertenece a una determinada Unidad Educativa
Representante	Clase	Persona que representa legalmente al alumno ante la Unidad Educativa
Padres	Clase	Personas que tienen a su cargo el cuidado del alumno
Función	Clase	Tareas que puede realizar un determinado Clase de usuario
Clase Usuario	Clase	Agrupación de usuarios que cumple determinadas características
Usuario	Clase	Persona que tiene acceso al sistema
Acceso	Clase	Entrada y salida de un usuario al sistema
Profesor	Clase	Persona encargada de impartir los conocimientos a los alumnos
Calificación	Clase	Acreditación de una materia dictada por determinado profesor
Dominio	Clase	Parámetro en el cual se basan cada uno de los reportes
Calificación Configuración	Clase	Calificación obtenida por un alumno en el período académico
Calificación de Dominio	Clase	Calificación obtenida por un alumno en un determinado dominio
Calificación por Periodo	Clase	Calificación obtenida por un alumno en un determinado periodo
Calificación por Reporte	Clase	Calificación obtenida por un alumno en un determinado reporte
Calificación de Recuperación	Clase	Calificación obtenida por un alumno en la recuperación
Calificación de Supletorio	Clase	Calificación obtenida por un alumno en el supletorio
Pensum	Clase	Conjunto de materias que se deben dictar en un determinado nivel
Materia	Clase	Conocimientos relacionados a un tópico específico bajo una sola denominación para determinado curso
Especialidad	Clase	Rama de una ciencia cuyo objeto es una parte limitada de ella
Recuperación	Clase	Calificación adicional cuando un estudiante no ha cumplido con el puntaje mínimo para aprobar un período en una determinada materia y el cual no debe ser el último

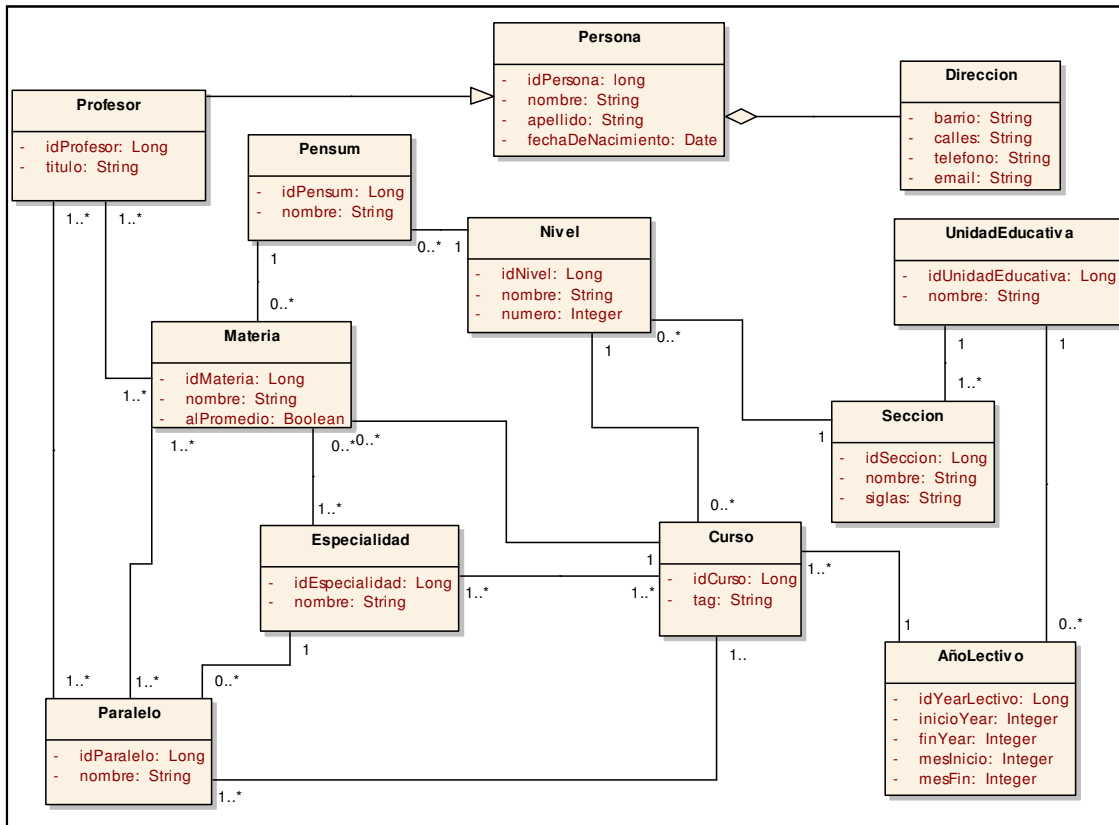


Reporte	Clase	Estructura que compone el período
Periodo	Clase	Estructura que compone el año lectivo de un determinado nivel
Parametriza	Clase	Notas mínimas y máximas a obtener en un supletorio o recuperación

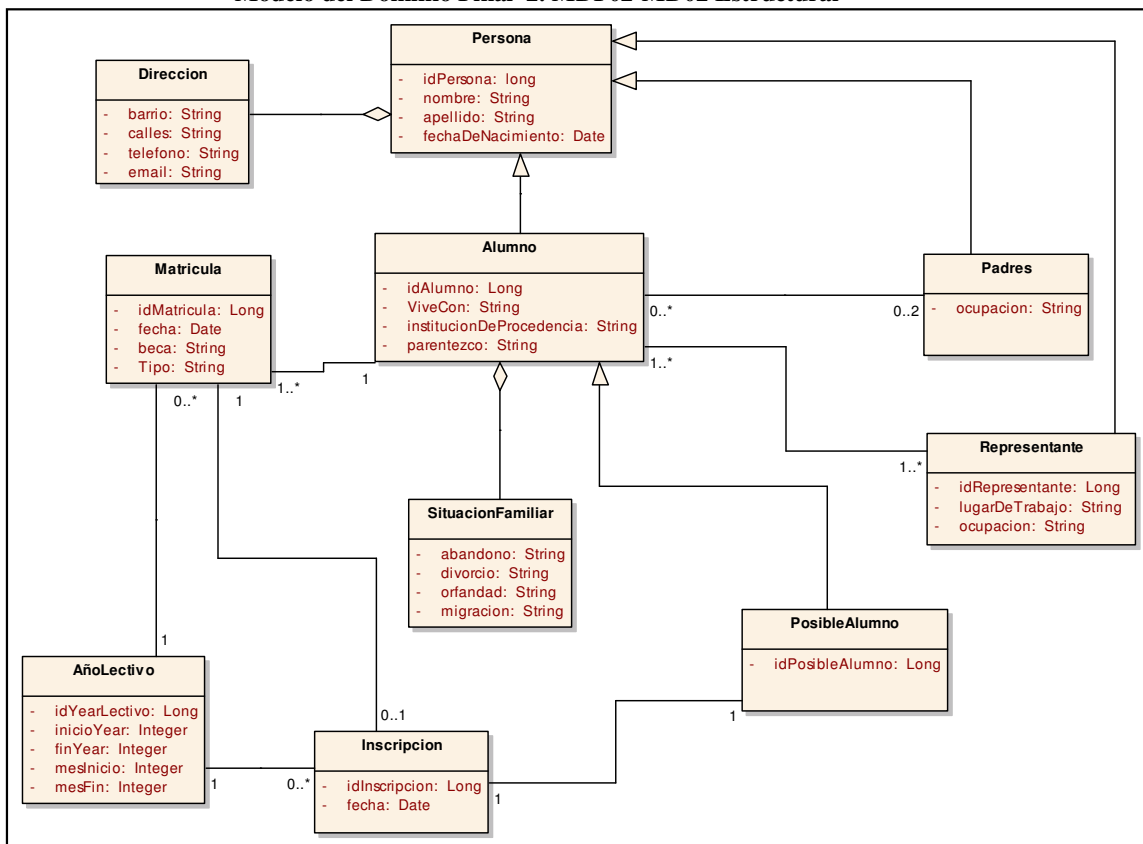
## 7.2.5 Modelo del Dominio.



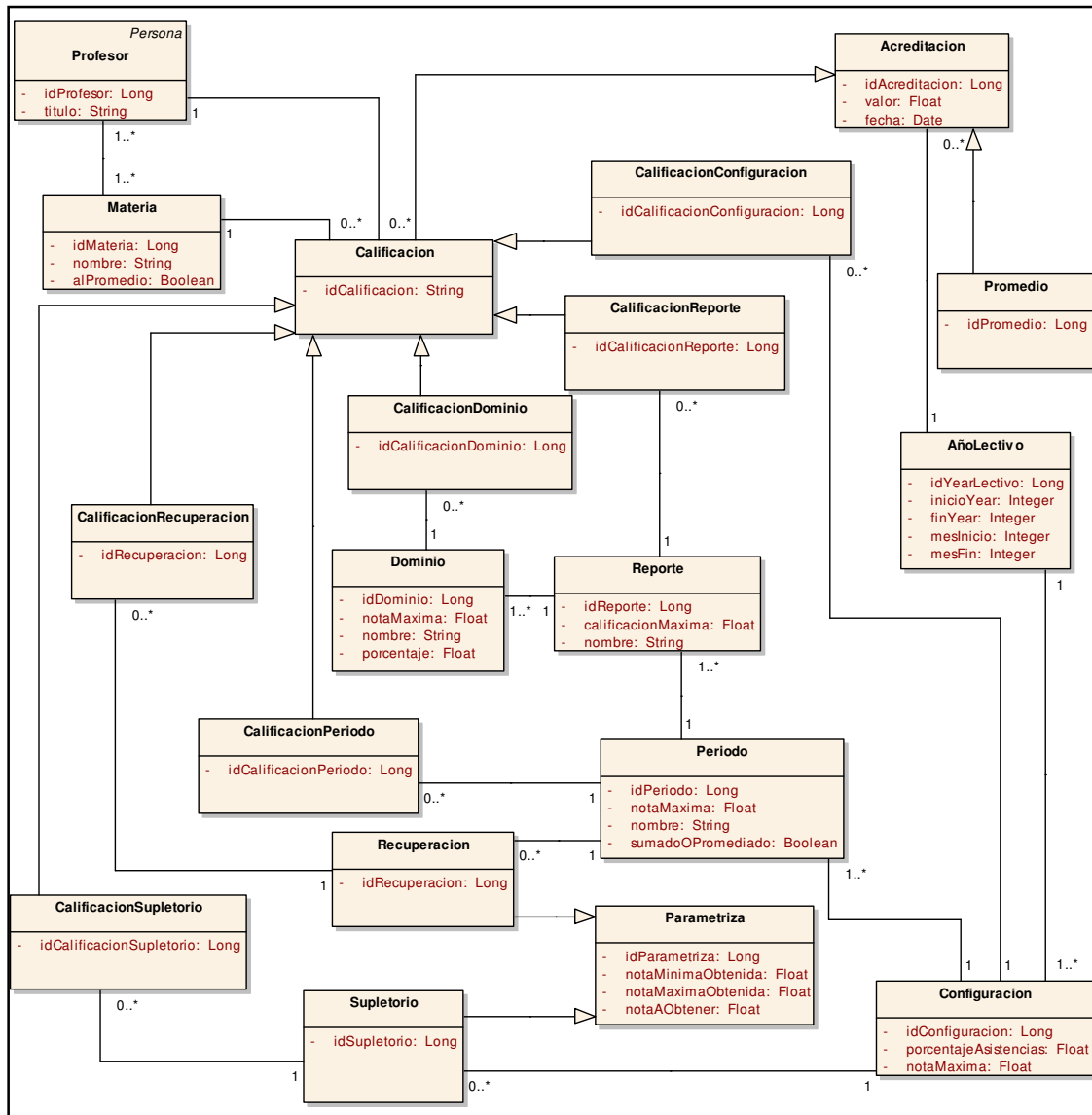
Modelo del Dominio Final 1. MDF01-MD01 Configuración del Sistema



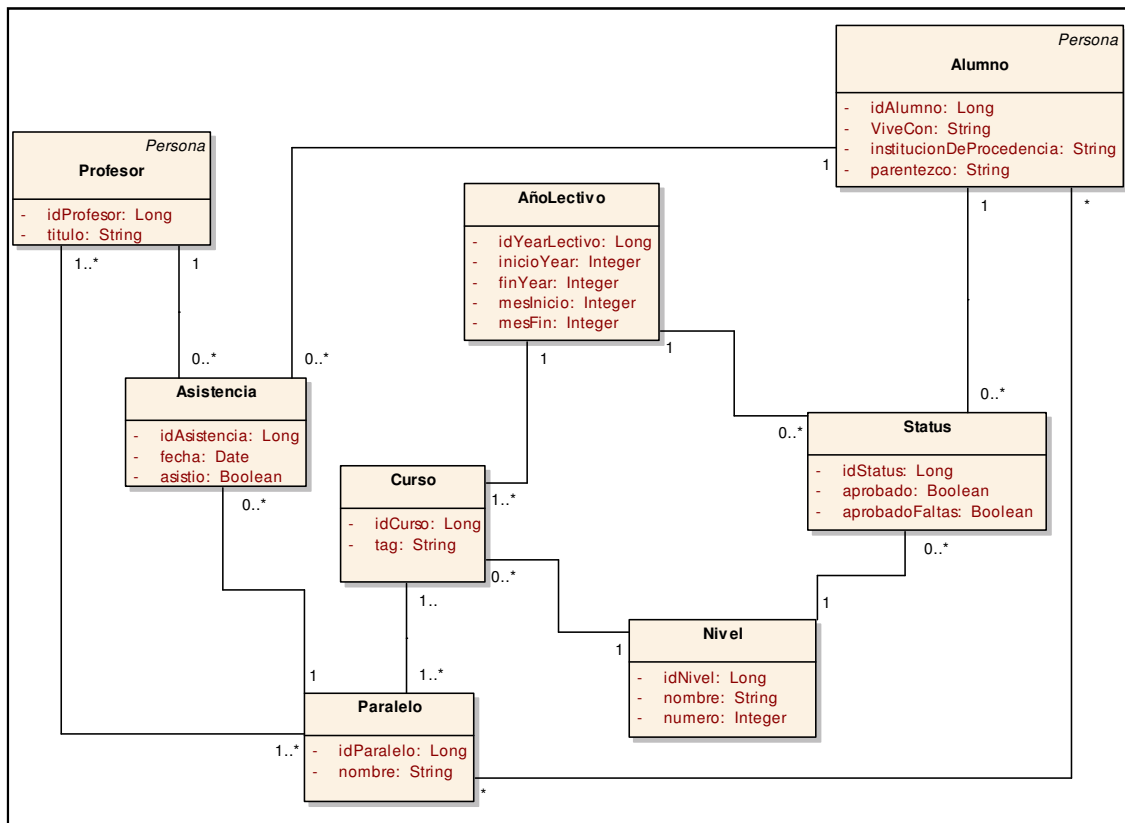
Modelo del Dominio Final 2. MDF02-MD02 Estructural



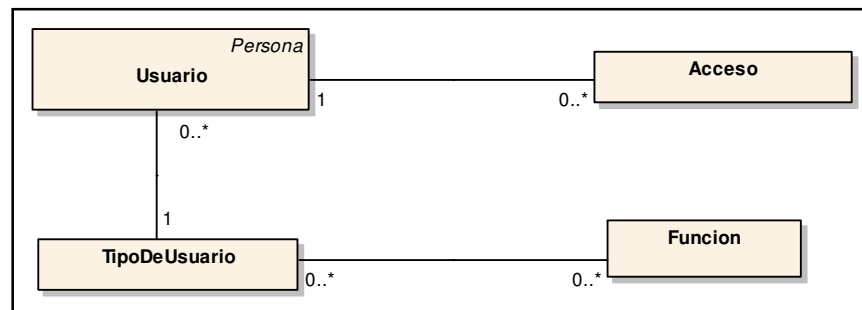
Modelo del Dominio Final 3. MDF03-MD03 Matriculación



Modelo del Dominio Final 4. MDF04-MD04 Acreditación



Modelo del Dominio Final 5. MDF05-MD05 Asistencias



Modelo del Dominio Final 6. MDF06-MD06 Usuarios

### 7.3 Diseño.

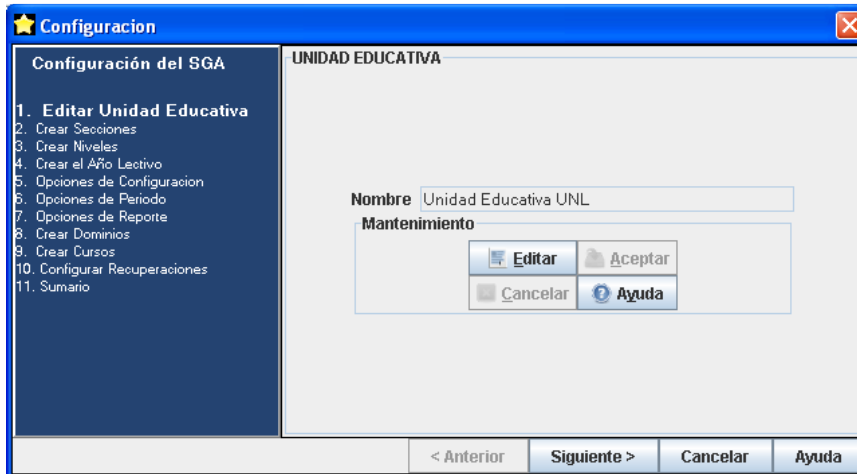
El diseño se compone de las siguientes partes:

#### 7.3.1 Descripción de los Casos de Uso.

##### 7.3.1.1 CU01 - Caso de uso Ingresar al Sistema

<b>Nombre</b>	Ingresar al Sistema.		
<b>Actor</b>	Usuarios.		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF36 Registrar y los accesos al sistema.</li> </ul>		
<b>Breve Descripción</b>	El <b>Usuario</b> ingresa el nombre de <b>Usuario</b> y contraseña para ingresar al Sistema y poder acceder al mismo.		
<b>PRE Condiciones</b>	1. Que el <b>Usuario</b> haya sido creado previamente en el Sistema.		
<b>Post Condición</b>	2. El <b>Usuario</b> ingresó al Sistema.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	El <b>Usuario</b> presiona el icono de <b>Acceso</b> al Sistema.	
	2.		El Sistema presenta la <u>Pantalla Ingresar al Sistema.</u>
	3.	El <b>Usuario</b> ingresa los datos del <b>Usuario</b> y presiona el <u>Botón Aceptar.</u>	
	4.		El Sistema valida los datos ingresados.
	5.		El Sistema presenta la <u>Pantalla Principal</u> de Sistema de gestión académica.
6.		El caso de uso finaliza.	
<b>Curso Alternativo A</b>	5. El Sistema presenta un mensaje “Nombre de <b>Usuario</b> o Clave Incorrectos”. El <b>Usuario</b> presiona el <u>Botón Aceptar.</u>		

### 7.3.1.2 CU02 – Caso de Uso Configurar Sistema

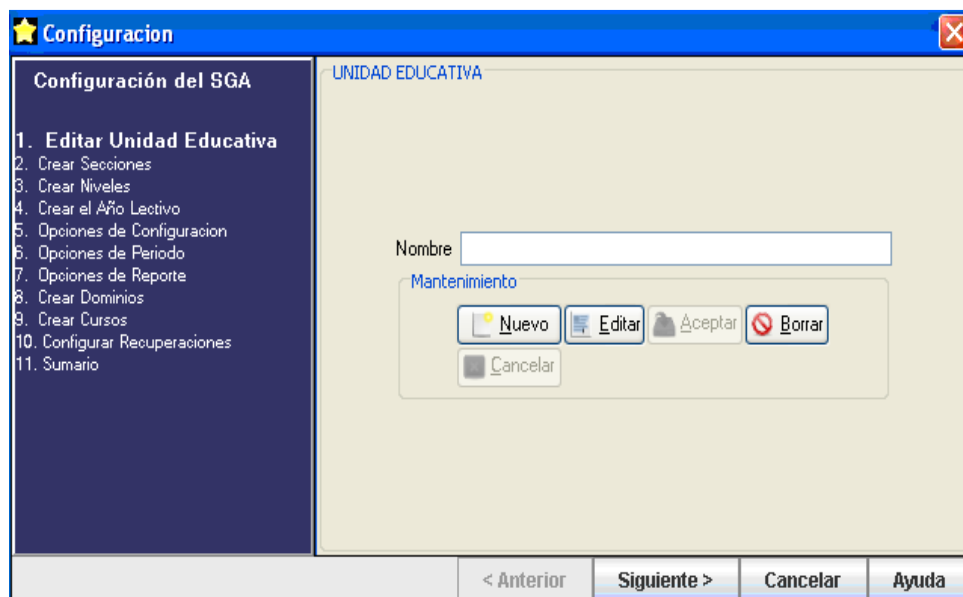


<b>Nombre</b>	Configurar Sistema		
<b>Actor</b>	Administrador		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF20</b> Configurar del sistema de acuerdo a la forma de calificar de cada uno de los niveles</li> </ul>		
<b>Breve Descripción</b>	El <b>Administrador</b> del Sistema podrá configurar el Sistema de acuerdo a la forma de evaluación de cada uno de los diversos <b>Cursos</b>		
<b>Pre Condición</b>	Ingresar al sistema por primera vez.		
<b>Post Condición</b>	Se creara una nueva <b>Configuración</b> para determinados <b>Cursos</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.		El sistema presenta la <u>Pantalla Configuración</u> automáticamente al iniciar por primera vez el sistema.
	2.		<<include>> CU03 - Mantener <b>Unidad Educativa</b>
	3.	El <b>Administrador</b> presiona el <u>Botón Siguiete</u>	
	4.		<<include>> CU04 - Mantener Secciones
	5.	El <b>Administrador</b> presiona el <u>Botón Siguiete</u>	
	6.		<<include>> CU05 - Mantener <b>Niveles</b>
	7.	El <b>Administrador</b> presiona el <u>Botón Siguiete</u>	
	8.		<<include>> CU06 - Mantener Años Lectivos
	9.	El <b>Administrador</b> presiona el <u>Botón Siguiete</u>	
	10.		<<include>> CU07 - Mantener <b>Configuración</b>
11.	El <b>Administrador</b> presiona el		

		<u>Botón Siguiente</u>	
	12.		<<include>> CU08 - Mantener <b>Periodos</b>
	13.	El <b>Administrador</b> presiona el <u>Botón Siguiente</u>	
	14.		<<include>> CU09 - Mantener <b>Reporte</b>
	15.	El <b>Administrador</b> presiona el <u>Botón Siguiente</u>	
	16.		<<include>> CU10 - Mantener <b>Dominio</b>
	17.	El <b>Administrador</b> presiona el <u>Botón Siguiente</u>	
	18.		<<include>> CU11 - Mantener <b>Curso</b>
	19.	El <b>Administrador</b> presiona el <u>Botón Siguiente</u>	
	20.		<<include>> CU12 - Mantener <b>Recuperación</b>
	21.		<<include>> CU13 - Mantener <b>Supletorios</b>
	22.	El <b>Administrador</b> presiona el <u>Botón Siguiente</u>	
	23.		El Sistema valida los datos ingresados
	24.		El Sistema presenta el resumen de la <b>Configuración</b> realizada
	25.	El <b>Administrador</b> presiona el <u>Botón Finalizar</u>	
	26.		El Sistema guarda la <b>Configuración</b>
	27.		El caso de Uso finaliza
<b>Curso Alterno A</b>	4, 6, 8, 10, 12, 14, 16, 18, 20, 23, 26. El <b>Administrador</b> presiona el <u>Botón Ayuda</u> El Sistema presenta la ayuda sobre la creación de configuraciones		
<b>Curso Alterno B</b>	15, 18, 21, 24, 27, 30, 33, 36, 40.- El Sistema presenta un mensaje de error "Error al ingresar datos" El <b>Administrador</b> presiona el <u>Botón Aceptar</u> Vuelve al paso en el que se genero el error		

### 7.3.1.3 CU03 – Caso de Uso Mantener Unidad Educativa

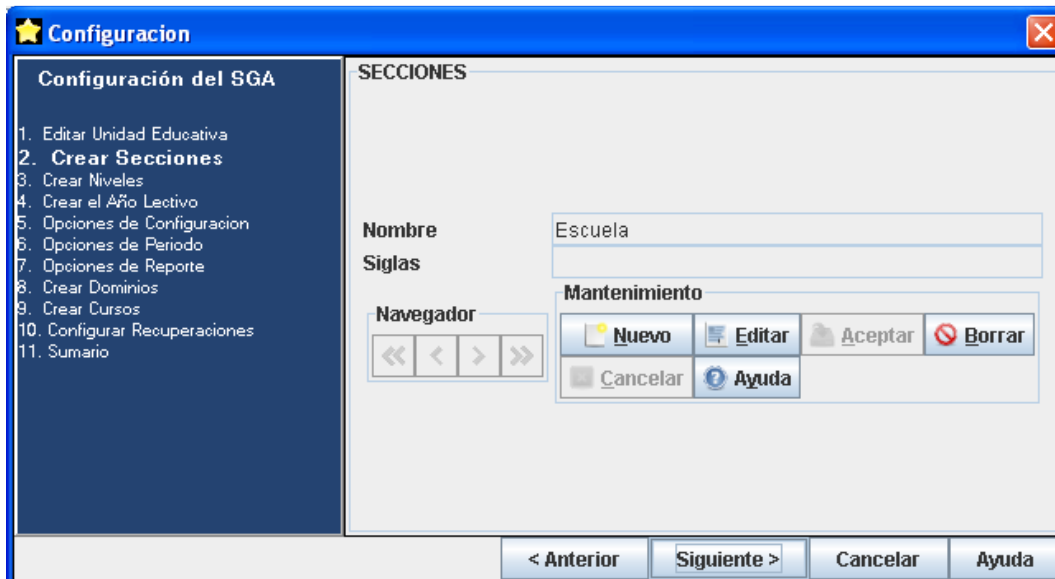




<b>Nombre</b>	Mantener <b>Unidad Educativa</b>		
<b>Actor</b>	<b>Administrador</b>		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF39</b> Crear Unidades Educativas</li> </ul>		
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear y editar la <b>Unidad Educativa</b> para la cual tendrá efecto todas las configuraciones posteriores.		
<b>Pre Condiciones</b>			
<b>Post Condición</b>	Se creara la <b>Unidad Educativa</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.		El Sistema presenta el <u>Panel Unidad Educativa</u> .
	2.	El <b>Administrador</b> presiona el <u>Botón Nuevo</u>	
	3.		El Sistema actualiza el <u>Panel Unidad Educativa</u> en modo edición
	4.	El <b>Administrador</b> ingresa los datos de la <b>Unidad Educativa</b> y presiona el <u>Botón Aceptar</u>	
	5.		El Sistema valida los datos ingresados
	6.		El Sistema guarda la <b>Unidad Educativa</b>
	7.		El Sistema quita el modo edición de las entradas de datos
8.	El caso de uso finaliza		
<b>Curso Alternativo A</b>	2. El <b>Administrador</b> presiona el <u>Botón Editar</u> El Sistema fija las entradas de datos en modo edición El <b>Administrador</b> Edita los datos de la <b>Unidad Educativa</b> El <b>Administrador</b> presiona el <u>Botón Aceptar</u>		

	El Sistema valida los datos ingresado El Sistema actualiza la <b>Unidad Educativa</b>
--	--

### 7.3.1.4 CU04 – Caso de Uso Mantener Secciones



<b>Nombre</b>	Mantener Secciones	
<b>Actor</b>	Administrador	
<b>Requerimientos</b>	• RQF38 Crear Secciones	
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear Secciones y de esta manera poder llevar un mejor control sobre los <b>Cursos</b>	
<b>PRE Condiciones</b>	Que exista una <b>Unidad Educativa</b> creadas	
<b>Post Condición</b>	Se crearan nuevas Secciones	
<b>Curso Normal</b>	<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	El Sistema presenta el <u>Panel Secciones</u>
	2.	El Sistema inicializa la <u>Lista de Secciones</u> existentes en la <b>Unidad Educativa</b>
	3.	El <b>Administrador</b> presiona el <u>Botón Nuevo</u>
	4.	El Sistema fija las entradas de datos en modo edición
	5.	El <b>Administrador</b> ingresa los datos de la <b>Sección</b> y presiona el <u>Botón Aceptar</u>
6.	El Sistema valida los datos ingresados	

	7.		El Sistema guarda los datos ingresados de la <b>Sección</b>
	8.		El Sistema quita el modo edición de las entradas de datos
	9.	El caso de uso finaliza	
<b>Curso Alterno A</b>	3. El <b>Administrador</b> presiona el <u>Botón Editar</u> El Sistema fija las entradas de datos en modo edición El <b>Administrador</b> Edita los datos de la <b>Sección</b> El <b>Administrador</b> presiona el <u>Botón Aceptar</u> El Sistema valida los datos ingresado El Sistema actualiza la <b>Sección</b>		
<b>Curso Alterno B</b>	3. El <b>Administrador</b> presiona el <u>Botón Borrar</u> El Sistema presenta un mensaje de confirmación El <b>Administrador</b> presiona el <u>Botón Si</u> El Sistema borra la <b>Sección</b>		
<b>Curso Alterno C</b>	1.El <b>Administrador</b> presiona el <u>Botón Primero</u> El Sistema presenta los datos de la primera <b>Sección</b> existente en la <u>Lista de Secciones</u> El Sistema presenta los resultados en el <u>Panel Mantener Sección</u>		
<b>Curso Alterno D</b>	3.El <b>Administrador</b> presiona el <u>Botón Anterior</u> El Sistema presenta los datos de la <b>Sección</b> anterior existente en la <u>Lista de Secciones</u> El Sistema presenta los resultados en el <u>Panel Mantener Sección</u>		
<b>Curso Alterno E</b>	3 El <b>Administrador</b> presiona el <u>Botón Siguiente</u> El Sistema presenta los datos de la siguiente <b>Sección</b> existente en la <u>Lista de Secciones</u> El Sistema presenta los resultados en el <u>Panel Mantener Sección</u>		
<b>Curso Alterno F</b>	3.El <b>Administrador</b> presiona el <u>Botón Ultimo</u> El Sistema presenta los datos de la ultima <b>Sección</b> existente en la <u>Lista de Secciones</u> El Sistema presenta los resultados en el <u>Panel Mantener Sección</u>		

## 7.3.1.5 CU05 – Caso de Uso Mantener Niveles

<b>Nombre</b>	Mantener Niveles	
<b>Actor</b>	Administrador	
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF10</b> Crear niveles de estudios</li> </ul>	
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear <b>Niveles</b> de estudio y en base a estos se puedan crear <b>Cursos</b>	
<b>PRE Condiciones</b>	Que exista al menos una <b>Sección</b> creada	
<b>Post Condición</b>	Se crearan nuevos <b>Niveles</b> de Estudio	
<b>Curso Normal</b>		<b>Entrada del Actor</b>
	1.	
	2.	
	3.	El <b>Administrador</b> presiona el <b>Botón Nuevo</b>
	4.	
	5.	El <b>Administrador</b> ingresa los datos del <b>Nivel</b> y selecciona la <b>Sección</b> a la cual va a pertenecer y presiona el <b>Botón Aceptar</b>
	6.	
7.		
	<b>Respuesta del Sistema</b>	
		El Sistema presenta el <b>Panel Nivel</b>
		El Sistema inicializa la <b>Lista de Niveles</b> existentes en la <b>Unidad Educativa</b>
		El Sistema actualiza el <b>Panel Nivel</b> en modo edición
		El Sistema valida los datos ingresados
		El Sistema guarda los datos del <b>Nivel</b> ingresado

	8.		El Sistema quita el modo edición de las entradas de datos
	9.	El caso de uso finaliza	
<b>Curso Alternativo A</b>	<p>3. El <b>Administrador</b> presiona el <u>Botón Editar</u>  El Sistema fija las entradas de datos en modo edición  El <b>Administrador</b> Edita los datos del <b>Nivel</b>  El <b>Administrador</b> presiona el <u>Botón Aceptar</u>  El Sistema valida los datos ingresado  El Sistema actualiza el <b>Nivel</b></p>		
<b>Curso Alternativo B</b>	<p>3. El <b>Administrador</b> presiona el <u>Botón Borrar</u>  El Sistema presenta un mensaje de confirmación  El <b>Administrador</b> presiona el <u>Botón Si</u>  El Sistema borra el <b>Nivel</b></p>		
<b>Curso Alternativo C</b>	<p>3.El <b>Administrador</b> presiona el <u>Botón Primero</u>  El Sistema obtiene los datos del primer <b>Nivel</b> existente, en la <u>Lista de Niveles</u>  El Sistema presenta los datos del <b>Nivel</b></p>		
<b>Curso Alternativo D</b>	<p>3.El <b>Administrador</b> presiona el <u>Botón Anterior</u>  El Sistema obtiene los datos del <b>Nivel</b> anterior, en la <u>Lista de Niveles</u>  El Sistema presenta los datos del <b>Nivel</b></p>		
<b>Curso Alternativo E</b>	<p>3.El <b>Administrador</b> presiona el <u>Botón Siguiente</u>  El Sistema obtiene los datos de la siguiente <b>Sección</b>, en la <u>Lista de Niveles</u>  El Sistema presenta los datos del <b>Nivel</b></p>		
<b>Curso Alternativo F</b>	<p>3.El <b>Administrador</b> presiona el <u>Botón Ultimo</u>  El Sistema obtiene los datos de la ultima <b>Sección</b>, en la <u>Lista de Niveles</u>  El Sistema presenta los datos del <b>Nivel</b></p>		

## 7.3.1.6 CU06 – Caso de Uso Mantener Año Lectivo

The screenshot shows a software window titled 'Configuración' with a sub-header 'Mantener Año Lectivo'. On the left is a sidebar menu with 11 items, where '4. Crear el Año Lectivo' is selected. The main area contains four dropdown menus: 'Año de Inicio' (2006), 'Año de Fin' (2007), 'Mes de Inicio' (Julio), and 'Mes de Fin' (Marzo). Below these are four buttons: 'Crear', 'Editar', 'Aceptar', and 'Cancelar'. At the bottom of the window are navigation buttons: '< Anterior', '> Siguiente', 'Cancelar', and 'Ayuda'.

<b>Nombre</b>	Mantener Año Lectivo	
<b>Actor</b>	Administrador	
<b>Requerimiento</b>	<ul style="list-style-type: none"> <li>• RQF19 Crear años lectivos</li> </ul>	
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear y configurar el periodo de tiempo del <b>Año Lectivo</b>	
<b>PRE Condiciones</b>	Que exista la <b>Unidad Educativa</b> creada No este activo otro <b>Año Lectivo</b> Que existan niveles creados	
<b>Post Condición</b>	Se crearan un nuevo <b>Año Lectivo</b>	
<b>Curso Normal</b>		<b>Entrada del Actor</b>
	1	
	2	El <b>Administrador</b> ingresa los datos del <b>Año Lectivo</b> en el <u>Panel Mantener Año Lectivo</u> y presiona el <u>Botón Crear</u>
	4	
		<b>Respuesta del Sistema</b>
		El sistema presenta el <u>Panel Mantener Año Lectivo</u> y actualiza la pantalla en modo edición
		El Sistema crea el <b>Año Lectivo</b>
<b>Curso Alternativo</b>	1. El <b>Administrador</b> presiona el <u>Botón Editar</u> El Sistema fija las entradas de datos en modo edición El <b>Administrador</b> Edita los datos del <b>Año Lectivo</b> El <b>Administrador</b> presiona el <u>Botón Aceptar</u> El Sistema actualiza el <b>Año Lectivo</b> El Sistema quita el modo edición de las entradas de datos	

## 7.3.1.7 CU07 – Caso de Uso Mantener Configuración

<b>Nombre</b>	Mantener Configuración									
<b>Actor</b>	Administrador									
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF20</b> Configurar del sistema de acuerdo a la forma de calificar de cada uno de los niveles</li> <li>• <b>RQF22</b> Fijar el porcentaje de asistencias con cual se aprueba un año lectivo</li> <li>• <b>RQF23</b> Definir el número de periodos de cada año lectivo</li> <li>• <b>RQF24</b> Determinar la calificación máxima por cada año lectivo</li> <li>• <b>RQF25</b> Determinar la calificación mínima para aprobar un año lectivo</li> <li>• <b>RQF26</b> Determinar la calificación mínima para aprobar un año lectivo</li> <li>• <b>RQF27</b> Determinar la calificación de cada uno de los periodos</li> </ul>									
<b>Breve Descripción</b>	El <b>Administrador</b> podrá configurar los datos necesarios para la <b>Configuración</b>									
<b>Pre Condiciones</b>	1. Que exista al menos un <b>Año Lectivo</b> creado Que exista un <b>Año Lectivo</b> activo									
<b>Post Condición</b>	Se configura los datos de la <b>Configuración</b>									
<b>Curso Normal</b>	<table border="1"> <thead> <tr> <th></th> <th><b>Entrada del Actor</b></th> <th><b>Respuesta del Sistema</b></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td></td> <td>El Sistema presenta el <u>Panel Mantener Configuración</u></td> </tr> <tr> <td>2.</td> <td>El <b>Administrador</b> ingresa los datos de <b>Configuración</b> en el <u>Panel Mantener Configuración</u></td> <td></td> </tr> </tbody> </table>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>	1.		El Sistema presenta el <u>Panel Mantener Configuración</u>	2.	El <b>Administrador</b> ingresa los datos de <b>Configuración</b> en el <u>Panel Mantener Configuración</u>	
		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>							
1.		El Sistema presenta el <u>Panel Mantener Configuración</u>								
2.	El <b>Administrador</b> ingresa los datos de <b>Configuración</b> en el <u>Panel Mantener Configuración</u>									

	3.		El Sistema calcula la Calificación para cada <b>Periodo</b> basado en la Calificación del <b>Año Lectivo</b> dividido al numero de <b>Periodos</b>
	4.		El Sistema crea la <b>Configuración</b>

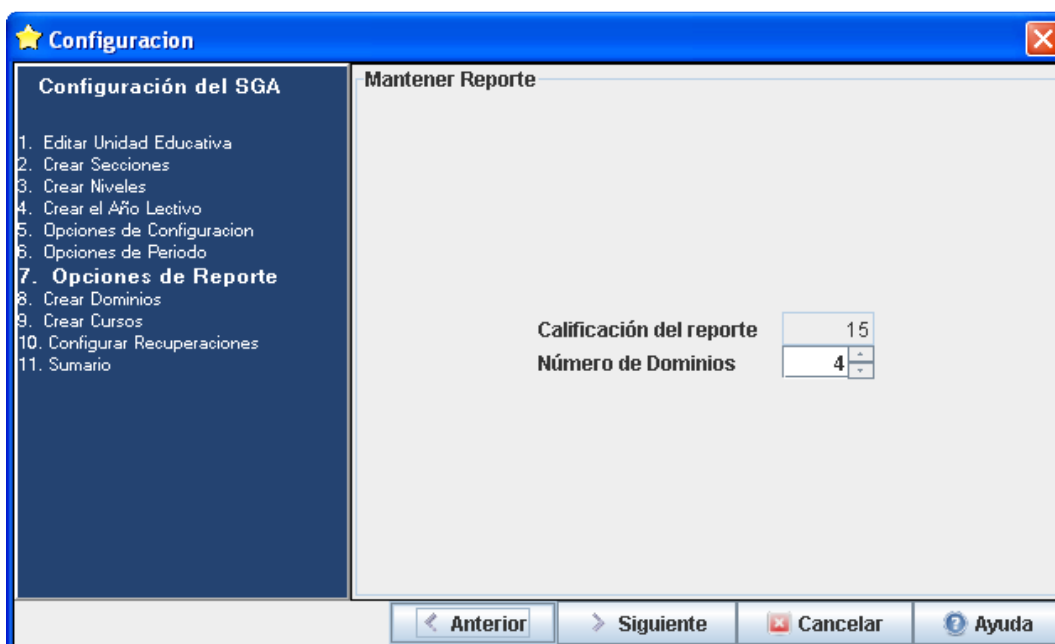
### 7.3.1.8 CU08 – Caso de Uso Mantener Periodos

<b>Nombre</b>	Mantener <b>Periodos</b>	
<b>Actor</b>	<b>Administrador</b>	
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF21</b> Crear periodos</li> <li>• <b>RQF28</b> Determinar el número de reportes de cada uno de los periodos</li> <li>• <b>RQF29</b> Determinar la forma de calificación de los reportes</li> <li>• <b>RQF32</b> Determinar el nombre de los periodos</li> </ul>	
<b>Breve Descripción</b>	El <b>Administrador</b> podrá configurar los datos del <b>Periodo</b>	
<b>PRE Condiciones</b>	Que se haya ingresado los datos de la <b>Configuración</b>	
<b>Post Condición</b>	Se configura los datos del <b>Periodo</b>	
<b>Curso Normal</b>		<b>Entrada del Actor</b>
		<b>Respuesta del Sistema</b>
	1.	El Sistema presenta el <u>Panel Mantener Periodo</u>
	2.	El <b>Administrador</b> ingresa los



		datos del <b>Periodo</b> el <u>Panel Mantener Periodo</u>	
	3.		El Sistema calcula la Calificación de los <b>Reportes</b> , basado en: la Calificación del <b>Periodo</b> dividido para el numero de <b>Reportes</b>
	4.		El Sistema crea el <b>Periodo</b>

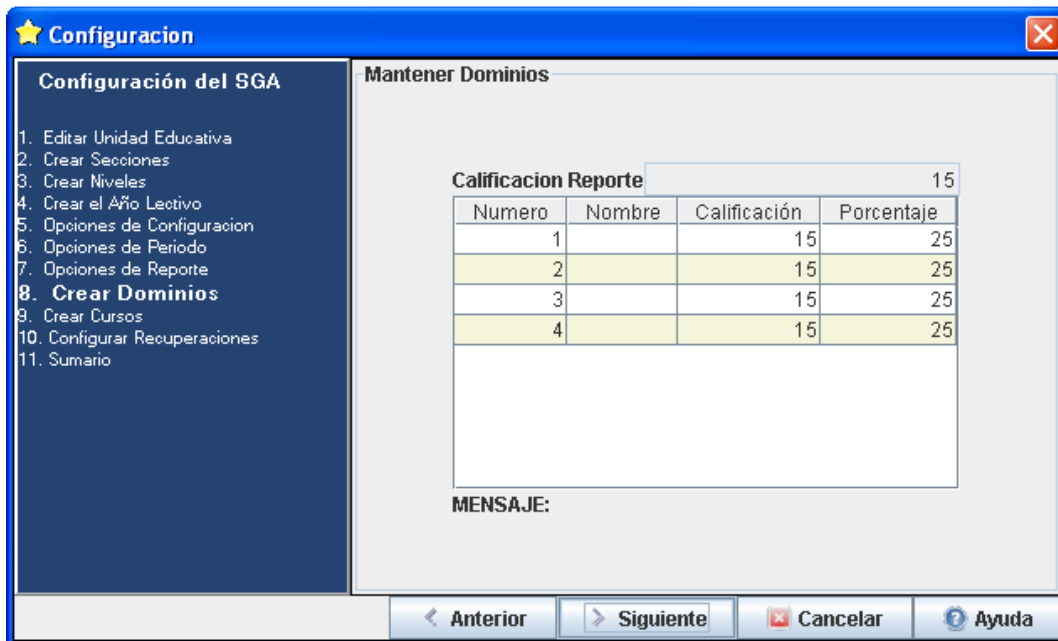
### 7.3.1.9 CU09 – Caso de Uso Mantener Reportes



<b>Nombre</b>	Mantener Reportes									
<b>Actor</b>	Administrador									
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF30</b> Determinar la calificación de cada uno de los reportes</li> <li>• <b>RQF33</b> Determinar el número de dominios de cada uno de los reportes (parámetros)</li> </ul>									
<b>Breve Descripción</b>	El <b>Administrador</b> podrá configurar los datos de los <b>Reportes</b>									
<b>PRE Condiciones</b>	Que se haya configurado los datos del <b>Periodo</b>									
<b>Post Condición</b>	Se configura los datos de los <b>Reportes</b>									
<b>Curso Normal</b>	<table border="1"> <thead> <tr> <th></th> <th>Entrada del Actor</th> <th>Respuesta del Sistema</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td></td> <td>El Sistema presenta el <u>Panel Mantener Reportes</u></td> </tr> <tr> <td>2.</td> <td>El <b>Administrador</b> ingresa el número de dominios que va a contener cada reporte en el <u>Panel Mantener Reportes</u></td> <td></td> </tr> </tbody> </table>		Entrada del Actor	Respuesta del Sistema	1.		El Sistema presenta el <u>Panel Mantener Reportes</u>	2.	El <b>Administrador</b> ingresa el número de dominios que va a contener cada reporte en el <u>Panel Mantener Reportes</u>	
		Entrada del Actor	Respuesta del Sistema							
1.		El Sistema presenta el <u>Panel Mantener Reportes</u>								
2.	El <b>Administrador</b> ingresa el número de dominios que va a contener cada reporte en el <u>Panel Mantener Reportes</u>									

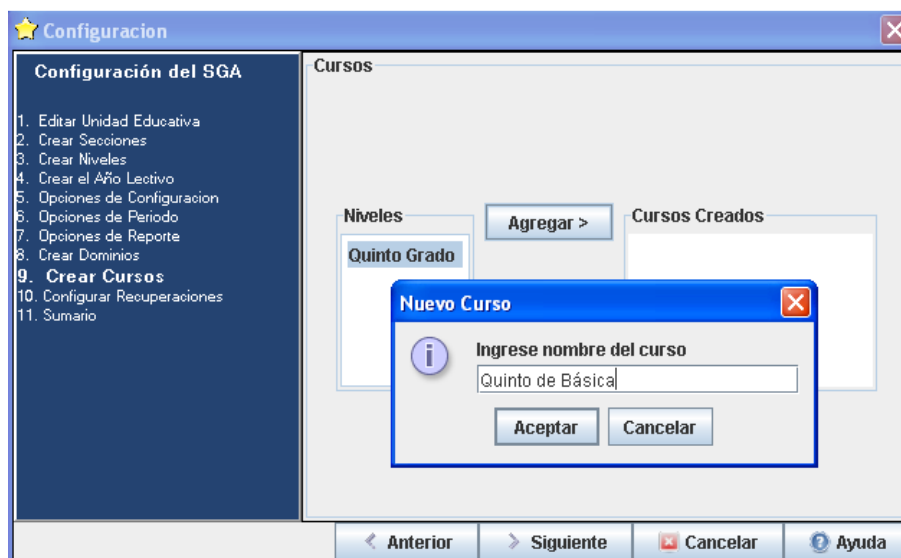
	3.	El Sistema crea el <b>Reporte</b>
--	----	-----------------------------------

### 7.3.1.10 CU10 – Caso de Uso Mantener Dominios



<b>Nombre</b>	Mantener Dominios															
<b>Actor</b>	Administrador															
<b>Requerimiento</b>	<ul style="list-style-type: none"> <li>• RQF31 Fijar el nombre de cada uno de los Dominios</li> </ul>															
<b>Breve Descripción</b>	El <b>Administrador</b> podrá configurar los datos de los Dominios															
<b>PRE Condiciones</b>	Que se haya configurado los datos del <b>Reporte</b>															
<b>Post Condición</b>	Se configura los datos de los Dominios															
<b>Curso Normal</b>	<table border="1"> <thead> <tr> <th></th> <th>Entrada del Actor</th> <th>Respuesta del Sistema</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td></td> <td>El Sistema presenta el <u>Panel Mantener Dominios</u> con la entrada para dominios de acuerdo al número ingresado en el <u>Panel Mantener Reporte</u></td> </tr> <tr> <td>2.</td> <td>El <b>Administrador</b> ingresa los datos del Dominio en el <u>Panel Mantener Dominios</u></td> <td></td> </tr> <tr> <td>3.</td> <td></td> <td>El Sistema valida los datos tomando como límite inferior y superior 100 %</td> </tr> <tr> <td>4.</td> <td></td> <td>El Sistema guarda los Dominios</td> </tr> </tbody> </table>		Entrada del Actor	Respuesta del Sistema	1.		El Sistema presenta el <u>Panel Mantener Dominios</u> con la entrada para dominios de acuerdo al número ingresado en el <u>Panel Mantener Reporte</u>	2.	El <b>Administrador</b> ingresa los datos del Dominio en el <u>Panel Mantener Dominios</u>		3.		El Sistema valida los datos tomando como límite inferior y superior 100 %	4.		El Sistema guarda los Dominios
		Entrada del Actor	Respuesta del Sistema													
	1.		El Sistema presenta el <u>Panel Mantener Dominios</u> con la entrada para dominios de acuerdo al número ingresado en el <u>Panel Mantener Reporte</u>													
	2.	El <b>Administrador</b> ingresa los datos del Dominio en el <u>Panel Mantener Dominios</u>														
3.		El Sistema valida los datos tomando como límite inferior y superior 100 %														
4.		El Sistema guarda los Dominios														

## 7.3.1.11 CU11 – Caso de Uso Mantener Cursos



<b>Nombre</b>	Mantener Cursos																		
<b>Actor</b>	Administrador																		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF06 Crear cursos</li> </ul>																		
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear <b>Cursos</b>																		
<b>PRE Condiciones</b>	Que exista Niveles de estudio creado Que los Niveles de estudios no tengan <b>Cursos</b> creados																		
<b>Post Condición</b>	Se crearan nuevos <b>Cursos</b>																		
<b>Curso Normal</b>	<table border="1"> <thead> <tr> <th></th> <th>Entrada del Actor</th> <th>Respuesta del Sistema</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td></td> <td>El Sistema presenta el <u>Panel Cursos</u></td> </tr> <tr> <td>2.</td> <td>El <b>Administrador</b> selecciona un <b>Nivel</b> de estudio de la <u>lista de Niveles</u> en el <u>Panel Cursos</u> y presiona el <u>Botón Agregar</u> del <u>Panel Cursos</u></td> <td></td> </tr> <tr> <td>3.</td> <td></td> <td>El Sistema presenta un cuadro de dialogo de Entrada Nuevo <b>Curso</b></td> </tr> <tr> <td>4.</td> <td>El <b>Administrador</b> ingresa un nombre para el <b>Curso</b> y presiona el <u>Botón Aceptar</u></td> <td></td> </tr> <tr> <td>5.</td> <td></td> <td>El Sistema crea el <b>Curso</b> con el nombre dado</td> </tr> </tbody> </table>		Entrada del Actor	Respuesta del Sistema	1.		El Sistema presenta el <u>Panel Cursos</u>	2.	El <b>Administrador</b> selecciona un <b>Nivel</b> de estudio de la <u>lista de Niveles</u> en el <u>Panel Cursos</u> y presiona el <u>Botón Agregar</u> del <u>Panel Cursos</u>		3.		El Sistema presenta un cuadro de dialogo de Entrada Nuevo <b>Curso</b>	4.	El <b>Administrador</b> ingresa un nombre para el <b>Curso</b> y presiona el <u>Botón Aceptar</u>		5.		El Sistema crea el <b>Curso</b> con el nombre dado
		Entrada del Actor	Respuesta del Sistema																
	1.		El Sistema presenta el <u>Panel Cursos</u>																
	2.	El <b>Administrador</b> selecciona un <b>Nivel</b> de estudio de la <u>lista de Niveles</u> en el <u>Panel Cursos</u> y presiona el <u>Botón Agregar</u> del <u>Panel Cursos</u>																	
	3.		El Sistema presenta un cuadro de dialogo de Entrada Nuevo <b>Curso</b>																
4.	El <b>Administrador</b> ingresa un nombre para el <b>Curso</b> y presiona el <u>Botón Aceptar</u>																		
5.		El Sistema crea el <b>Curso</b> con el nombre dado																	
<b>Curso Alternativo A</b>	El <b>Administrador</b> selecciona un <b>Curso</b> creado de la lista <b>Cursos</b> creados El <b>Administrador</b> presiona el <u>Botón Quitar</u> El Sistema elimina el <b>Curso</b> seleccionado																		

## 7.3.1.12 CU12 – Caso de Uso Mantener Recuperación

**Configuración**

**Configuración del SGA**

1. Editar Unidad Educativa
2. Crear Secciones
3. Crear Niveles
4. Crear el Año Lectivo
5. Opciones de Configuración
6. Opciones de Período
7. Opciones de Reporte
8. Crear Dominios
9. Crear Cursos
10. **Configurar Recuperacion...**
11. Sumario

**Recuperación Y Supletorio**

**Recuperación**

Número de Recuperaciones: 3

Nota Mínima	Nota Máxima	Nota a obtener
17	19	12
15	16	14
0	14	16

**Supletorio**

No Seleccionado

Nota Mínima	Nota Máxima	Nota a obtener
-------------	-------------	----------------

< Anterior    > Siguiente    Cancelar    Ayuda

<b>Nombre</b>	Mantener <b>Recuperación</b>		
<b>Actor</b>	Administrador		
<b>Requerimientos</b>			
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear Recuperaciones		
<b>Pre Condiciones</b>	Que existan <b>Periodos</b> en la <b>Configuración</b> actual que admitan recuperaciones		
<b>Post Condición</b>	Se crearán nuevos parámetros para las <b>Recuperaciones</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.		El Sistema presenta el <u>Panel Recuperación y Supletorio</u>
	2.		El Sistema activa la sección <u>Recuperación</u> del <u>Panel Recuperación y Supletorio</u>
	3.	El <b>Administrador</b> fija el número de recuperaciones a configurar	
	4.		El Sistema presenta una tabla de acuerdo al número de recuperaciones seleccionadas
	5.	El <b>Administrador</b> llena cada uno de los datos de <b>Recuperación</b>	
6.		El Sistema crea recuperaciones temporales	

## 7.3.1.13 CU13 – Caso de Uso Mantener Supletorio

**Configuración**

**Configuración del SGA**

1. Editar Unidad Educativa
2. Crear Secciones
3. Crear Niveles
4. Crear el Año Lectivo
5. Opciones de Configuración
6. Opciones de Período
7. Opciones de Reporte
8. Crear Dominios
9. Crear Cursos
- 10. Configurar Recuperación...**
11. Sumario

**Recuperación Y Supletorio**

**Recuperación**

No Seleccionado

Nota Mínima	Nota Máxima	Nota a obtener

**Supletorio**

Número de Supletorios: 3

Nota Mínima	Nota Máxima	Nota a obtener
35	39	12
30	34	14
25	29	16

< Anterior    > Siguiente    Cancelar    Ayuda

<b>Nombre</b>	Mantener Supletorio		
<b>Actor</b>	Administrador		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF34 Configurar supletorios nota mínima, nota a obtener</li> </ul>		
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear parámetros para los <b>Supletorios</b>		
<b>Pre Condiciones</b>	Este visible el <u>Panel Recuperación y Supletorio</u> Que la <b>Configuración</b> actual admita <b>Supletorios</b>		
<b>Post Condición</b>	Se crearán nuevos parámetros para los <b>Supletorios</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.		El Sistema activa la sección Supletorio del <u>Panel Recuperación y Supletorio</u>
	2.	El <b>Administrador</b> fija el número de <b>Supletorios</b> a configurar	
	3.		El Sistema presenta una tabla de acuerdo al número de Supletorios seleccionadas
	4.	El <b>Administrador</b> llena cada uno de los datos de <b>Supletorio</b>	
5.		El Sistema crea <b>Supletorios</b> Temporales	

## 7.3.1.14 CU14 – Caso de Uso Mantener Inscripción

**Inscripcion**

**Datos del Alumno**

Fecha: 23 de Mayo del 2006

**Datos Alumno**

Nombres: Edison  
 Apellidos: Coronel  
 Lugar De Nacimiento: Loja  
 Vive con:  Padre y Madre  Padre  Madre  Abuelos  Otros  
 Institución de procedencia: tecnico

**Fecha de Nacimiento**

Año: 1981 Mes: Febrero Dia: 25

**Dirección:**

Barrio: Loja  
 Calles: I  
 Telefono:  
 e-mail:

**Mantenimiento**

Navegador: << < > >>

Nuevo Editar Aceptar Borrar  
 Cancelar Buscar Imprimir Ayuda

<b>Nombre</b>	Mantener <b>Inscripción</b> .		
<b>Actor</b>	<b>Secretaria</b> .		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF01</b> Permitir la inscripción de alumnos.</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá crear Inscripciones, editarlas y borrarlas según como sea el requerimiento.		
<b>PRE Condiciones</b>	Que se haya configurado en su totalidad el <b>Año Lectivo</b> .		
<b>Post Condición</b>	Se inscribió un Alumno en la Unidad Educativa.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <b>Matricula</b> y selecciona el ítem <b>Inscripción</b> de la <b>Pantalla Principal</b>	
	2.		El Sistema presenta la <b>Pantalla Inscripción</b> .
	3.		El Sistema inicializa la Lista de Inscripciones
	4.	La <b>Secretaria</b> presiona el <b>Botón Nuevo</b> en la pestaña <b>Alumnos</b> de la <b>Pantalla Inscripción</b> .	
	5.		El Sistema fija las entradas de datos en modo edición.
6.	La <b>Secretaria</b> ingresa los		

		datos de la <b>Inscripción</b> , del <b>Alumno</b> y de los <b>Padres</b> y presiona el <b>Botón Aceptar</b> en la <b>Pantalla Inscripción</b> .	
	7.		El Sistema valida los datos ingresados de la <b>Inscripción</b> y del <b>Alumno</b>
	8.		El Sistema guarda los datos ingresados de la <b>Inscripción</b> y del <b>Alumno</b> .
	9.		El Sistema quita el modo edición de las entradas de datos
	10.	La <b>Secretaria</b> presiona el <b>Botón Imprimir</b>	
	11.		El Sistema genera el <b>Reporte de Inscripción de Alumno</b>
	12.		<<Include>> Imprimir
	13.		El caso de uso finaliza.
<b>Curso Alterno A</b>		3. La <b>Secretaria</b> presiona el <b>Botón Editar</b> . El Sistema fija las entradas de datos en modo edición. La <b>Secretaria</b> edita los datos deseados y presiona el <b>Botón Aceptar</b> . El Sistema valida los datos ingresados El Sistema Actualiza la <b>Inscripción del Alumno</b> El Sistema quita el modo edición de las entradas de datos	
<b>Curso Alterno B</b>		3. La <b>Secretaria</b> presiona el <b>Botón Borrar</b> . El Sistema le presenta un mensaje de confirmación de eliminación. La <b>Secretaria</b> presiona el <b>Botón Si</b> . El Sistema valida la eliminación. El Sistema elimina los datos solicitados.	
<b>Curso Alterno C</b>		3. La <b>Secretaria</b> presiona el <b>Botón Buscar</b> . El Sistema le presenta la Pantalla buscar <<Use Case >> Buscar	
<b>Curso Alterno D</b>		3. La <b>Secretaria</b> presiona el <b>Botón Ayuda</b> . El Sistema le presenta la Pantalla de la ayuda de <b>Inscripción</b> .	
<b>Curso Alterno E</b>		3. La <b>Secretaria</b> presiona el <b>Botón Siguiente</b> . El Sistema obtiene los datos de la siguiente <b>Inscripción</b> de la Lista El Sistema presenta los resultados	
<b>Curso Alterno F</b>		3. La <b>Secretaria</b> presiona el <b>Botón Anterior</b> El Sistema obtiene los datos de la <b>Inscripción</b> anterior de la Lista El Sistema presenta los resultados	
<b>Curso Alterno G</b>		3. La <b>Secretaria</b> presiona el botón. Primero El Sistema obtiene los datos de la primera <b>Inscripción</b> de la Lista El Sistema presenta los resultados	
<b>Curso Alterno H</b>		3. La <b>Secretaria</b> presiona el botón último. El Sistema le presenta los datos de la última <b>Inscripción</b> de la Lista El Sistema presenta los resultados	
<b>Curso Alterno I</b>		5. La <b>Secretaria</b> presiona el <b>Botón Buscar Padres</b> <<include>> Buscar	

## 7.3.1.15 CU15 – Caso de Uso Mantener Matricula

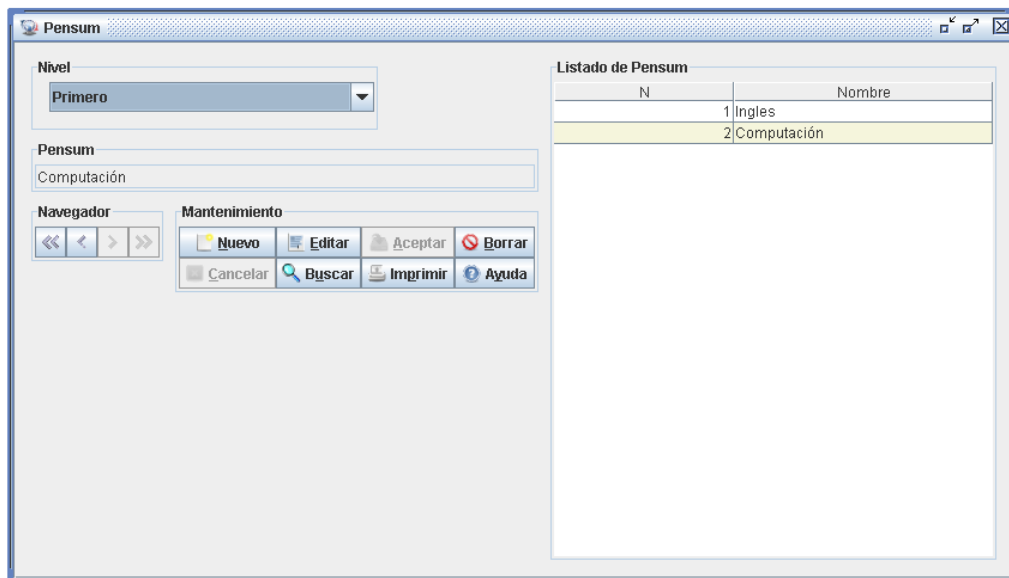
<b>Nombre</b>	<b>Mantener Matricula.</b>		
<b>Actor</b>	<b>Secretaria.</b>		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF02</b> Permitir la matriculación de los alumnos en determinado curso</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá crear <b>Matriculas</b> , editarlas y borrarlas según como sea el requerimiento.		
<b>PRE Condiciones</b>	Que se haya configurado en su totalidad el <b>Año Lectivo</b> .		
<b>Post Condición</b>	Se <b>Matricula Alumno</b> en la <b>Unidad Educativa</b> .		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <u>Matriculación</u> y selecciona el ítem <u>Matricula de la Pantalla Principal</u>	
	2.		El Sistema presenta la <u>Pantalla Matricula</u> .
	3.		El Sistema Inicializa la <u>Lista de Matriculas</u>
	4.	La <b>Secretaria</b> presiona el <u>Botón Nuevo</u> de la <u>Pantalla Matricula</u> .	
5.		El Sistema fija las entradas de datos en modo edición y activa	



		el botón <u>Buscar Inscripción, Alumno, Representante</u>
	6.	La <b>Secretaria</b> ingresa los datos de la <b>Matricula</b>
	7.	La <b>Secretaria</b> Ingresa los datos del <b>Alumno</b>
	8.	La <b>Secretaria</b> Ingresa los datos de los <b>Padres</b>
	9.	La <b>Secretaria</b> ingresa los datos del <b>Representante</b>
	10.	La <b>Secretaria</b> selecciona un <b>Curso</b> y presiona el <u>Botón Aceptar</u>
	11.	El Sistema Valida los datos ingresados de la <b>Matricula, Alumno, Padres, y del Representante</b>
	12.	El Sistema guarda los datos de la del <b>Alumno, los Padres, El Representante</b> y de la <b>Matricula</b> en determinado <b>Curso del Año Lectivo Actual</b>
	13.	El Sistema quita el modo edición de las entradas de datos
	14.	El caso de uso finaliza.
<b>Curso Alterno A</b>	3. La <b>Secretaria</b> presiona el <u>Botón Editar</u> . El Sistema fija las entradas de datos en modo edición. La <b>Secretaria</b> modifica los datos de la <b>Matricula, los Padres, el Representante</b> y del <b>Alumno</b> , y presiona el <u>Botón Aceptar</u> El Sistema actualiza los datos de la <b>Matricula, Padres, Representante</b> y del <b>Alumno</b> El Sistema quita el modo edición de las entradas de datos	
<b>Curso Alterno B</b>	3. La <b>Secretaria</b> presiona el <u>Botón Borrar</u> . El Sistema le presenta un mensaje de confirmación de eliminación. La <b>Secretaria</b> presiona el <u>Botón Si</u> . El Sistema elimina la <b>Matricula</b> .	
<b>Curso Alterno C</b>	3. La <b>Secretaria</b> presiona el <u>Botón Siguiente</u> . El Sistema le obtiene la <b>Matricula</b> siguiente	
<b>Curso Alterno D</b>	3. La <b>Secretaria</b> presiona el <u>Botón Anterior</u> . El Sistema obtiene los datos de la <b>Matricula</b> anterior de la <u>Lista de Matriculas</u> El Sistema presenta los resultados	
<b>Curso Alterno E</b>	3. La <b>Secretaria</b> presiona el <u>Botón Primero</u> . El Sistema obtiene los datos de la primera <b>Matricula</b> . anterior de la <u>Lista de Matriculas</u> El Sistema presenta los resultados	
<b>Curso Alterno F</b>	3. La <b>Secretaria</b> presiona el botón. ultimo	

	El Sistema obtiene los datos de la ultima <b>Matricula</b> . anterior de la <u>Lista de Matriculas</u> El Sistema presenta los resultados
<b>Curso Alterno G</b>	3. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> . El Sistema presenta la Pantalla de la ayuda de la <b>Matricula</b> anterior de la <u>Lista de Matriculas</u> El Sistema presenta los resultados
<b>Curso Alterno H</b>	3. La <b>Secretaria</b> presiona el <u>Botón Buscar</u> . El Sistema le presenta la Pantalla buscar <<Include >> Buscar El Sistema presenta los resultados de la búsqueda de la <b>Matricula</b>
<b>Curso Alterno I</b>	6. La <b>Secretaria</b> presiona el <u>Botón Buscar Inscripción</u> <<include>> Buscar El Sistema presenta los datos de la <b>Inscripción del Pasible Alumno</b>
<b>Curso Alterno J</b>	7. La <b>Secretaria</b> presiona el <u>Botón Buscar Alumno</u> <<include>> Buscar El Sistema carga el <b>Alumno</b> El Sistema valida el estado del <b>Alumno</b> El Sistema selecciona el <b>Curso</b> de acuerdo al estado del <b>Alumno</b>
<b>Curso Alterno J</b>	9. La <b>Secretaria</b> presiona el <u>Botón Buscar Representante</u> <<include>> Buscar El Sistema carga y presenta los datos del <b>Representante</b> producto de la búsqueda
<b>Curso Alterno L</b>	9. La <b>Secretaria</b> presiona el <u>Botón Imprimir</u> El Sistema genera el <b>Reporte</b> de la <b>Matricula</b> <<include>> Imprimir

### 7.3.1.16 CU16 – Caso de Uso Mantener Pensum

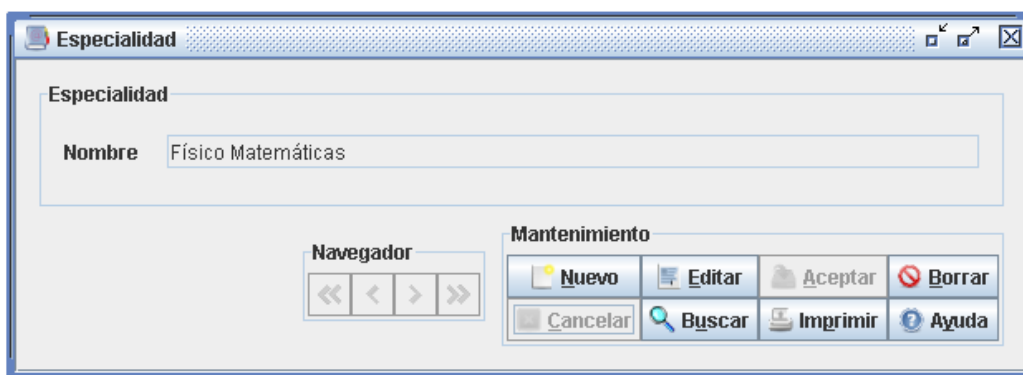


<b>Nombre</b>	<b>Mantener Pensum</b>
<b>Actor</b>	<b>Secretaria</b>

<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF11</b> Crear el pensum de estudios para cada uno de los niveles</li> </ul>		
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear diferentes <b>Pensum</b> para cada uno de los <b>Niveles</b>		
<b>Pre Condiciones</b>	1. Que exista al menos un <b>Nivel</b> de Educación creado		
<b>Post Condición</b>	Se creo un <b>Pensum</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace Clic en el <u>menú Materia</u> y Selecciona el <u>ítem Pensum</u> de la <u>Pantalla Principal</u>	
	2.		El Sistema presenta la <u>Pantalla Pensum</u>
	3.		El Sistema inicializa la <u>lista de Niveles</u> y el <u>Listado de Pensum</u> para el determinado <b>Nivel</b>
	4.	La <b>Secretaria</b> selecciona el <b>Nivel</b> para el cual se va a crear el <b>Pensum</b> y presiona el <u>Botón Nuevo</u>	
	5.		El Sistema actualiza las entradas de datos en modo edición
	6.	La <b>Secretaria</b> ingresa los datos del <b>Pensum</b> en el campo de entrada de texto <b>pensum</b> y presiona el <u>Botón Aceptar</u>	
	7.		El Sistema valida los datos ingresados
	8.		El Sistema guarda los datos ingresados
	9.		El Sistema quita el modo edición de las entradas de datos
	10.	El caso de Uso Finaliza	
<b>Curso Alterno A</b>	4. La <b>Secretaria</b> presiona el <u>Botón Editar</u> El Sistema fija las entradas de datos en modo edición La <b>Secretaria</b> Edita los datos del <b>Pensum</b> y presiona el <u>botón Aceptar</u> El Sistema valida los datos ingresados El Sistema actualiza el <b>Pensum</b> El sistema quita el modo edición		
<b>Curso Alterno B</b>	4. La <b>Secretaria</b> presiona el <u>Botón Borrar</u> El Sistema presenta un mensaje de confirmación La <b>Secretaria</b> presiona el <u>Botón Si</u> El Sistema borra el <b>Pensum</b>		
<b>Curso Alterno C</b>	4. La <b>Secretaria</b> presiona el <u>Botón Primero</u> El Sistema obtiene los datos del primer <b>Pensum</b> de la Lista		

	El Sistema presenta los resultados
<b>Curso Alterno D</b>	4.El <b>Administrador</b> presiona el <u>Botón Anterior</u> El Sistema obtiene los datos del <b>Pensum</b> anterior de la Lista El Sistema presenta los resultados
<b>Curso Alterno E</b>	4.La <b>Secretaria</b> presiona el <u>Botón Siguiente</u> El Sistema obtiene los datos del siguiente <b>Pensum</b> de la Lista El Sistema presenta los resultados
<b>Curso Alterno F</b>	4.La <b>Secretaria</b> presiona el <u>Botón Ultimo</u> El Sistema obtiene los datos del ultimo <b>Pensum</b> de la Lista El Sistema presenta los resultados
<b>Curso Alterno G</b>	4. el <b>Administrador</b> presiona el <u>Botón Ayuda</u> El Sistema presenta la Pantalla Ayuda de Tipos de <b>Usuarios</b>

### 7.3.1.17 CU17 – Caso de Uso Mantener Especialidades



<b>Nombre</b>	Mantener Especialidades		
<b>Actor</b>	<b>Secretaria</b>		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF12</b> Crear especialidades</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá crear diferentes Especialidades		
<b>PRE Condiciones</b>	Que exista la <b>Unidad Educativa</b>		
<b>Post Condición</b>	Se creo una <b>Especialidad</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace Clic en el <u>menú Materia</u> y Selecciona el <u>ítem Especialidad</u> de la <u>Pantalla Principal</u>	
	2.		El Sistema presenta la <u>Pantalla Especialidad</u>
	3.		El Sistema inicializa la <u>Lista de Especialidades</u> creadas en la <b>Unidad Educativa</b>
	4.	La <b>Secretaria</b> presiona el	

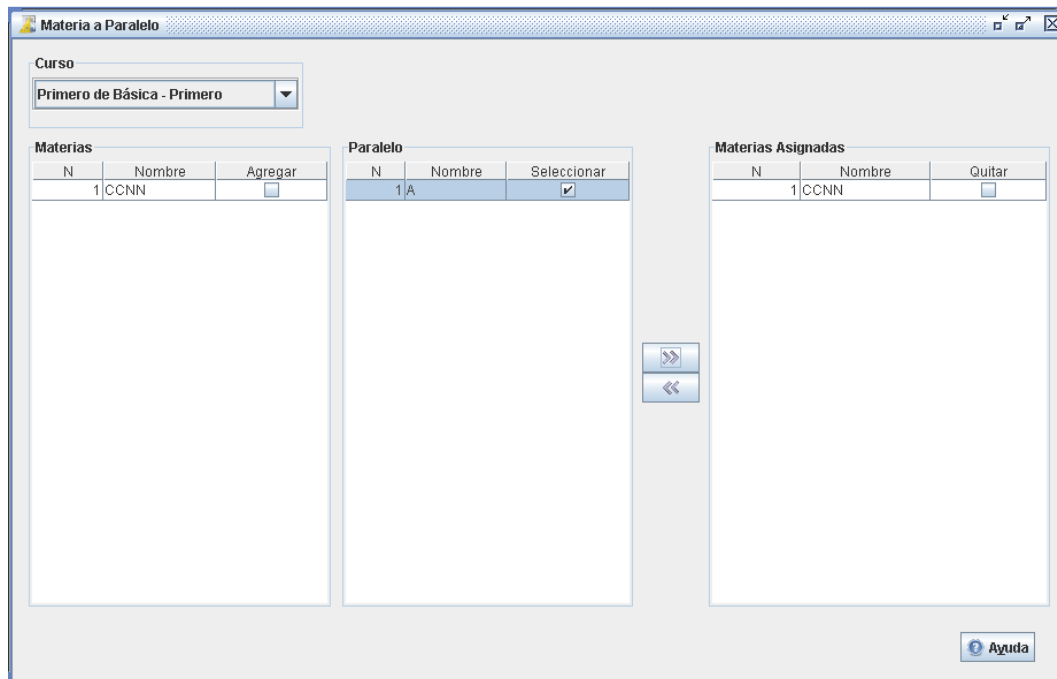
	<u>Botón Nuevo</u> de la <u>Pantalla Especialidad</u>	
	5.	El Sistema fija las entradas de datos en modo edición
	6.	La <b>Secretaria</b> ingresa los datos de la <b>Especialidad</b> y presiona el <u>Botón Aceptar</u> de la <u>Pantalla Especialidad</u>
	7.	El Sistema valida los datos ingresados
	8.	El Sistema guarda los datos ingresados
	9.	El Sistema quita el modo edición de las entradas de datos
	10.	El caso de Uso Finaliza
<b>Curso Alternativo A</b>	<p>4. La <b>Secretaria</b> presiona el <u>Botón Editar</u>  El Sistema fija las entradas de datos en modo edición  La <b>Secretaria</b> Edita los datos de la <b>Especialidad</b> y presiona el <u>Botón Aceptar</u>  El Sistema valida los datos ingresado  El Sistema actualiza la <b>Especialidad</b>  El Sistema quita el modo edición</p>	
<b>Curso Alternativo B</b>	<p>4. La <b>Secretaria</b> presiona el <u>Botón Borrar</u>  El Sistema presenta un mensaje de confirmación  La <b>Secretaria</b> presiona el <u>Botón Si</u>  El Sistema borra la <b>Especialidad</b></p>	
<b>Curso Alternativo C</b>	<p>4. La <b>Secretaria</b> presiona el <u>Botón Primero</u>  El Sistema obtiene los datos de la primera <b>Especialidad</b> de la Lista  El Sistema presenta los resultados</p>	
<b>Curso Alternativo D</b>	<p>4. El <b>Administrador</b> presiona el <u>Botón Anterior</u>  El Sistema obtiene los datos de la <b>Especialidad</b> anterior de la Lista  El Sistema presenta los resultados</p>	
<b>Curso Alternativo E</b>	<p>4. La <b>Secretaria</b> presiona el <u>Botón Siguiente</u>  El Sistema obtiene los datos de la siguiente <b>Especialidad</b> de la Lista  El Sistema presenta los resultados</p>	
<b>Curso Alternativo F</b>	<p>4. La <b>Secretaria</b> presiona el <u>Botón Ultimo</u>  El Sistema obtiene los datos de la ultima <b>Especialidad</b> de la Lista  El Sistema presenta los resultados</p>	
<b>Curso Alternativo G</b>	<p>4. el <b>Administrador</b> presiona el <u>Botón Ayuda</u>  El Sistema presenta la Pantalla Ayuda de <b>Especialidad</b></p>	

## 7.3.1.18 CU18 – Caso de Uso Mantener Materias

<b>Nombre</b>	Mantener Materias		
<b>Actor</b>	Secretaria.		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF14</b> Crear materias para cada uno de los cursos en base al pensum existente de cada nivel</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> ingresa en el menú <b>Materia</b> y selecciona la <b>Materia</b> que desea crear del <b>Pensum</b> .		
<b>Pre Condiciones</b>	Que el <b>Pensum</b> de <b>Nivel</b> exista. Que existan <b>Cursos</b> . Que el <b>Nivel</b> del <b>Curso</b> tenga <b>Pensum</b> .		
<b>Post Condición</b>	Se cree la <b>Materia</b> en el Sistema.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> selecciona el ítem <b>Materia</b> del menú <b>Mantener Materia</b> de la <b>Pantalla Principal</b>	
	2.		El Sistema presenta la <b>Pantalla Materia</b> .
3.	La <b>Secretaria</b> selecciona el <b>curso</b> en el cual se va a crear la <b>materia</b> en la <b>lista</b>		

	<u>desplegable curso</u>	
	4.	El Sistema presenta el <b>Pensum</b> del <b>Nivel</b> al que pertenece el <b>Curso</b> seleccionado.
	5.	La <b>Secretaria</b> selecciona de la <u>lista Pensum</u> , el <b>Pensum</b> que se desea agregar como <b>Materia</b> para el <b>Curso</b> Seccionado
	6.	El Sistema activa el botón <u>agregar</u>
	7.	La <b>Secretaria</b> presiona el <u>Botón Agregar</u>
	8.	El Sistema agrega la <b>Materia</b> al <b>Curso</b> y presenta la <b>Materia</b> creada en la lista <b>Materias</b> .
	9.	El caso de uso finaliza.
<b>Curso Alterno A</b>	3, 5. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> . <<include>> CU35 – Caso de Uso Mostrar Ayuda	
<b>Curso Alterno B</b>	La <b>Secretaria</b> selecciona la <b>Materia</b> de la lista El Sistema activa el <u>Botón Quitar</u> La <b>Secretaria</b> presiona el <u>Botón Quitar</u> El Sistema quita la <b>Materia</b> del <b>Curso</b> El Sistema actualiza la Pantalla y presenta la <b>Materia</b> seleccionada en la lista <b>Pensum</b> .	

## 7.3.1.19 CU19 – Caso de Uso Asignar Materia a Paralelo



<b>Nombre</b>	Asignar Materia a Paralelo.		
<b>Actor</b>	Secretaria.		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF16 Asignar Materias a cada uno de los paralelos</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> ingresa en el menú <b>Materia</b> y selecciona <b>Materia a Paralelo</b> para asignar las <b>Materias</b> que le pertenecen a un determinado <b>Paralelo</b> .		
<b>PRE Condiciones</b>	<ol style="list-style-type: none"> <li>1. Que la Materia exista.</li> <li>2. Que existan Paralelos.</li> </ol>		
<b>Post Condición</b>	Se asigna la <b>Materia</b> a un <b>Paralelo</b> .		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <b>Materias</b> y selecciona el ítem <b>Materia a Paralelo</b> de la <u>Pantalla Principal</u>	
	2.		El Sistema presenta la Pantalla <b>Materia a Paralelo</b> .
	3.	La <b>Secretaria</b> hace clic sobre la <u>lista desplegable Curso</u> para seleccionar el <b>Curso</b> al cual se le va asignar una <b>Materia</b> .	
	4.		El Sistema carga la lista de <b>Materias</b> asignadas al <b>Curso</b>
5.		El Sistema carga la lista de <b>Paralelos</b> del <b>Curso</b> seleccionado	



	6.	La <b>Secretaria</b> selecciona la(s) <b>Materia(s)</b> de la <u>lista materias</u> y el (los) <b>Paralelo(s)</b> de la <u>lista paralelos</u>	
	7.		El Sistema activa el <u>Botón Agregar</u>
	8.	La <b>Secretaria</b> presiona el <u>Botón Agregar</u> .	
	9.		El Sistema asigna la(s) <b>Materia(s)</b> al (los) <b>Paralelo(s)</b> seleccionados y presenta las Materias asignadas en la <u>lista Materias Asignadas</u>
	10		El caso de uso finaliza.
<b>Curso Alterno A</b>	1,3, 6. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> . <<include>> CU35 – Caso de Uso Mostrar Ayuda.		
<b>Curso Alterno B</b>	La <b>Secretaria</b> selecciona la(s) <b>Materia(s)</b> de la <u>lista Materias Asignadas</u> El Sistema activa el <u>Botón Quitar</u> La <b>Secretaria</b> presiona el <u>Botón Quitar</u> . El Sistema quita la <b>Materia</b> seleccionada de los <b>Paralelo</b> asignado en la <u>lista Materias Asignadas</u> .		

## 7.3.1.20 CU20 – Caso de Uso Mantener Profesor

**Profesor**

**Datos Profesor**

Nombres: Jorge

Apellidos: Sanchez

Título: Licenciado

Hombre  Mujer

**Dirección :**

Barrio : Tebaida

Calles : Argentina

Telefono : 2575757

e-mail : profe@hotmail.com

**Navegador**

« < > »

**Mantenimiento**

Nuevo Editar Aceptar Borrar

Cancel... Buscar Imprimir Ayuda

<b>Nombre</b>	Mantener <b>Profesor</b> .		
<b>Actor</b>	<b>Secretaria</b> .		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF08</b> Registrar y mantener los datos de cada uno de los profesores</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> ingresa en el menú <b>Profesor</b> e ingresa los datos del <b>Profesor</b> que pertenece a la <b>Unidad Educativa</b> .		
<b>PRE Condiciones</b>	1. Que el <b>Profesor</b> pertenezca a la <b>Unidad Educativa</b> .		
<b>Post Condición</b>	2. El <b>Profesor</b> queda registrado en el Sistema.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <b>Profesor</b> y selecciona el ítem <b>Profesor</b> de la <b>Pantalla Principal</b> .	
	2.		El Sistema presenta la <b>Pantalla Profesor</b> .
	3.		El Sistema carga la <b>Lista De Profesores</b> y presenta los datos del primer Profesor
	4.	La <b>Secretaria</b> presiona el <b>Botón Nuevo</b> en la <b>Pantalla</b>	

	<u>Profesor.</u>	
	5.	El Sistema fija los datos en modo edición.
	6.	La <b>Secretaria</b> ingresa los datos del <b>Profesor</b> y presiona el <u>Botón Aceptar</u> .
	7.	El Sistema valida los datos ingresados.
	8.	El Sistema guarda los datos ingresados.
	9.	El caso de uso finaliza.
<b>Curso Alterno A</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Editar</u>.  El Sistema fija los datos en modo edición.  La <b>Secretaria</b> modifica los datos deseados y presiona el <u>Botón Aceptar</u>.  El Sistema valida los datos  El Sistema actualiza el registro del <b>Profesor</b>  El Sistema quita el modo edición de las entradas de datos</p>	
<b>Curso Alterno B</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Borrar</u>.  El Sistema le presenta un mensaje de confirmación de eliminación.  La <b>Secretaria</b> presiona el <u>Botón Si</u>.  El Sistema valida la eliminación.  El Sistema elimina el registro del <b>Profesor</b></p>	
<b>Curso Alterno C</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Buscar</u>.  El Sistema le presenta la Pantalla buscar &lt;&lt;include&gt;&gt; Buscar</p>	
<b>Curso Alterno D</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u>.  &lt;&lt;include&gt;&gt; CU35 – Caso de Uso Mostrar Ayuda</p>	
<b>Curso Alterno E</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Siguiente</u>.  El Sistema obtiene los datos del siguiente <b>Profesor</b> de la Lista  El Sistema presenta los Resultados</p>	
<b>Curso Alterno F</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Anterior</u>.  El Sistema obtiene los datos del <b>Profesor</b> anterior de la Lista  El Sistema presenta los Resultados</p>	
<b>Curso Alterno G</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Primero</u>.  El Sistema obtiene los datos del primer <b>Profesor</b> de la Lista  El Sistema presenta los Resultados</p>	
<b>Curso Alterno H</b>	<p>3. La <b>Secretaria</b> presiona el botón último.  El Sistema obtiene los datos del último <b>Profesor</b> de la Lista  El Sistema presenta los Resultados</p>	
<b>Curso Alterno I</b>	<p>3. La <b>Secretaria</b> presiona el <u>Botón Imprimir</u>  El Sistema genera el</p>	

## 7.3.1.21 CU21 – Caso de Uso Asignar Alumno a Paralelo

<b>Nombre</b>	<b>Asignar Alumno a Paralelo</b>		
<b>Actor</b>	<b>Secretaria.</b>		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF05</b> Crear de paralelos</li> <li>• <b>RQF07</b> Asignar alumnos a un curso y cambio de paralelos</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> crea los diferentes <b>Paralelos</b> y .asigna los diferentes <b>Alumnos</b> a determinado <b>Paralelo</b>		
<b>PRE Condiciones</b>			
<b>Post Condición</b>	Se asigna crea el <b>Paralelo</b> y le asigna <b>Alumnos</b> al <b>Paralelo</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <b>Materias</b> y selecciona el ítem <b>Asignar Alumno a Paralelo</b> de la <b>Pantalla Principal</b>	
	2.		El Sistema presenta la <b>Pantalla Asignar a Paralelo</b>
3.		El Sistema inicializa la <b>Lista de Paralelos</b>	

	4.	La <b>Secretaria</b> selecciona un <b>Curso</b> de la <u>lista desplegable Curso</u>	
	5.		El Sistema carga la lista de <b>Alumnos</b> que no tienen <b>Paralelo en la lista Alumnos sin Paralelo</b>
	6.		El Sistema carga la <u>lista de Paralelos</u>
	7.	La <b>Secretaria</b> selecciona el <b>Alumno(s)</b>	
	8.		El Sistema activa el <u>Botón Agregar</u>
	9.	La <b>Secretaria</b> presiona el <u>Botón Agregar</u>	
	10		El sistema agrega el (los) <b>Alumno(s) al Paralelo</b>
	11		El Sistema presenta los resultados
<b>Curso Alternativo A</b>	4, 7. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> . <<include>> CU35 – Caso de Uso Mostrar Ayuda.		
<b>Curso Alternativo B</b>	La <b>Secretaria</b> selecciona el(los) <b>Alumnos(s)</b> de la lista <b>Alumnos Asignados</b> El Sistema activa el <u>Botón Quitar</u> La <b>Secretaria</b> presiona el <u>Botón Quitar</u> . El Sistema quita el <b>Alumno del Paralelo</b>		
<b>Curso Alternativo C</b>	4, 7. La <b>Secretaria</b> presiona el <u>Botón Nuevo</u> El sistema pone las entradas de datos en modo edición La <b>Secretaria</b> ingresa los datos del <b>Paralelo</b> La <b>Secretaria</b> presiona el <u>Botón Aceptar</u> El Sistema valida los datos ingresados El Sistema guarda el <b>Paralelo</b>		
<b>Curso Alternativo D</b>	4, 7. La <b>Secretaria</b> presiona el <u>Botón Editar</u> El sistema pone las entradas de datos en modo edición La <b>Secretaria</b> edita los datos del <b>Paralelo</b> La <b>Secretaria</b> presiona el <u>Botón Aceptar</u> El Sistema valida los datos ingresados El Sistema actualiza el <b>Paralelo</b>		
<b>Curso Alternativo E</b>	4, 7. La <b>Secretaria</b> presiona el <u>Botón Borrar</u> El Sistema valida si no tiene <b>Alumnos</b> El Sistema presenta un Mensaje de Confirmación La <b>Secretaria</b> presiona el <u>Botón Si</u> El Sistema borra el <b>Paralelo</b>		
<b>Curso Alternativo F</b>	4, 7 La <b>Secretaria</b> presiona el <u>Botón Imprimir</u> El Sistema genera el <b>Reporte de Paralelo</b> <<include>> Imprimir <b>Reporte</b>		
<b>Curso Alternativo G</b>	4, 7. La <b>Secretaria</b> presiona el <u>Botón Buscar</u> <<include>> Buscar		

<b>Curso Alternativo H</b>	4, /. La <b>Secretaria</b> presiona el <u>Botón Primero</u> El Sistema obtiene los datos del primer <b>Paralelo</b> de la Lista El Sistema presenta los resultados
<b>Curso Alternativo J</b>	4, /. La <b>Secretaria</b> presiona el <u>Botón Siguiente</u> El Sistema obtiene los datos del siguiente <b>Paralelo</b> de la Lista El Sistema presenta los resultados
<b>Curso Alternativo K</b>	4, /. La <b>Secretaria</b> presiona el <u>Botón Anterior</u> El Sistema obtiene los datos del anterior <b>Paralelo</b> de la Lista El Sistema presenta los resultados
<b>Curso Alternativo L</b>	4, /. La <b>Secretaria</b> presiona el <u>Botón Ultimo</u> El Sistema obtiene los datos del ultimo <b>Paralelo</b> de la Lista El Sistema presenta los resultados

### 7.3.1.22 CU22 – Caso de Uso Asignar Paralelos a Especialidad

<b>Nombre</b>	Asignar <b>Paralelos a Especialidad</b>
<b>Actor</b>	<b>Secretaria</b>
<b>Requerimientos</b>	
<b>PRE Condiciones</b>	1. Que exista <b>Paralelos</b> 2. Que exista Al menos una <b>Especialidad</b>

Post Condición	Se asigno <b>Paralelos</b> a las diferentes <b>Especialidades</b>	
	Entrada del Actor	Respuesta del Sistema
<b>Curso Normal</b>	1. La <b>Secretaria</b> hace Clic en el menú <b>Materia</b> y Selecciona el ítem <b>Asignar Paralelos a Especialidad</b> de la <b>Pantalla Principal</b>	
	2.	El Sistema presenta la <b>Pantalla Asignar Paralelos A Especialidad</b> con las <b>Especialidades</b> existentes
	3. La <b>Secretaria</b> selecciona un <b>Curso</b> de la <b>lista desplegable Cursos</b>	
	4.	El Sistema presenta el listado de <b>Paralelos</b> para dicho <b>Curso</b>
	5. La <b>Secretaria</b> selecciona el <b>Paralelo</b> del listado <b>Paralelos</b>	
	6.	El Sistema activa el <b>Botón &gt;&gt;</b> ( <b>Agregar</b> )
	7. La <b>Secretaria</b> presiona el botón <b>&gt;&gt;</b> ( <b>Agregar</b> )	
	8.	El Sistema agrega los <b>Paralelos</b> seleccionados a la <b>Especialidad</b>
	9.	El Caso de Uso finaliza
<b>Curso Alterno A</b>	3, 5, 7. La <b>Secretaria</b> presiona el <b>Botón Ayuda</b> El Sistema presenta la <b>Pantalla Ayuda de Paralelos</b>	
<b>Curso Alterno B</b>	3. La <b>Secretaria</b> selecciona los <b>Paralelos</b> de la lista <b>Paralelos</b> asignados El Sistema activa el <b>Botón &lt;&lt;</b> ( <b>Quitar</b> ) La <b>Secretaria</b> presiona el <b>Botón &lt;&lt;</b> ( <b>Quitar</b> ) El Sistema quita de la <b>Especialidad</b> los <b>Paralelos</b> seleccionados	
<b>Curso Alterno C</b>	3. La <b>Secretaria</b> presiona el <b>Botón Primero</b> El Sistema carga la primera <b>Especialidad</b> existente	
<b>Curso Alterno D</b>	3. La <b>Secretaria</b> presiona el <b>Botón Anterior</b> El Sistema carga <b>Especialidad</b> anterior existente	
<b>Curso Alterno E</b>	3. La <b>Secretaria</b> presiona el <b>Botón Siguiente</b> El Sistema carga la siguiente <b>Especialidad</b> existente	
<b>Curso Alterno F</b>	3. La <b>Secretaria</b> presiona el <b>Botón Ultimo</b> El Sistema carga la ultima <b>Especialidad</b> existente	

## 7.3.1.23 CU23 – Caso de Uso Asignar Materia a Profesor

<b>Nombre</b>	Asignar <b>Materia</b> a <b>Profesor</b> .		
<b>Actor</b>	<b>Secretaria</b> .		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF09</b> Asignar las materias a los profesores</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> ingresa en el menú asignar <b>Materia</b> a <b>Profesor</b> y asigna una determinada <b>Materia</b> de un <b>Paralelo</b> y <b>Curso</b> específico al <b>Profesor</b> deseado.		
<b>PRE Condiciones</b>	<ol style="list-style-type: none"> <li>1. Que el <b>Profesor</b> exista en el Sistema.</li> <li>2. Que existan <b>Paralelos</b>.</li> <li>3. Que existan <b>Materias</b>.</li> </ol>		
<b>Post Condición</b>	La <b>Materia</b> queda asignada a un <b>Profesor</b> del Sistema.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	
	1.	La <b>Secretaria</b> hace clic en el menú <b>Profesor</b> y selecciona el ítem <b>Asignar Materia a Profesor</b> de la <b>Pantalla Principal</b> .	
	2.		
		<b>Respuesta del Sistema</b>	
	3.	La <b>Secretaria</b> hace clic sobre	El Sistema presenta la <b>Pantalla Asignar Materia a Profesor</b> .



		la <u>lista desplegable Curso</u> para seleccionar el <b>Curso</b> del cual se le va asignar la <b>Materia</b> al <b>Profesor</b> .	
	4.		El Sistema carga los <b>Paralelos</b> del <b>Curso</b> seleccionado
	5.	La <b>Secretaria</b> selecciona el paralelo de la <u>lista desplegable Paralelo</u>	
	6.		El Sistema carga las <b>Materias</b> asignadas del <b>Paralelo</b> seleccionado en la <u>tabla Materias</u>
	7.	La <b>Secretaria</b> selecciona de la <u>lista Profesores</u> el <b>Profesor</b> al que se desea asignar la <b>Materia</b> .	
	8.		El Sistema presenta el <b>Profesor</b> seleccionado.
	9.	La <b>Secretaria</b> selecciona la(s) <b>Materia(s)</b> a asignar al <b>Profesor</b> del listado de la <u>tabla Materias</u>	
	10		El Sistema activa el <u>Botón Agregar</u>
	11	La <b>Secretaria</b> presiona el <u>Botón Agregar</u>	
	12		El Sistema agrega la <b>Materia</b> al <b>Profesor</b> seleccionado.
	13		El caso de uso finaliza.
<b>Curso Alternativo A</b>	3, 5, 7,9. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> . <<include>> CU35 – Caso de Uso Mostrar Ayuda		
<b>Curso Alternativo B</b>	7. La <b>Secretaria</b> selecciona la <b>Materia</b> de la <u>lista Materias</u> asignadas y presiona el <u>Botón Quitar</u> . El Sistema quita las <b>Materias</b> del <b>Profesor</b> El Sistema actualiza la Pantalla y presenta la <b>Materia</b> seleccionada en la <u>lista Materias</u> .		

## 7.3.1.24 CU24 – Caso de Uso Registrar Asistencia

<b>Nombre</b>	Registrar Asistencias		
<b>Actor</b>	Secretaria		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF04 Registrar la asistencia de cada uno de los alumnos</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá registrar las acreditaciones de cada uno de los <b>Alumnos Matriculados</b> en la <b>Unidad Educativa</b>		
<b>PRE Condiciones</b>	<ol style="list-style-type: none"> <li>1. Que existan <b>Curso</b></li> <li>2. Que el <b>Curso</b> tenga <b>Paralelos</b></li> <li>3. Que existan <b>Profesor</b> asignados a dicho <b>Paralelo</b></li> <li>4. Que existan <b>Materias</b> Asignadas al <b>Paralelo</b></li> </ol>		
<b>Post Condición</b>	Se registro las Acreditaciones		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace Clic en el <u>menú Paralelo</u> y Selecciona el <u>ítem Asistencias</u> de la <u>Pantalla Principal</u>	
	2.		El Sistema presenta la <u>Pantalla Asistencias</u>
	3.		El Sistema inicializa la Lista de <u>Cursos</u> existentes en la <b>Unidad Educativa</b>
4.	La <b>Secretaria</b> selecciona la <u>fecha</u> , el <b>Curso</b> , el <b>Paralelo</b> , el		

		<b>Profesor y la Materia</b> en la cual se van a registrar las <b>Asistencias</b>	
	5.		El Sistema presenta el listado de los <b>Alumnos</b> del <b>Paralelo</b> seleccionado
	6.	La <b>Secretaria</b> presiona el <u>Botón Registrar</u>	
	7.		El Sistema carga y presenta el listado de <b>Alumnos</b> y sus <b>Asistencias</b> en modo edición
	8.	La <b>Secretaria</b> registra cada una de las <b>Asistencias</b> de los diferentes <b>Alumnos</b> y presiona el <u>Botón Guardar</u>	
	9.		El Sistema valida los datos ingresados
	10.		El Sistema guarda los datos ingresados
	11.		El Sistema quita el modo edición del listado de <b>Alumnos</b> y sus <b>Asistencias</b>
	12.	El caso de Uso Finaliza	
<b>Curso Alterno A</b>	3, 5, 7, 8. La <b>Secretaria</b> presiona el <u>Botón Ayuda</u> El Sistema presenta la Pantalla Ayuda de <b>Asistencias</b>		
<b>Curso Alterno B</b>	6. El Sistema presenta un mensaje de error “No se pueden registrar <b>Asistencias</b> de Sábados y Domingos”		

## 7.3.1.25 CU25 – Caso de Uso Registrar Acreditación

<b>Nombre</b>	Registrar Acreditación		
<b>Actor</b>	Secretaria		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF03 Registrar la acreditación de cada uno de los alumnos</li> </ul>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá registrar las acreditaciones de cada uno de los <b>Alumnos Matriculados</b> en la <b>Unidad Educativa</b>		
<b>Pre Condiciones</b>	<ol style="list-style-type: none"> <li>1. Que exista al menos un <b>Curso</b></li> <li>2. Que el <b>Curso</b> tenga <b>Paralelos</b></li> <li>3. Que existan <b>Profesor</b> asignados a dicho <b>Paralelo</b></li> <li>4. Que existan <b>Materias</b> Asignadas al <b>Paralelo</b> y a dicho <b>Profesor</b></li> <li>5. Que el <b>Curso</b> este Configurado</li> <li>6. Que existan <b>Alumnos</b> en el <b>Paralelo</b></li> </ol>		
<b>Post Condición</b>	Se registro las Acreditaciones		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace Clic en el <u>menú Paralelo</u> y Selecciona el <u>ítem Acreditación</u> de la <u>Pantalla Principal</u>	
	2.		El Sistema presenta la <u>Pantalla Acreditación</u> e inicializa la <u>Lista Curso</u> con la Lista de <b>Cursos</b> existentes en la <b>Unidad Educativa</b>
3.	La <b>Secretaria</b> selecciona el <b>Curso</b> ,		

4.		El Sistema carga y presenta los <b>Paralelos</b> del <b>Curso</b> seleccionado
5.		El Sistema carga la <b>Configuración</b> para dicho <b>Curso</b>
6.		El Sistema presenta los <b>Periodos</b> de dicho <b>Curso</b>
7.		El Sistema presenta los <b>Reportes</b> de los <b>Periodos</b>
8.	La <b>Secretaria</b> selecciona un <b>Paralelo</b>	
9.		El Sistema carga el Listado de <b>Alumnos</b> del <b>Paralelo</b>
10.		El Sistema carga los <b>Profesores</b> asignados a dicho <b>Paralelo</b>
11.	La <b>Secretaria</b> selecciona un <b>Profesor</b>	
12.		El Sistema carga y Presenta las <b>Materias</b> del <b>Paralelo</b> y <b>Profesor</b> seleccionados
13.	La <b>Secretaria</b> selecciona una <b>Materia</b>	
14.	La <b>Secretaria</b> selecciona el <b>Periodo</b>	
15.		El Sistema carga y presenta los <b>Reportes</b> para dicho <b>Periodo</b>
16.	La <b>Secretaria</b> selecciona un <b>Reporte</b>	
17.		El Sistema presenta el listado de los <b>Alumnos</b> del <b>Paralelo</b> seleccionado, con sus respectivos <b>Dominios</b> y <b>acreditaciones</b>
18.	La <b>Secretaria</b> presiona el <b>Botón Registrar</b>	
19.		El Sistema pone el <u>listado de Alumnos</u> y sus <u>Acreditaciones</u> en modo edición
20.	La <b>Secretaria</b> ingresa cada una de las <u>acreditaciones</u> de los diferentes <b>Alumnos</b> en cada uno de los <b>Dominios</b> y presiona el <u>Botón Guardar</u>	
21.		El Sistema valida los datos ingresados
22.		El Sistema guarda las

		acreditaciones ingresadas
	23.	El Sistema registra la <b>Acreditación</b> de los Dominios y calcula las acreditaciones del <b>Reporte</b> , el <b>Periodo</b> y el <b>Promedio</b> general basado en la <b>Configuración</b>
	24.	El Sistema quita el modo edición del listado de <b>Alumnos</b> y sus <b>Acreditaciones</b>
	25.	El caso de Uso Finaliza
<b>Curso Alterno A</b>	21. El Sistema presenta un Mensaje de error “Acreditaciones no validas”	
<b>Curso Alterno B</b>	3, 9, 11, 14,16, 18, 20, 21. La <b>Secretaria</b> presiona el <b>Botón Ayuda</b> <<include>> Mostrar Ayuda	

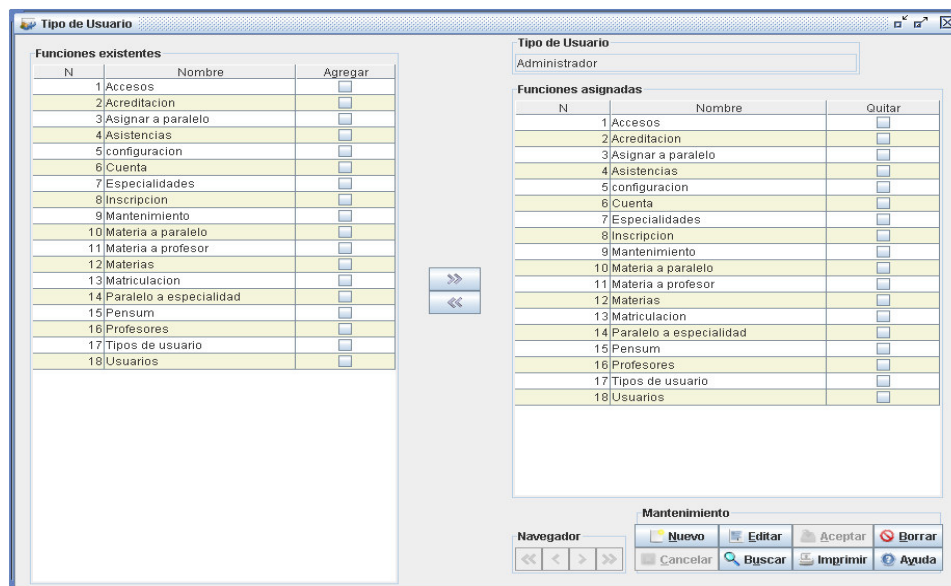
### 7.3.1.26 CU26 – Caso de Uso Mantener Tipo de Usuario

<b>Nombre</b>	Mantener Tipo de Usuario
<b>Actor</b>	<b>Administrador</b>
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF35</b> Crear tipos de usuarios, usuarios propiamente, los cuales van a tener determinado acceso a ciertas funciones</li> </ul>
<b>Breve Descripción</b>	El <b>Administrador</b> podrá crear diferentes Tipos de <b>Usuarios</b>

<b>PRE</b>	Que el usuario al que se desea asignar funciones exista.	
<b>Condiciones</b>	Que el usuario que asigna las funciones tenga permisos para hacerlo.	
<b>Post Condición</b>	Se creo un <b>Tipo de Usuarios</b>	
<b>Curso Normal</b>		<b>Entrada del Actor</b>
		<b>Respuesta del Sistema</b>
	1.	El <b>Administrador</b> hace Clic en el menú <b>Mantener Usuarios</b> y Selecciona el <u>ítem Tipo de Usuario</u> de la <u>Pantalla Principal</u>
	2.	El Sistema presenta la <u>Pantalla Tipo de Usuario</u>
	3.	El Sistema inicializa la Lista de Tipos de <b>Usuarios</b> existentes
	4.	El <b>Administrador</b> presiona el <u>Botón Nuevo</u> de la <u>Pantalla Tipo de Usuario</u>
	5.	El Sistema fija el campo <b>Tipo de Usuario</b> en modo edición
	6.	El <b>Administrador</b> ingresa los datos del <b>Tipo de Usuario</b> y presiona el <u>Botón Aceptar</u>
	7.	El Sistema valida los datos ingresados
	8.	El Sistema guarda los datos ingresados
	9.	El Sistema quita el modo edición de las entradas de datos
10.	<<include>> Asignar <b>Función</b>	
11.	El caso de Uso Finaliza	
<b>Curso Alterno A</b>	3. El <b>Administrador</b> presiona el <u>Botón Editar</u> El Sistema fija las entradas de datos en modo edición El <b>Administrador</b> Edita los datos de <b>Tipo de Usuarios</b> El Sistema valida los datos ingresado El Sistema actualiza el <b>Tipo de Usuarios</b>	
<b>Curso Alterno B</b>	3. El <b>Administrador</b> presiona el <u>Botón Borrar</u> El Sistema presenta un mensaje de confirmación El <b>Administrador</b> presiona el <u>Botón Si</u> El Sistema borra el <b>Tipo de Usuarios</b>	
<b>Curso Alterno C</b>	3.El <b>Administrador</b> presiona el <u>Botón Primero</u> El Sistema obtiene los datos del primer <b>Tipo de Usuarios</b> de la Lista El Sistema presenta los resultados	
<b>Curso Alterno D</b>	3.El <b>Administrador</b> presiona el <u>Botón Anterior</u> El Sistema obtiene los datos del <b>Tipo de Usuarios</b> anterior de la Lista El Sistema presenta los resultados	
<b>Curso Alterno E</b>	3.El <b>Administrador</b> presiona el <u>Botón Siguiente</u> El Sistema obtiene los datos del siguiente <b>Tipo de Usuarios</b> de la	

	Lista El Sistema presenta los resultados
<b>Curso Alterno F</b>	3.El <b>Administrador</b> presiona el <u>Botón Ultimo</u> El Sistema obtiene los datos del ultimo <b>Tipo de Usuarios</b> de la Lista El Sistema presenta los resultados
<b>Curso Alterno G</b>	3. El <b>Administrador</b> presiona el <u>Botón Buscar</u> <<include>> buscar
<b>Curso Alterno H</b>	3. el <b>Administrador</b> presiona el <u>Botón Ayuda</u> El Sistema presenta la Pantalla Ayuda de Tipos de <b>Usuarios</b>

### 7.3.1.27 CU27 – Caso de Uso Asignar Función



<b>Nombre</b>	Asignar Función	
<b>Actor</b>	Administrador	
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF35</b> Crear tipos de usuarios, usuarios propiamente, los cuales van a tener determinado acceso a ciertas funciones</li> </ul>	
<b>Breve Descripción</b>	El <b>Administrador</b> podrá asignar funciones a los diferentes Tipos de <b>Usuarios</b>	
<b>PRE Condiciones</b>	Que exista al menos un <b>Tipo de Usuarios</b> Creado	
<b>Post Condición</b>	Se asigno funciones a un <b>Tipo de Usuarios</b>	
<b>Curso Normal</b>	<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>1. Selecciona el <u>ítem Tipo de Usuario</u> de la <u>Pantalla Principal</u></li> <li>2. El <b>Administrador</b> selecciona</li> </ol>	



		las funciones de la <u>lista Funciones Existentes</u>	
	3.		El Sistema activa el <u>botón &gt;&gt;</u> (Agregar)
	4.	El <b>Administrador</b> presiona el <u>botón &gt;&gt;</u> (Agregar)	
	5.		El Sistema asigna las funciones seleccionadas al <b>Tipo de Usuarios</b> seleccionado
	6.		El Sistema presenta en el listado <u>Funciones asignadas</u> las funciones del <b>Tipo de Usuarios</b> asignadas
	7.	El caso de uso finaliza	
<b>Curso Alterno A</b>	1. El <b>Administrador</b> selecciona funciones del listado de Funciones asignadas El Sistema activa el <u>Botón &lt;&lt;</u> (Quitar) El <b>Administrador</b> presiona el <u>Botón &lt;&lt;</u> (Quitar) El Sistema quita las funciones seleccionadas del <b>Tipo de Usuarios</b>		

### 7.3.1.28 CU28 – Caso de Uso Mantener Usuarios

<b>Nombre</b>	Mantener Usuarios
<b>Breve</b>	El <b>Administrador</b> podrá crear diferentes <b>Usuarios</b>

<b>Descripción</b>																																					
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF37</b> Mantener los datos de la cuenta de cada uno de los usuarios</li> </ul>																																				
<b>PRE Condiciones</b>	Que exista al menos un <b>Tipo de Usuarios</b>																																				
<b>Post Condición</b>	Se creó un <b>Usuario</b>																																				
<b>Curso Normal</b>	<table border="1"> <thead> <tr> <th></th> <th><b>Entrada del Actor</b></th> <th><b>Respuesta del Sistema</b></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>El <b>Administrador</b> hace Clic en el menú <b>Mantener Usuarios</b> y Selecciona el ítem <u>Usuarios</u> de la <u>Pantalla Principal</u></td> <td></td> </tr> <tr> <td>2.</td> <td></td> <td>El Sistema presenta la <u>Pantalla Usuarios</u></td> </tr> <tr> <td>3.</td> <td></td> <td>El Sistema inicializa Los tipos de <b>Usuarios</b> e inicializa la lista de <b>Usuarios</b> para el <b>Tipo de Usuario</b> seleccionado</td> </tr> <tr> <td>4.</td> <td>El <b>Administrador</b> selecciona un <b>Tipo de Usuarios</b> de la lista desplegable <u>Tipos de Usuarios</u></td> <td></td> </tr> <tr> <td>5.</td> <td>El <b>Administrador</b> presiona el <u>Botón Nuevo</u></td> <td></td> </tr> <tr> <td>6.</td> <td></td> <td>El Sistema actualiza y pone las entradas de datos en modo edición</td> </tr> <tr> <td>7.</td> <td>El <b>Administrador</b> ingresa los datos del <b>Usuarios</b> y presiona el <u>Botón Aceptar</u></td> <td></td> </tr> <tr> <td>8.</td> <td></td> <td>El Sistema valida los datos ingresados</td> </tr> <tr> <td>9.</td> <td></td> <td>El Sistema guarda el <b>Usuario</b></td> </tr> <tr> <td>10.</td> <td></td> <td>El Sistema quita el modo edición de las entradas de datos</td> </tr> <tr> <td>11.</td> <td>El caso de Uso Finaliza</td> <td></td> </tr> </tbody> </table>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>	1.	El <b>Administrador</b> hace Clic en el menú <b>Mantener Usuarios</b> y Selecciona el ítem <u>Usuarios</u> de la <u>Pantalla Principal</u>		2.		El Sistema presenta la <u>Pantalla Usuarios</u>	3.		El Sistema inicializa Los tipos de <b>Usuarios</b> e inicializa la lista de <b>Usuarios</b> para el <b>Tipo de Usuario</b> seleccionado	4.	El <b>Administrador</b> selecciona un <b>Tipo de Usuarios</b> de la lista desplegable <u>Tipos de Usuarios</u>		5.	El <b>Administrador</b> presiona el <u>Botón Nuevo</u>		6.		El Sistema actualiza y pone las entradas de datos en modo edición	7.	El <b>Administrador</b> ingresa los datos del <b>Usuarios</b> y presiona el <u>Botón Aceptar</u>		8.		El Sistema valida los datos ingresados	9.		El Sistema guarda el <b>Usuario</b>	10.		El Sistema quita el modo edición de las entradas de datos	11.	El caso de Uso Finaliza	
		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>																																		
	1.	El <b>Administrador</b> hace Clic en el menú <b>Mantener Usuarios</b> y Selecciona el ítem <u>Usuarios</u> de la <u>Pantalla Principal</u>																																			
	2.		El Sistema presenta la <u>Pantalla Usuarios</u>																																		
	3.		El Sistema inicializa Los tipos de <b>Usuarios</b> e inicializa la lista de <b>Usuarios</b> para el <b>Tipo de Usuario</b> seleccionado																																		
	4.	El <b>Administrador</b> selecciona un <b>Tipo de Usuarios</b> de la lista desplegable <u>Tipos de Usuarios</u>																																			
	5.	El <b>Administrador</b> presiona el <u>Botón Nuevo</u>																																			
	6.		El Sistema actualiza y pone las entradas de datos en modo edición																																		
	7.	El <b>Administrador</b> ingresa los datos del <b>Usuarios</b> y presiona el <u>Botón Aceptar</u>																																			
	8.		El Sistema valida los datos ingresados																																		
	9.		El Sistema guarda el <b>Usuario</b>																																		
10.		El Sistema quita el modo edición de las entradas de datos																																			
11.	El caso de Uso Finaliza																																				
<b>Curso Alternativo A</b>	<p>4. El <b>Administrador</b> presiona el <u>Botón Editar</u>  El Sistema fija las entradas de datos en modo edición  El <b>Administrador</b> Edita los datos del <b>Usuarios</b> y presiona el <u>Botón Aceptar</u>  El Sistema valida los datos ingresado  El Sistema actualiza del <b>Usuarios</b>  El Sistema quita el modo edición de las entradas de datos</p>																																				
<b>Curso Alternativo B</b>	<p>4. El <b>Administrador</b> presiona el <u>Botón Borrar</u>  El Sistema presenta un mensaje de confirmación  El <b>Administrador</b> presiona el <u>Botón Si</u>  El Sistema borra el <b>Usuario</b></p>																																				

<b>Curso Alterno C</b>	4.El <b>Administrador</b> presiona el <u>Botón Primero</u> El Sistema obtiene los datos del primer <b>Usuarios</b> de la Lista <b>Usuarios</b> El sistema presenta los resultados
<b>Curso Alterno D</b>	4.El <b>Administrador</b> presiona el <u>Botón Anterior</u> El Sistema obtiene los datos del <b>Usuarios</b> anterior de la Lista <b>Usuarios</b> El sistema presenta los resultados
<b>Curso Alterno E</b>	4.El <b>Administrador</b> presiona el <u>Botón Siguiente</u> El Sistema obtiene los datos del siguiente <b>Usuarios</b> de la Lista <b>Usuarios</b> El sistema presenta los resultados
<b>Curso Alterno F</b>	4.El <b>Administrador</b> presiona el <u>Botón Ultimo</u> El Sistema obtiene los datos del ultimo <b>Usuarios</b> de la Lista <b>Usuarios</b> El sistema presenta los resultados
<b>Curso Alterno G</b>	4. El <b>Administrador</b> presiona el <u>Botón Buscar</u> <<include>> buscar
<b>Curso Alterno H</b>	4. el <b>Administrador</b> presiona el <u>Botón Ayuda</u> El Sistema presenta la Pantalla Ayuda de <b>Usuarios</b>

### 7.3.1.29 CU29 – Caso de Uso Mantener Cuenta

The screenshot shows a web application window titled "Cuenta". The window contains a form for user information. The form is divided into two main sections: "Información del Usuario" and "Dirección".

**Información del Usuario:**

- User Name:** Administrador
- Nombre:** Edison
- Apellido:** Coronel
- Clave Anterior:** \*\*\*\*\*
- Nueva Clave:** \*\*\*\*\*
- Confirmar nueva clave:** \*\*\*\*\*

**Dirección:**

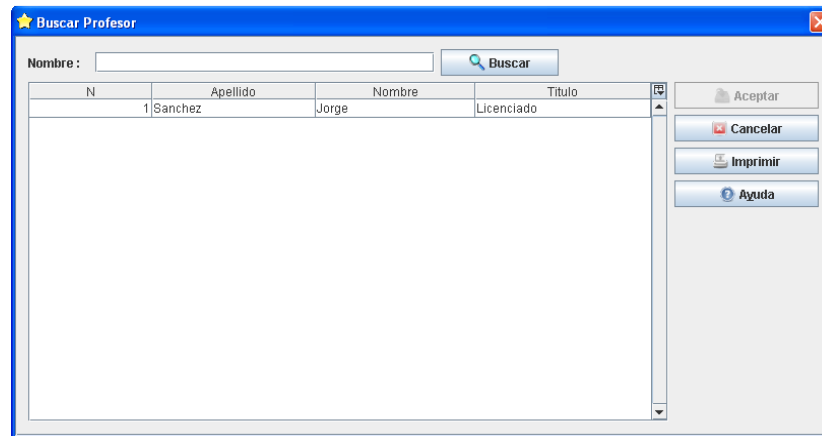
- Barrio:** [Empty text box]
- Calles:** [Empty text box]
- Telefono:** [Empty text box]
- e-mail:** [Empty text box]

At the bottom of the form, there are four buttons: "Editar", "Cancelar", "Aceptar", and "Ayuda".

<b>Nombre</b>	Mantener Cuenta
---------------	-----------------

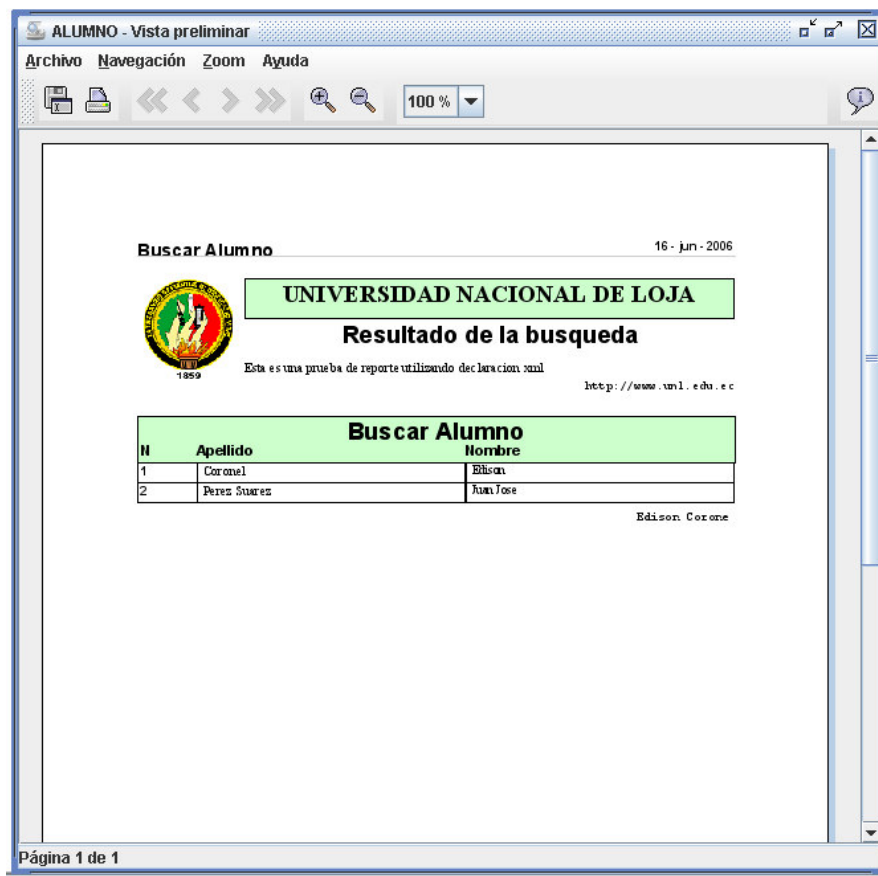
<b>Actor</b>	<b>Usuario</b>		
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• <b>RQF37</b> Mantener los datos de la cuenta de cada uno de los usuarios</li> </ul>		
<b>Breve Descripción</b>	El <b>Usuarios</b> podrá editar los datos de su Cuenta		
<b>Pre Condiciones</b>	Que sea un <b>Usuario</b> registrado		
<b>Post Condición</b>	Se edito la Cuenta del <b>Usuarios</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	El <b>Usuario</b> hace Clic en el menú <b>Usuarios</b> y Selecciona el ítem <b>Cuenta</b> de la <b>Pantalla Principal</b>	
	2.		El Sistema presenta la <b>Pantalla Cuenta</b> y la inicializa con los datos del <b>Usuario</b> que ingresó al Sistema
	3.	El <b>Usuario</b> presiona el <b>Botón Editar</b> de la <b>Pantalla Cuenta</b>	
	4.		El Sistema actualiza los campos en modo edición
	5.	El <b>Usuarios</b> ingresa los datos de su Cuenta y presiona el <b>Botón Aceptar</b> de la <b>Pantalla Cuenta</b>	
	6.		El Sistema valida los datos ingresados
	7.		El Sistema actualiza los datos ingresados
	8.		El Sistema quita el modo edición de las entradas de datos
	9.	El caso de Uso Finaliza	
<b>Curso Alternativo A</b>	3, 6, 7, el <b>Administrador</b> presiona el <b>Botón Ayuda</b> El Sistema presenta la Pantalla Ayuda de Cuenta		

### 7.3.1.30 FR01 – Fragmento Buscar



<b>Nombre</b>	Buscar.		
<b>Actor</b>	<b>Secretaria.</b>		
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá buscar todo tipo de elementos que se hayan ingresado en el Sistema, pero solo los que tienen relación al tópicos en el que se encuentre como activo.		
<b>PRE Condiciones</b>	1. Que se haya ingresado datos en el Sistema.		
<b>Post Condición</b>	2. Se presenten los datos en la Pantalla deseada.		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.		El Sistema presenta la <u>Pantalla Buscar.</u>
	2.		El Sistema carga el criterio de búsqueda en la <u>Pantalla Buscar</u>
	3.	La <b>Secretaria</b> ingresa en el campo nombre el texto de la consulta deseada y presiona el <u>Botón Buscar.</u>	
	4.		El Sistema valida la consulta ingresada.
	5.		El Sistema presenta los datos coincidentes.
	6.	La <b>Secretaria</b> selecciona de las coincidencias encontradas el objeto buscado y presiona el <u>Botón Aceptar.</u>	
	7.		El Sistema carga el objeto seleccionado
8.		El caso de uso finaliza.	
<b>Curso Alternativo</b>	6. La <b>Secretaria</b> presiona el <u>Botón Imprimir.</u> El Sistema genera el <b>Reporte</b> de búsqueda <<include>> Imprimir <b>Reporte.</b>		

## 7.3.1.31 FR02 – Fragmento imprimir Reporte



<b>Nombre</b>	Imprimir Reporte	
<b>Actor</b>	Secretaria.	
<b>Requerimientos</b>	<ul style="list-style-type: none"> <li>• RQF17 Consultar e imprimir reportes de los alumnos</li> <li>• RQF18 Consultar e imprimir reportes de los profesores</li> </ul>	
<b>Breve Descripción</b>	La <b>Secretaria</b> podrá imprimir todos los tópicos que conforman el Sistema de gestión académica según como sea su requerimiento y el tópico que se encuentre activo.	
<b>PRE Condiciones</b>	<ul style="list-style-type: none"> <li>• Que existan datos ingresados en el Sistema.</li> <li>• Que el reporte haya sido generado previamente</li> <li>• Un <b>Usuario</b> del sistema acceda a la opción imprimir en cualquier pantalla del Sistema</li> </ul>	
<b>Post Condición</b>	Se imprimió el <b>Reporte</b> requerido.	
<b>Curso Normal</b>		<b>Entrada del Actor</b>
	1.	
	2.	La <b>Secretaria</b> presiona el botón <u>imprimir</u> .
3.		<b>Respuesta del Sistema</b>
		El Sistema presenta la Pantalla <u>vista preliminar</u> del comprobante a emitir.
		El Sistema presenta el cuadro de dialogo Imprimir

	4.	La <b>Secretaria</b> presiona el <b>Botón Aceptar</b>	
	5.		El Sistema imprime el <b>Reporte.</b>
	6.		El caso de uso finaliza.
<b>Curso Alterno A</b>	3. La <b>Secretaria</b> presiona el icono guardar. El Sistema presenta el cuadro de dialogo Guardar Como La <b>Secretaria</b> ingresa el nombre y presiona el <b>Botón Aceptar</b> El Sistema guarda el <b>Reporte</b>		
<b>Curso Alterno B</b>	3. La <b>Secretaria</b> presiona el botón Acerca De El Sistema presenta el cuadro de Dialogo Acerca de...		

### 7.3.1.32 FR03 –Fragmento Administrar Ayuda

**Ayuda del SGA**

**INSCRIPCIÓN**

- Al seleccionar la opción Inscripción se presenta la siguiente pantalla:

**Datos del Alumno**

Fecha

Fecha

Datos Alumno

Nombres

Apellidos

Lugar De Nacimiento

Vive con  Padre y Madre  Padre  Madre  Abuelos  Otros

Institución de procedencia

Fecha de Nacimiento

Año: 1978 Mes: Enero Día: 1

Dirección:

Barrio:

Calles:

Telefono:

e-mail:

Navegador

Mantenimiento

- La pantalla inscripción consta de las siguientes pestañas: Alumnos y Padres.
- La pestaña Alumnos consta de los siguientes bloques:
  1. Fecha, se presenta la fecha actual del sistema, este campo es solo informativo no puede ser modificado.

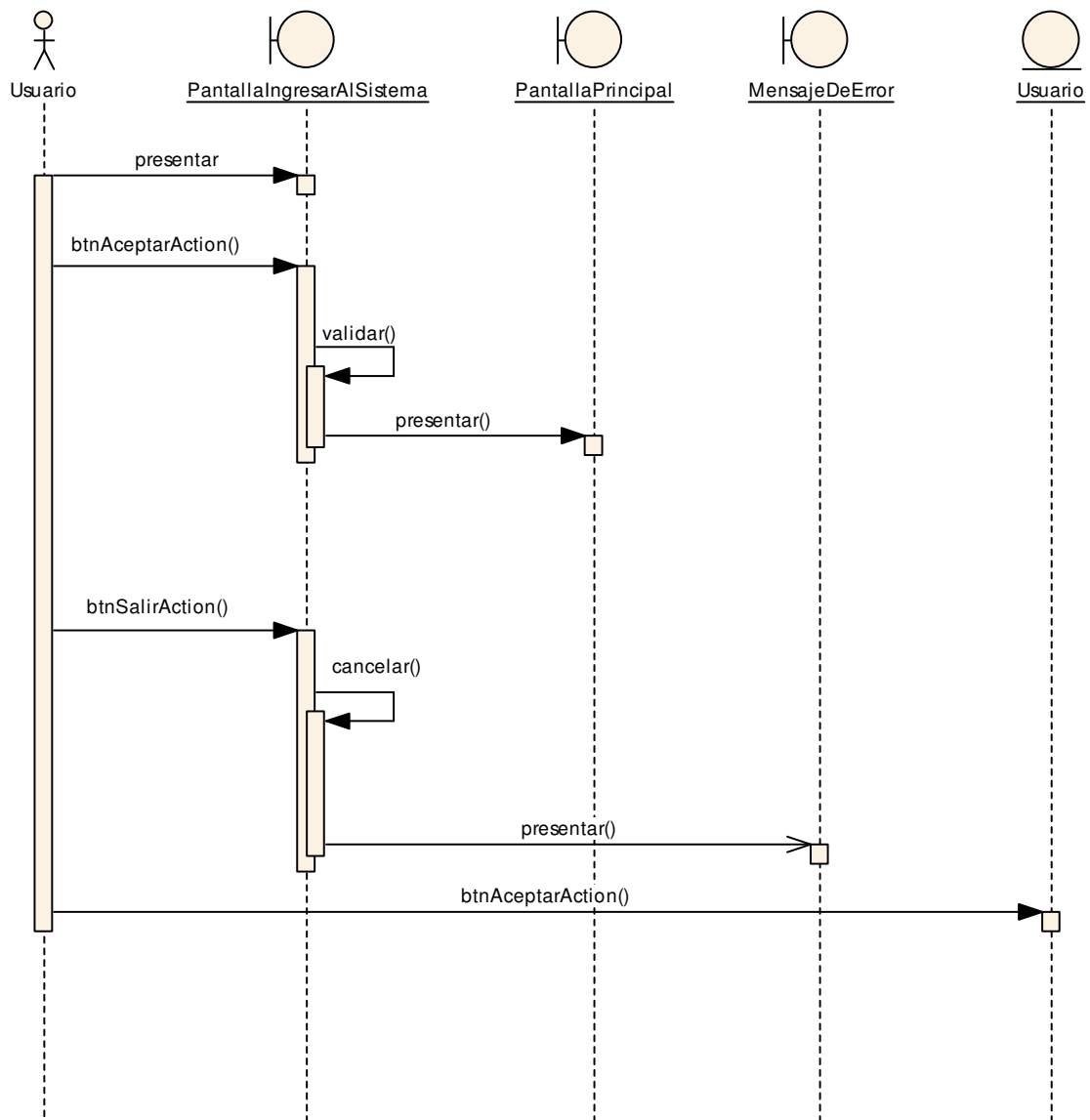
<b>Nombre</b>	Administrar Ayuda.		
<b>Actor</b>	<b>Secretaria.</b>		
<b>Breve Descripción</b>	La <b>Secretaria</b> ingresa en el menú ayuda y selecciona el ítem ayuda cuando desee información del funcionamiento de un determinado tópico.		
<b>PRE Condiciones</b>			
<b>Post Condición</b>	Se asigna la <b>Materia</b> a un <b>Paralelo.</b>		
<b>Curso Normal</b>		<b>Entrada del Actor</b>	<b>Respuesta del Sistema</b>
	1.	La <b>Secretaria</b> hace clic en el menú <b>ayuda</b> y selecciona el	

	ítem ayuda.	
	2.	El Sistema presenta la <u>Pantalla de Ayuda</u>
	3.	La <b>Secretaria</b> hace clic sobre uno de los tópicos mostrados en la pestaña tabla de contenidos.
	4.	El Sistema presenta el detalle del tópico seleccionado.
<b>Curso Alterno A</b>	<p>La <b>Secretaria</b> hace clic en la pestaña buscar.</p> <p>La <b>Secretaria</b> ingresa el criterio de búsqueda deseado y presiona la tecla enter.</p> <p>El Sistema presenta las coincidencias del criterio de búsqueda.</p>	
<b>Curso Alterno B</b>	<p>La <b>Secretaria</b> hace clic en la pestaña favoritos.</p> <p>El Sistema presenta los tópicos agregados en favoritos.</p>	
<b>Curso Alterno C</b>	<p>La <b>Secretaria</b> presiona el <u>Botón Imprimir</u>.</p> <p>El Sistema presenta la Pantalla de impresión.</p>	
<b>Curso Alterno D</b>	<p>La <b>Secretaria</b> presiona el <u>Botón Anterior</u>.</p> <p>El Sistema presenta el tópico anterior.</p>	
<b>Curso Alterno E</b>	<p>La <b>Secretaria</b> presiona el <u>Botón Siguiente</u>.</p> <p>El Sistema presenta el tópico siguiente.</p>	

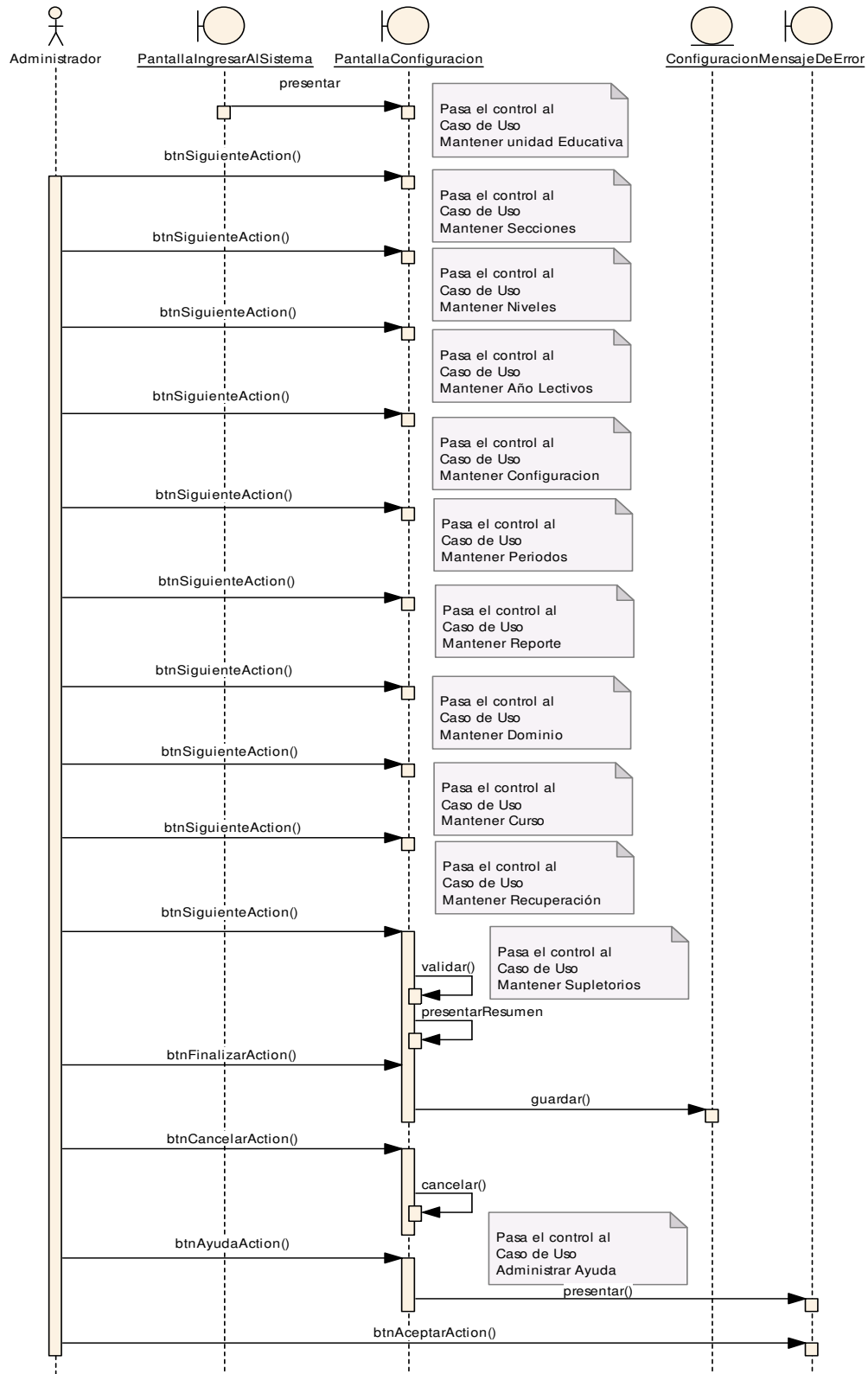


## 7.3.2 Diagramas de Secuencia.

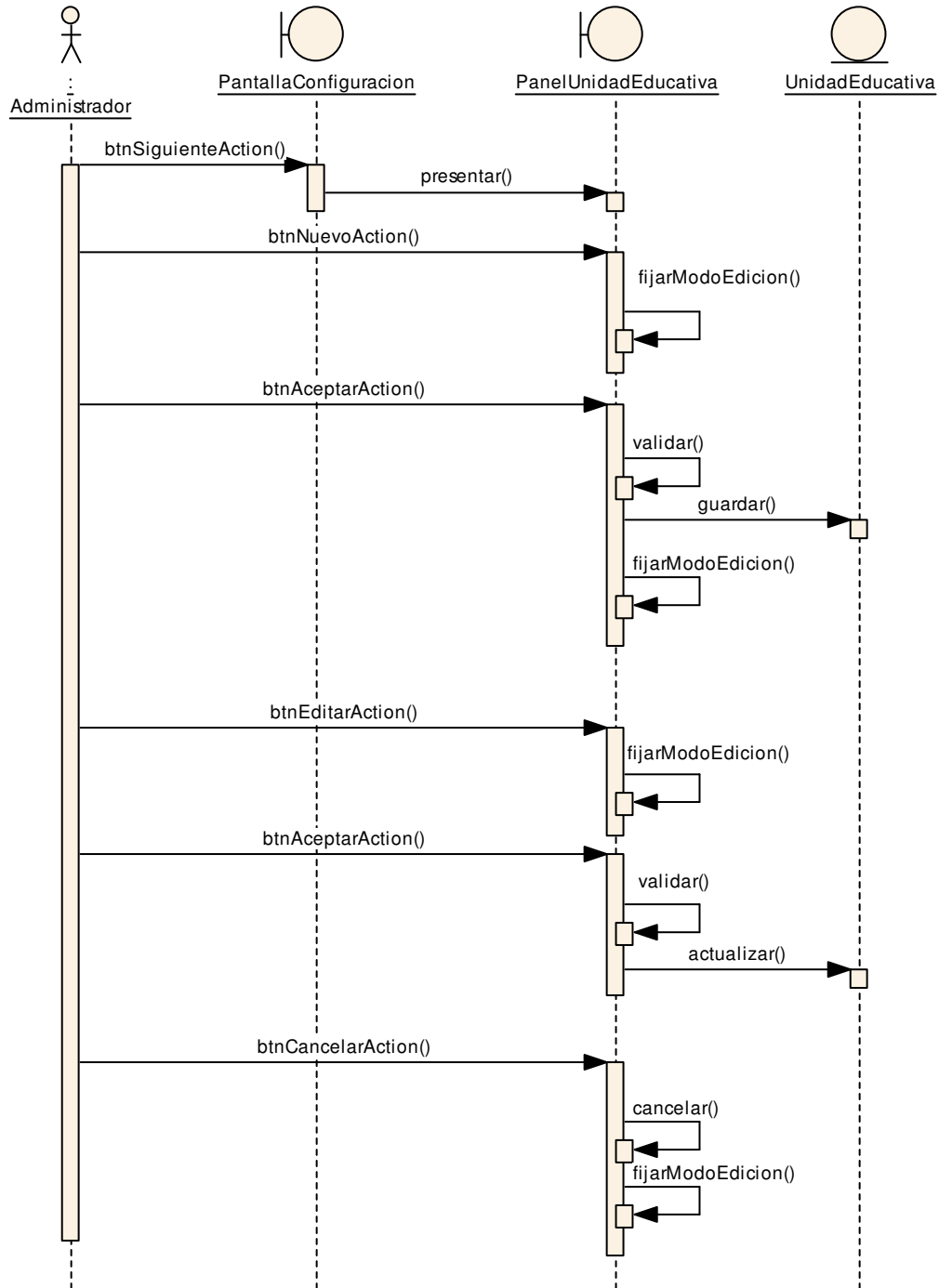
### 7.3.2.1 DS01-DR01 Ingresar al Sistema



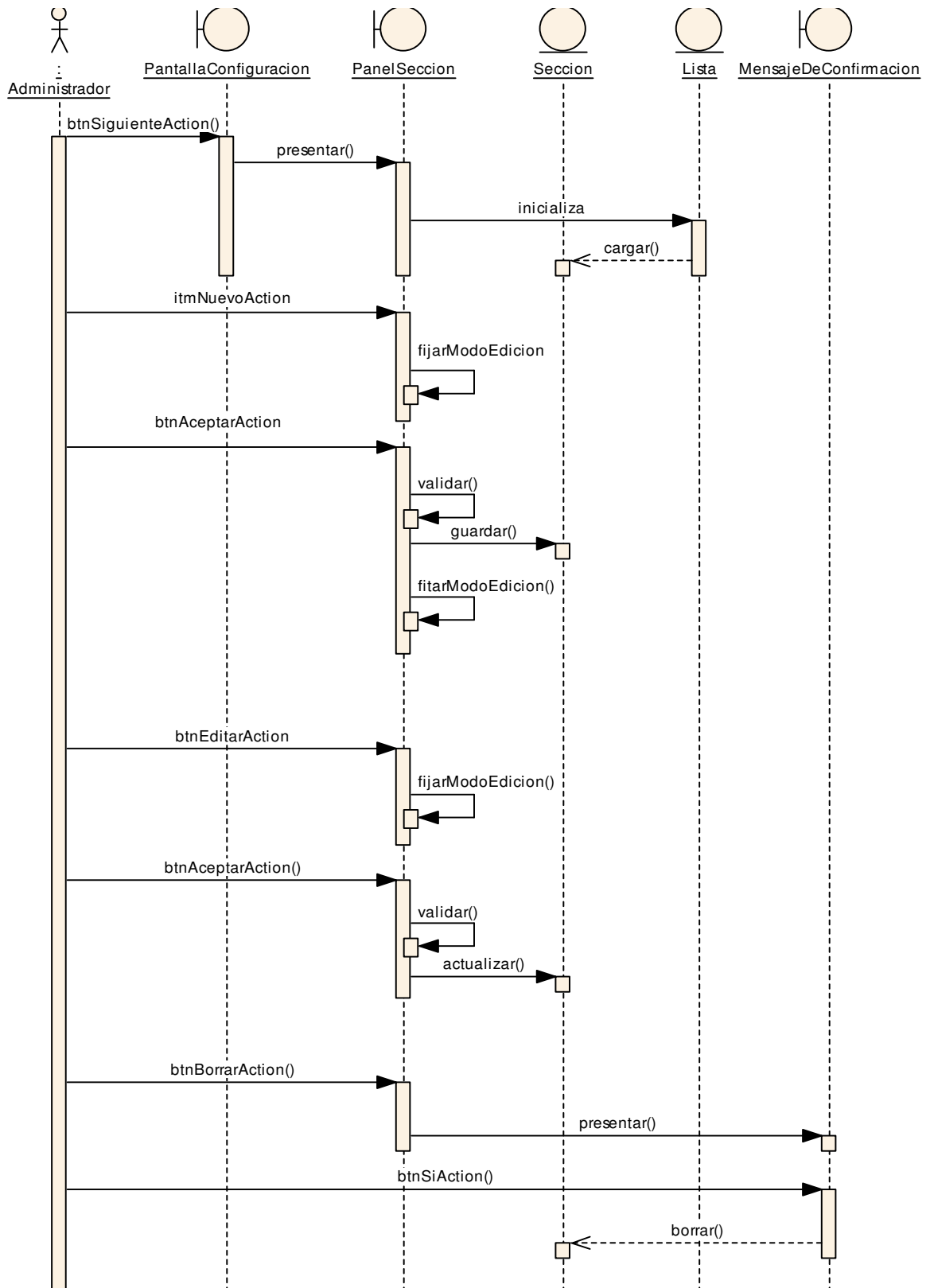
### 7.3.2.2 DS02- DR02 Configurar Sistema

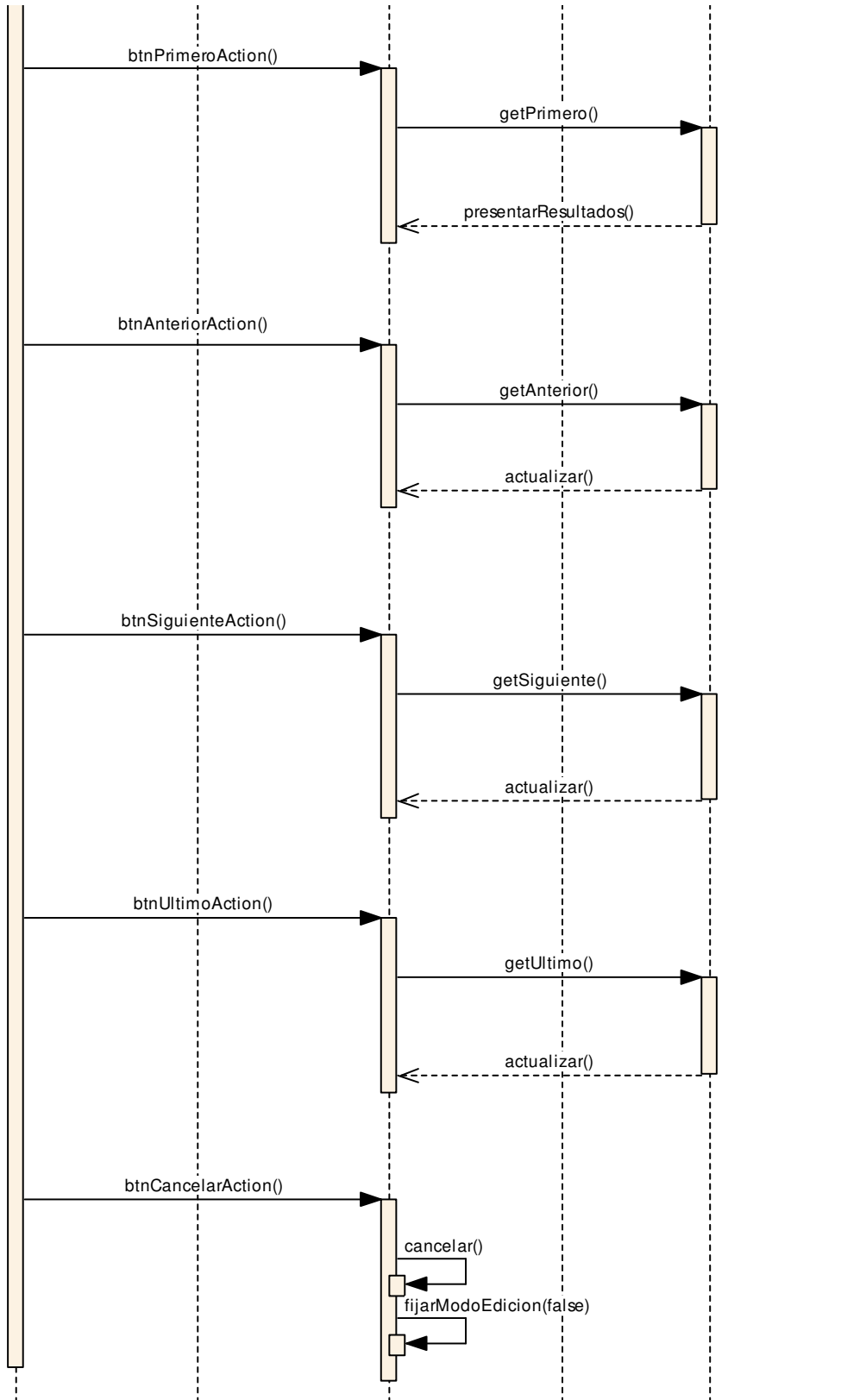


### 7.3.2.3 DS03-DR03 Mantener Unidad Educativa

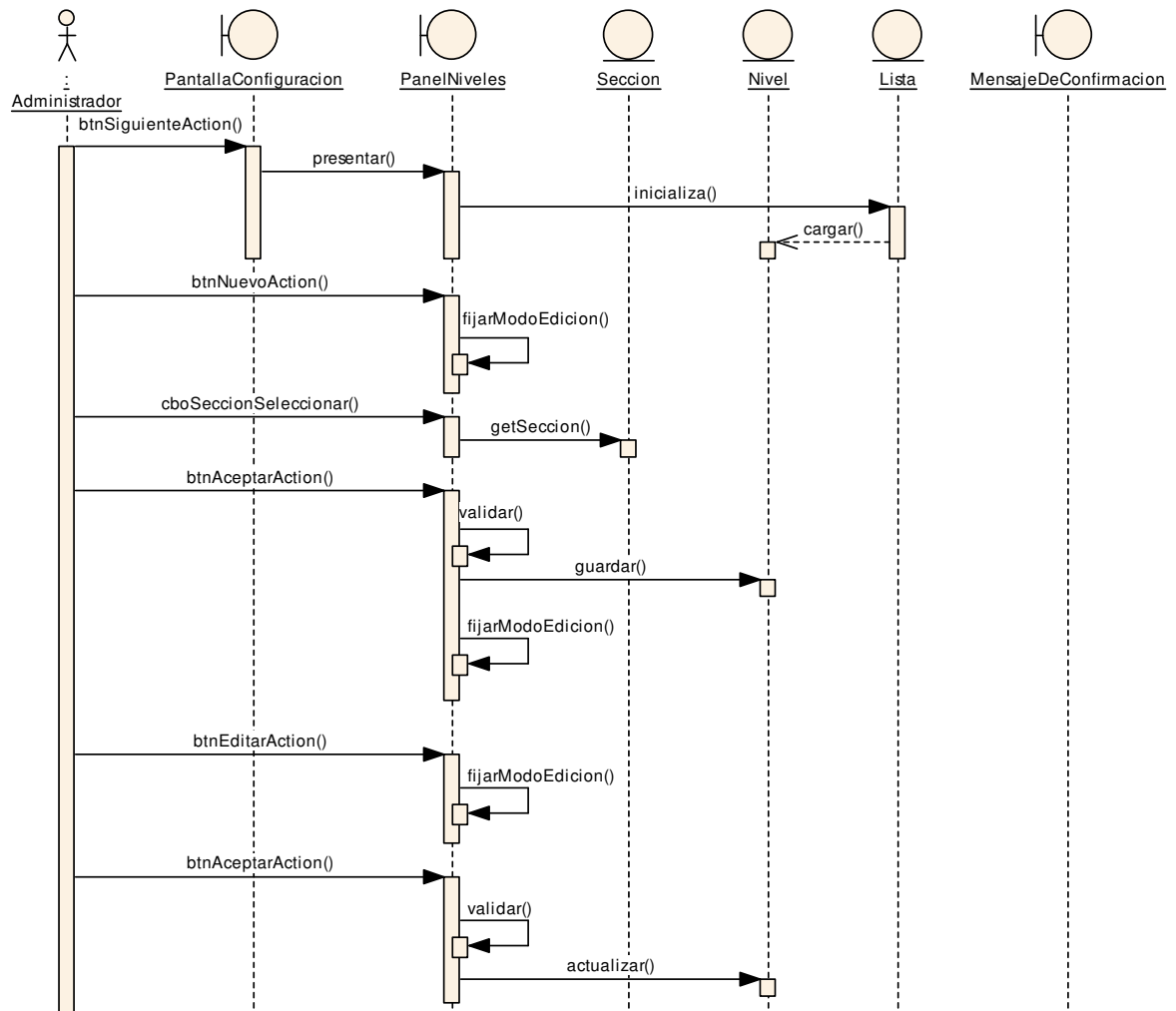


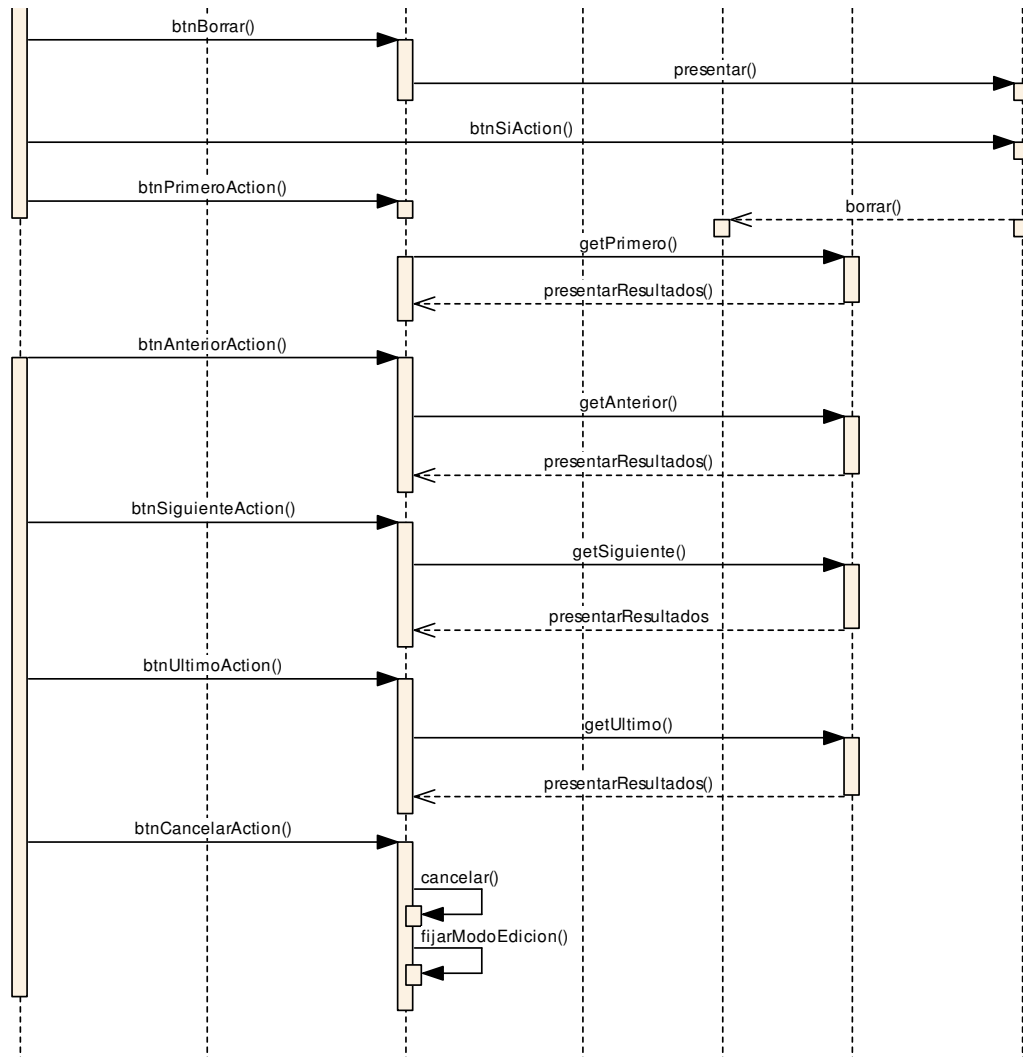
## 7.3.2.4 DS04-DR04 Mantener Secciones



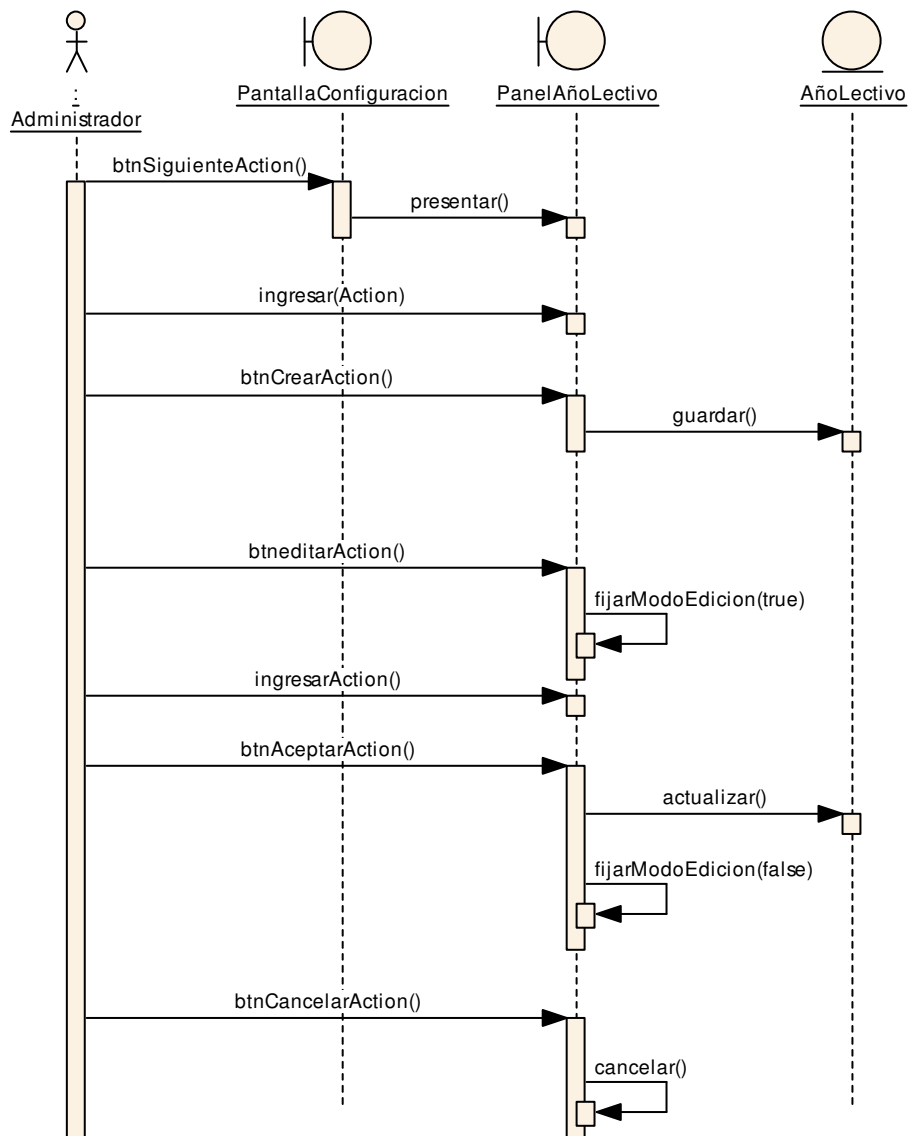


### 7.3.2.5 DS05–DR05 Mantener Niveles



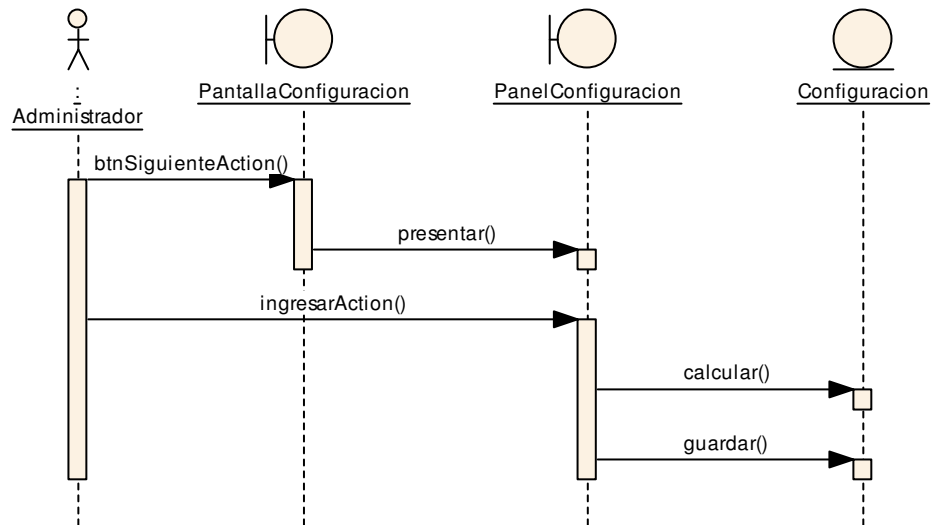


## 7.3.2.6 DS06-DR06 Mantener Año Lectivo

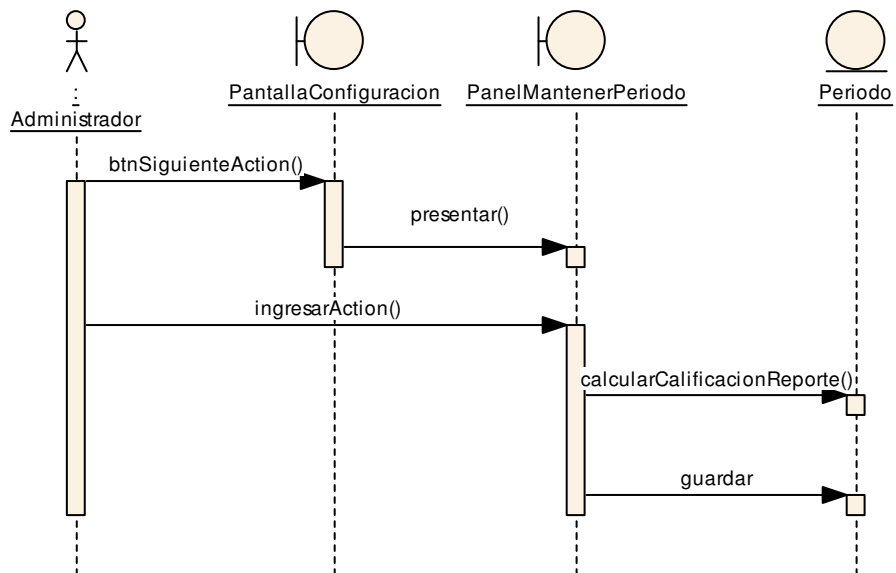




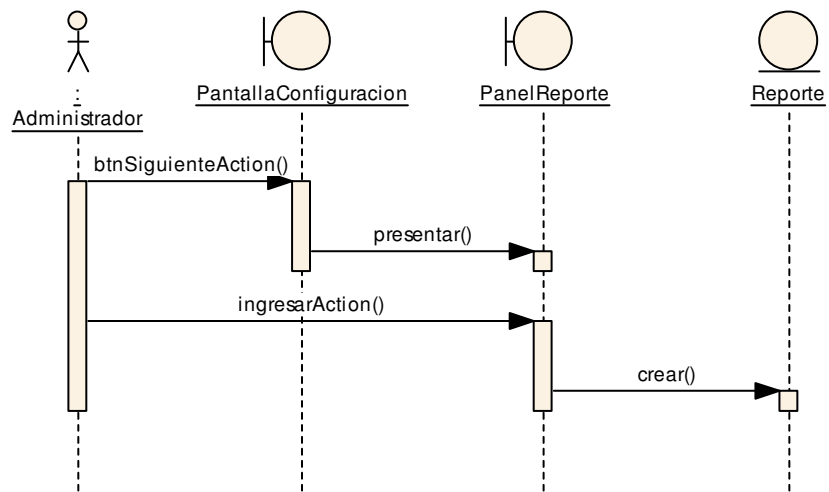
### 7.3.2.7 DS07-DR07 Mantener Configuración



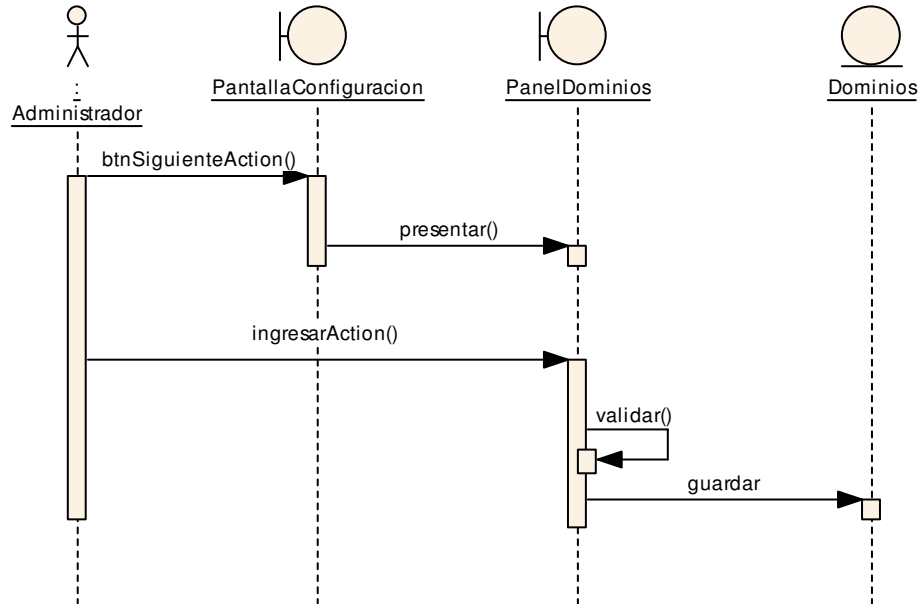
### 7.3.2.8 DS08-DR08 Mantener Periodos



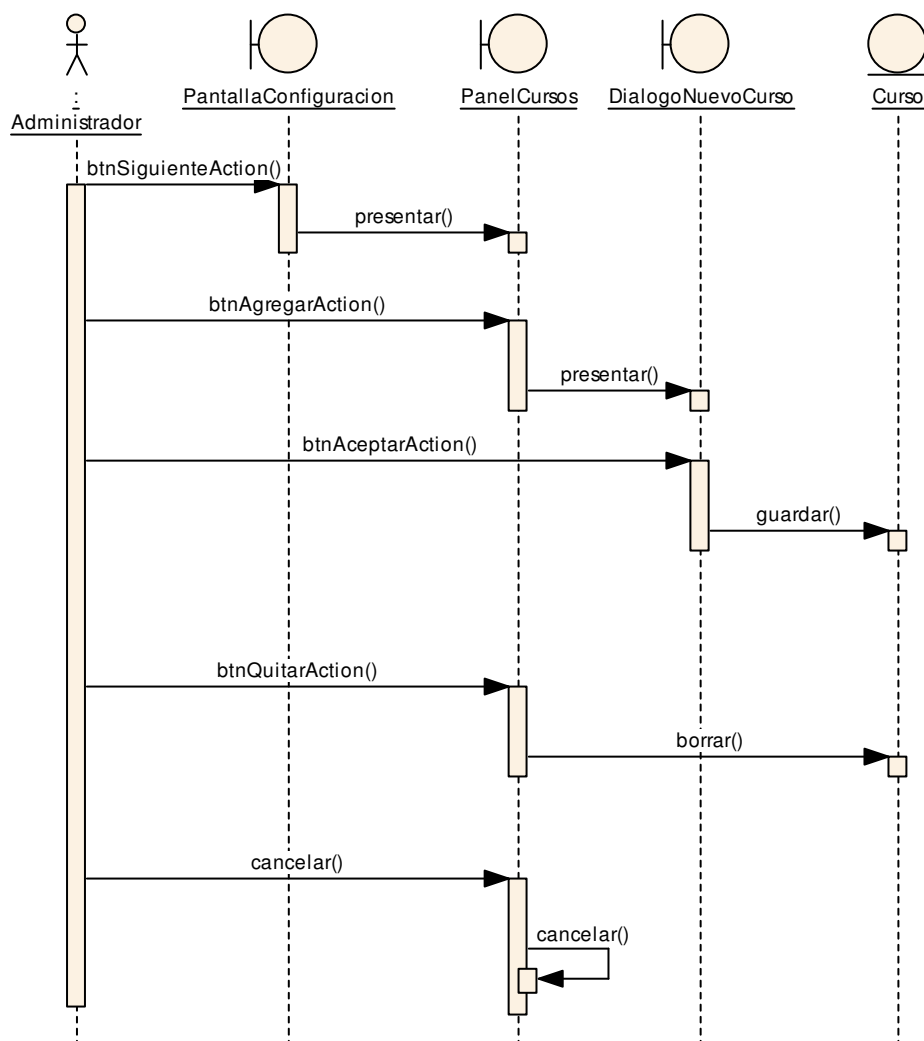
### 7.3.2.9 DS09–DR09 Mantener Reportes



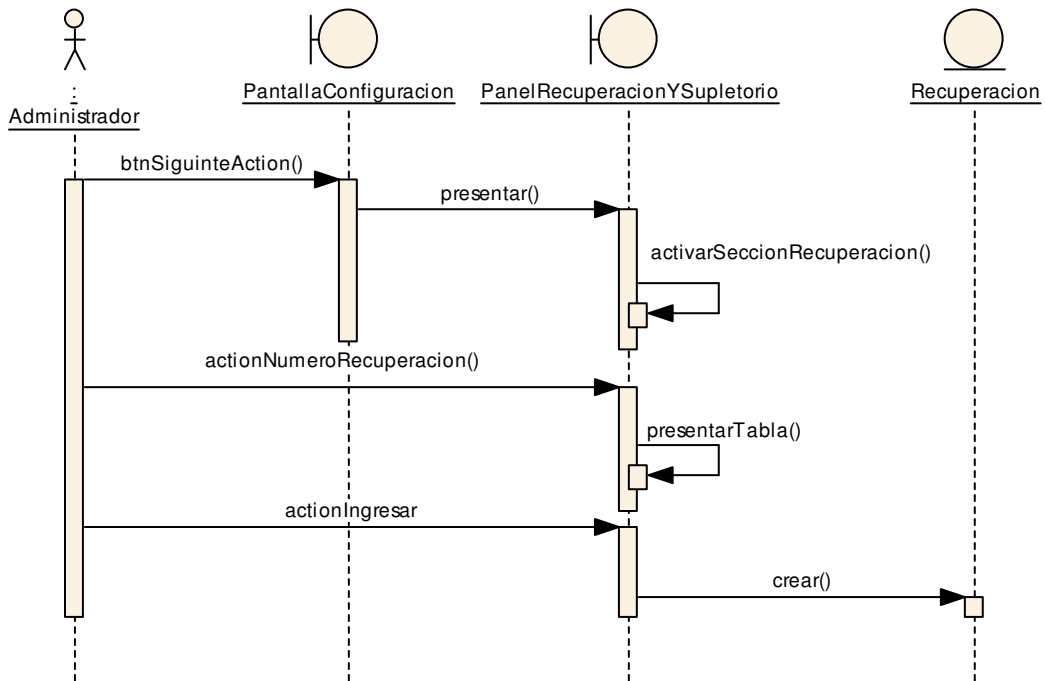
### 7.3.2.10 DS10–DR10 Mantener Dominios



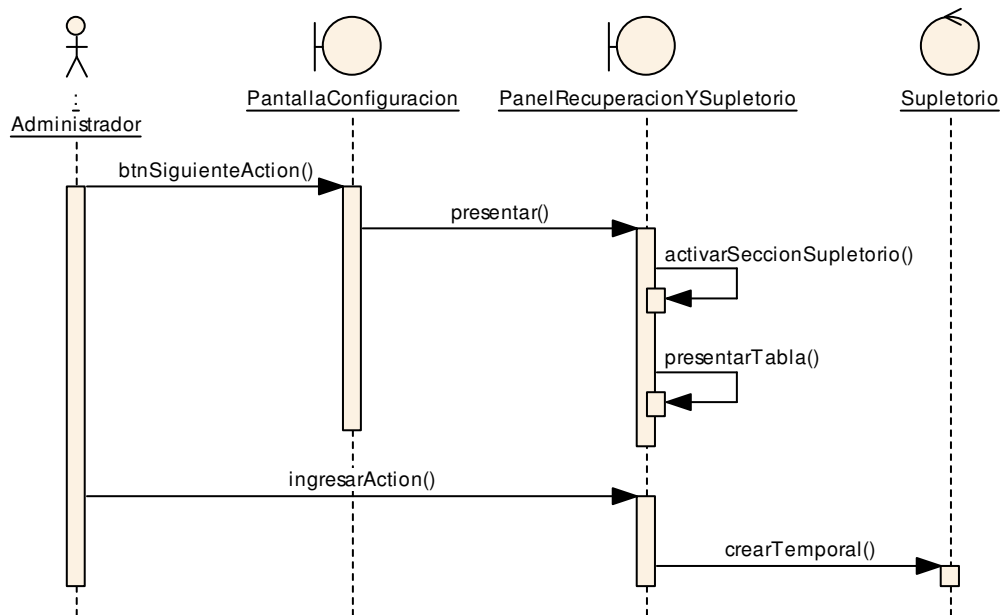
## 7.3.2.11 DS11-DR11 Mantener Cursos



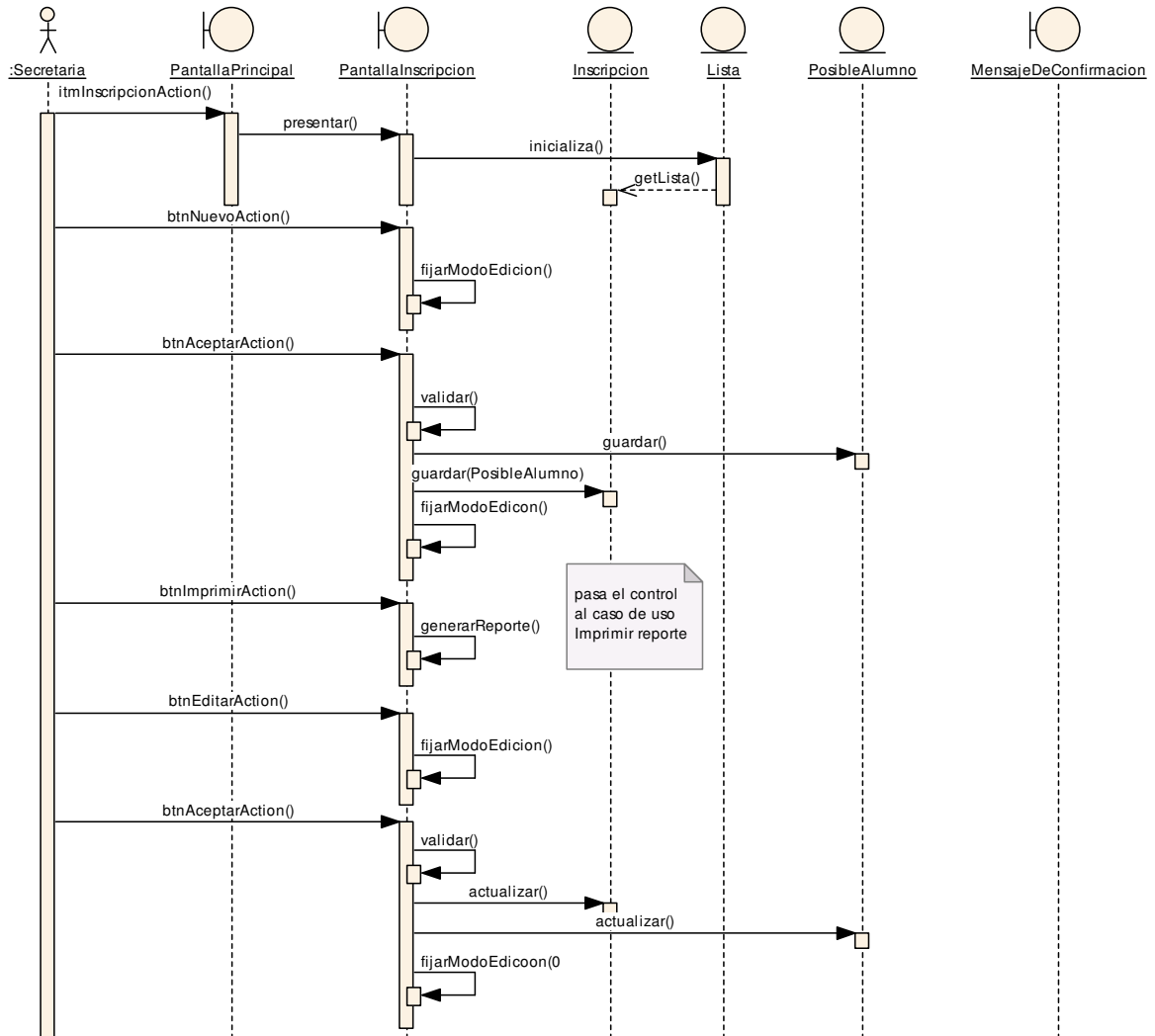
### 7.3.2.12 DS12-DR12 Mantener Recuperación

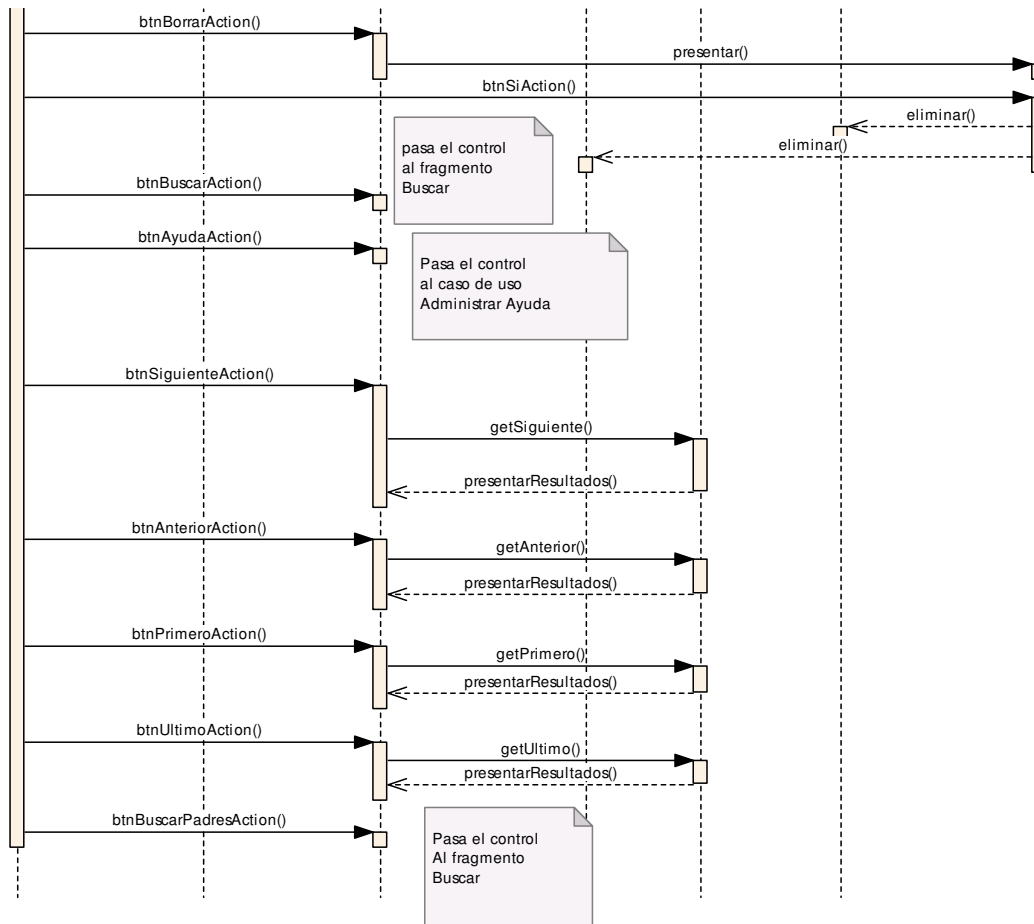


### 7.3.2.13 DS13-DR13 Mantener Supletorio

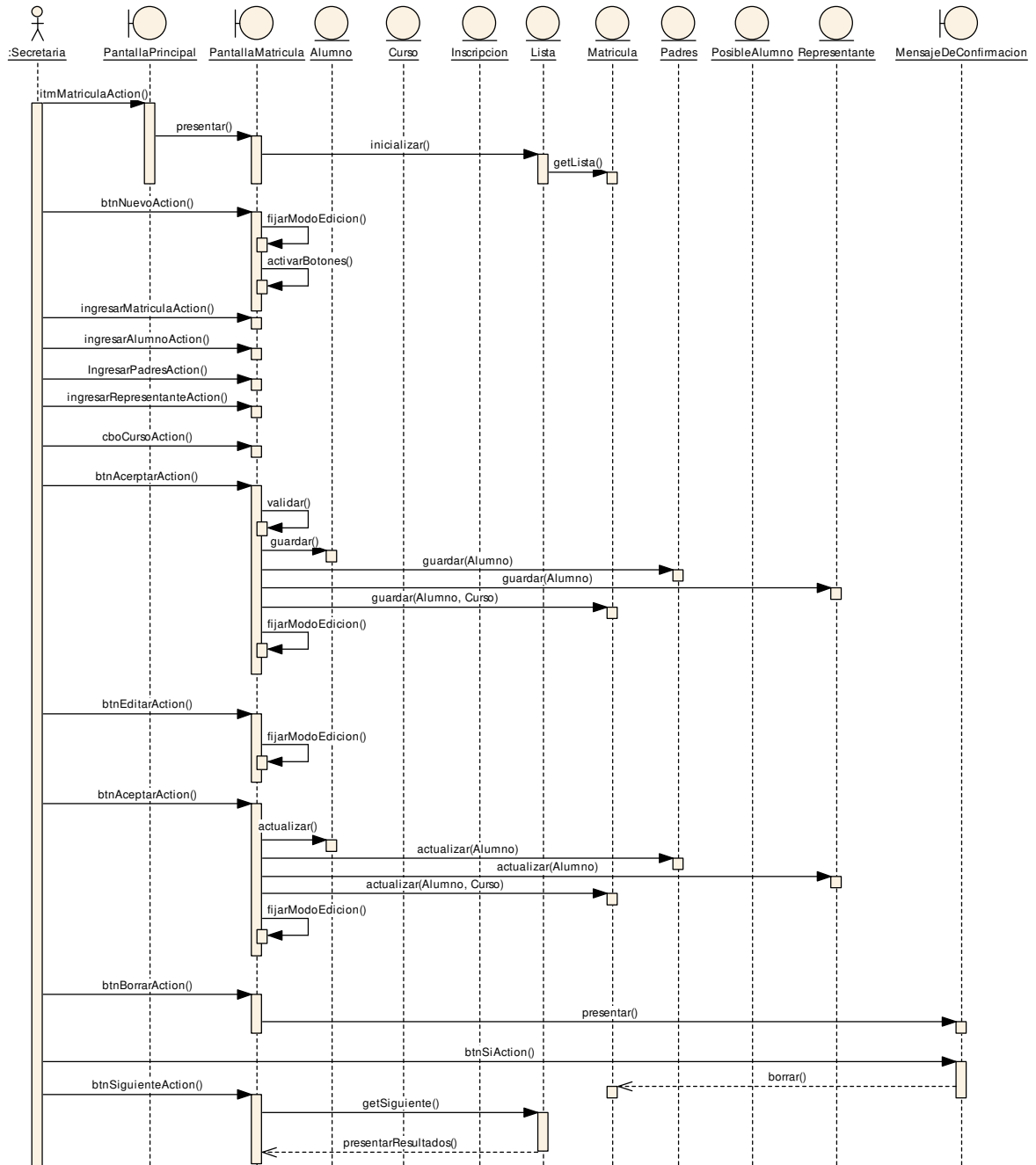


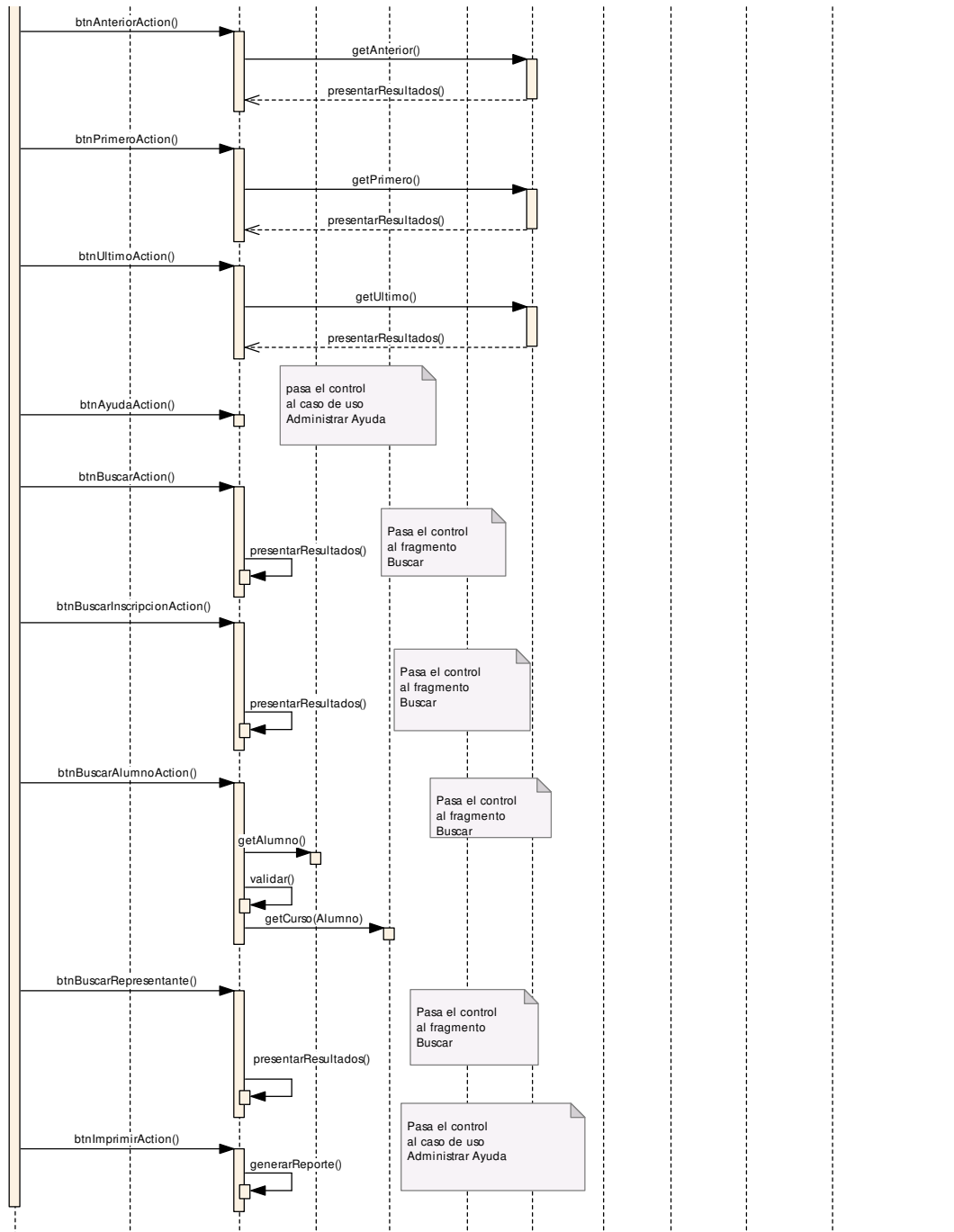
### 7.3.2.14 DS14-DR14 Mantener Inscripción





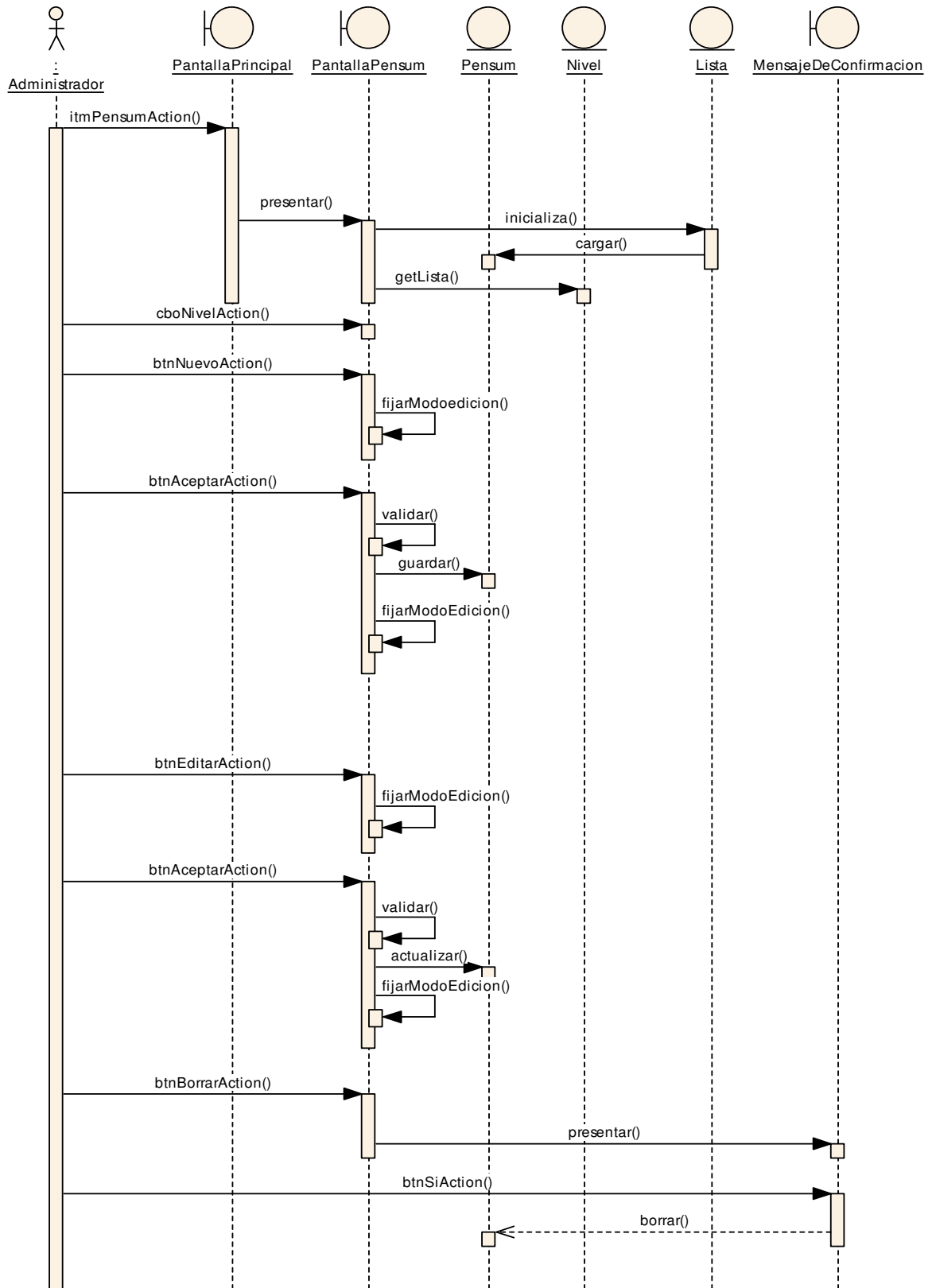
### 7.3.2.15 DS15-DR15 Mantener Matricula

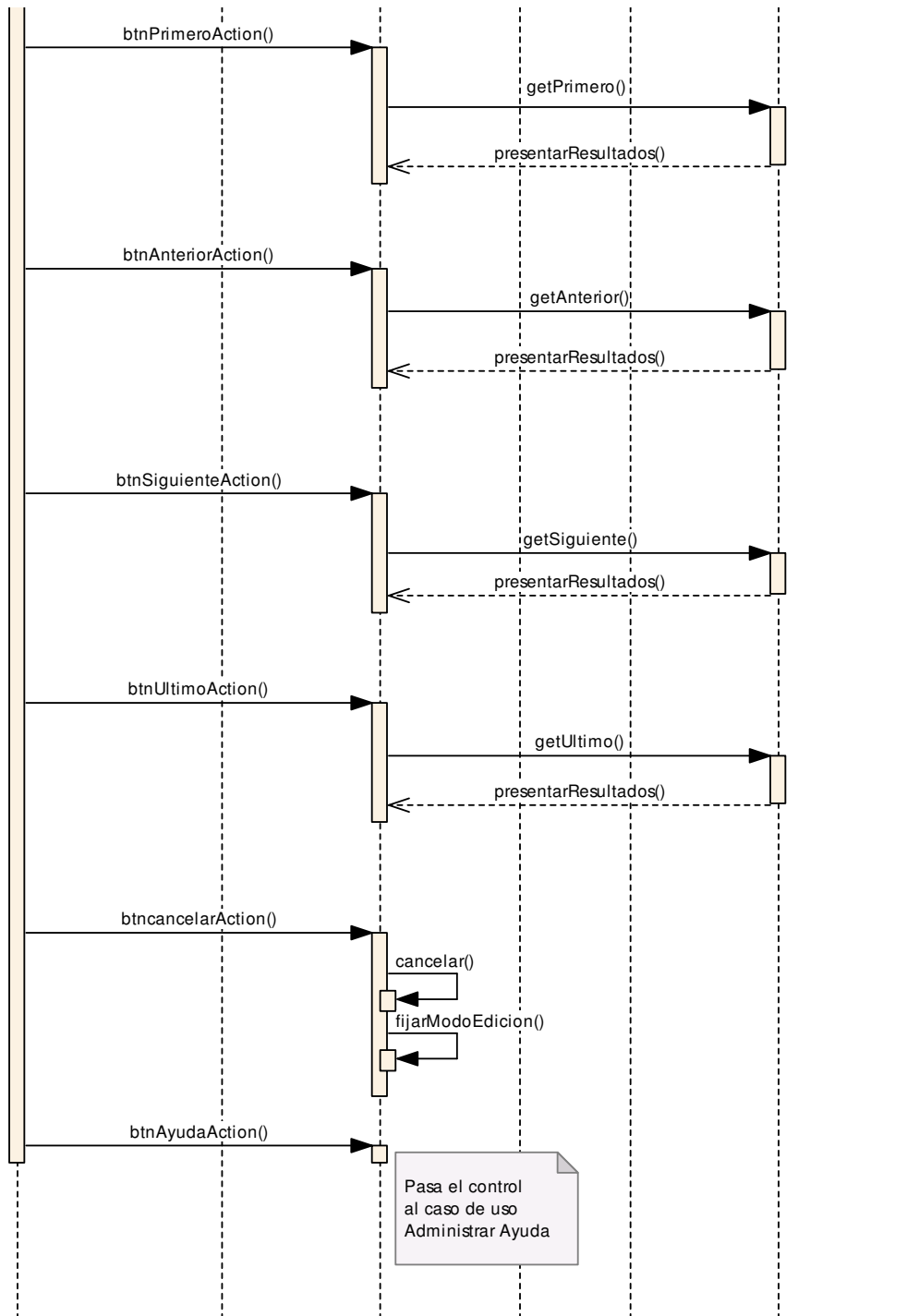




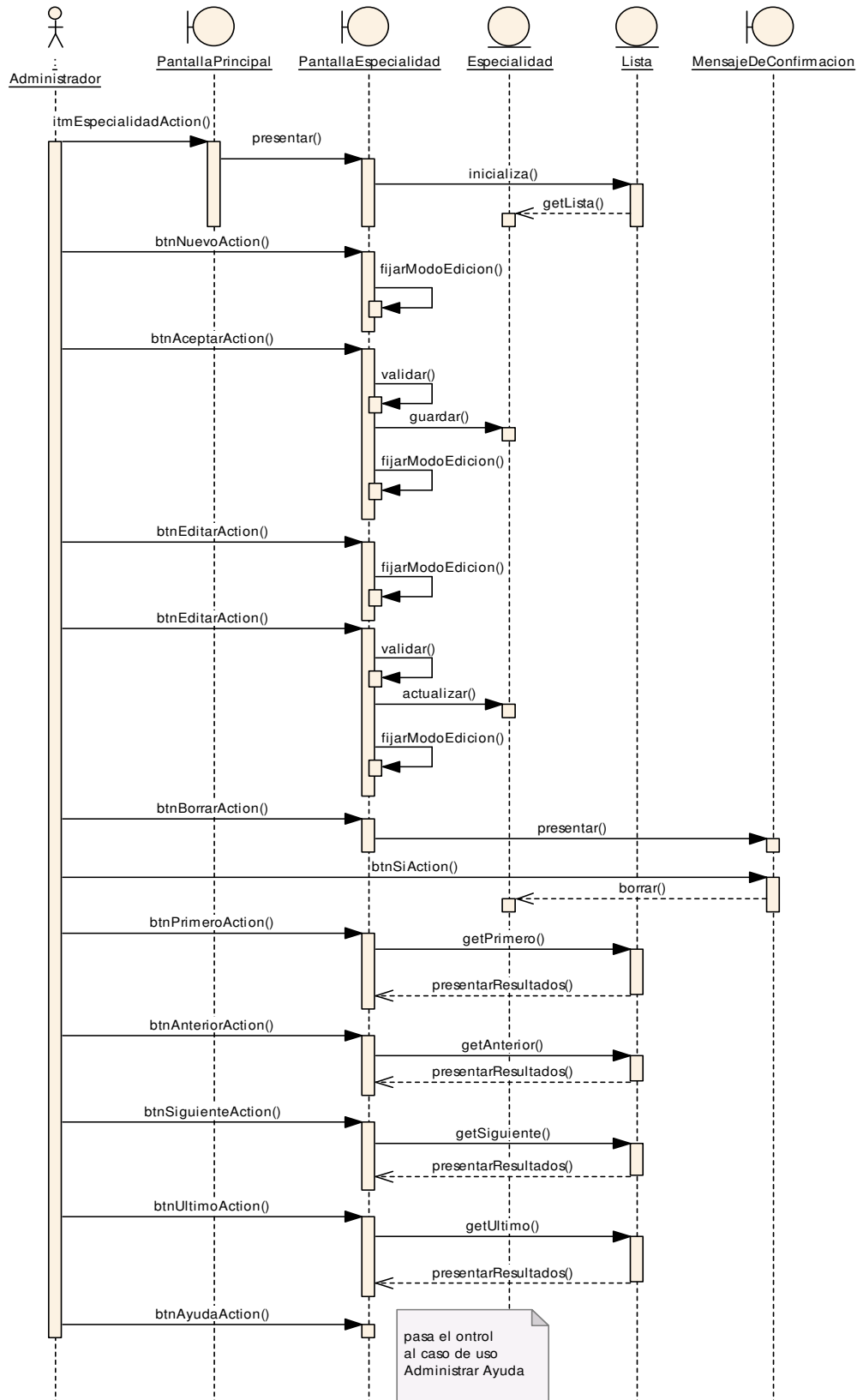


## 7.3.2.16 DS16-DR16 Mantener Pensum

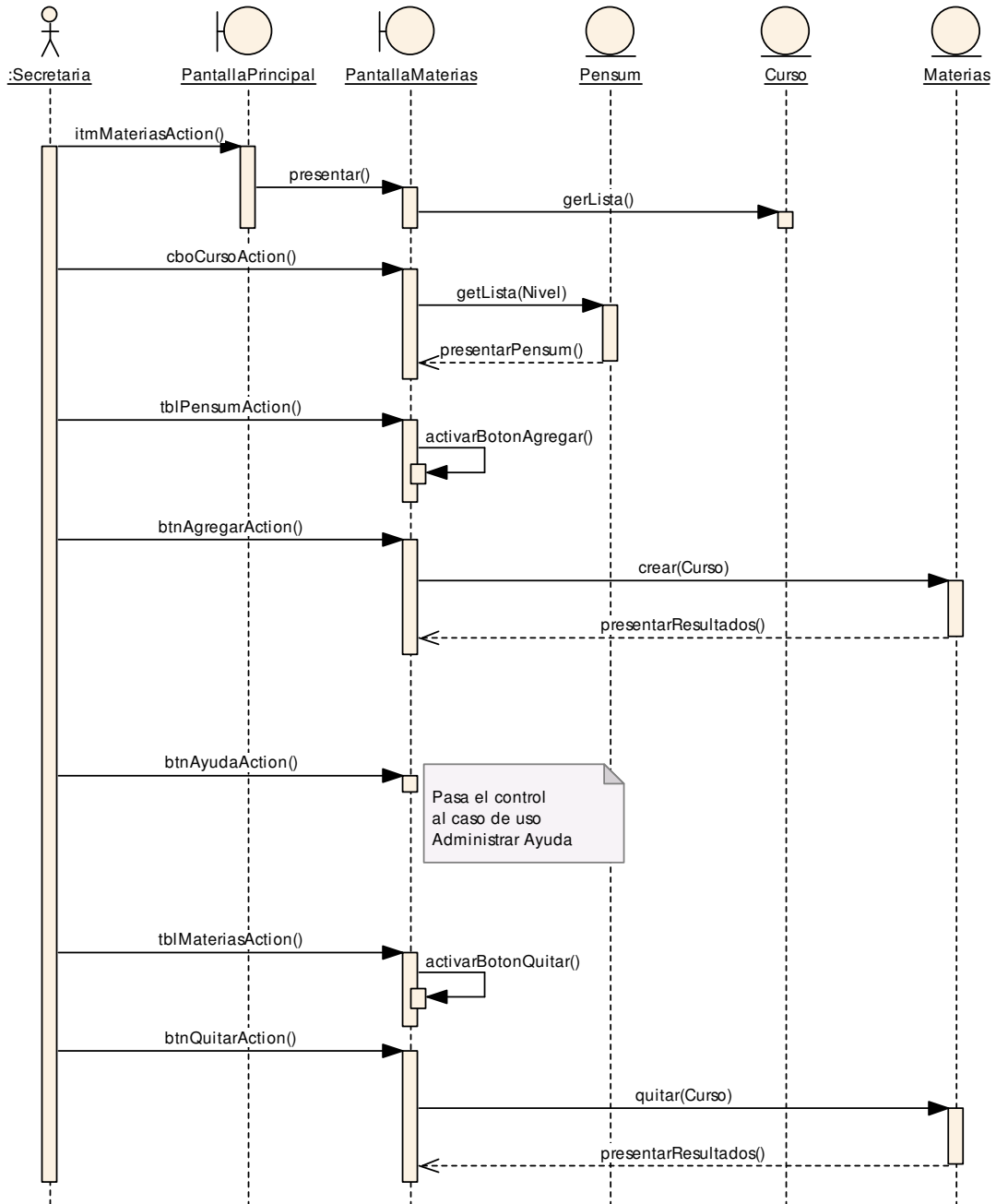




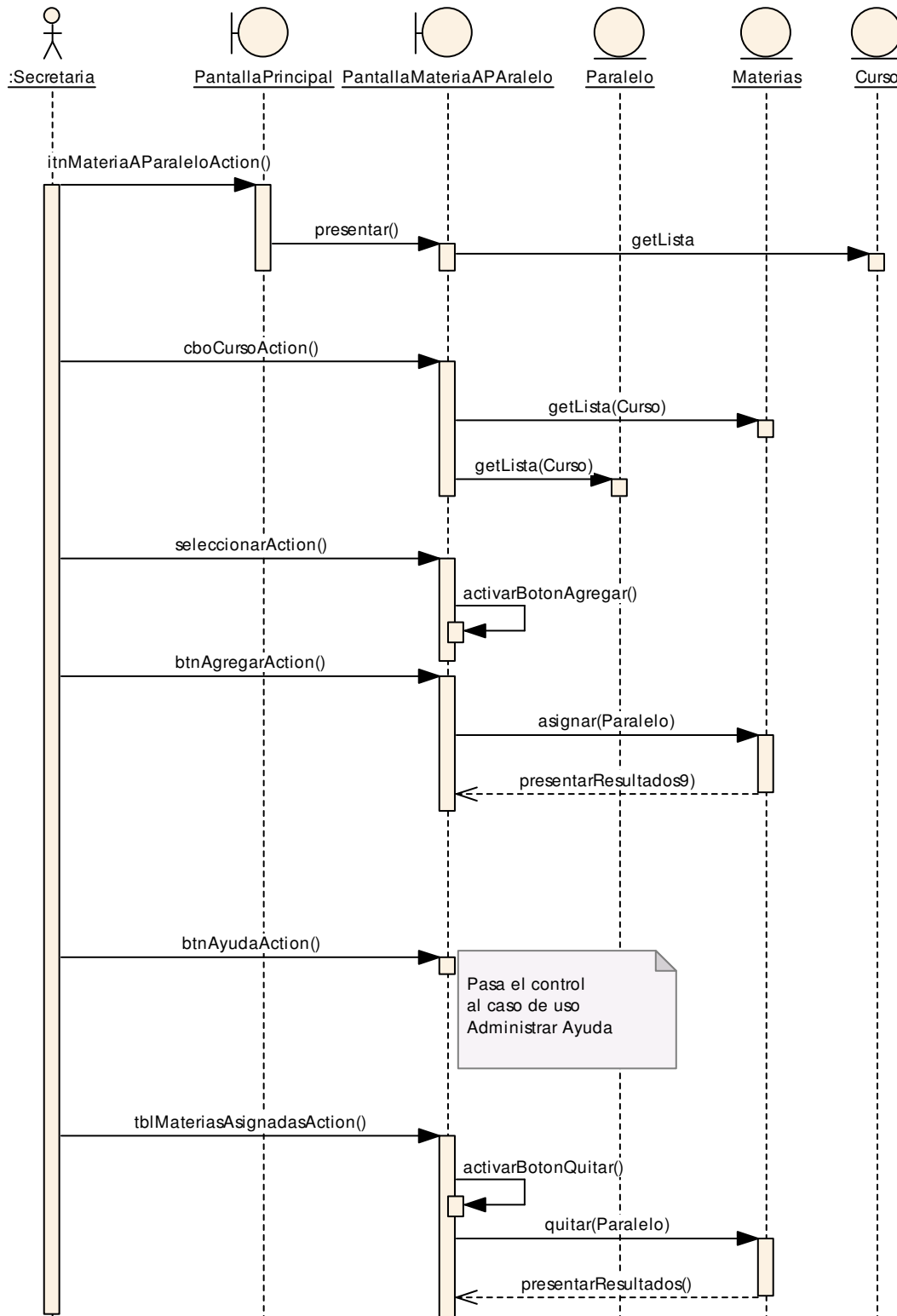
## 7.3.2.17 DS17-DR17 Mantener Especialidades



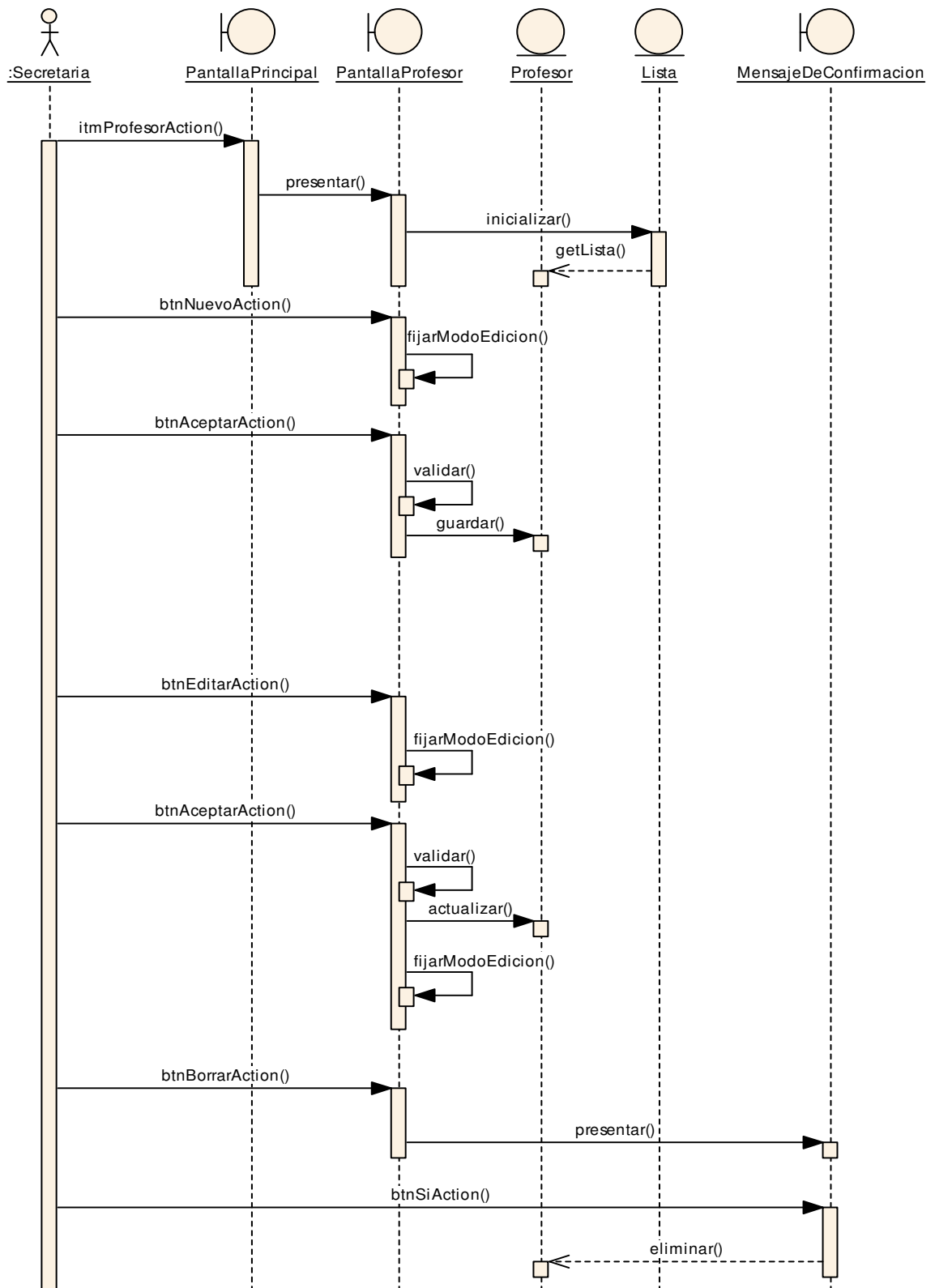
## 7.3.2.18 DS18-DR18 Mantener Materias

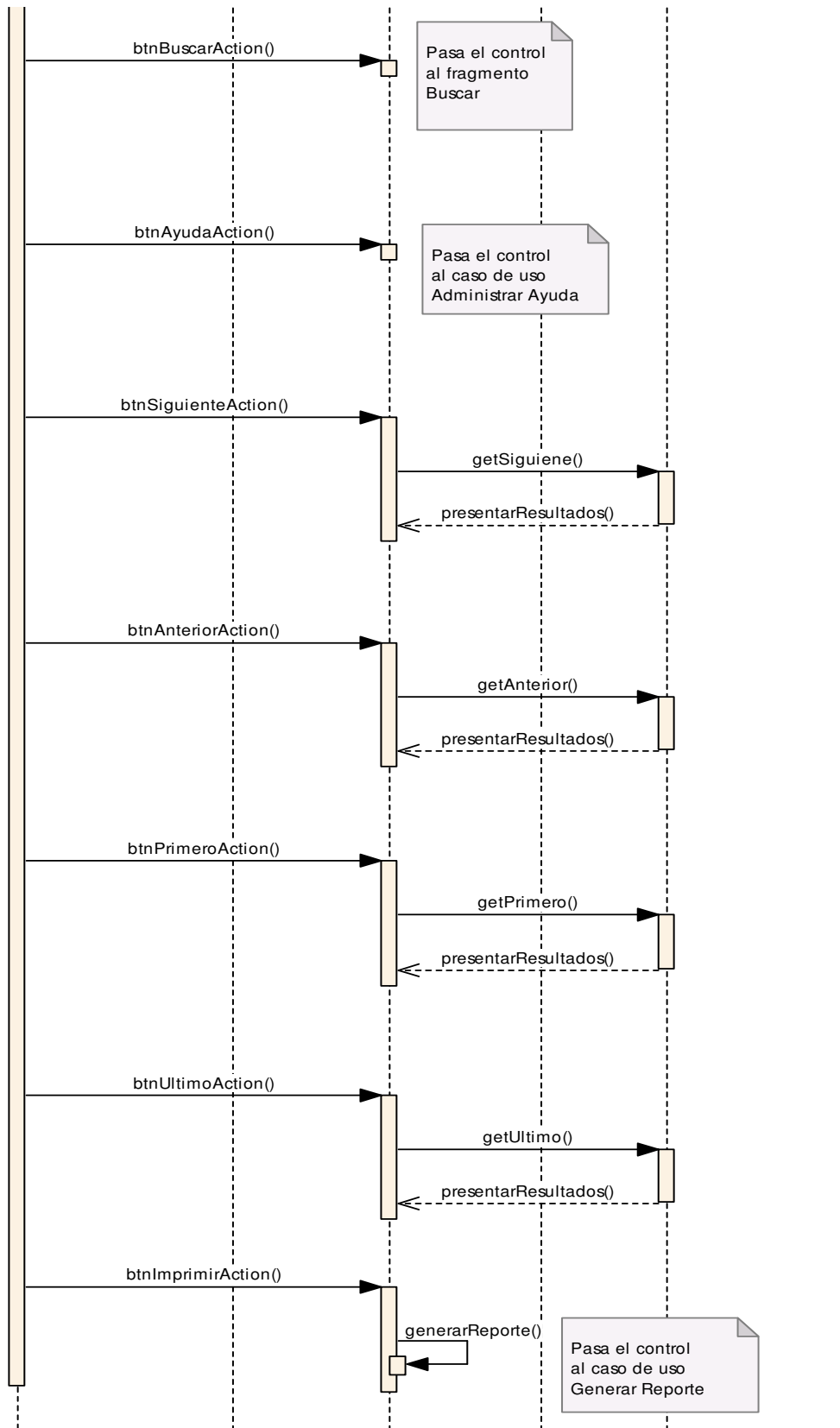


### 7.3.2.19 DS19-DR19 Asignar Materia a Paralelo

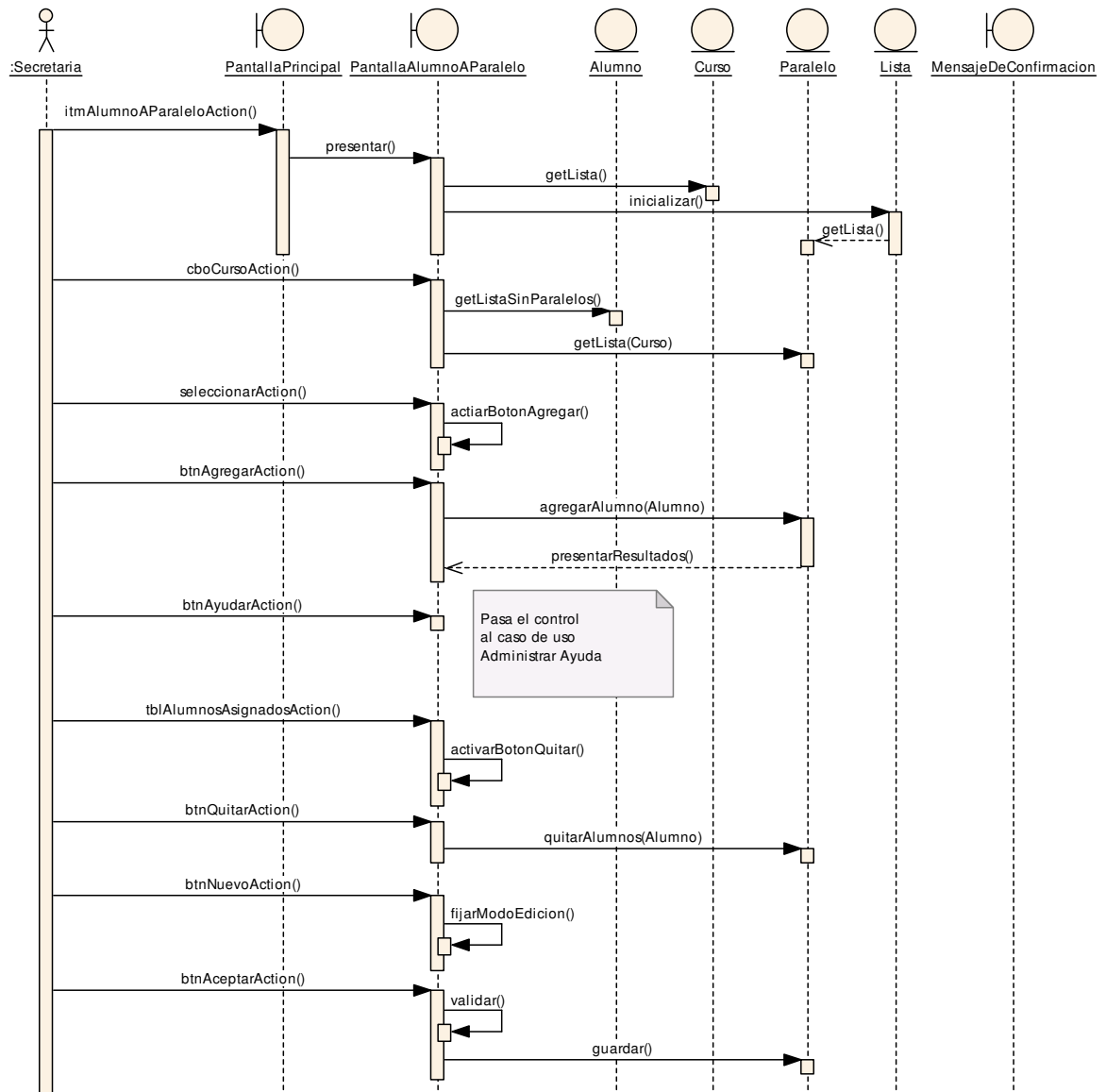


## 7.3.2.20 DS20-DR20 Mantener Profesor

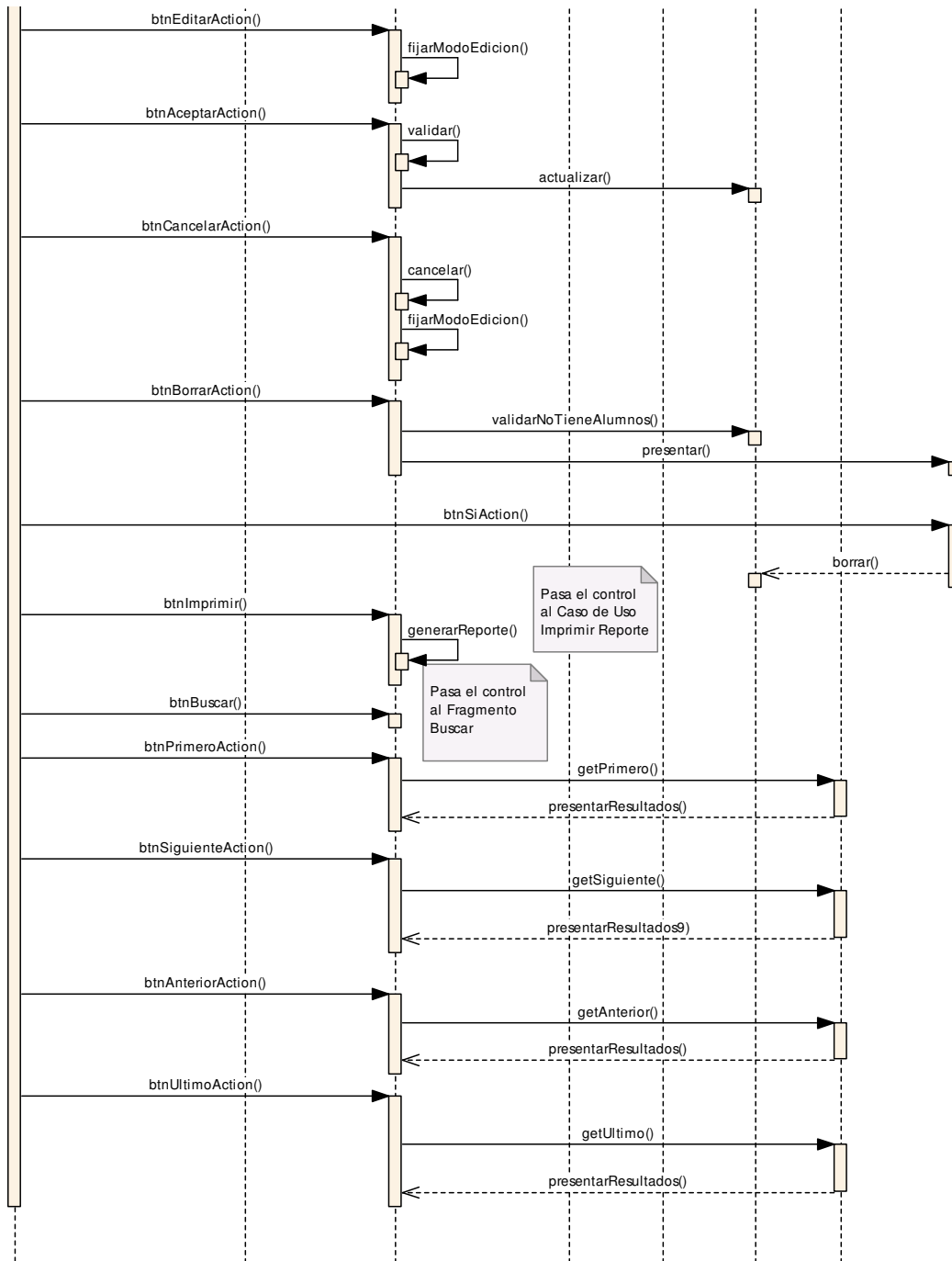




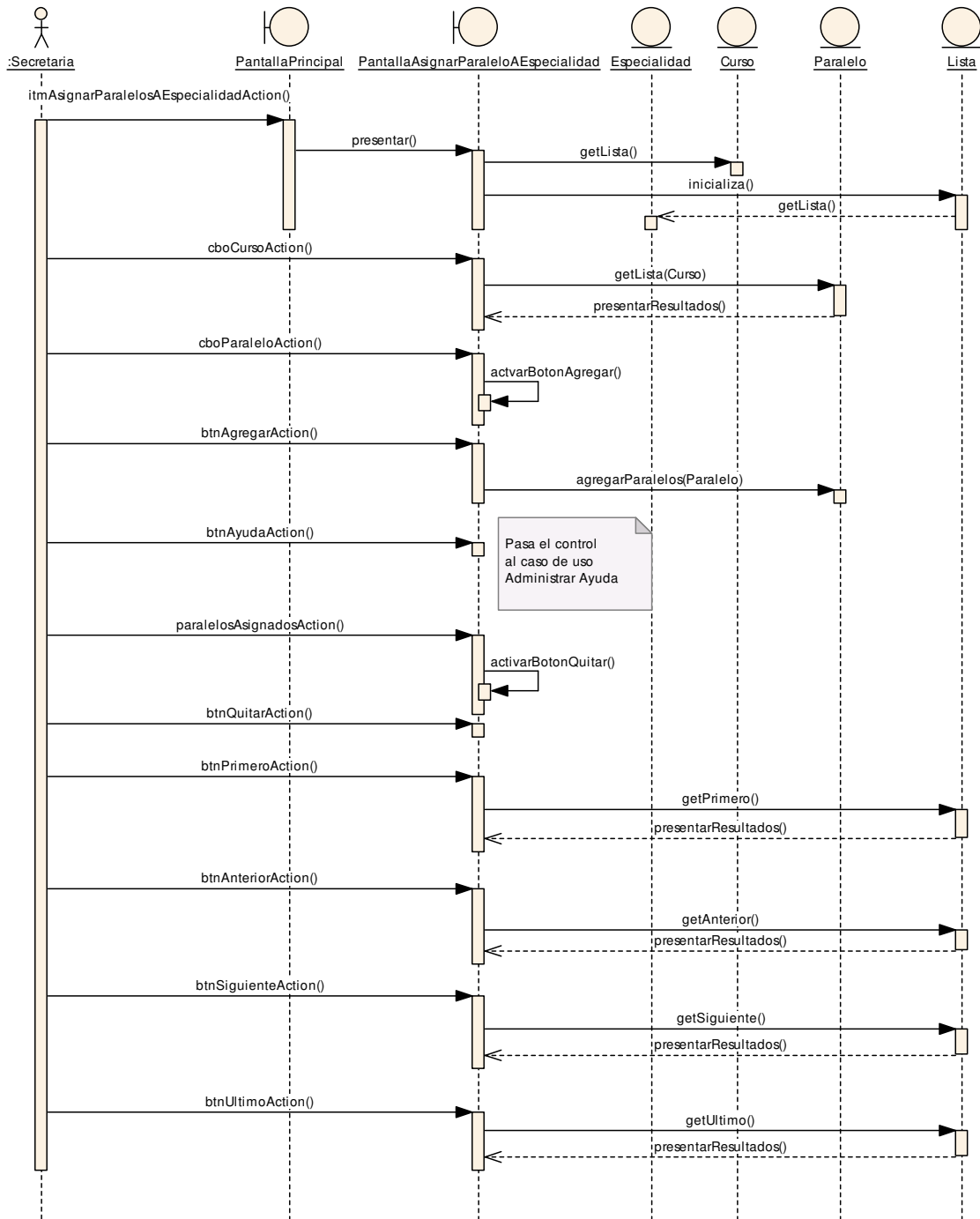
## 7.3.2.21 DS21-DR21 Asignar Alumno a Paralelo



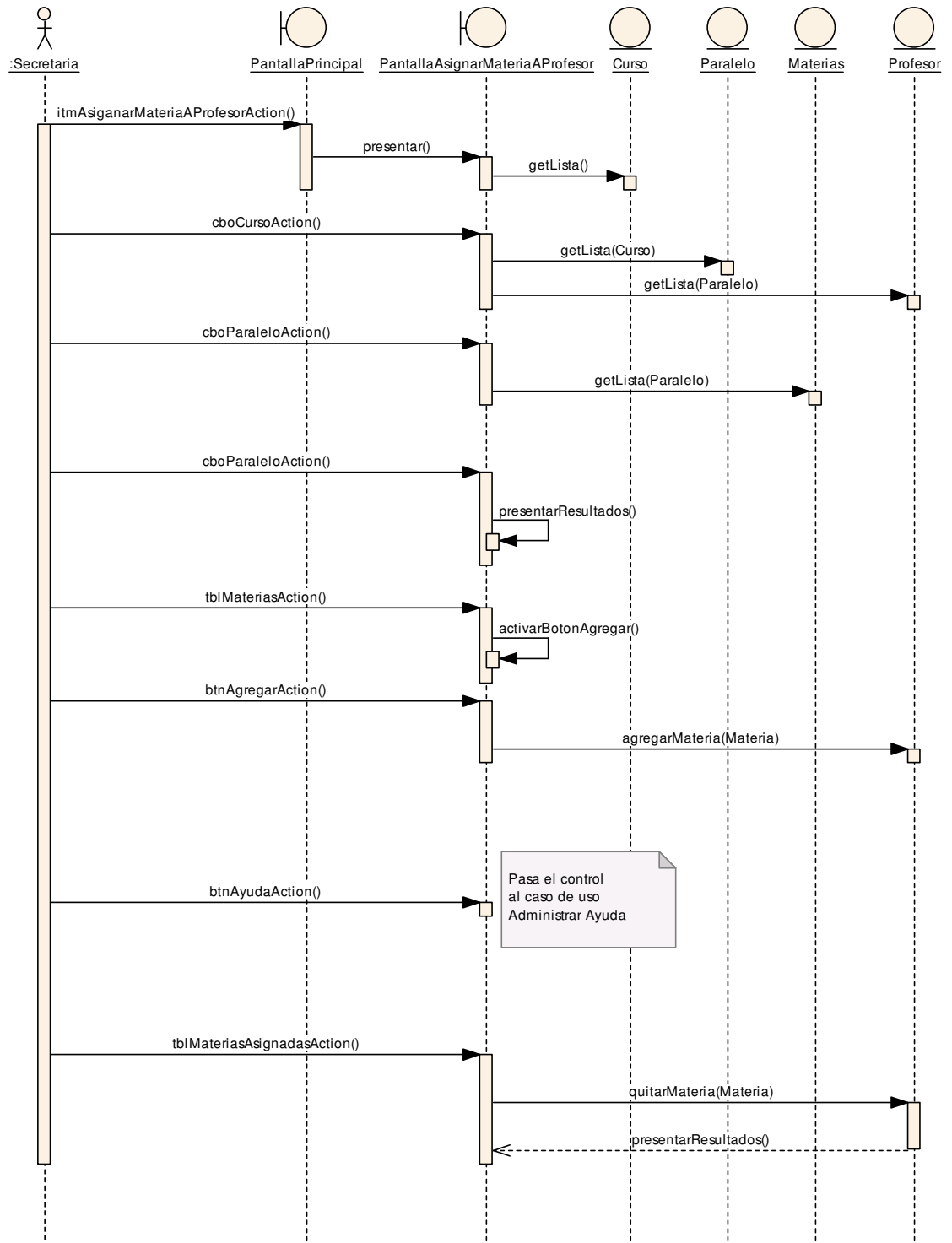




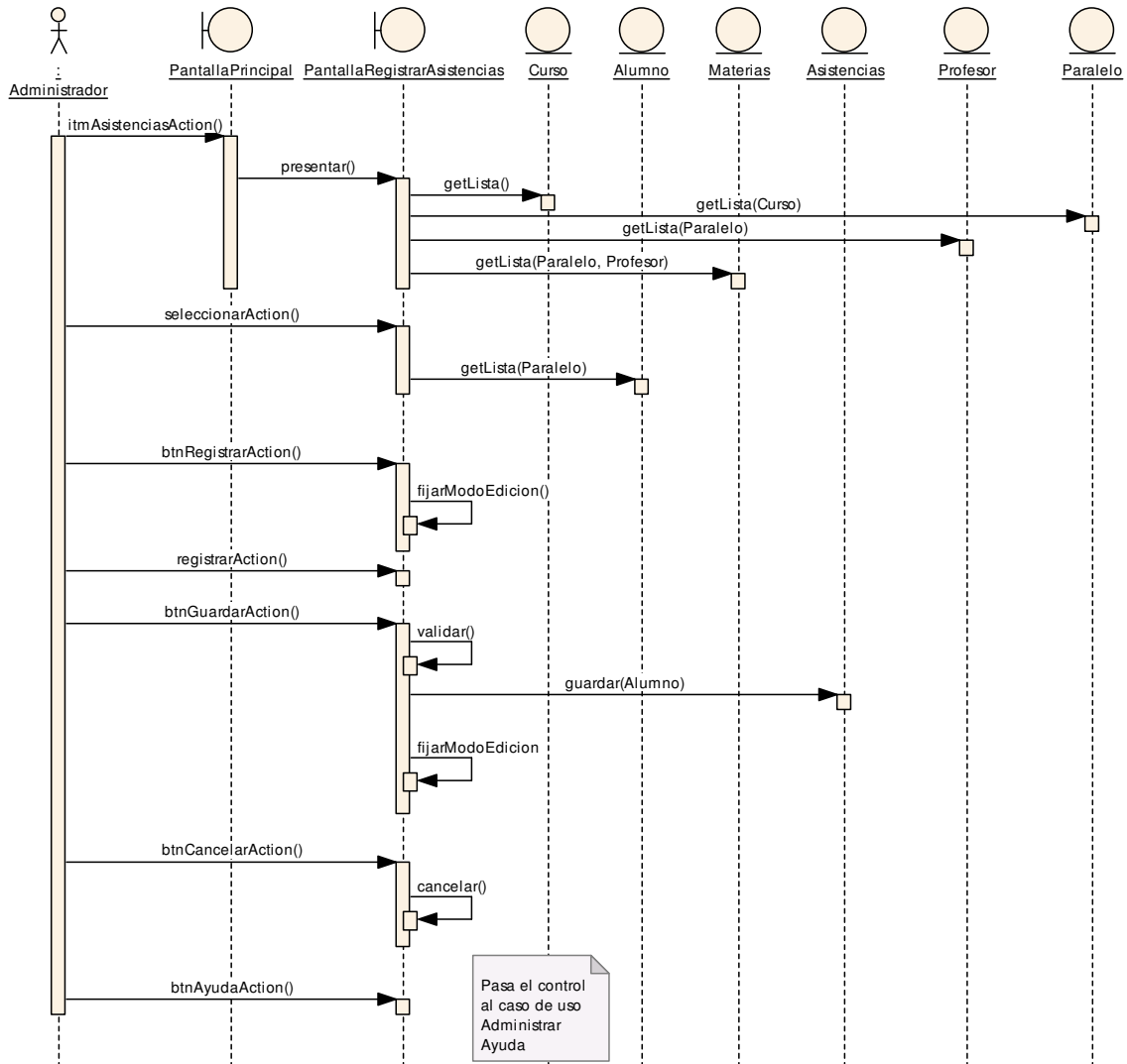
### 7.3.2.22 DS22-DR22 Asignar Paralelos a Especialidad



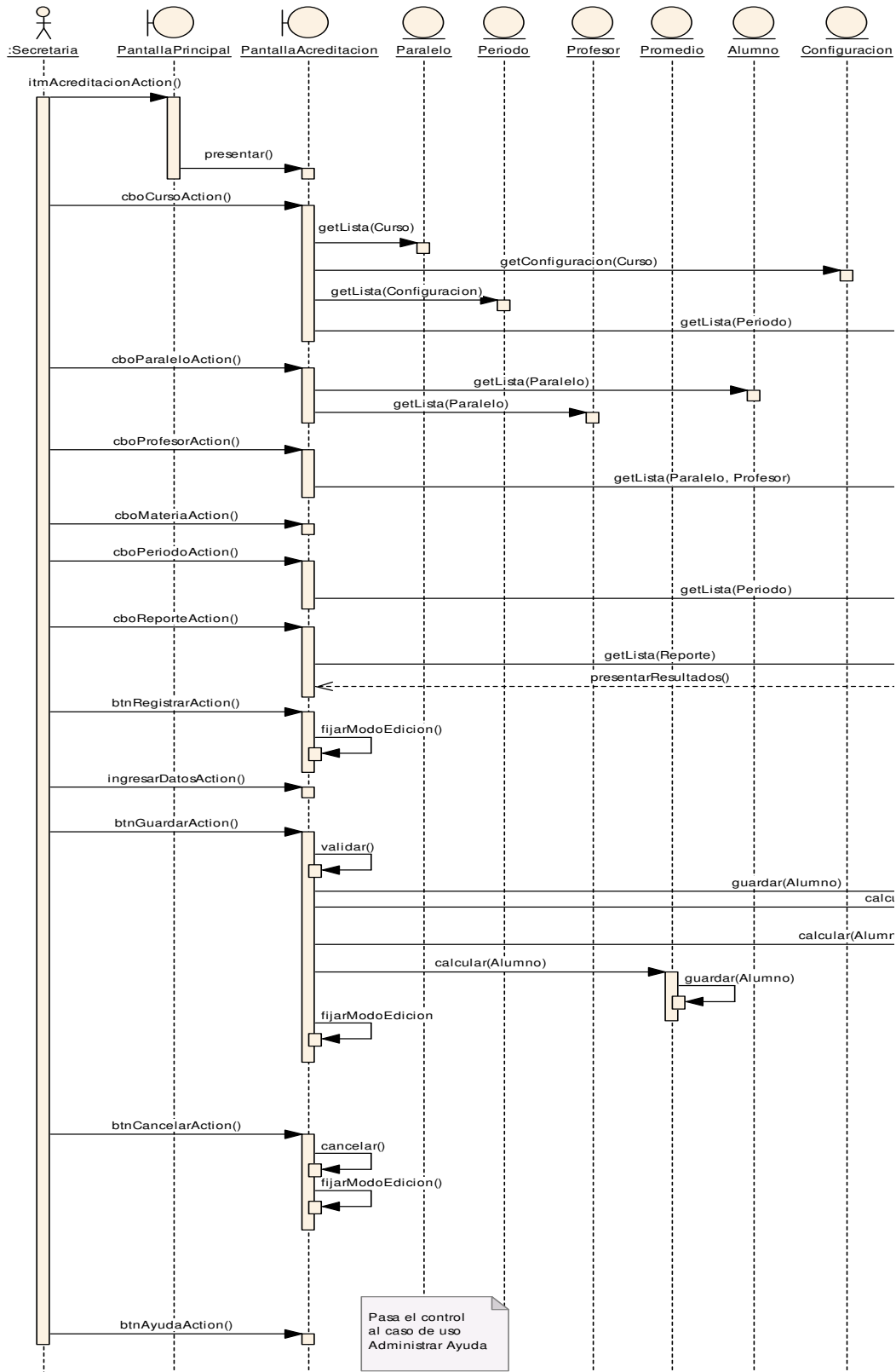
## 7.3.2.23 DS23-DR23 Asignar Materia a Profesor

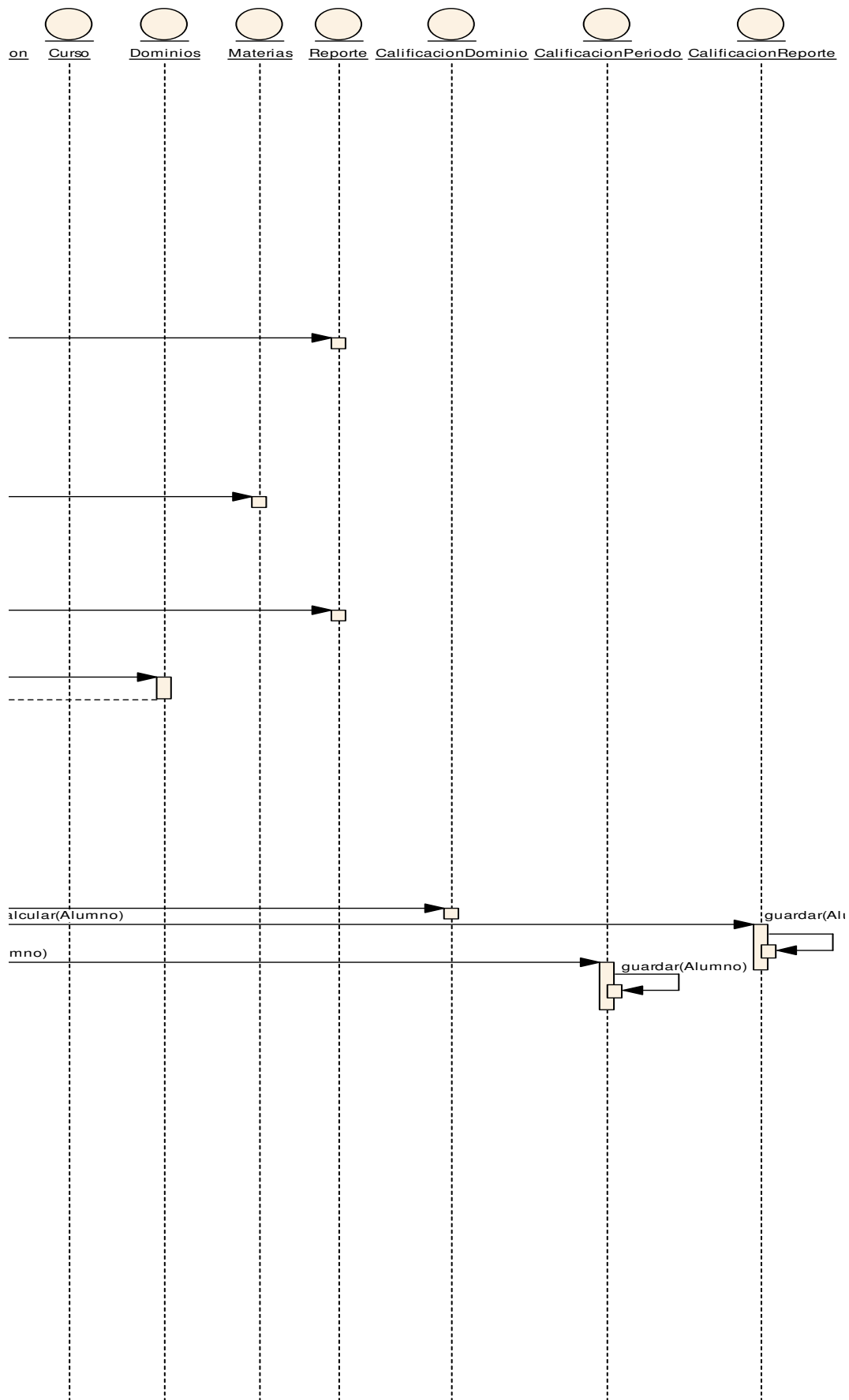


## 7.3.2.24 DS24-DR24 Registrar Asistencia

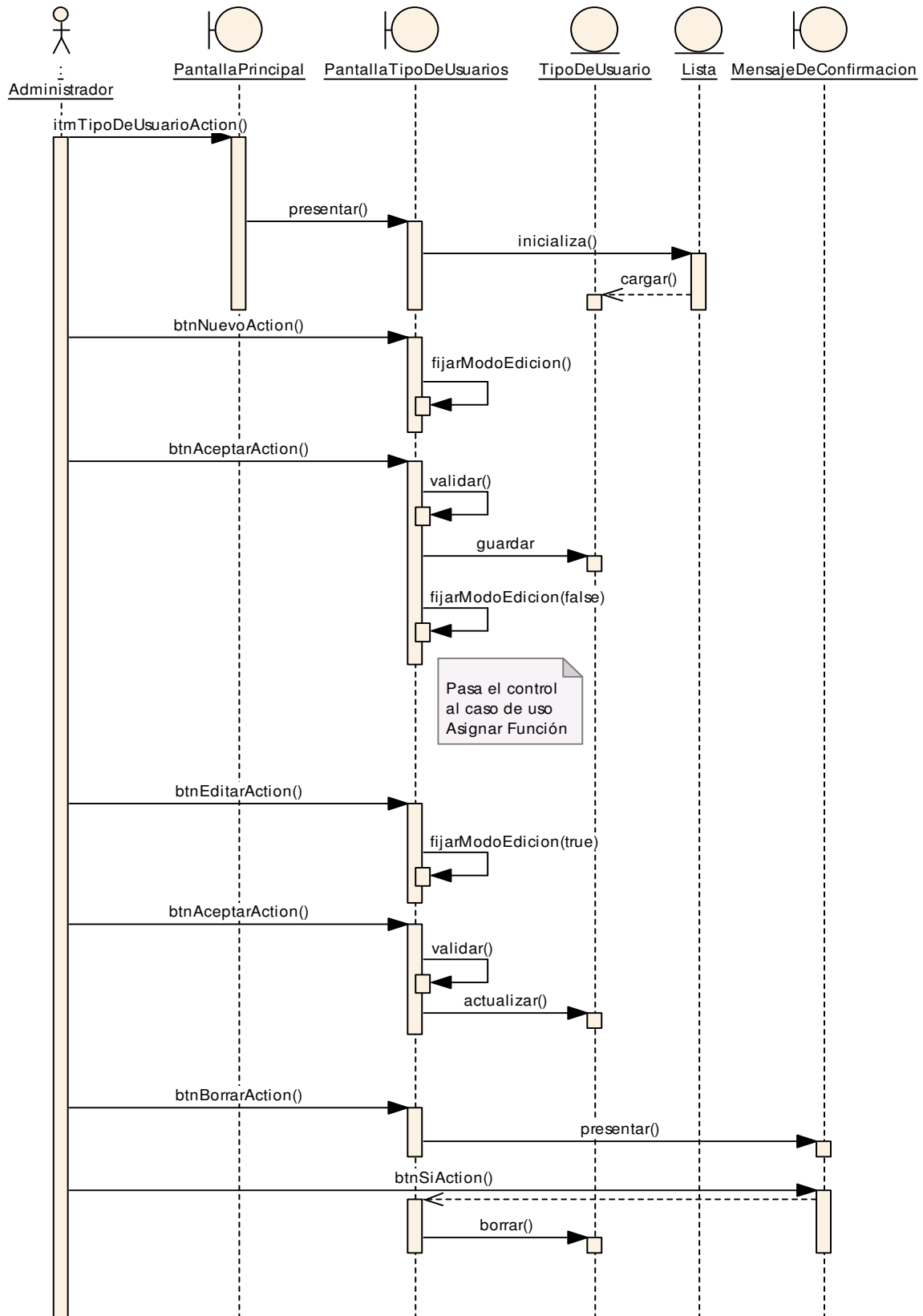


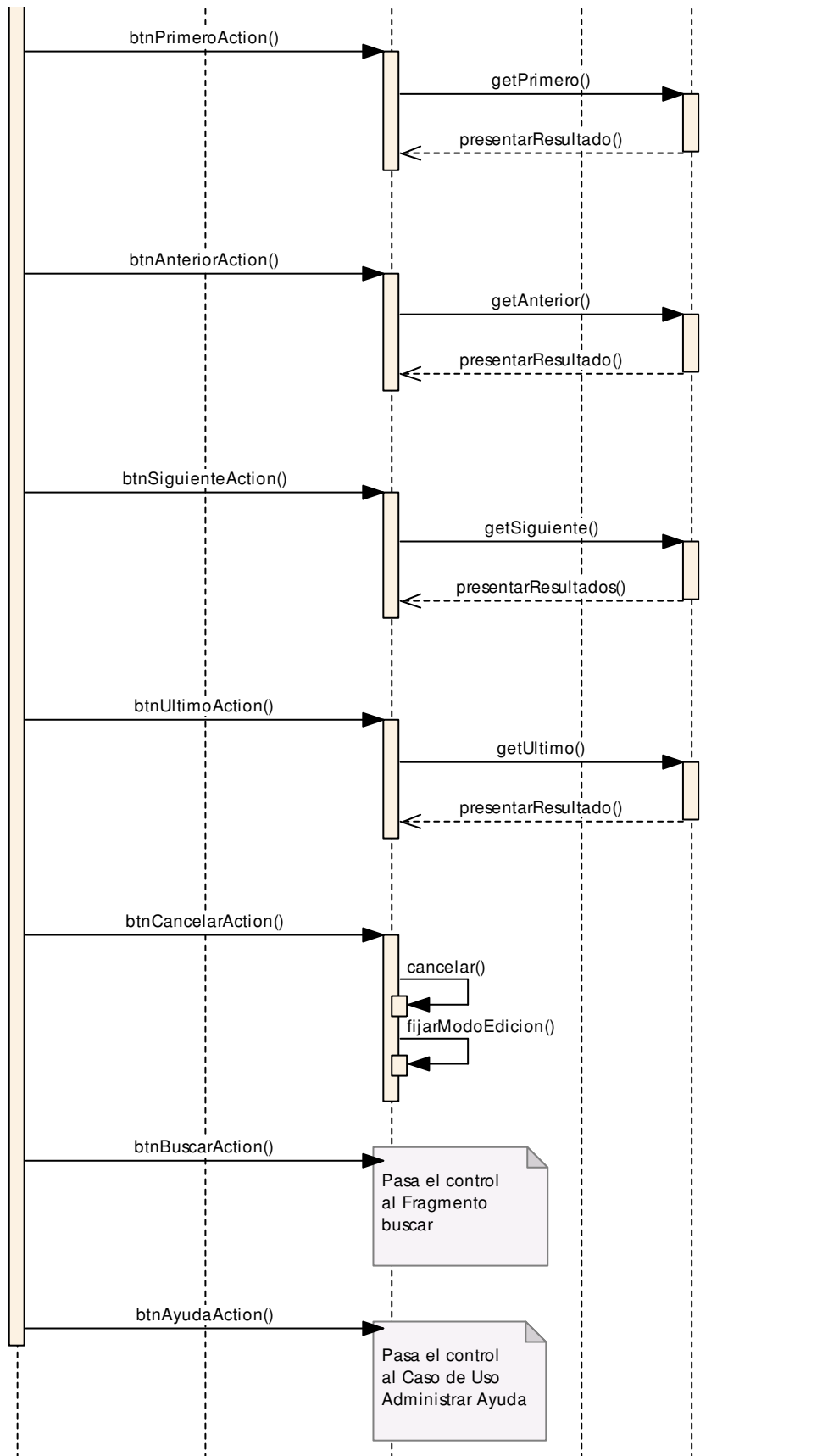
### 7.3.2.25 DS25-DR25 Registrar Acreditación





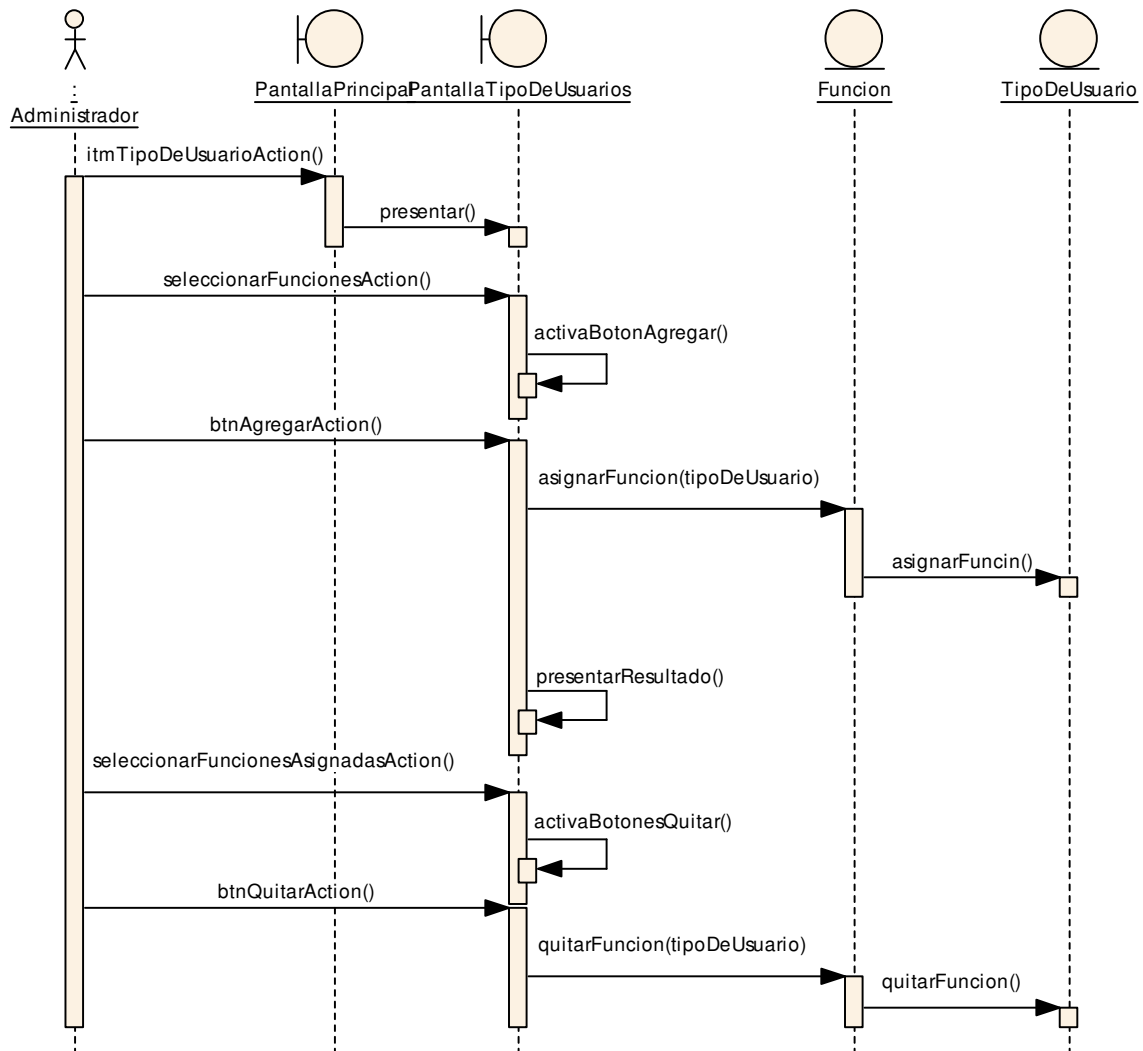
## 7.3.2.26 DS26-DR26 Mantener Tipos de Usuario



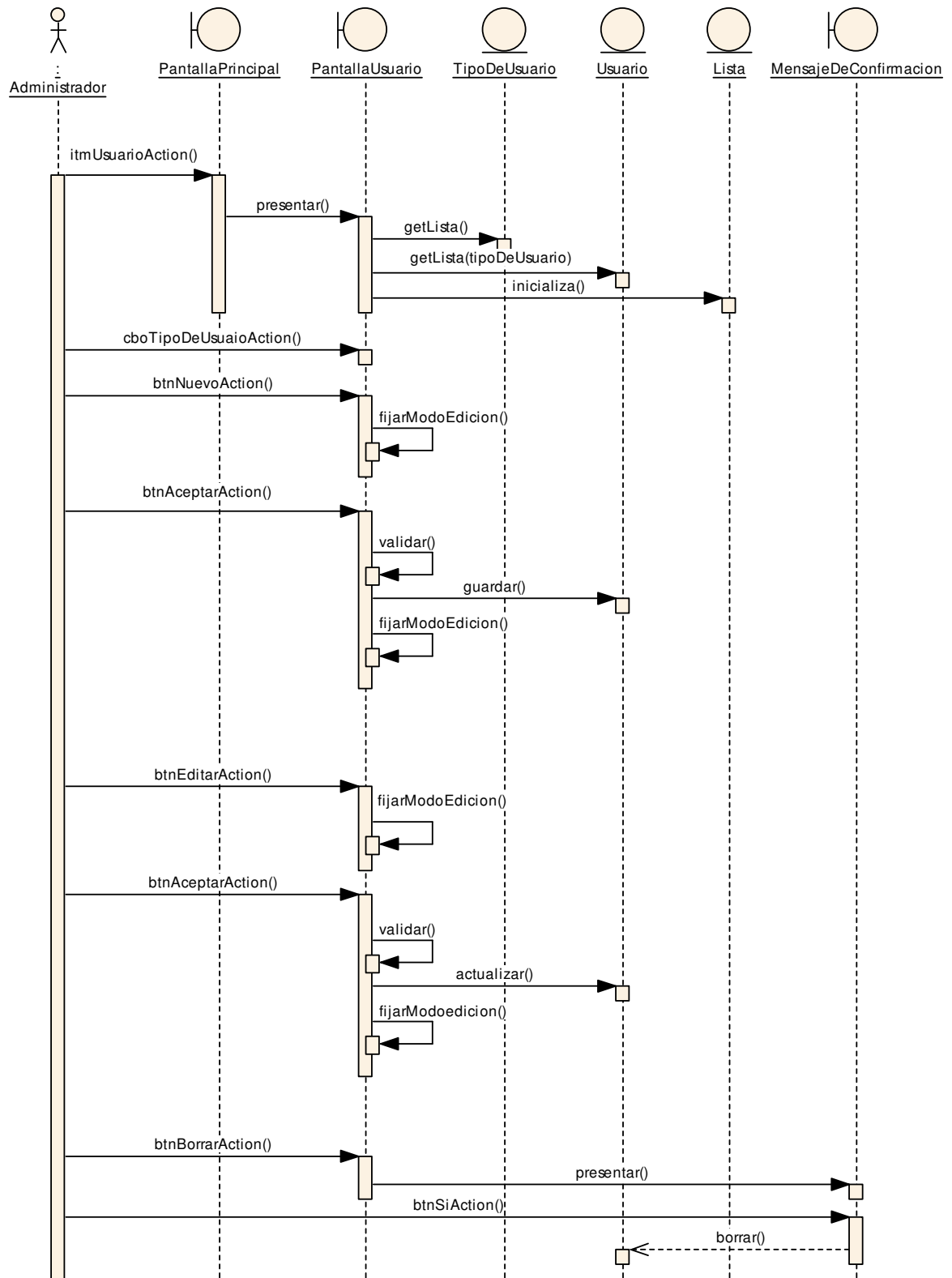


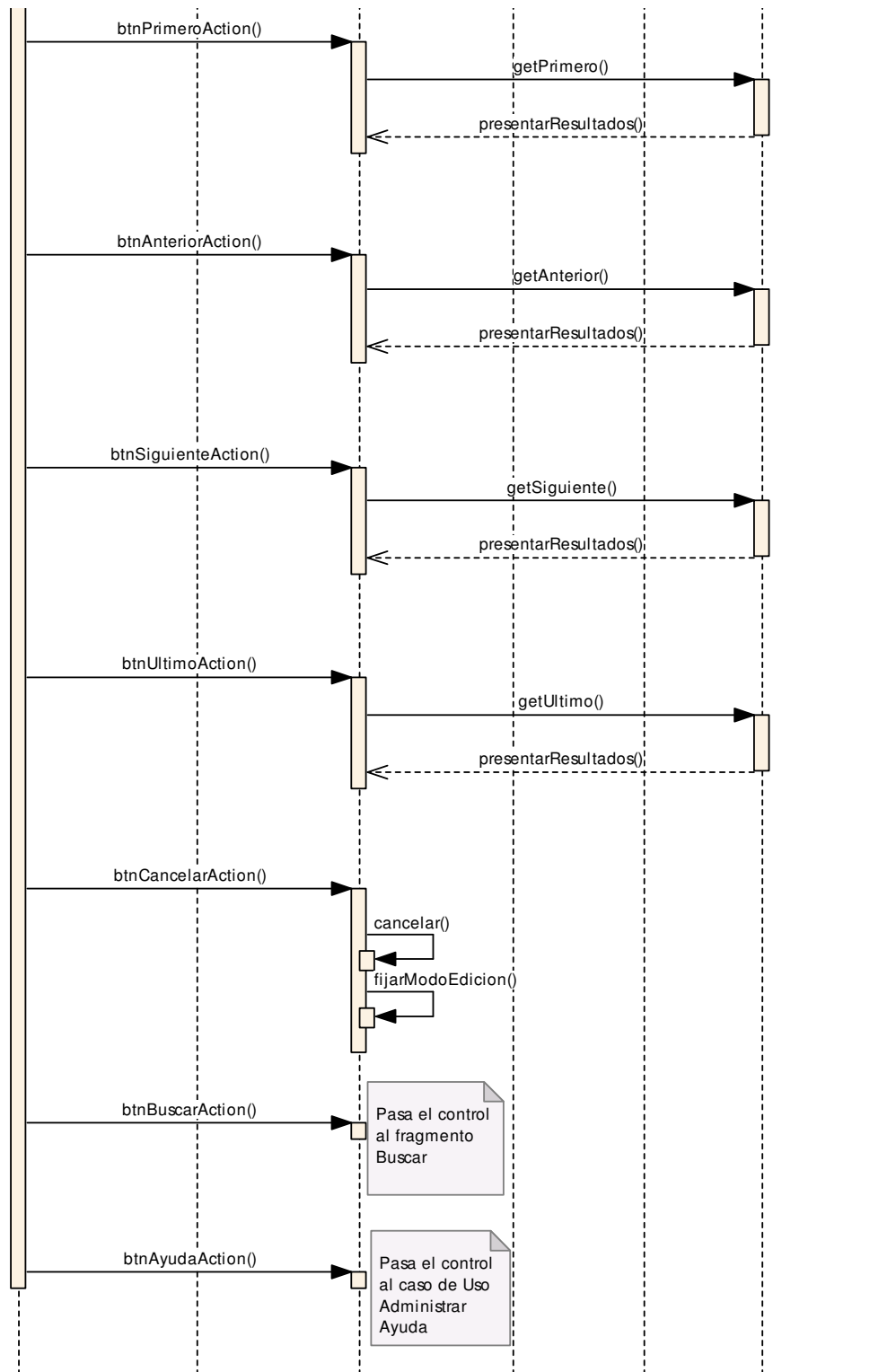


## 7.3.2.27 DS27-DR27 Asignar Función

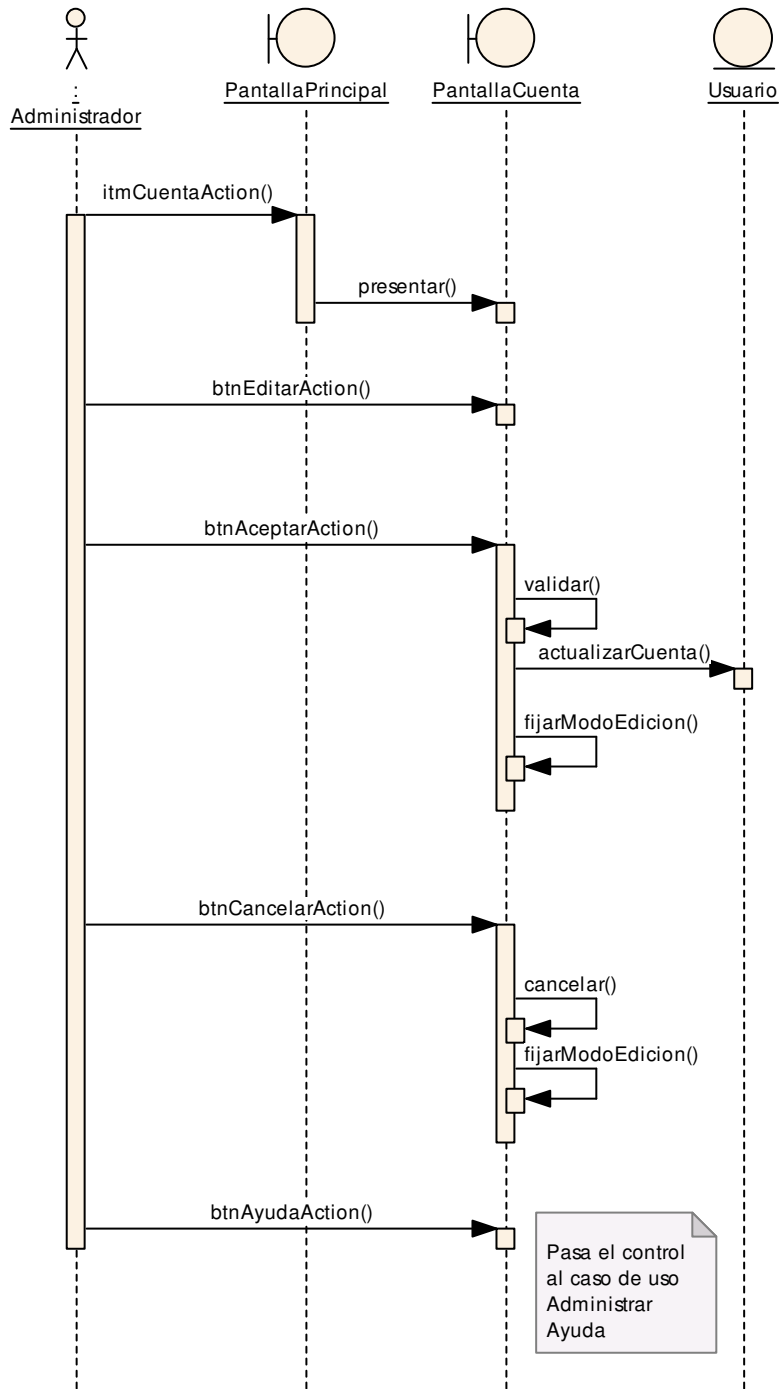


## 7.3.2.28 DS28-DR28 Mantener Usuario.





## 7.3.2.29 DS29-DR29 Mantener Cuenta



### 7.3.3 Diagrama de Clases.

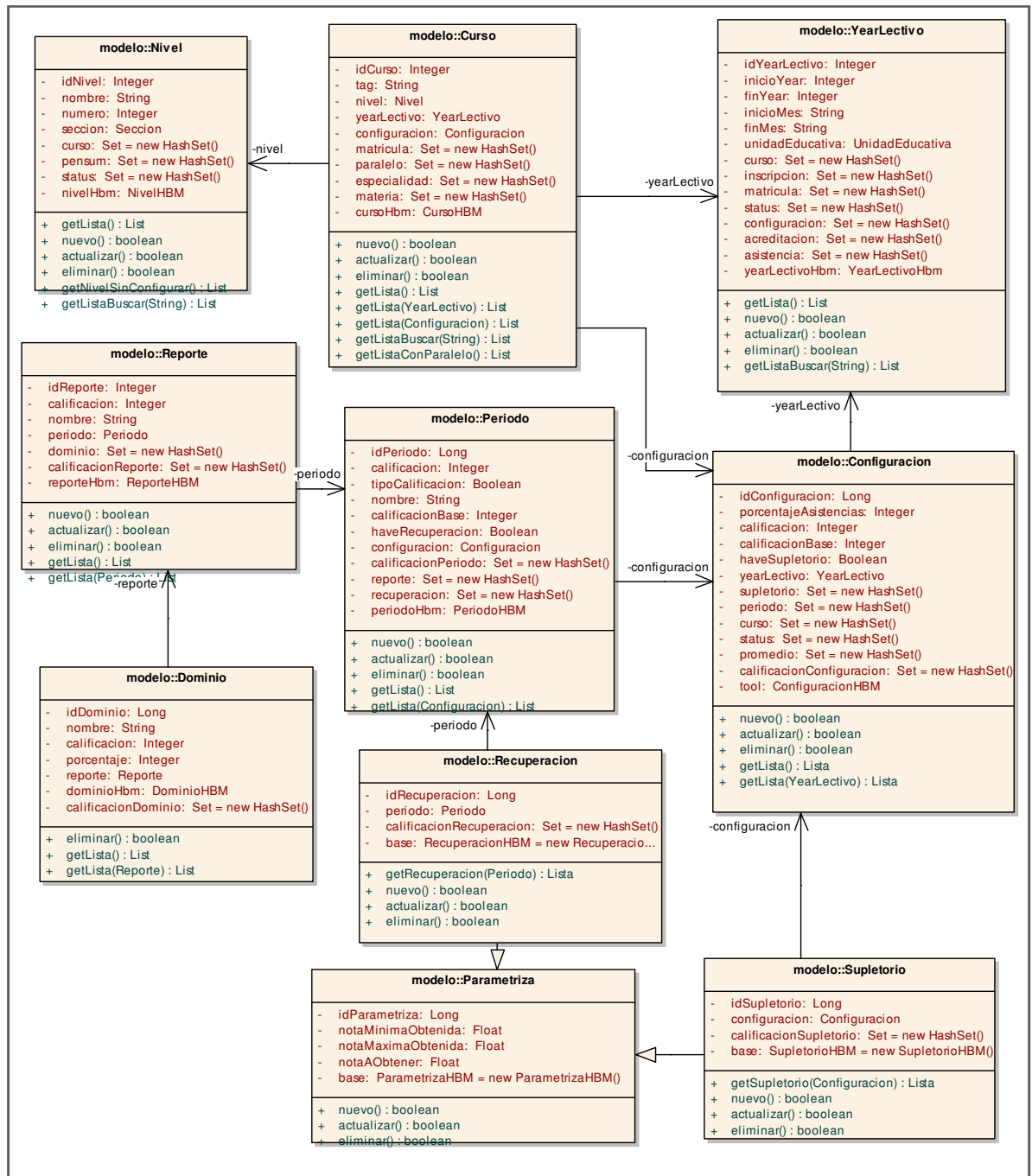


Diagrama de Clases 1. DC01 Configuración

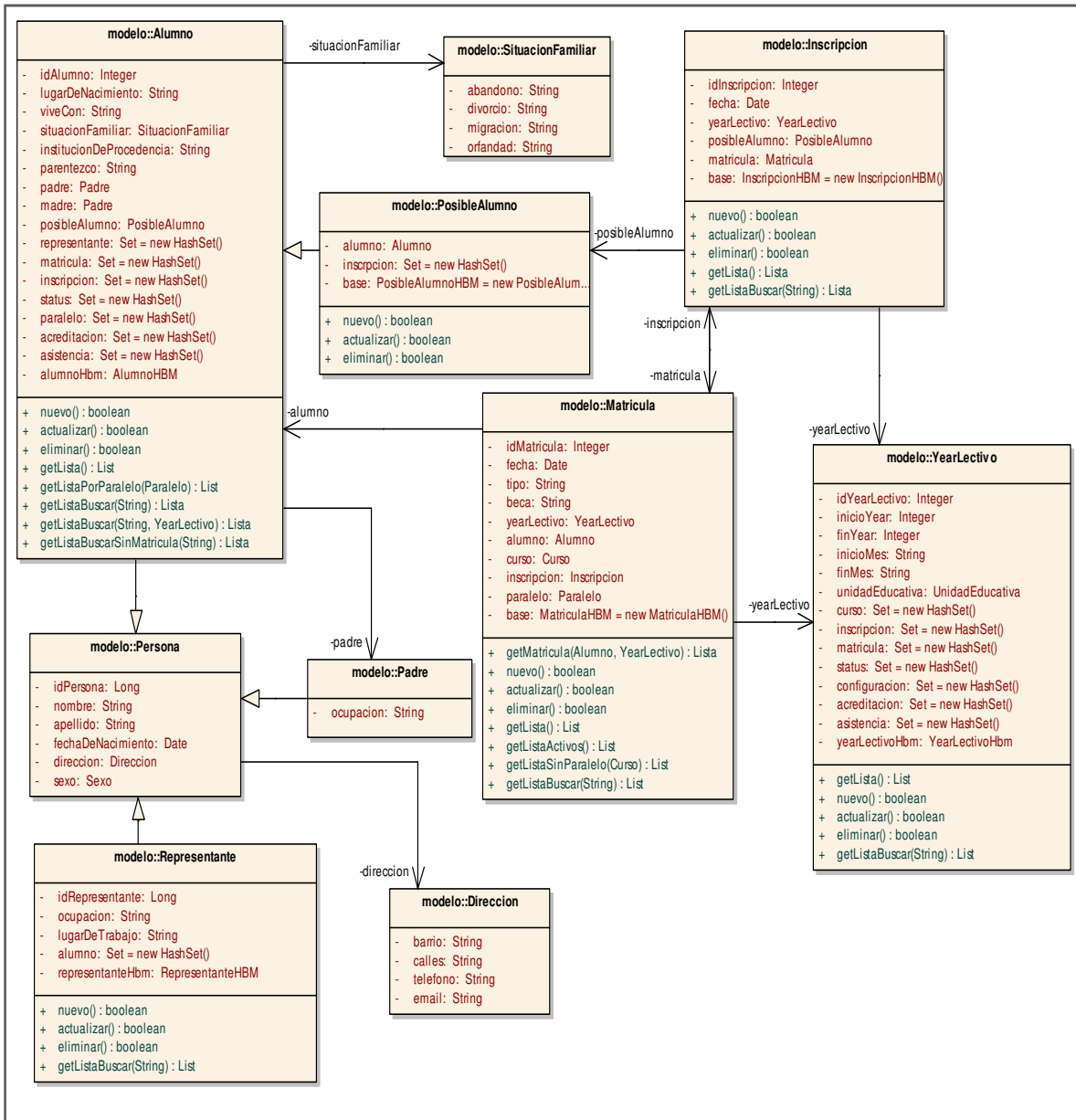


Diagrama de Clases 2. DC02 Matriculación

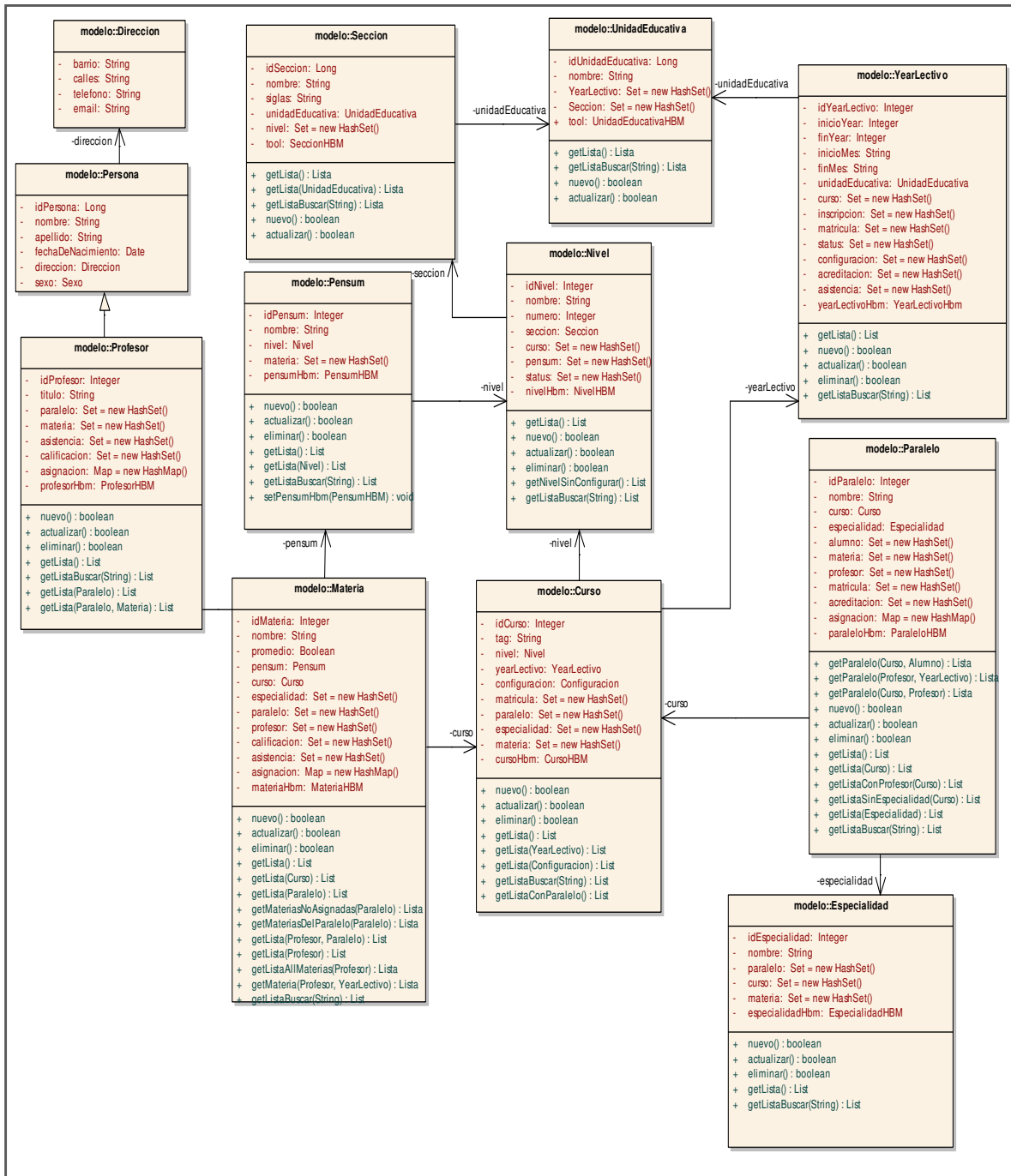


Diagrama de Clases 3. DC03 Estructural





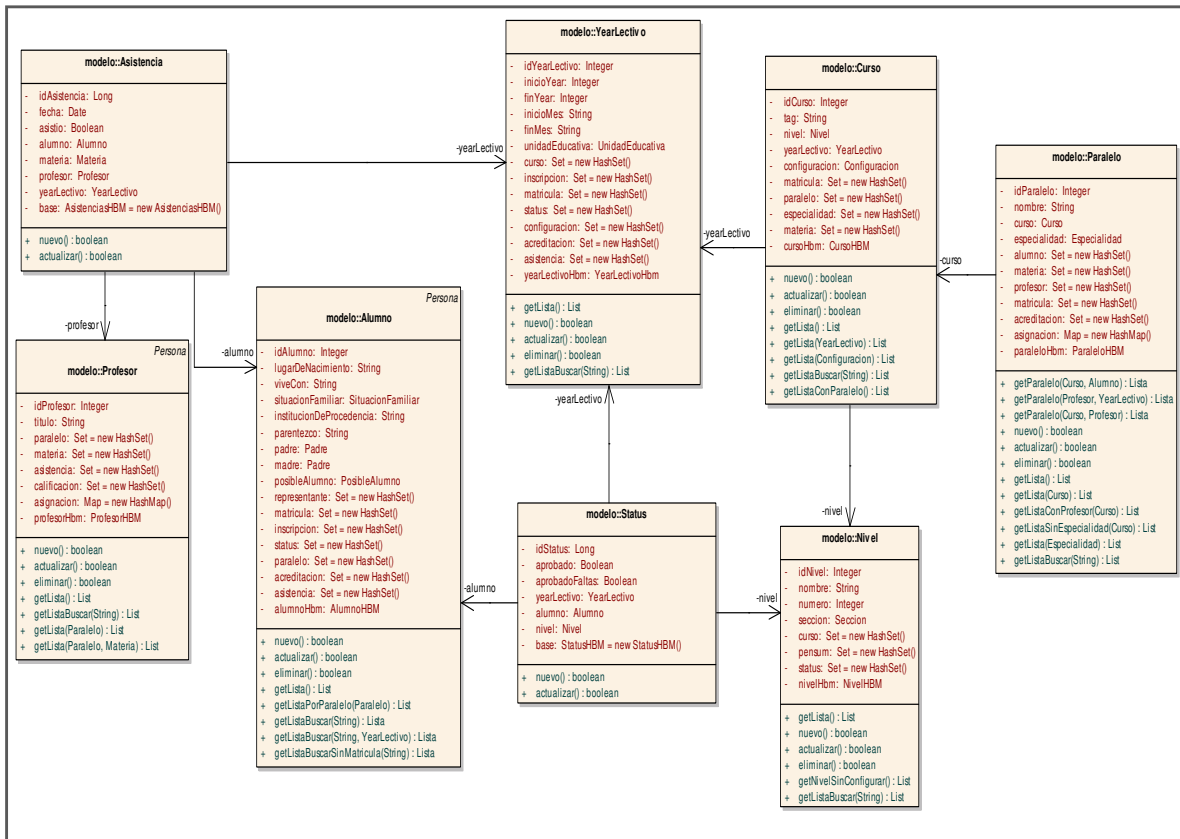


Diagrama de Clases 5. DC05 Asistencia

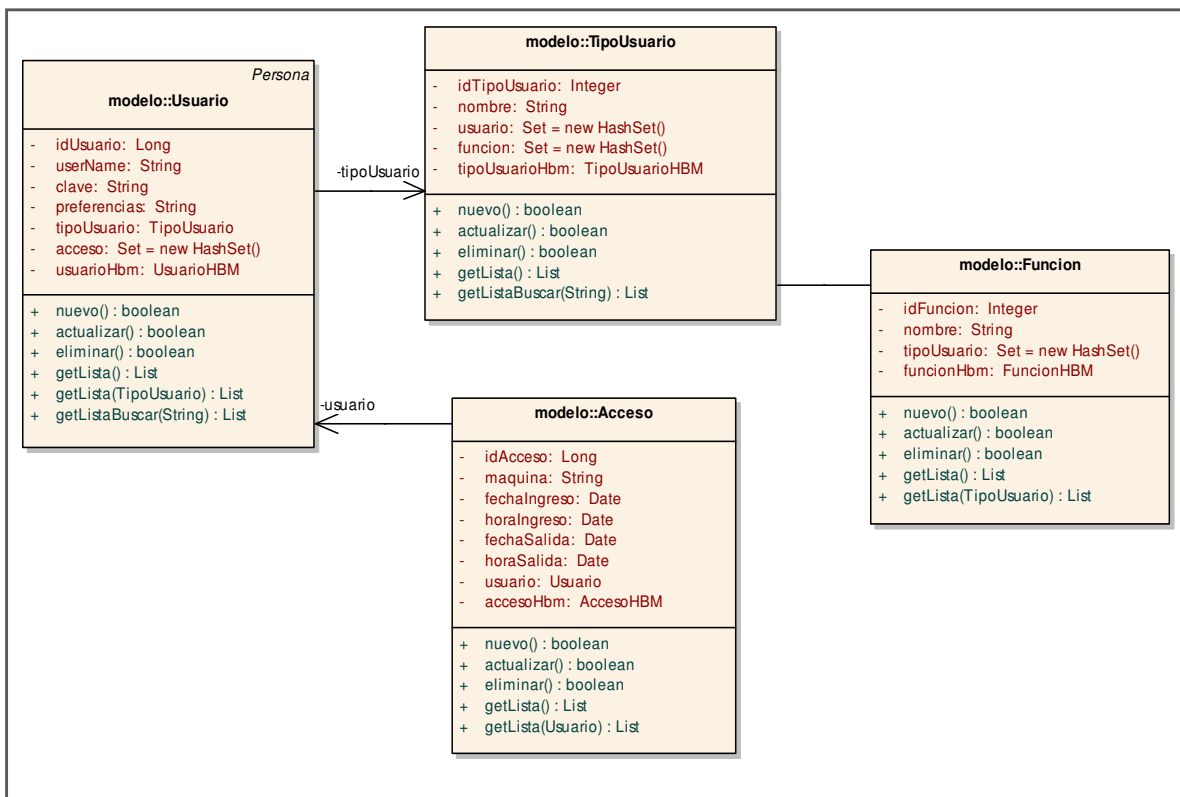


Diagrama de Clases 6. DC06 Usuarios

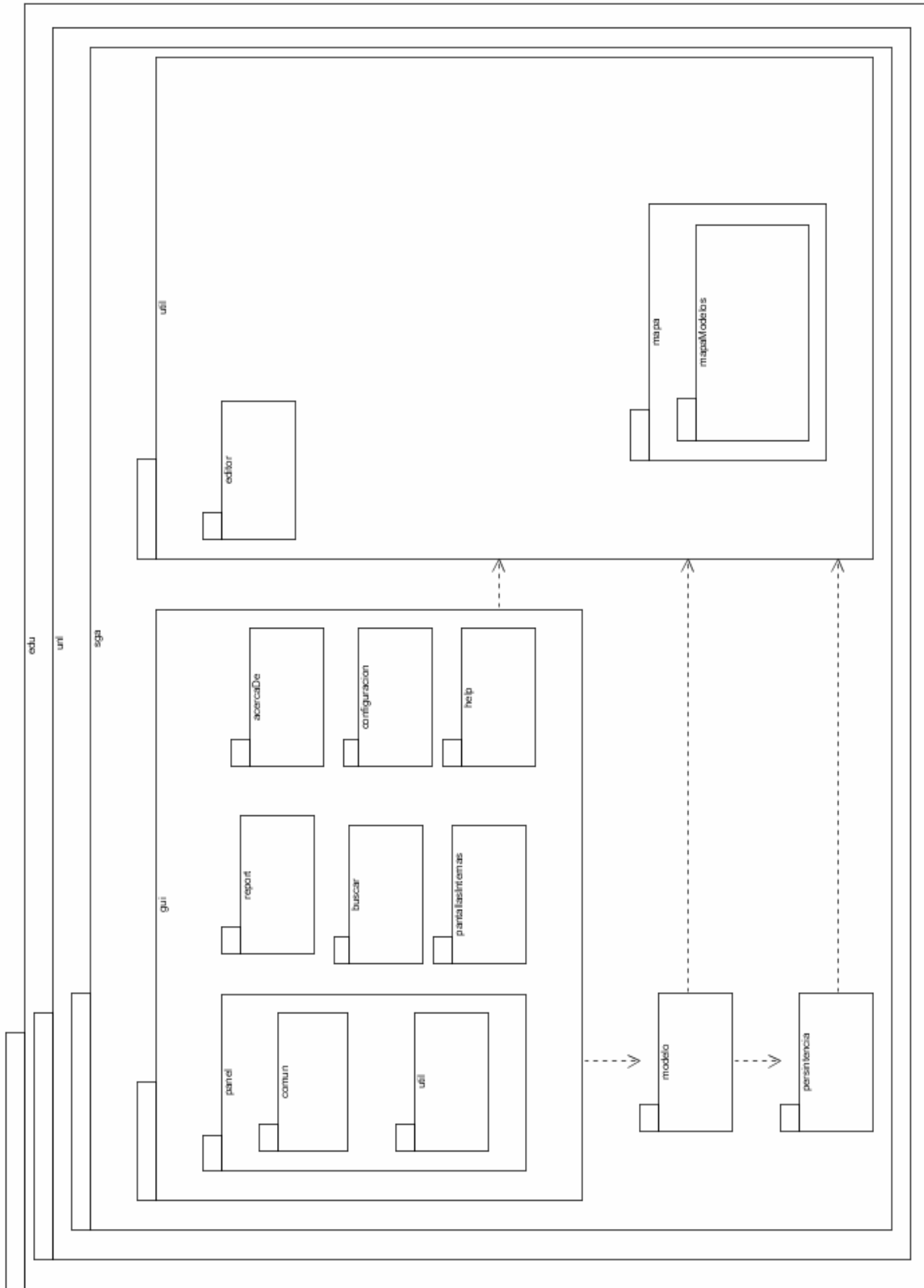
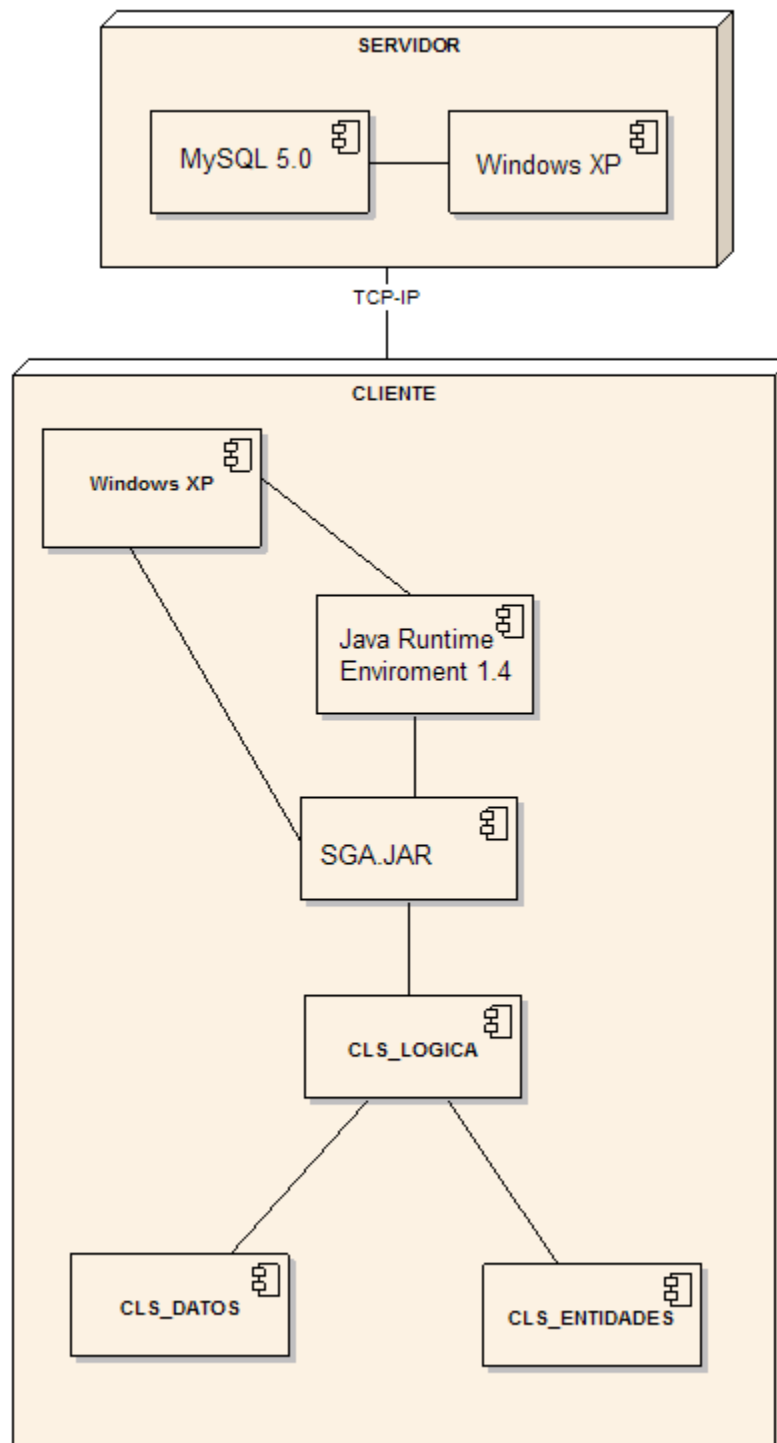


Diagrama de Paquetes 7. Diagrama de paquetes de Clases

## 8 Implementación.

### 8.1 Diagrama de Componentes.



## **8.2 Pruebas del Sistema.**

Para la elaboración de las pruebas del sistema se plantea la generación de un plan de pruebas el cual se describe a continuación:

### **8.2.1 Plan de pruebas**

#### **8.2.1.1 Introducción**

El proceso de prueba comienza con la generación de un plan de pruebas en base a la documentación sobre el proyecto y la documentación sobre el software a probar. A partir de dicho plan, se entra en detalle diseñando de pruebas específicas basándose en la documentación del software a probar. Una vez detalladas las pruebas (especificaciones de casos y de procedimientos), se toma la configuración del software (revisada, para confirmar que se trata de la versión apropiada del programa) que se va a probar para ejecutar sobre ella los casos. En algunas situaciones, se puede tratar de re ejecuciones de pruebas, por lo que es conveniente tener constancia de los defectos ya detectados aunque aún no corregidos. A partir de los resultados de salida, se pasa a su evaluación mediante comparación con la salida esperada. A partir de ésta, se pueden realizar dos actividades:

- La depuración (localización y corrección de defectos).
- El análisis de la estadística de errores.

La depuración puede corregir o no los defectos. Si no consigue localizarlos, puede ser necesario realizar pruebas adicionales para obtener más información. Si se corrige un defecto, se debe volver a probar el software para comprobar que el problema está resuelto.

Por su parte, el análisis de errores puede servir para realizar predicciones de la fiabilidad del software y para detectar las causas más habituales de error y mejorar los procesos de desarrollo.

Las técnicas de diseño de casos de prueba tienen como objetivo conseguir una confianza aceptable en que se detectarán los defectos existentes (ya que la seguridad total sólo puede obtenerse de la prueba exhaustiva, que no es practicable) sin necesidad de consumir una cantidad excesiva de recursos (por ejemplo, tiempo para probar o tiempo de ejecución). Toda la disciplina de pruebas debe moverse, por lo tanto, en un equilibrio entre la disponibilidad de recursos y la confianza que aportan los casos para descubrir los defectos existentes. Este equilibrio no es sencillo, lo que convierte a las pruebas en una disciplina difícil que está lejos de parecerse a la imagen de actividad rutinaria que suele sugerir.

En este documento se reúne la información necesaria, para planear y controlar el desarrollo de las pruebas de verificación y validación del Sistema de Gestión académica para la Unidad Educativa de la Universidad Nacional de Loja utilizando las pruebas de enfoque aleatorio.

#### **8.2.1.2 Alcance.**

El Sistema de Gestión Académico para la Unidad Educativa Anexa a la Universidad Nacional de Loja consta de los siguientes módulos, cuya funcionalidad se describe a continuación:

- **Configuración.** Permite el ingreso de la configuración básica inicial para el funcionamiento de un año lectivo donde comprende, inicio y fin del año lectivo, puntajes mínimos de asistencia, acreditación, niveles, recuperaciones, supletorios.
- **Usuarios.** Permite el ingreso de nuevos tipos usuarios para el sistema como la administración de niveles de seguridad de acceso a la información, asignación de tipos de usuarios a cuentas, edición de cuentas.
- **Pensum.** Permite la administración de un determinado pensum para el año lectivo vigente.
- **Materias.** Permite la administración de materias según el determinado pensum vigente.
- **Inscripción.** Permite la administración de las inscripciones realizadas en el año lectivo vigente.
- **Matrícula.** Permite la administración de matrículas realizadas en el año lectivo vigente.

- **Paralelos.** Permite la administración de paralelos en el año lectivo vigente.
- **Profesores.** Permite la administración de datos de profesores en el año lectivo vigente.
- **Asignación de Materias a Profesor.** Permite la administración de la asignación de materias a profesor en el año lectivo vigente.
- **Asignación de Alumnos a Paralelos.** Permite la administración de la asignación de alumnos a paralelos en el año lectivo vigente.
- **Asignación de Materias a Paralelos.** Permite la administración de la asignación de materias a paralelos en el año lectivo vigente.
- **Asistencias.** Permite el registro de asistencias diarias por alumno, materia y paralelo de los estudiantes que pertenecen a la Unidad Educativa.
- **Acreditaciones.** Permite el registro de acreditaciones parciales y generación de nota final por alumno, materia y paralelo de los estudiantes que pertenecen a la Unidad Educativa.
- **Reportes.** Permite la generación de reportes de impresión como matrícula, certificados, libretas.
- **Ayuda.** Se muestra una pantalla adicional donde se visualiza el tema específico del que se requiere la ayuda o permite ver los tópicos de la ayuda de forma global.

Los módulos antes mencionados serán evaluados tomando en cuenta los siguientes parámetros:

- Pruebas unitarias (componentes/ unidades simples).
- Pruebas de construcción (comportamiento de los componentes/ unidades al integrar datos reales).
- Pruebas del sistema (validan la función del negocio y analizan el rendimiento del sistema completo).
- Pruebas de aceptación (realizadas por el usuario final), para verificar la confiabilidad del sistema antes de ser puesto oficialmente en marcha.

Para cada una de las fases indicadas anteriormente, existirán los siguientes Tipos de Prueba:

- Funciones del negocio.

- Interfaz de usuario.
- Base de datos.

### **8.2.1.3 Equipo de Trabajo**

El equipo de trabajo que desarrollará el plan de pruebas en el presente proyecto es el siguiente:

- Desarrolladores.
- Usuarios (funcionarios de la Unidad Educativa Anexa a la Universidad Nacional de Loja: Jardín José Alejo Palacios Directora, Escuela Pedro Víctor Falconí Secretaria y Colegio Manuel Cabrera Lozano Secretaria).

### **8.2.1.4 Referencias**

Se toma como referencia la siguiente información:

- Casos de Uso (llevan implícitos los requerimientos funcionales).
- Requerimientos no funcionales.

### **8.2.1.5 Estrategia de pruebas**

La estrategia de pruebas presenta la aproximación recomendada para las pruebas de acuerdo al objetivo que persigue. Para cada tipo de prueba, se deberá suministrar una descripción de la misma y por qué está siendo implementada y ejecutada.

Las principales consideraciones para la estrategia de pruebas son las técnicas a ser usadas y el criterio de conocimiento cuando la prueba se haya completado.

Adicionalmente a las consideraciones provistas para cada prueba, las pruebas sólo deberían ser ejecutadas usando bases de datos controladas, en ambientes seguros.



### **8.2.1.6 Tipos de pruebas**

#### **Pruebas de Integridad de Datos**

Las pruebas de integridad de datos, buscan comprobar que el acceso y manipulación de los datos generados a través del Sistema de Gestión Académica, están de acuerdo a los datos de prueba utilizados. Cabe mencionar que en software en prueba la base de datos es manejada por una capa intermedia entre la capa del negocio y la base de datos donde la creación de sentencias sql es totalmente innecesaria.

Para realizar las pruebas de integridad de datos se utilizó una aplicación que permite manejar la base de datos directamente denominada SQLYog de licencia libre y gratuito y verificar los datos ingresados están almacenados en sus tablas correspondientes, como también el mismo software al momento de recuperara datos para consultas y reportes.

#### **Pruebas de Funcionamiento**

Los casos de uso son utilizados para validar las pruebas de funcionamiento, para comprobar que el sistema cumple con los requerimientos planteados al inicio de la elaboración del mismo y que dichos requerimientos han sido implementados con veracidad y eficacia.

#### **Pruebas de Aceptación de Usuario**

Las pruebas de aceptación nos permiten validar la interfaz con el usuario (pantallas, etiquetas, mensajes), como también que el software cumple con todas las funcionalidades necesitadas; para la ejecución de estas pruebas se necesita la presencia del usuario final. Para lo cual se laboró una encuesta (Ver Anexo 6.Encuestas.)

### **8.2.1.7 Metodología**

La metodología que se utiliza para el desarrollo de las pruebas antes descritas, comprende:

- **Bitácora de casos de prueba:** Se construirá en base a la especificación detallada de los casos de uso, que contenga la siguiente información:
  - Número de caso de prueba.
  - Código de escenario.
  - Código de caso de prueba.
  - Descripción del escenario.
  - Descripción del caso de prueba.
  - Confirmación o negación de la ejecución del caso de prueba.
  - Confirmación o negación de si genera error la ejecución del caso de prueba.
  - Fecha.
  - Estado Final.
  
- **Bitácora de errores:** Se generará con los resultados de la ejecución de los casos de prueba, ésta contendrá la siguiente información:
  - Número de error.
  - Código que vincula el caso de prueba y el escenario mediante la concatenación de sus códigos.
  - Nombre del error.
  - Descripción del error.
  - Tipo de error.
  - Prioridad.
  - Estado Inicial del error.

Los errores detectados en el desarrollo de los casos de prueba, se han tipificado como:

- **Error:** Aquellos que no permiten continuar con la ejecución del sistema, o su presencia producirá resultados incorrectos en la ejecución de otras funcionalidades de la aplicación. Tienen prioridad de corrección alta.

- **Fallo:** Son errores que no detienen el funcionamiento del sistema; sin embargo, producen datos ligeramente incorrectos, como por ejemplo al realizar el cálculo del saldo de una cuenta contable el valor obtenido difiere en decimales del valor correcto. Tienen prioridad de corrección media.
- **Defecto:** Son situaciones que pueden ser mejoradas, no afectan en el funcionamiento del sistema. Tienen prioridad de corrección baja.

Los estados que se han definido para los errores son:

- **Pendiente:** Significa que no se han tomado medidas correctivas para el error y que este continúa presente.
- **Corregido:** En este estado se encuentran los errores que han sido debidamente corregidos y los casos de prueba que los generan se han ejecutado nuevamente, para verificarlos.
- **Encuestas:** Para la evaluación de la aprobación por parte de usuario.

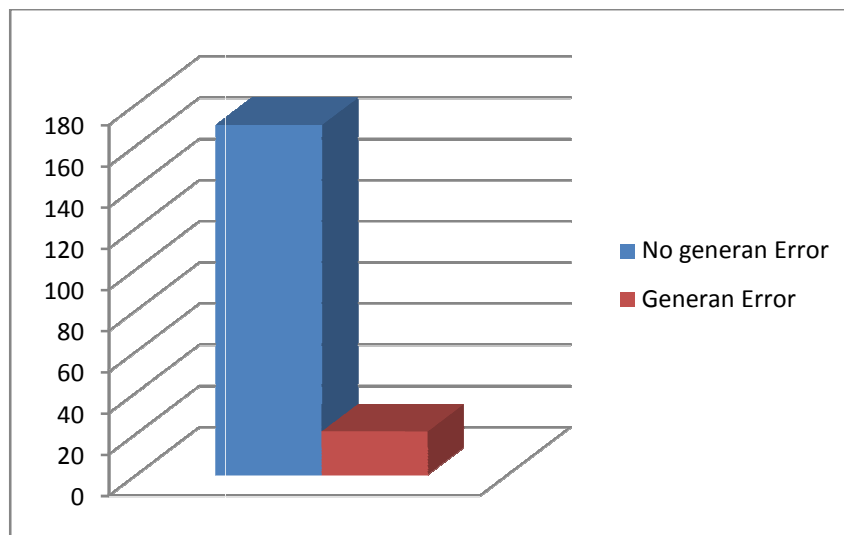
### 8.2.1.8 Informe de pruebas

El informe de pruebas, contendrá los resultados de la aplicación de los casos de prueba, los cuales se listan a continuación:

Para las pruebas de funcionalidad se determinó luego de la aplicación de bitácora de casos de pruebas con la bitácora de errores (Ver Anexo 5. Registros de Casos y Bitácoras de Prueba) los siguientes resultados:

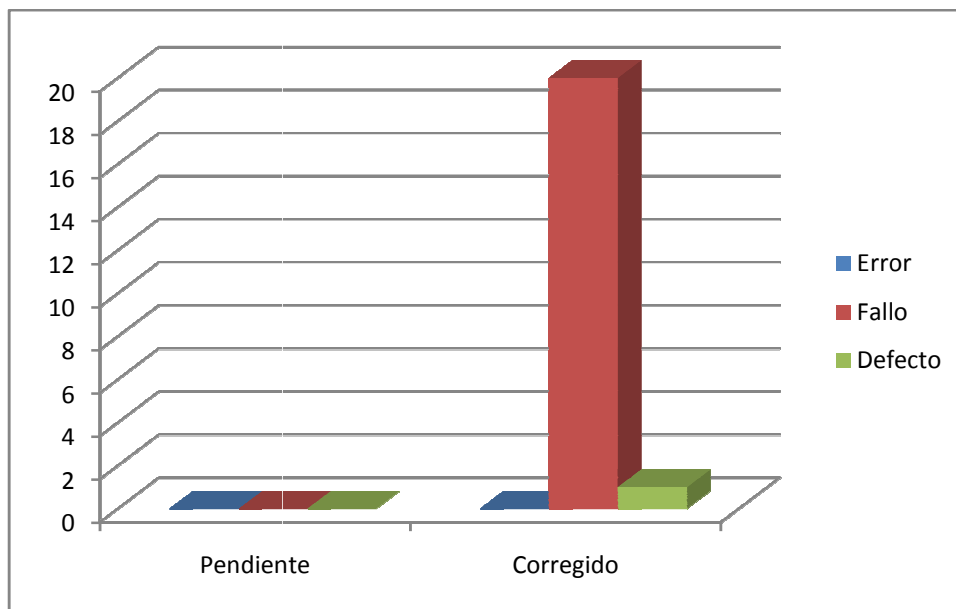
En el análisis de la bitácora de casos de prueba se pudo evidenciar que de los mismos un mínimo número de 21 presentaron errores mientras que 171 no presentaron error de los 191 casos de prueba planteados.

ANÁLISIS POR CASO DE PRUEBA	TOTAL	%
No generan Error	170	89.01%
Generan Error	21	10.99%
<b>TOTAL</b>	<b>191</b>	<b>100.00%</b>



En el análisis del tipo de error presentado se pudo evidenciar que la mayor cantidad de errores eran del tipo fallo al ser 21 mientras que del tipo defecto se encontró uno y del tipo error ninguno, de los cuales todos han sido corregidos como se muestra en el cuadro adjunto.

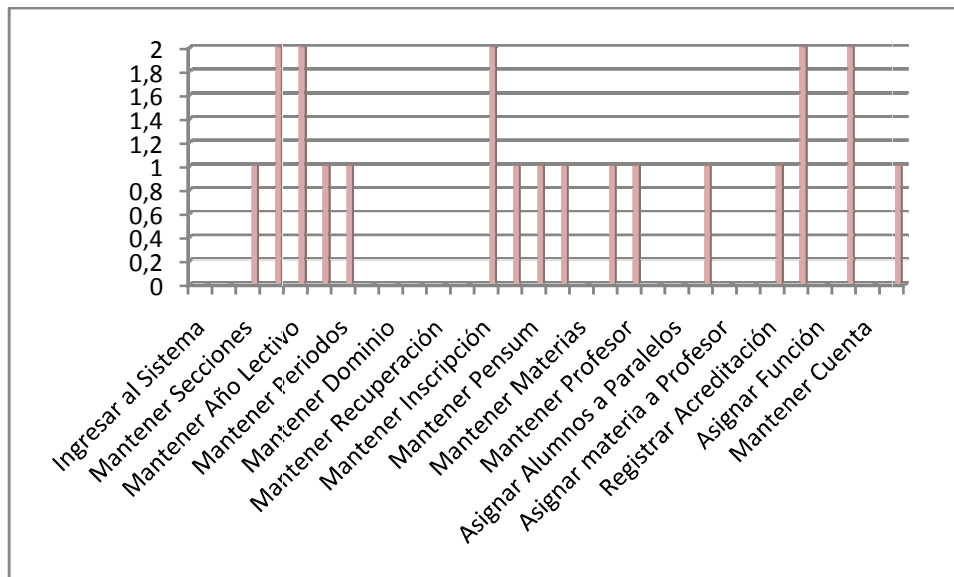
ANÁLISIS POR TIPO DE ERROR	ESTADO		TOTAL	%
	Pendiente	Corregido		
Error	0	0	0	0.00%
Fallo	0	20	20	95.24%
Defecto	0	1	1	4.76%
<b>TOTAL</b>	0	21	21	100.00%



En el análisis por caso de uso se evidenció en cual de ellos se presentaron mayor cantidad de errores como en los cuales se evidenció la menor cantidad, como se muestra en el cuadro adjunto:

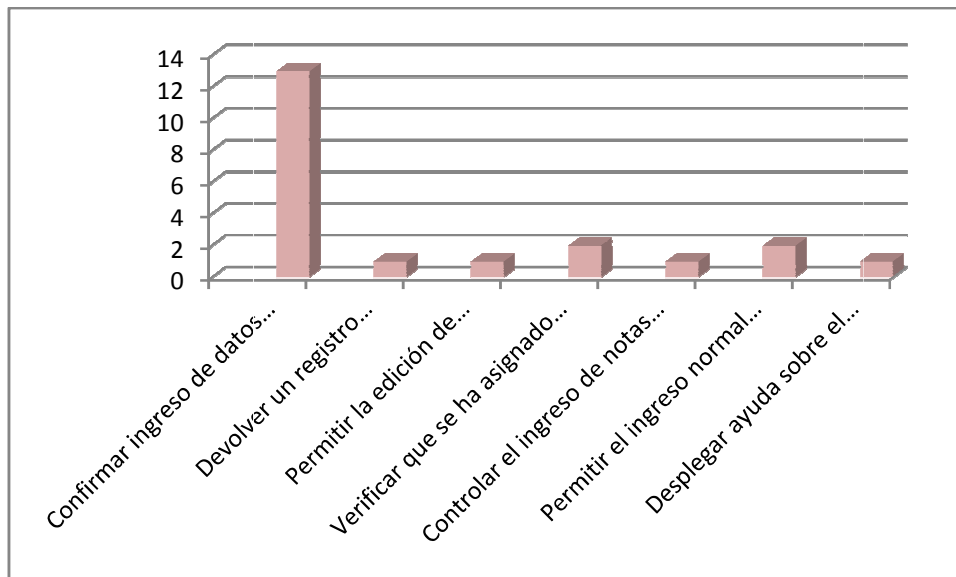
ANÁLISIS POR CASO DE USO	ERRORES	%
Ingresar al Sistema	0	0.00%
Mantener Unidad Educativa	0	0.00%
Mantener Secciones	1	4.76%

Mantener Niveles de Estudio	2	9.52%
Mantener Año Lectivo	2	9.52%
Mantener Configuración	1	4.76%
Mantener Periodos	1	4.76%
Mantener Reporte	0	0.00%
Mantener Dominio	0	0.00%
Mantener Cursos	0	0.00%
Mantener Recuperación	0	0.00%
Mantener Supletorio	0	0.00%
Mantener Inscripción	2	9.52%
Mantener Matrícula	1	4.76%
Mantener Pensum	1	4.76%
Mantener Especialidad	1	4.76%
Mantener Materias	0	0.00%
Asignar Materia a Paralelo	1	4.76%
Mantener Profesor	1	4.76%
Mantener Paralelos	0	0.00%
Asignar Alumnos a Paralelos	0	0.00%
Asignar Paralelos a Especialidad	1	4.76%
Asignar materia a Profesor	0	0.00%
Registrar Asistencia	0	0.00%
Registrar Acreditación	1	4.76%
Mantener Tipo de Usuario	2	9.52%
Asignar Función	0	0.00%
Mantener Usuario	2	9.52%
Mantener Cuenta	0	0.00%
Administrar Ayuda	1	4.76%
<b>TOTAL</b>	<b>21</b>	<b>100.00%</b>



En el análisis por funcionalidad se pudo detectar que la mayor cantidad de errores se generaban cuando no se validaba datos mínimos para grabar un registro como se muestra en el cuadro adjunto:

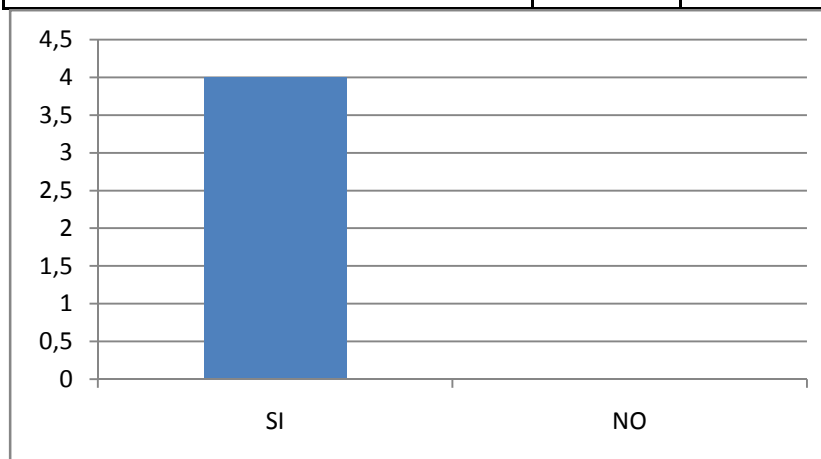
ANÁLISIS POR FUNCIONALIDAD	ERRORES	%
Confirmar ingreso de datos esenciales	13	61.90%
Devolver un registro específico que se está consultando	1	4.76%
Permitir la edición de registros ya almacenados	1	4.76%
Verificar que se ha asignado un registro específico	2	9.52%
Controlar el ingreso de notas máximas permitidas según la configuración	1	4.76%
Permitir el ingreso normal de datos	2	9.52%
Desplegar ayuda sobre el software en mención	1	4.76%
<b>TOTAL</b>	<b>21</b>	<b>100.00%</b>



Para las pruebas de aceptación del usuario se realizó una encuesta (Ver Anexos 6. Encuestas) a cada uno de los usuarios que manipularán el sistema de gestión académico y las respuestas a dicha encuesta se muestran a continuación:

Con relación al acceso al sistema se pudo evidenciar que el total de encuestados afirman que el acceso al sistema fácil como se muestra en el cuadro adjunto:

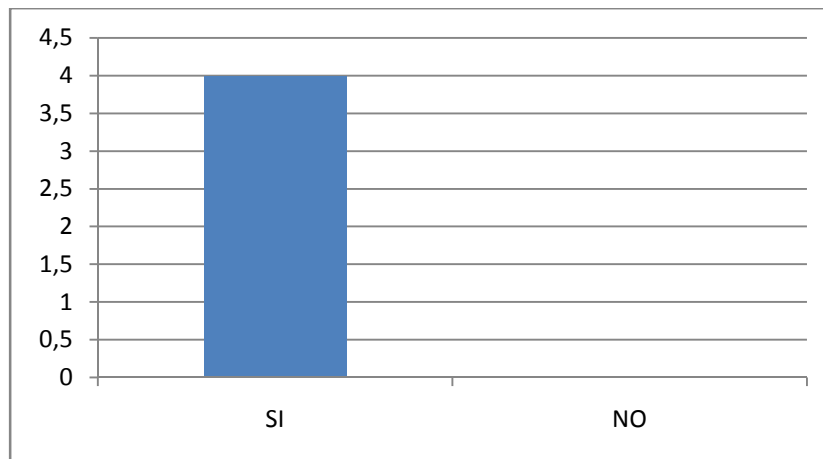
EL ACCESO AL SISTEMA ES FÁCIL	CANTIDAD	%
SI	4	100.00%
NO	0	0.00%
TOTAL	4	100.00%





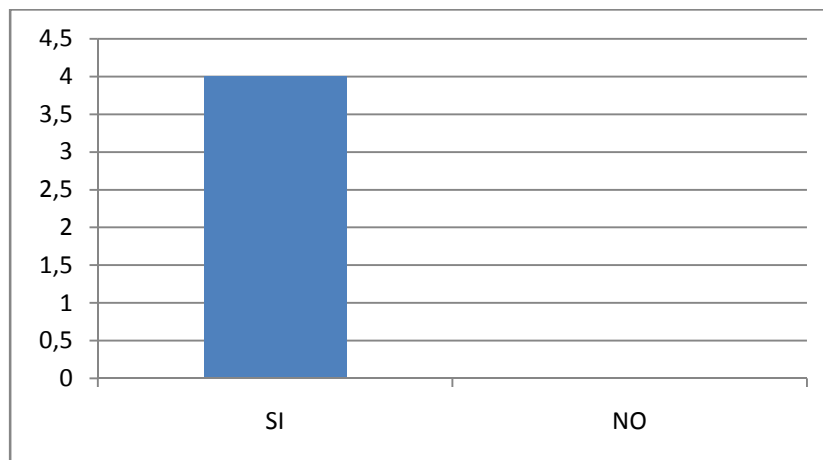
Con relación a la facilidad de exploración de menús como su sencillez para su entendimiento se pudo evidenciar que el total de encuestados afirman que los menús son sencillos y fáciles de usar como se muestra en el cuadro adjunto:

<b>LOS MENÚS DE EXPLORACIÓN SON SENCILLOS Y FÁCILES DE UTILIZAR</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
<b>TOTAL</b>	<b>4</b>	<b>100.00%</b>



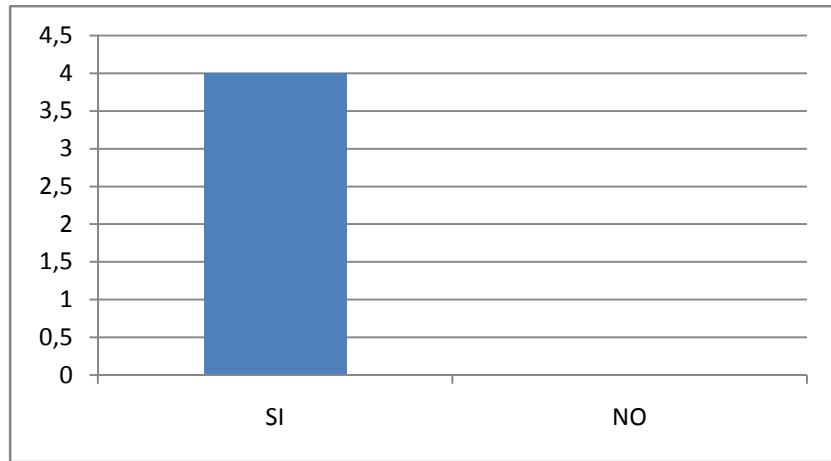
Con relación a la interfaz gráfica de software se pudo evidenciar que el total de encuestados afirman que las pantallas diseñadas para el software en mención son elegantes y atractivas como se muestra en el cuadro adjunto:

<b>EL DISEÑO DE LA PANTALLA ES ELEGANTE Y ATRACTIVO</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
TOTAL	4	100.00%



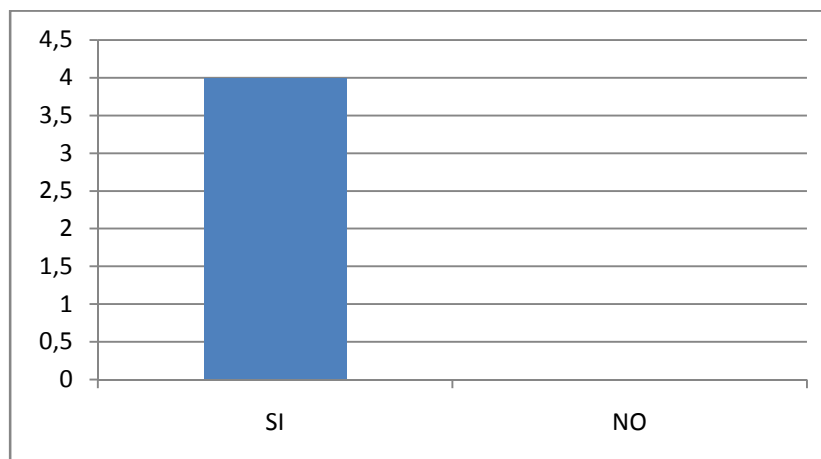
Con relación al ingreso de la información en el software se pudo evidenciar que el total de encuestados afirman que el ingreso de información es sencillo y específicamente lo necesario como se muestra en el cuadro adjunto:

<b>EL INGRESO DE LA INFORMACIÓN ES SENCILLO</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
TOTAL	4	100.00%



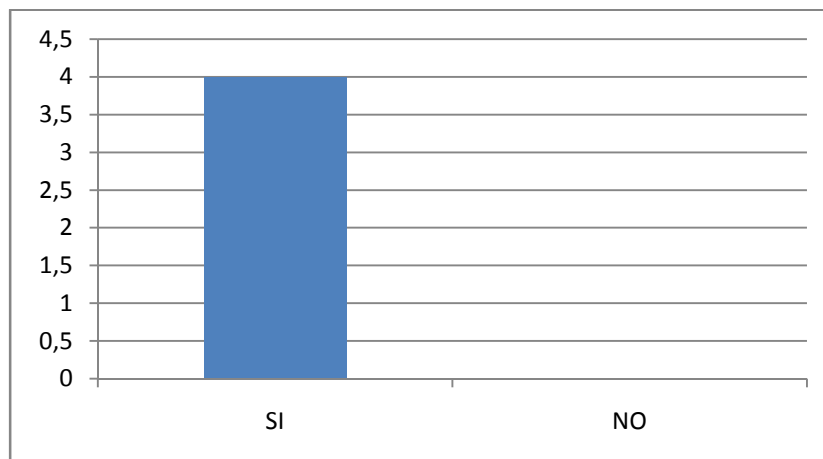
Con relación a la consulta de datos que permite el software se pudo evidenciar que el total de encuestados afirman que la cantidad de información que refleja cada una de las consultas que se realizan a través del software es la necesaria y correcta como se muestra en el cuadro adjunto:

<b>LA CONSULTA DE DATOS REFLEJA LA CANTIDAD DE INFORMACIÓN NECESITADA</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
<b>TOTAL</b>	<b>4</b>	<b>1</b>



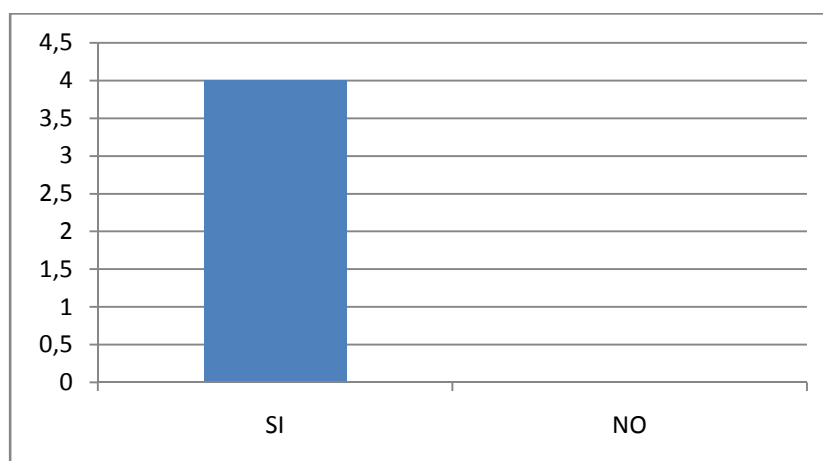
Con relación al menú de navegación que permite visualizar listados dentro del software se pudo evidenciar que el total de encuestados afirman que el menú de navegación es de fácil comprensión y muy sencillo de utilizar como se muestra en el cuadro adjunto:

<b>EL MENÚ DE NAVEGACIÓN ES DE FÁCIL COMPRENSIÓN</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
<b>TOTAL</b>	<b>4</b>	<b>1</b>



Con relación a los botones del panel de administración dentro del software se pudo evidenciar que el total de encuestados afirman que sus etiquetas como sus funciones son claras al momento de ser utilizados como se muestra en el cuadro adjunto:

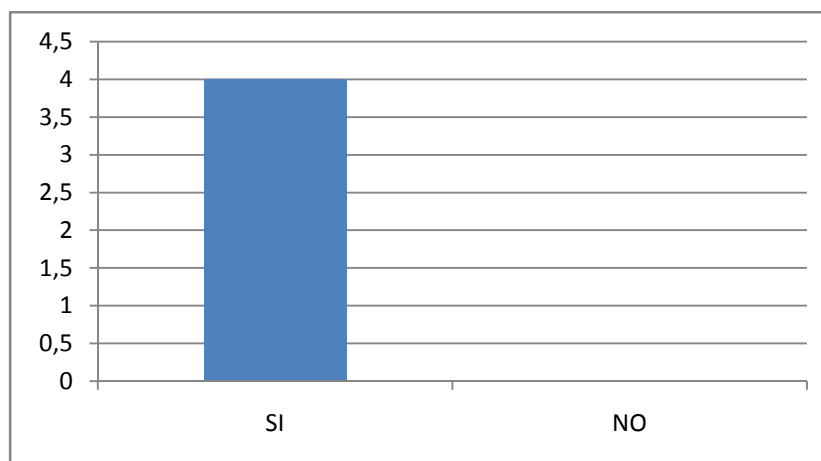
<b>LOS BOTONES DEL PANEL DE ADMINISTRACIÓN SON CLAROS</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
<b>TOTAL</b>	<b>4</b>	<b>1</b>



Finalmente se pregunto si el software implementado cumple con todas sus expectativas y requerimientos para lo cual se pudo evidenciar que el total de encuestados afirman que sus

que están contentos con el software ya que cumple sus expectativas y principalmente con los requerimientos de la Unidad Educativa Anexa a la Universidad Nacional de Loja como se muestra en el cuadro adjunto:

<b>EL SISTEMA CUMPLE CON LOS REQUERIMIENTOS Y LAS ESPECTATIVAS DEL USUARIO</b>	<b>CANTIDAD</b>	<b>%</b>
SI	4	100.00%
NO	0	0.00%
TOTAL	4	100.00%



## **9 Diseño de Red Tentativo**

### **9.1 Introducción**

La necesidad de compartir recursos e intercambiar información fue una inquietud permanente desde los primeros tiempos de la informática hasta la actualidad. Es por ello y gracias a los avances tecnológicos actuales, que los desarrolladores de la presente tesis plantean un diseño tentativo de red para la Unidad Educativa Anexa a la Universidad Nacional de Loja orientando dicho proyecto a la sistematización como el intercambio de datos de cada uno de los estamentos que lo conforman. Para lograr este objetivo, nada mejor que implementar una excelente red de datos, que permita la fácil y fluida circulación de información por todos y cada uno de sus estamentos que conforman dicha Unidad Educativa, por todos y cada uno de los eslabones que constituyen la pirámide jerárquica de la gestión académica, con apego a las mejores especificaciones técnicas y a la mejor visión futurista que en el momento sea posible.

La Unidad educativa anexa a la Universidad Nacional de Loja cuenta con la sede principal ubicada en la ciudadela universitaria.

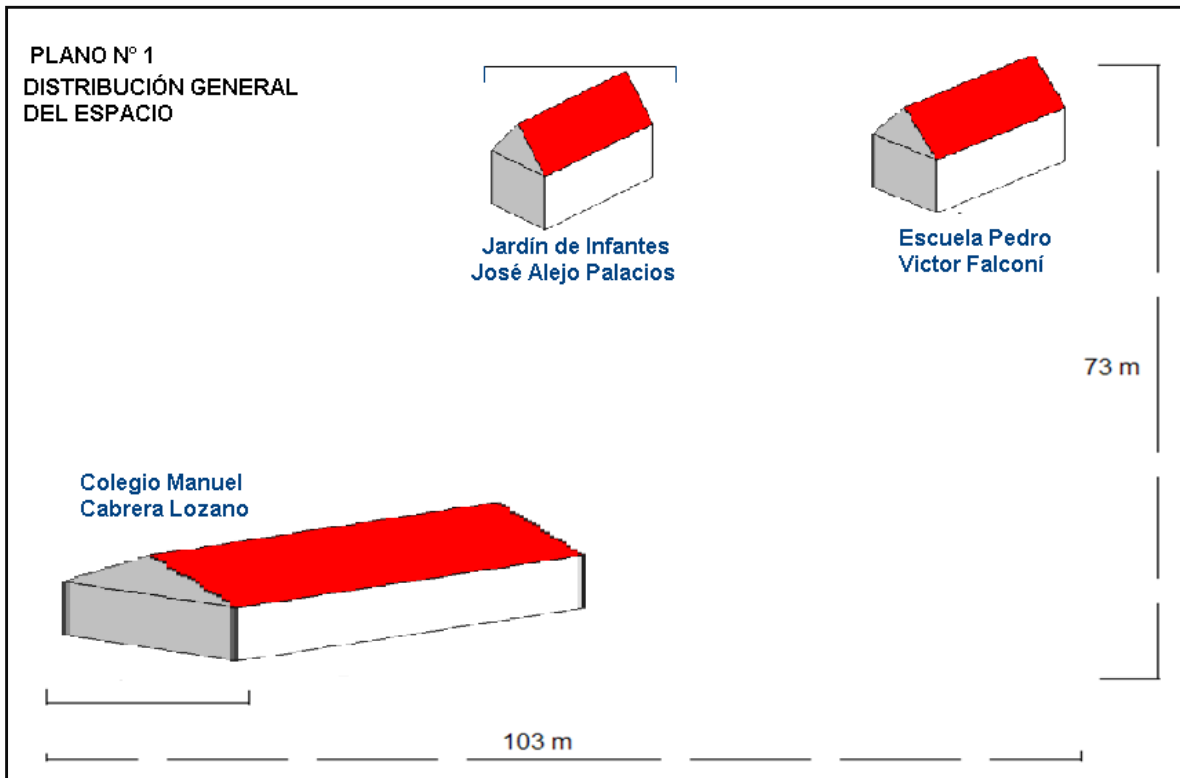
La sede principal donde opera la Unidad educativa mide 103 m x 73 m y posee 3 edificios con 1 piso administrativo cada uno. Allí se deben organizar las posibles estaciones de trabajo en las posibles dependencias.

Para realizar el diseño tentativo de red se ha definido los siguientes parámetros:

- Estructura física de la sede.
- Distribución de canaletas.
- Resumen de tendido de cables y tomas.
- Esquema de topología de red física y tecnología a utilizar.
- Estructura general del cableado horizontal.
- Diagrama jerárquico de los dispositivos de red.
- Selección de cada uno de los dispositivos y sus referencias.

## 9.2 Estructura Física

La Unidad Educativa anexa a la Universidad nacional de Loja consta con 3 edificios. Posee un área de 103 m x 73 m. La distribución específica de esta área se puede apreciar en el Plano Nro. 1, sobre “Distribución General de Espacio”. En este plano puede apreciarse con claridad sus dimensiones generales, ubicación de columnas, ventanas y entrada principal.

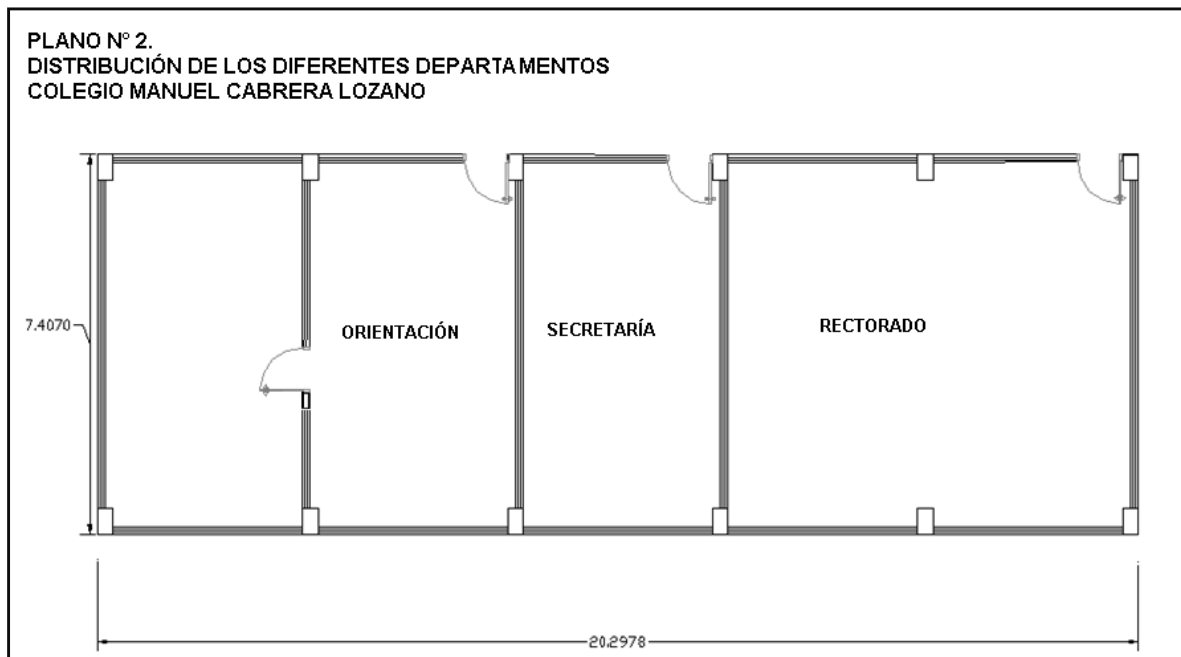


- La altura de los techos en todos los pisos, salvo especificación en contrario, es de 3.5 metros.
- Todos los pisos son de hormigón,
- El centro principal de cableado, se ubicará en el edificio del Colegio “Manuel Cabrera Lozano”.
- El cableado para las áreas de trabajo es distribuido por medio de canaletas o ductos marca LEGRAND, las cuales permiten aislar los cableados de datos de los cableados de potencia AC, o cualquier otro sistema que se implemente en un futuro. Estas canaletas tienen una dimensión de 75 x 50 milímetros.

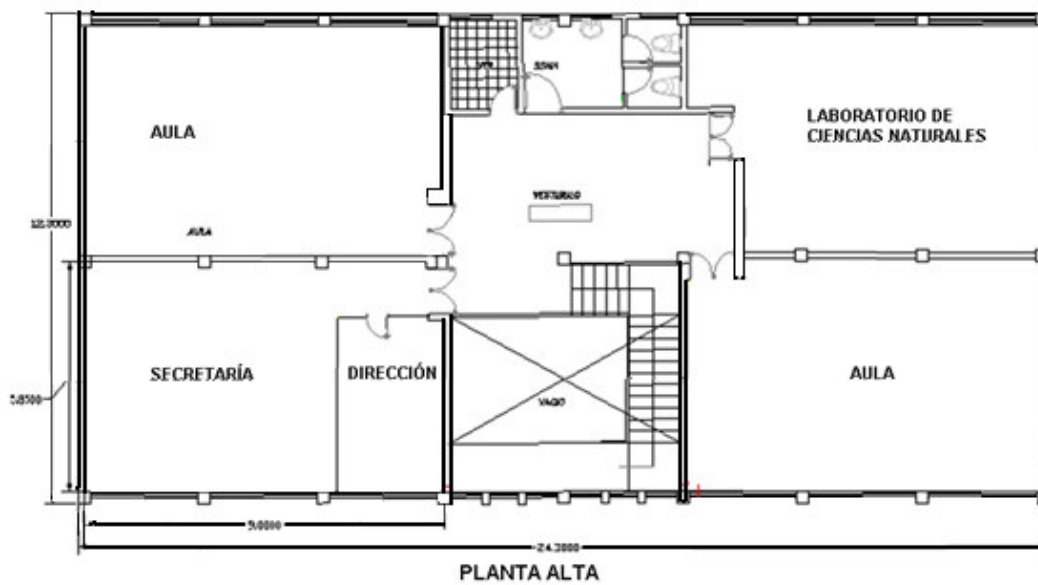


- En el *Plano Nro. 2*, *Plano Nro. 3* y *Plano Nro. 4*, Mostrados a continuación se puede apreciar en detalle la distribución de las diferentes áreas.

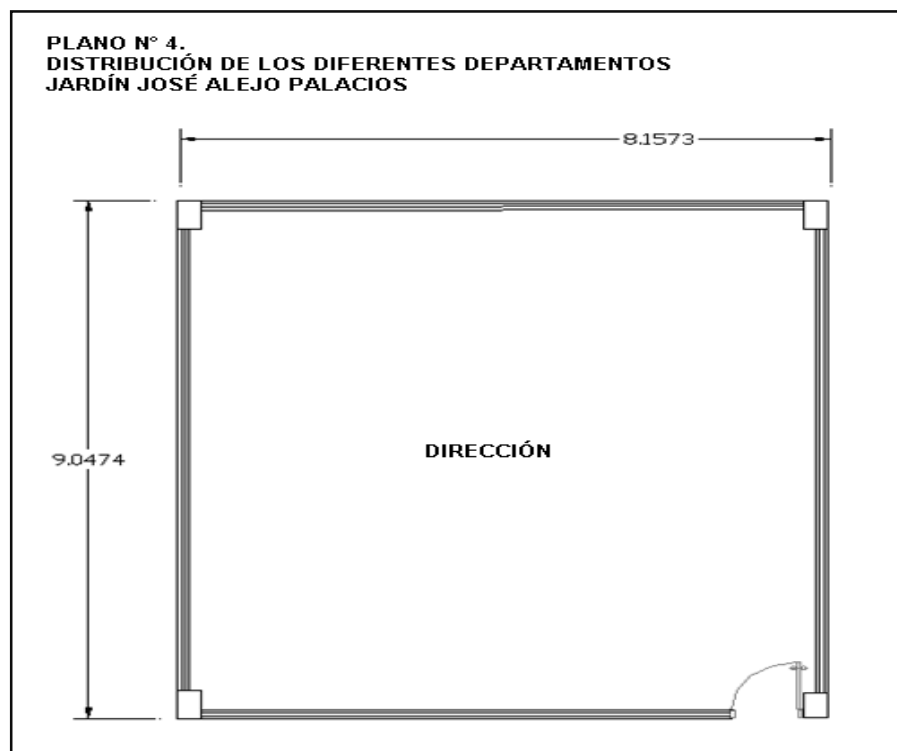
Plano Número	Dependencia	Tipo de Equipo	Cantidad
2	Rectorado Colegio “Manuel Cabrera Lozano”	Computador de escritorio	1
2	Secretaría Colegio “Manuel Cabrera Lozano”	Computador de escritorio Impresora Laser	1 1
2	Orientación Colegio “Manuel Cabrera Lozano”	Computador de escritorio Impresora Matricial	1 1
3	Dirección Escuela “Pedro Víctor Falconí”	Computador de escritorio Impresora matricial	1 1
3	Secretaría Escuela “Pedro Víctor Falconí”	Computador de escritorio Impresora matricial	1 1
4	Dirección Jardín “José Alejo Palacios”	Computador de escritorio Impresora matricial	1 1



PLANO N° 3.  
DISTRIBUCIÓN DE LOS DIFERENTES DEPARTAMENTOS  
ESCUELA PEDRO VÍCTOR FALCONÍ

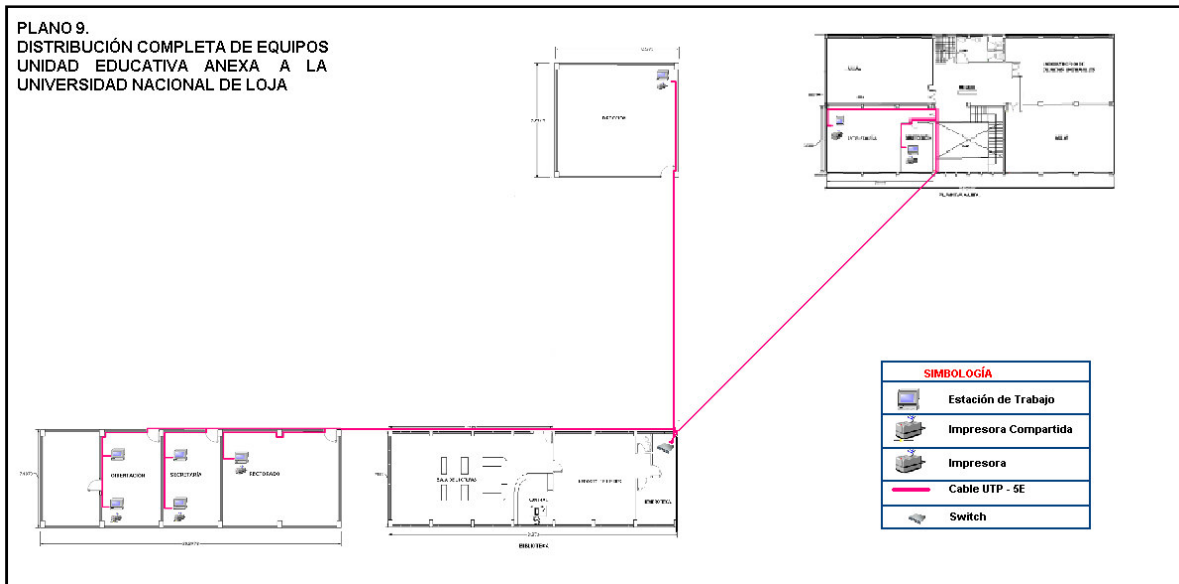


PLANO N° 4.  
DISTRIBUCIÓN DE LOS DIFERENTES DEPARTAMENTOS  
JARDÍN JOSÉ ALEJO PALACIOS



### 9.3 Cableado Horizontal

Plano Nro. 9 Distribución Completa de Equipos Unidad Educativa Anexa a la Universidad Nacional de Loja, muestra al detalle cómo se halla la distribución del cableado UPT-5E para los diferentes nodos, así mismo los circuitos que quedan disponibles ya que el switch es de 16 puertos.



La canaleta utilizada es de tipo doble cavidad, marca LEGRAND.

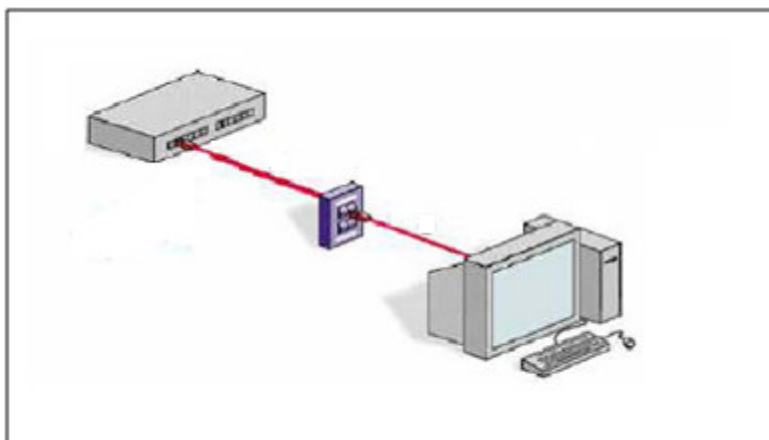


Gráfico 9-1. Cableado Horizontal.

#### **9.4 Cable UTP**

El cable UTP para el cableado horizontal de datos será Categoría 5, deberá estar conformado de 4 pares (8 hilos) de conductores sólidos de cobre. El cable debe permitir la transmisión de datos a altas velocidades (100Mbps, 155 Mbps, 1000 Mbps) y presentar un ancho de banda aprobada de 250 MHz, deberá soportar los siguientes estándares: LAN 100 BASE TX, ATM, Gigabit Ethernet.

#### **9.5 Jacks Y Tomas**

Acorde a la norma EIA/TIA-568-A, se utilizará Jacks y tomas tipo RJ45 con el fin de conectar el cable UTP.

La terminación mecánica de los cables horizontales en el área de trabajo será en conectores tipo jack modular RJ45 Categoría 5 de 8 posiciones. El jack modular RJ45 deberá permitir configuraciones 568A y 568 B. La conexión mecánica entre los 8 conductores del UTP horizontal y las 8 posiciones del Jack RJ45 deberá ser en contactos IDC (**I**nsulation **D**isplacement **C**onector = “Conector con puntos de contacto aislados”) con corte diferente a 90°, con recubrimiento de plata (no estaño) y/o de capa de oro de 50 micro pulgadas en el área de contacto para proporcionar una conexión libre de corrosión en el transcurso del tiempo y proporcionar un desempeño confiable en ambientes hostiles (calor y frío extremo, humedad y exposición a sales).

Los IDC (Insulation Displacement Connector = “Conector con puntos de contacto aislados”) de los Jack´s Modulares RJ45 deben tener la capacidad de terminar cables de calibres 22, 24 y 26 AWG y un mínimo de 2 conductores del mismo calibre por contacto, así como permitir la terminación de cables filamentados. El contacto debe soportar la terminación de cables de calibres más pequeños después de insertar y remover cables de calibre mayor sin necesidad de modificarlos o ajustarlos.

El cuerpo del Jack Modular debe estar construido de material termoplástico de alto impacto, retardante de flama, inhibidor de los rayos UV y clasificado bajo la UL 94V-0 y debe tener espacio suficiente para soportar placas de pared de 1, 2, 4 o 8 ventanas, además que permita la entrada a plug de 2, 4, 6 o 8 hilos en forma indistinta sin presentar deformación.



Gráfico 9-2. Jack Rj-45<sup>32</sup>

## 9.6 Distribución de Cableado:

Siguiendo el estándar TIA/EIA-568-A sobre especificaciones para una LAN-ETHERNET, el tendido del cableado horizontal estará conectado a un punto central de la Unidad Educativa Anexa a la Universidad Nacional de Loja, conformando una topología en estrella.

## 9.7 Racks

- Se debe adquirir un rack para colocar y proteger los equipos de comunicación ya que no existe un cuarto o buitrón específico para comunicaciones, en el cual además se podrá incluir equipo de comunicaciones que en un futuro sea necesario. Este Racks es el elemento sobre el cual se soporta toda la estructura del cableado; el tipo de rack seleccionado es un mueble rack de 19" cuyas características son:
- El rack profesional de seguridad para redes RACK LAN de 19", ha sido especialmente diseñado para las cajas de servidores de 1U, 2U y 4U. Su estructura lo hace ideal para la instalación de clusters en centros de procesos de datos. Al aplicar la tecnología del Switch, racionaliza extraordinariamente el espacio disponible, a la vez que facilita al máximo su administración.

<sup>32</sup> [www. www.controlp.com/Jacks.asp](http://www.controlp.com/Jacks.asp)

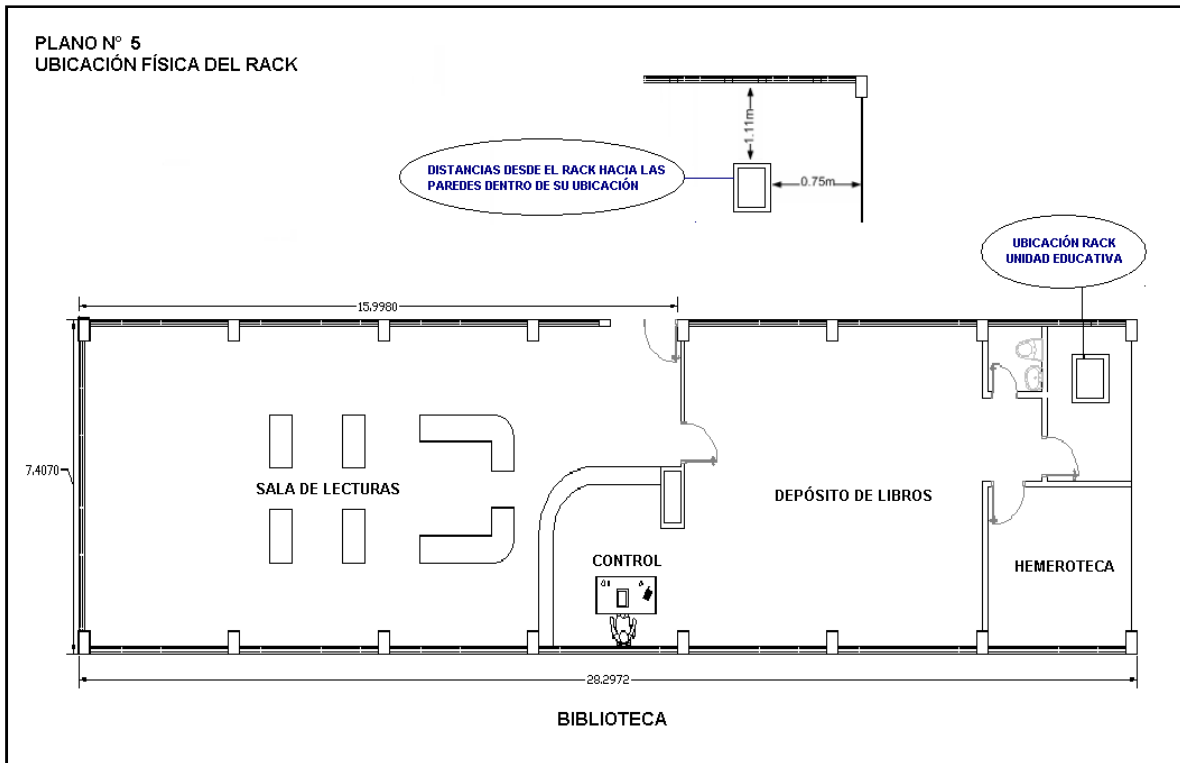


**Gráfico 9-3. Racks<sup>33</sup>**

- Este rack tiene la capacidad adicional de incluir dentro del mismo el servidor como los equipos de comunicaciones cuyas características adicionales son:
- Bandejas extraíbles para teclado y ratón, Sistema de ventilación en la parte superior regulable de gran potencia, rejillas de ventilación tanto en la puerta delantera como en la trasera, estabilizadores deslizantes para la extracción total de las bandejas sin peligro de vuelco, aperturas laterales para un rápido acceso al sistema, canalización para los cables, que evitan enredos y errores.
- Entre la pared y la parte posterior de los racks, como mínimo debe haber una distancia de 75 cm. Y a los costados un mínimo de 1.11 metros, espacio suficiente para que el encargado del mismo pueda moverse y manipular el rack.
- Paredes pisos y techos. El cuarto no tiene techo falso, garantizando así la seguridad de la instalación.

A continuación se muestra en el plano N° 5, la ubicación física del rack dentro de la Unidad Educativa Anexa a la Universidad nacional de Loja.

<sup>33</sup> [www.flytech.es/ArmariosRack/ArmarioRack.htm](http://www.flytech.es/ArmariosRack/ArmarioRack.htm)



### 9.8 Especificaciones Ambientales. Temperatura y Humedad.

- Temperatura aproximada 21°C.
- Humedad relativa entre el 30% y el 50%
- No hay cañerías de agua ni de vapor que atraviesen o pasen por encima del punto central
- A un costado de los rack hay 1.11 metros.

### 9.9 Topologías de la Red

La topología de una red, define básicamente la distribución del cable que interconecta los diferentes equipos, las estaciones de trabajo (WS). A la hora de instalar una red es importante seleccionar la topología más adecuada a las necesidades del usuario, debiendo tenerse en cuenta entre otros los siguientes factores:

- La distribución de los equipos a interconectar.
- El tipo de aplicaciones que se va a ejecutar.
- La inversión que se quiere hacer.

- El costo presupuestado para actualizaciones y mantenimiento de red.
- El tráfico que la red deba soportar.
- La capacidad de crecimiento o expansión futura.

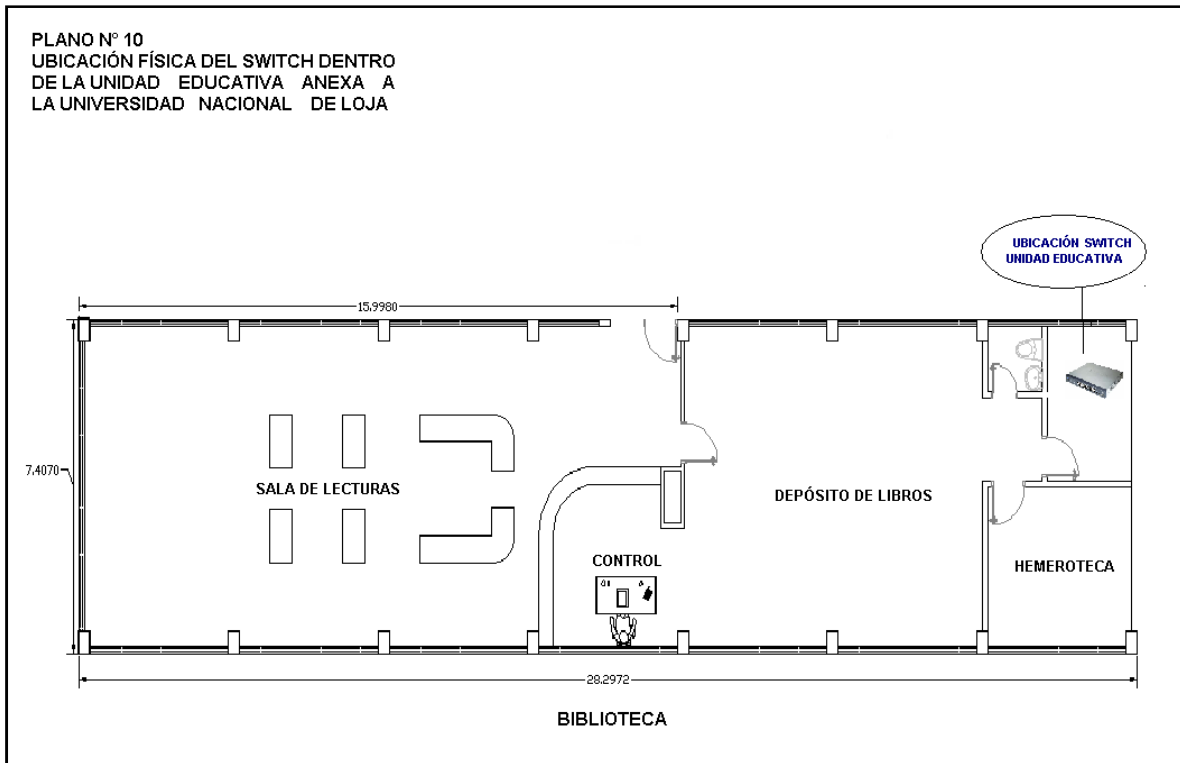
La topología es la estructura que forman el medio de transmisión y las estaciones conectadas al medio.

### **9.10 Topología Física**

Se utilizará una topología en estrella extendida (cableada en forma de estrella). El punto central de esta estrella estará ubicado en el Colegio Manuel Cabrera Lozano. Las estaciones de trabajo (PC), se distribuyen a lo largo y ancho de todas las dependencias de la Unidad educativa, cada una de estas WS (Work Station = Estación de Trabajo), poseen una NIC (Tarjeta de interface de red), la cual les permite intercambiar archivos, enviar correo electrónico etc., es decir compartir dispositivos y servicios. Cada NIC se conecta a su respectivo socket RJ-45 (Wall Plate), y de allí a un puerto en el switch a través de un medio UTP-5.

La decisión de usar switch se hizo con base en primera instancia en crecimientos futuros, que sean fáciles y dinámicos, que no requiera costos mayores. Y con el fin de garantizar seguridad y transparencia en el manejo de los datos, y esto se logra fácilmente usando este tipo de switch el cual será descrito en el siguiente punto, en el plano N° 10 se muestra la ubicación física del switch dentro de la Unidad educativa Anexa a la Universidad Nacional de Loja.





### 9.10.1 18SWITCH GIGABIT ETHERNET D-LINK 10/100/1000 16 PUERTOS SOHO DGS-1216T

“Las características por las cuales se ha escogido este tipo de switch se describen a continuación incluyendo como una de las principales su costo relativamente bajo de 256 dólares sin iva con relación a otros equipos con las mismas características.

El Switch Gigabit Ethernet DGS-1216T ofrece 16 puertos 10/100/1000Mbps + 2 Mini GBIC (SFP) y un puerto de consola RS-232, está especialmente diseñado para una fácil instalación y alto rendimiento en ambientes donde el tráfico de la red y el número de usuarios crece constantemente. Creando una variedad de sistemas administradores de segmentos de la red, el Switch DGS-1216T puede soportar el alto tráfico de datos que se produce al utilizar aplicaciones críticas tales como: CAD, Vídeo conferencia, Voz sobre IP, etc.

## CARACTERISTICAS TECNICAS

- Estándares. El Switch DGS-1216T es compatible con IEEE 802.3 10BASE-T, 802.3u 100BASE-TX, 802.3ab 1000BASE-T, 802.3z Gigabit Ethernet (fibra) y 803.3x control de flujo. Este switch esta equipado para auto negociar velocidades a 10Mbps, 100Mbps y 1000Mbps.
- Control de flujo. El Switch DGS-1216T, en modo Full-Dúplex, permite proteger a los usuarios frente a posibles pérdidas de datos durante la transmisión en la red. Cuando están conectados a una tarjeta LAN (en un servidor o PC) que soporte control de flujo, y cuando el buffer de datos está por llenarse, el switch envía una señal al PC indicando tal situación. Luego, el PC demora la transmisión hasta que el buffer se haya liberado y sea posible el envío de más información.
- Agregación de puertos. Ofrece la posibilidad de poder combinar desde 2 hasta 8 puertos y transformarlos en un ancho de banda de alto rendimiento, a través de una conexión switch-to-switch o una conexión switch-to-servidor.
- VLAN's. Este switch cuenta con soporte de VLANs, Port VLAN y VLAN Tagging, para extender el dominio de Broadcast y el tráfico en segmentos de red, mejorando el rendimiento y facilitando la administración de la red. Además cómo un mecanismo de seguridad, las VLANs pueden restringir el acceso a diferentes segmentos de red, creando subredes independientes unas de otras. Adicionalmente y con la característica de soporte de IEEE 802.1Q VLAN Tagging, es posible la interconexión con switches de otras marcas, posibilitando la creación de VLANs interswitches. Soporta también GVRP ( GARP VLAN protocolo de registro) para automáticas configuraciones VLAN.”<sup>34</sup>

### 9.11 Topología Lógica

El tipo de red es ETHERNET GIGABIT, con una topología lógica tipo Bus (es decir donde el flujo de la información se ubica en un bus lineal) y un método de acceso al medio

---

<sup>34</sup> [www.controlp.com /D-Link/productos.asp](http://www.controlp.com/D-Link/productos.asp)

llamado CSMA/CD (Carrier Sense Multiple Access con Detección de Colisión = Acceso al medio por sensor de portadora y/detección de colisiones).

La topología en estrella extendida es sumamente jerárquica, y busca que la Información se mantenga local.

Cada piso del edificio tendrá asignada una dirección de subred y las estaciones de trabajo que pertenezcan a él usarán una dirección IP, la cual les permitirá comunicarse en el momento de requerir transmitir información a otras subredes.

La comunicación en la red se realizará mediante el protocolo TCP/IP. En este protocolo el TCP maneja todo lo referente a la capa de transporte y el IP le da la identificación en la red. En esta tecnología si una estación de trabajo necesita enviar algún dato, verifica si el medio está libre de tráfico, escuchando la red. Si no está libre, continúa esperando, si está disponible comienza a transmitir enviando un paquete que incluye la dirección IP destino, origen, señales de control y la data misma. Las estaciones CSMA/CD pueden detectar colisiones, de modo que saben en qué momento pueden volver a transmitir, reduciendo así la pérdida de tiempo, energía e información redundante.

Esta red Ethernet Gigabit es tipo broadcast; lo que significa que cada estación puede ver todas las tramas, aunque una estación determinada no sea el destino propuesto para esos datos. Cada estación debe entonces examinar las tramas que recibe para determinar si corresponden al destino. De ser así, la trama pasa a una capa de protocolo superior dentro de la estación para su adecuado procesamiento.

## **9.12 Selección de Cada Uno de los Dispositivos y sus Especificaciones**

### **9.12.1 Dispositivos por Planta**

A continuación se especifica en forma detalla cada una de las secciones con sus posibles equipos a incorporar en la red.

### 9.12.1.1 Equipos para la Sección: Colegio

Como puntos base se han generado cinco puntos de red para el colegio, cabe indicar que actualmente solo poseen dos estaciones de trabajo para ser incorporados a la red. (Ver Plano N° 6).

<b>Tipo de Equipo</b>	<b>Cantidad</b>	<b>Marca</b>
PC de escritorio	5	Acer Inspire
Impresora local	1	Epson lx300+
	1	Laser HP

### 9.12.2 Equipos para la Sección: Escuela

Como puntos base se han generado dos puntos de red para la escuela, cabe indicar que actualmente solo poseen una estación de trabajo para ser incorporada a la red. (Ver Plano N° 7).

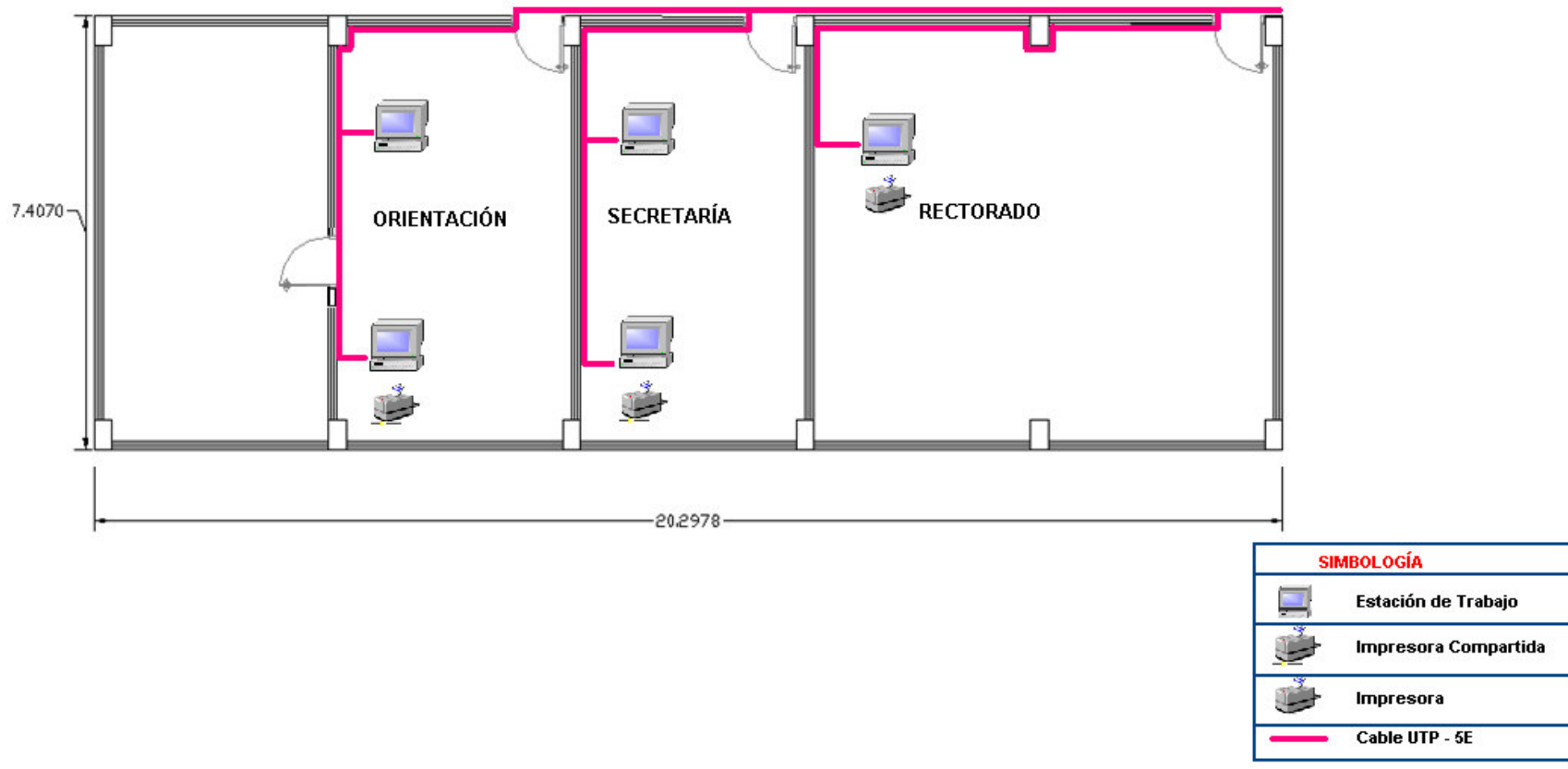
<b>Tipo de Equipo</b>	<b>Cantidad</b>	<b>Marca</b>
PC de escritorio	2	Acer Inspire
Impresora local	2	Epson lx300+

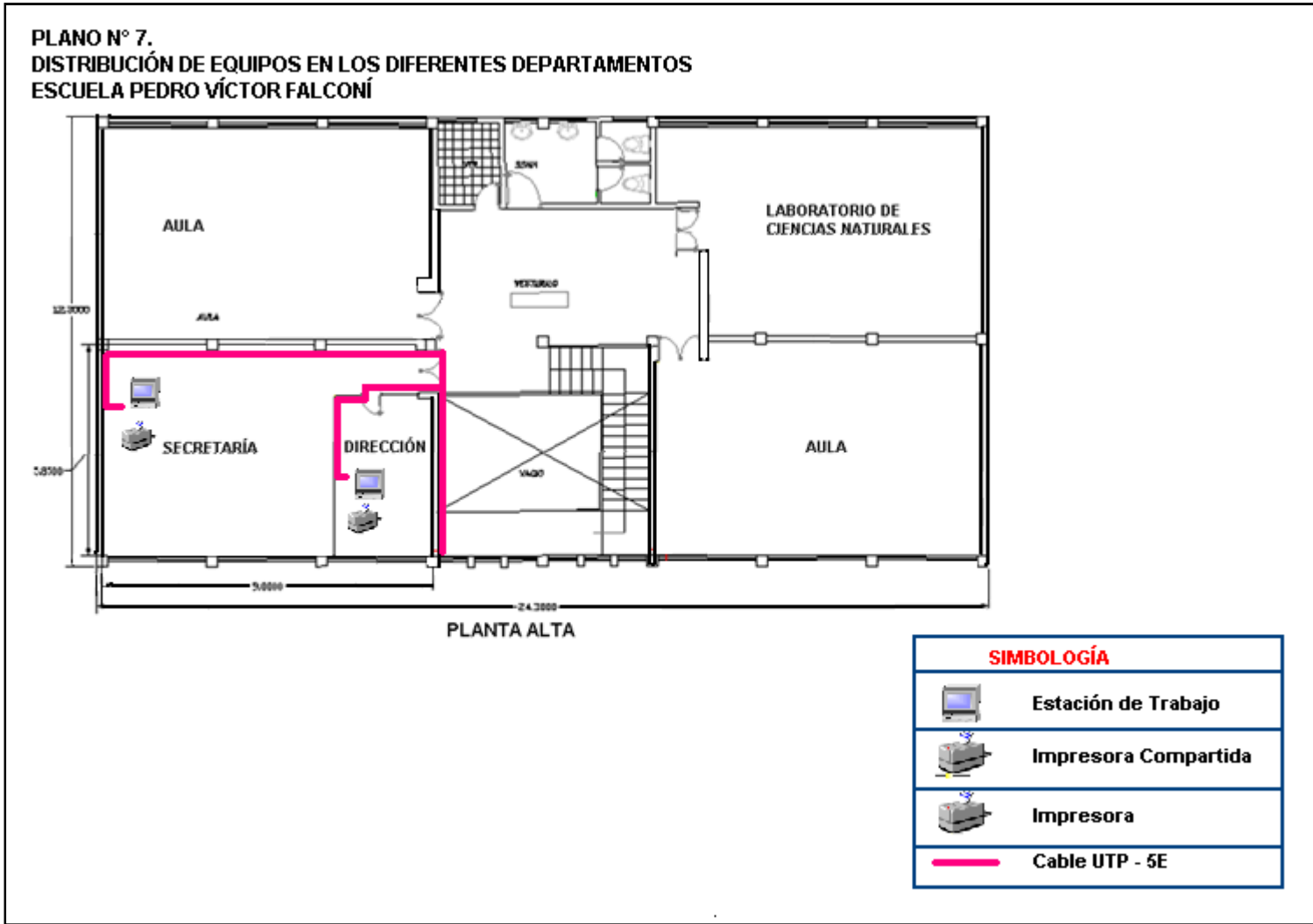
### 9.12.3 Equipos para la Sección: Jardín

Como puntos base se han generado un punto de red para el jardín, cabe indicar que actualmente solo poseen una estación de trabajo para ser incorporada a la red y el espacio físico como la distribución administrativa, no permiten más puntos de red. (Ver Plano N° 8).

<b>Tipo de Equipo</b>	<b>Cantidad</b>	<b>Marca</b>
PC de escritorio	1	Acer Inspire
Impresora local	1	Epson lx300+

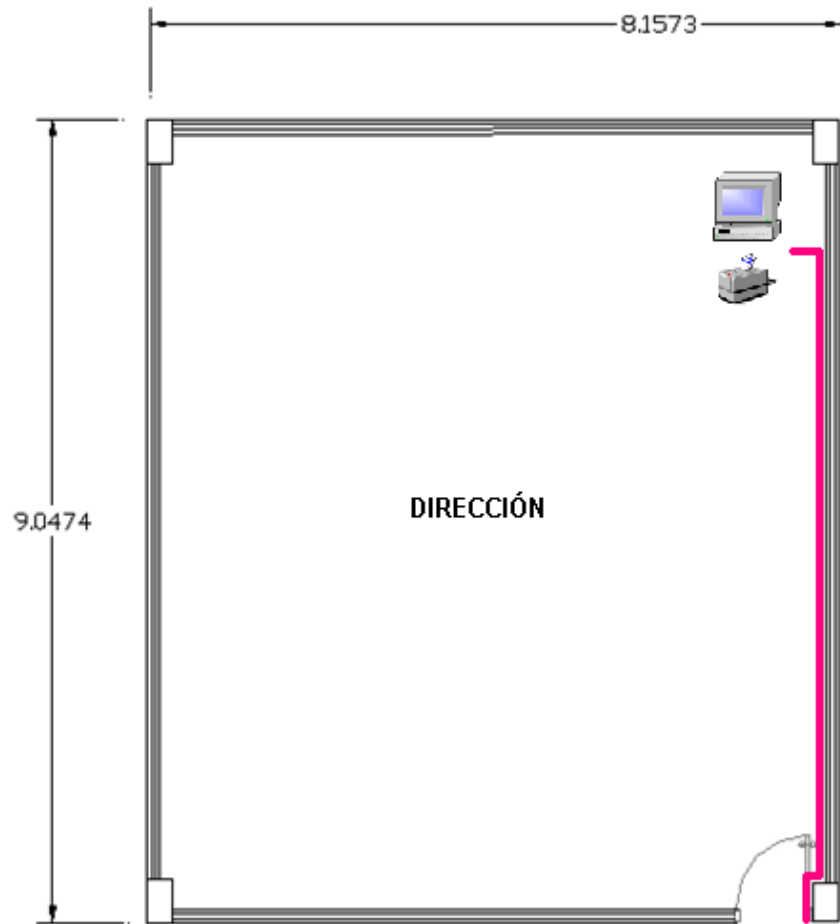
**PLANO N° 6**  
**DISTRIBUCIÓN DE EQUIPOS EN LOS DIFERENTES DEPARTAMENTOS**  
**COLEGIO MANUEL CABRERA LOZANO**









Plano 1. Red Interna del Jardín

PLANO N° 8.  
DISTRIBUCIÓN DE EQUIPOS EN LOS DIFERENTES DEPARTAMENTOS  
JARDÍN JOSÉ ALEJO PALACIOS



SIMBOLOGÍA	
	Estación de Trabajo
	Impresora Compartida
	Impresora
	Cable UTP - 5E

### 9.13 Asignación de Direcciones IP.

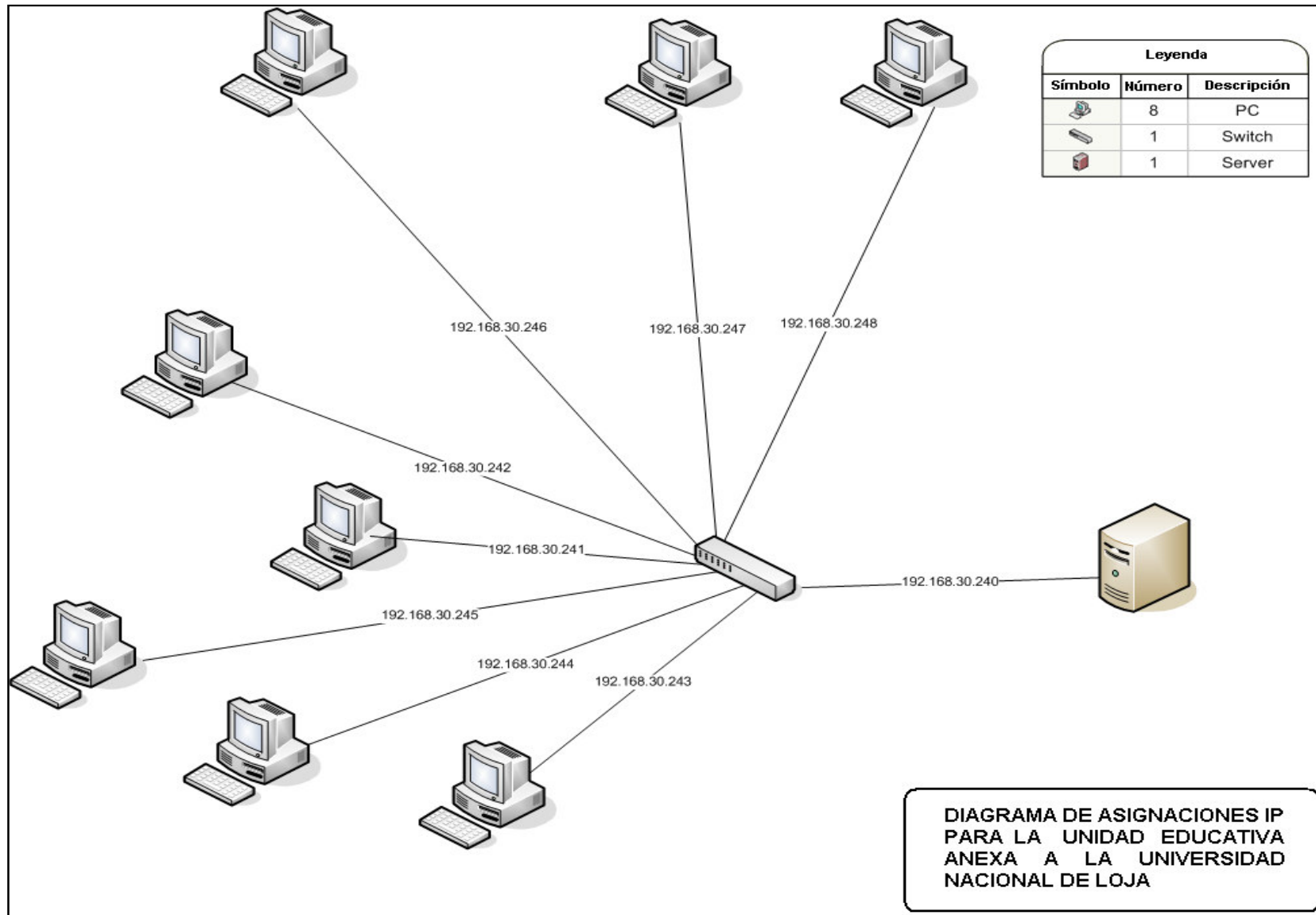
La asignación de direcciones IP dependen de la organización de la red interna de la Universidad Nacional de Loja, para lo cual se solicitó información en la Dirección General de Informática indicándonos que los IP's posibles a utilizar corresponden desde el 192.188.30.240 hasta el 192.188.30.248. Y como máscara de red 255.255.255.0

En el cuadro siguiente se muestra la distribución de IP's en la Unidad educativa Anexa a la Universidad nacional de Loja:

ESTAMENTO	DEPENDENCIA	IP	MÁSCARA
Colegio “Manuel Cabrera Lozano”	Biblioteca (Posible Ubicación Servidor)	192.168.30.240	255.255.255.0
Colegio “Manuel Cabrera Lozano”	Rectorado	192.168.30.241	255.255.255.0
Colegio “Manuel Cabrera Lozano”	Secretaría	192.168.30.242	255.255.255.0
Colegio “Manuel Cabrera Lozano”	Secretaría	192.168.30.243	255.255.255.0
Colegio “Manuel Cabrera Lozano”	Orientación	192.168.30.244	255.255.255.0
Colegio “Manuel Cabrera Lozano”	Orientación	192.168.30.245	255.255.255.0
Escuela “Pedro Víctor Falconí”	Dirección	192.168.30.246	255.255.255.0
Escuela “Pedro Víctor Falconí”	Orientación	192.168.30.247	255.255.255.0
Jardín “José Alejo Palacios”	Dirección	192.168.30.248	255.255.255.0

En el diagrama siguiente se muestra el diagrama de asignaciones IP para la Unidad educativa Anexa a la universidad Nacional de Loja.





## **10 VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL**

El proyecto planteado, pretende contribuir en parte con el desarrollo académico de la Universidad Nacional de Loja, el mismo que se encuentra enmarcado dentro de las líneas de investigación-desarrollo del Área de Energía, Industrias y Recursos Naturales no Renovables de la Universidad Nacional de Loja, en lo que hace referencia a tecnologías de informática y comunicación, planteando el desarrollo de software para usos y aplicaciones industriales, empresariales y académicos.

El libre mercado ha convertido a instituciones educativas de todos los niveles en verdaderas empresas donde se forman los futuros profesionales a través de métodos tradicionales y modernizantes. Debido a ello la educación no debe ser ajena, ni apartarse de la creciente innovación tecnológica en la que han incurrido otras áreas de la sociedad, la ausencia hasta hoy de un sistema único, oficial y confiable de recopilación y tratamiento de la información en distintos niveles de la Unidad Educativa anexa a la Universidad Nacional de Loja, requiere la necesidad de construir, con base sólida, un Sistema que facilite el manejo de la información que allí se genere y permita la automatización de los servicios que brinda.

Técnicamente sea obtenido un software de alta calidad que satisfice los requerimientos y necesidades de la Unidad educativa anexa a la Universidad nacional de Loja, ya que se utilizó metodología y herramientas y paquetes reutilizables adicionales que son de alta calidad para el desarrollo del software acorde con los objetivos planteados en el presente proyecto.

Económicamente se minimizó todo costo de producción al desarrollar el proyecto en mención con software y paquetes adicionales de licencia de libre distribución, cabe aclarar que para esto se debe adoptar la licencia GNU/GPL que indica que el código que se escribió en nuestro software de igual forma es de libre distribución. Con relación a la red se hizo el respectivo estudio de componentes y calidades que minimizarían al máximo el costo de implementación en el caso de realizarse dicha implementación.

Ambientalmente el impacto es mínimo ya que no se ocupa ningún área que no sea de actual uso de la Unidad Educativa para sus labores diarias y para la propuesta de la red se utilizaría

ductos ya creados para el cableado del teléfono, de forma que el medio ambiente no se ve afectado por la implementación del presente proyecto.

## 11 CONCLUSIONES

- El sistema de Gestión académica para la Unidad educativa Anexa a la Universidad Nacional de Loja es una herramienta informática que permite realizar todos los procesos académicos que se generan en un determinado año lectivo.
- El sistema de Gestión académica para la Unidad educativa Anexa a la Universidad Nacional de Loja tiene interfaces fáciles y amigables para el usuario lo que permite una fácil interacción entre el usuario y la aplicación.
- La utilización del software en los tres estamentos permitiría un adecuado flujo de información entre los mismos y reducción de la duplicidad de información almacenada.
- Los procesos de Gestión Académica en la Unidad Educativa Anexa a la Universidad nacional de Loja, se encuentran automatizados después de la implantación del software de Gestión Académico desarrollado en la presente Tesis.
- La información en la Unidad educativa Anexa a la Universidad Nacional de Loja se encuentra organizada por creación y cierres de años lectivos lo que permite el correcto manejo de los mismos para cada uno de los alumnos.
- Debido a que los procesos de Gestión Académica se encuentran automatizados se ha contribuido con el desarrollo Tecnológico en la Unidad Educativa Anexa a la Universidad Nacional de Loja.
- El proceso de obtención de la información (ingreso de datos, generación de datos, reportes para docentes, padres de familia) se da en forma eficiente y eficaz según sean los requerimientos de la Unidad Educativa Anexa a la Universidad Nacional de Loja tanto de los Docentes, los Alumnos como también de los Padres de Familia.
- El sistema de Gestión académica para la Unidad educativa Anexa a la Universidad Nacional de Loja permite la generación de los diferentes tipos de reportes que se necesiten en cada uno de los módulos como dentro del módulo específico para la generación de dichos reportes.

- La información que posee la Unidad Educativa es más segura ya que el ingreso al sistema se da por niveles de seguridad de ingreso y/u obtención de datos determinado por el Administrador de dicho sistema.
- La colaboración brindada por cada uno de los estamentos que conforman la Unidad educativa Anexa a la Universidad Nacional de Loja permitió la obtención de un software de calidad que cumple con los requerimientos y necesidades de la misma.
- El uso de la metodología ICONIX a través de todas sus fases nos permitió obtener resultados eficientes y satisfactorios el desarrollo del software Sistema de Gestión Académica.
- La utilización de netbeans como herramienta de trabajo para el desarrollo del software como el uso de librerías de libre distribución nos permitió obtener interfaces amigables, atractivas e interesantes para el uso de dicho software los usuarios.
- El uso de herramientas como Hibernate nos permitió reducción del tiempo y fiabilidad en la generación y obtención de datos en la elaboración del software ya que hibernate es una capa intermedia entre la capa del negocio y la base de datos ocupándose por completo de la administración de dicha base de datos.
- El Sistema de Gestión Académico para la Unidad Educativa Anexa a la Universidad Nacional de Loja según el plan de pruebas y validación por parte de usuario fue aceptado en su totalidad por cada uno de dichos usuarios.
- Enriquecimiento Intelectual y Tecnológico por parte de los desarrolladores de la presente Tesis.

## 12 RECOMENDACIONES

- Para un mejor rendimiento y uso del software se debe tener total conocimiento del manual de usuario por parte de todos los funcionarios de la Unidad Educativa Anexa a la Universidad Nacional de Loja que vayan a utilizar el software.
- El adecuado funcionamiento del sistema depende de la configuración creada para año lectivo en cada una de las entidades que pertenecen a la Unidad Educativa Anexa a la Universidad Nacional por lo que debe tener mucho cuidado al momento que se ingresen los valores de configuración para el año lectivo vigente.
- Preocuparse por el mantenimiento del software en el caso que deba realizar actualizaciones según los cambios en la Gestión Académica que se pudieran dar en la Unidad Educativa Anexa a la Universidad Nacional de Loja.
- Determinar las funciones que se asignará a un determinado usuario del software para garantizar la seguridad y validez de los datos en el sistema de Gestión Académica de la Unidad Educativa Anexa a la Universidad Nacional de Loja.
- Generar cuentas para cada uno de los usuarios que vayan a manejar el Sistema de Gestión Académico con sus respectivos niveles de usuario determinado por el Director de cada uno de los estamentos que sería el único autorizado a permitir la creación de nuevas cuentas como roles de usuarios.
- Designar por parte de la Unidad Educativa Anexa a la Universidad Nacional de Loja un administrador para el software Sistema de Gestión Académica para que realice la administración de usuarios y cuentas según se necesite en la entidad.
- La seguridad del sistema como de los datos almacenados se ve garantizada al momento que el administrador como usuarios mantengan sus cuentas en total confidencialidad.
- Implementar la red que en el presente proyecto queda como esquema tentativo, pero como base para dicha ejecución.
- Dar al Sistema de Gestión Académico el correspondiente periodo de mantenimiento por posibles cambios o mejoras que se puedan necesitar, según cómo evolucione la

Unidad Educativa Anexa a la Universidad Nacional de Loja para que no termine su periodo de vida útil como un software obsoleto e inservible.

- Así mismo brindar el correspondiente mantenimiento a los equipos que tienen instalado el software como al servidor de la aplicación que este caso es el mismo, para obtener un máximo rendimiento en la aplicación, así como mantener un antivirus actualizado para garantizar la seguridad de los datos.

## 13 BIBLIOGRAFÍA

- ◆ ALLEN Downey, How to Think Like a Computer Scientists, java version, third edition, 2000.
- ◆ BRUCE Eckel, Thinking in Java, editorial Prentice – Hall, second edition, mid- june 2000.
- ◆ DEITEL & Deitel, Como Programar en Java, segunda edición, 2000.
- ◆ GIBBONs Paul, Desarrollo.net para Programadores en Java, Ediciones Anaya Multimedia, 2002.
- ◆ GARCIA Thomás, Redes para Proceso Distribuido, editorial Alfaomega, segunda edición.
- ◆ ROSENBERG, Doug, Applying Use Case Driven Object Modeling with UML, Copyright Addison-Wesley, 2001.
- ◆ GILBERT, David, The JFreeReport Class Library Versión 0.8.3 Reference Guide, Object Refinery Limited, 2003.
- ◆ BEGNIS, Camille, Guía de Referencia Mandrake Linux 9.1, Copyright Mandrakesoft S.A., 2003.
- ◆ [www.cybercursos.net](http://www.cybercursos.net)
- ◆ [www.javahispano.com](http://www.javahispano.com)
- ◆ [www.molwick.com/tgecv.es](http://www.molwick.com/tgecv.es)
- ◆ [www.monografías.com](http://www.monografías.com)
- ◆ [www.monografiass.com](http://www.monografiass.com)
- ◆ [www.um.es](http://www.um.es)
- ◆ [www.unlar.estudio3.com](http://www.unlar.estudio3.com)
- ◆ [www.uoc.edu/ideasolutions.com](http://www.uoc.edu/ideasolutions.com)
- ◆ [www.iconixsw.org](http://www.iconixsw.org)
- ◆ [www.sun.com](http://www.sun.com)
- ◆ [www.mysql.com](http://www.mysql.com)
- ◆ [www.jgoodiess.org](http://www.jgoodiess.org)
- ◆ [www.hibernate.org](http://www.hibernate.org)
- ◆ [www.jfree.org](http://www.jfree.org)
- ◆ [www.swinglabs.org](http://www.swinglabs.org)
- ◆ [www.javahelp](http://www.javahelp)
- ◆ [www.apache.org](http://www.apache.org)
- ◆ [www.mandrakesoft.com](http://www.mandrakesoft.com)
- ◆ [www.pragmaconsultores.com](http://www.pragmaconsultores.com)



**15 ANEXOS.**

**15.1 ANEXO 1. ANTEPROYECTO.**

**15.2 ANEXO 2. MANUAL DE USUARIO.**

### **15.3 ANEXO 3. ENTREVISTAS.**

#### **15.4 ANEXO 4. REGISTRO DE CASOS Y BITÁCORA DE PRUEBAS.**

**15.5 ANEXO 5. ENCUESTAS.**

**15.6 ANEXO 6. CERTIFICACIONES.**