

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERIA EN SISTEMAS**

**Desarrollo del Sistema de Administración Estudiantil para el  
Colegio Segré usando el Desarrollo Dirigido por Modelos**

**PROYECTO PREVIO A LA OBTENCIÓN EL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**AUTOR:**

**Calero Romero Gerardo Alberto**

***pantereprn@hotmail.com***

**DIRECTOR:**

**Msc. Ing. Raúl Córdova**

***raul.cordova@epn.edu.ec***

**Quito, Enero 2009**

## **DECLARACIÓN**

Yo, Gerardo Alberto Calero Romero, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mi derecho de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Gerardo Alberto Calero Romero

## **CERTIFICACION**

Certifico que el presente trabajo fue desarrollado por Gerardo Alberto Calero Romero, bajo mi supervisión.

---

**Msc. Ing. Raúl Córdova**  
**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTO**

Agradezco a Dios, porque me ha dado las fuerzas y la sabiduría para afrontar los obstáculos y vivir plenamente los triunfos. A mi familia, que han confiado en mí siempre, y me han apoyado en todo momento de mi vida.

Gerardo





## **DEDICATORIA**

Dedico esta tesis a mis padres y hermanos, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron apoyándome en los momentos más críticos de mi vida. Por ustedes, por lo que representan, por su apoyo incondicional y por todo el esfuerzo que han hecho por mí.

Gerardo

# Índice De Contenido

<b>INTRODUCCIÓN</b> .....	¡Error! Marcador no definido.
<b>DEFINICIÓN DEL PROBLEMA</b> .....	<b>1</b>
<b>1.1. INVESTIGACIÓN BIBLIOGRÁFICA</b> .....	<b>1</b>
<b>1.2. Definición del proyecto</b> .....	<b>3</b>
1.2.1. Descripción de la organización.....	3
1.2.1.1. Misión.....	3
1.2.1.2. Principios Fundamentales .....	3
1.2.1.3. Organigrama de la institución.....	4
1.2.2. Planteamiento del Problema.....	4
1.2.3. Alcance.....	5
1.2.4. Exclusiones .....	6
1.2.5. Definiciones .....	6
<b>1.3. Desarrollo Dirigido Por Modelos</b> .....	<b>8</b>
1.3.1. Definición de MDD (Model Driven – Development ‘Desarrollo Dirigido Por Modelos’).....	8
1.3.2. Aspectos en los que MDD mejora el proceso.....	8
1.3.3. Paradigma de MDD .....	10
1.3.4. Modelos .....	10
1.3.4.1. PIM (Platform Independent Model ‘Modelo Independiente De La Plataforma’).....	10
1.3.4.2. PSM (Platform Specific Model ‘Modelo Específico De La Plataforma’).....	11
1.3.4.3. Implementación específica de la plataforma o Código .....	11
1.3.5. Elementos para MDD .....	11
1.3.5.1. Técnicas Para MDD .....	12
1.3.5.1.1. MDA .....	13
a) UML .....	15
b) MOF .....	16
c) OCL .....	17
d) XMI .....	17
e) QVT .....	18
1.3.5.1.2. Factorías de Software .....	19
1.3.5.1.3. DSM .....	19
1.3.6. Aplicación De MDD Para Los Sistemas De Software.....	20
1.3.6.1. Metodología Orientada a Objetos Para MDD .....	20

1.3.6.2.	Criterios de Evaluación para Herramientas MDD.....	25
1.3.6.2.1.	Herramientas a ser evaluadas.....	26
a)	MOFScript.....	26
b)	MTF (Model Transformation Framework) .....	27
c)	OptimalJ .....	27
d)	ArcStyler .....	28
1.3.6.2.2.	Evaluación de las herramientas descritas .....	28
1.3.6.2.3.	Conclusión de las herramientas evaluadas .....	29
1.3.7.	Fundamentos de la herramienta seleccionada .....	30
1.3.7.1.	Creación de un Proyecto en OptimalJ.....	30
1.3.7.2.	Elementos para el Modelo del Dominio.....	33
1.3.7.2.1.	 Tipo Enumeración .....	33
1.3.7.2.2.	 Tipo Estructura .....	36
1.3.7.2.3.	 Clases.....	38
1.3.7.2.4.	 Relaciones .....	40
1.3.7.3.	Creación de los Modelos Específicos.....	42
1.3.7.3.1.	Modelo de Presentación .....	43
a)	Elementos capa presentación.....	44
b)	Aumentar campos en la capa de presentación.....	44
c)	Validación de campos .....	45
d)	Cambios en Textos de Presentación .....	47
1.3.7.3.2.	Modelo del Negocio.....	48
1.3.7.3.3.	Modelo de Datos .....	49
1.3.7.4.	Generación de Código .....	49
1.3.7.5.	Creación de Base de Datos .....	50
1.3.7.6.	Inicialización de la aplicación .....	52
<b>Análisis y Diseño del Sistema .....</b>		<b>53</b>
<b>2.1.</b>	<b>Recolección de Datos.....</b>	<b>53</b>
2.1.1.	Introducción.....	56
2.1.1.1.	Ámbito .....	56
2.1.1.2.	Definición del sistema .....	56
2.1.1.3.	Objetivos Generales .....	57
2.1.1.4.	Entorno De Operación.....	57
2.1.1.5.	Convenciones.....	58
2.1.1.6.	Material De Referencia.....	58
2.1.2.	Requerimiento Funcionales Del Sistema de Administración Estudiantil 59	
2.1.2.1.	Funcionalidad Del Sistema # 1: Administración de Información del Estudiante.....	59
2.1.2.2.	Funcionalidad Del Sistema # 2: <i>Matriculación</i> .....	60
2.1.2.3.	Funcionalidad Del Sistema # 3: Administración de Información de Profesor.....	60

2.1.2.4.	Funcionalidad Del Sistema # 4: Administración de Información de Cursos	61
2.1.2.5.	Funcionalidad Del Sistema # 5: Administración de Información de Materias	62
2.1.2.6.	Funcionalidad Del Sistema # 6: Registro de Evaluaciones por Materia	63
2.1.2.7.	Funcionalidad Del Sistema # 7: Registro de Evaluaciones por Estudiante	63
2.1.2.8.	Funcionalidad Del Sistema # 8: Listado de Estudiantes por Curso	64
2.1.2.9.	Funcionalidad Del Sistema # 9: Listado de Notas de los Estudiantes por Materia	64
2.1.2.10.	Funcionalidad Del Sistema # 10: Listado de Notas de Materias por Estudiante	65
2.1.2.11.	Funcionalidad Del Sistema # 11: Listado de Profesores por Materia	65
2.1.3.	Requisitos De Interfaz	66
2.1.3.1.	Interfaces De Usuario	66
2.1.3.1.1.	Interface Inicial	66
2.1.3.1.2.	Acceso a las Funcionalidades	70
2.1.3.1.3.	Despliegue de Errores	70
2.1.3.2.	Interfaces De Software	70
<b>2.2.</b>	<b>Análisis</b>	<b>72</b>
2.2.1.	Casos de Uso	72
2.2.1.1.	Diagrama de Casos de Uso	72
2.2.1.2.	Detalle de Casos de Uso	76
2.2.1.2.1.	Paquete Gestión Estudiante	76
2.2.1.2.2.	Paquete Gestión Profesor	80
2.2.1.2.3.	Paquete Gestión Curso	84
2.2.1.2.4.	Paquete Gestión Materia	88
2.2.1.2.5.	Paquete Matricular Estudiante	91
2.2.1.2.6.	Paquete Evaluación	92
2.2.1.2.7.	Paquete Listar Información Estudiantil	94
2.2.2.	Clases	98
2.2.2.1.	Diagrama de Clases	98
2.2.2.2.	Detalle de Objetos	101
2.2.2.2.1.	Tipos de Datos	101
a)	Enumeración	101
b)	Estructura	102
2.2.2.2.2.	Clases	104
2.2.2.2.3.	Relaciones	106



<b>2.3.</b>	<b>Diseño Detallado .....</b>	<b>110</b>
2.3.1.	PIM aplicado a la herramienta .....	110
2.3.2.	PSM's generados .....	111
2.3.2.1.	Campos adicionales .....	115
2.3.2.1.1.	Evaluación Por Estudiante .....	115
2.3.2.1.2.	Evaluación Por Materia .....	115
2.3.2.2.	Validación de campos .....	116
<b>Construcción Y Pruebas del Sistema .....</b>		<b>118</b>
<b>3.1.</b>	<b>Codificación .....</b>	<b>118</b>
3.1.1.	Descripción de cambios en el código .....	118
3.1.1.1.	Función init() de EstudianteForm. java .....	118
3.1.1.2.	Función calcularNotaFinal() de EvaluationBean.java .....	118
3.1.2.	Creación de la Base de Datos .....	119
<b>3.2.</b>	<b>Pruebas .....</b>	<b>119</b>
<b>Conclusiones y recomendaciones .....</b>		<b>145</b>
<b>4.1.</b>	<b>Conclusiones .....</b>	<b>145</b>
<b>4.2.</b>	<b>Recomendaciones .....</b>	<b>146</b>
<b>Bibliografía .....</b>		<b>147</b>
<b>Anexos .....</b>		<b>148</b>

## Índice de gráficos

Figura 1.1 Organigrama de la Institución. ....	4
Figura 1.2: Proceso para el desarrollo de un DSL hasta el lenguaje de máquina. .....	13
Figura 1.3: Procesos de la transformación de modelos a modelos y modelos a código .....	14
Figura 1.4: Niveles de UML, tutorial de UML .....	15
Figura 1.5: Redefinición de subclase .....	17
Figura 1.6: Proceso de transformación con QVT .....	18
Figura 1.7: Ciclo de vida para MDD .....	25
Figura 1.8 : Modelo del Dominio. ....	31
Figura 1.9 : Modelo del Dominio. ....	31
Figura 1.10: Especifica el nombre del proyecto y localización. ....	31
Figura 1.11: Especifica el nombre del paquete e ID del proyecto. ....	32
Figura 1.12: Estructura del Modelo de Dominio. ....	32
Figura 1.13: Ambiente del paquete <i>class</i> . ....	33
Figura 1.14: Tipo Enumeración. ....	33
Figura 1.15: Especificación del nombre de la Enumeración. ....	34
Figura 1.16: Selección del módulo. ....	34
Figura 1.17: Selección del tipo de dato. ....	35
Figura 1.18: Ingreso de valores en la Enumeración. ....	35
Figura 1.19: Tipo Estructura. ....	36
Figura 1.20: Especificación del nombre de la Estructura. ....	36
Figura 1.21: Selección del módulo. ....	37
Figura 1.22: Ingreso de atributos en la Estructura. ....	37
Figura 1.23: Clases. ....	38
Figura 1.24: Especificación del nombre de la Clase. ....	38
Figura 1.25: Selección del módulo. ....	39
Figura 1.26: Ingreso de atributos en la Clase. ....	39
Figura 1.27: Especificación de valores únicos. ....	40
Figura 1.28: Relaciones. ....	40

Figura 1.29: Especificación de <i>Agregaciones y Multiplicidad</i> .....	41
Figura 1.30: Especificación del nombre de la Relación. ....	41
Figura 1.31: Selección del módulo.....	42
Figura 1.32: Paquetes del Modelo de Aplicación.....	43
Figura 1.33: Actualización de todos los modelos.....	43
Figura 1.34: Elementos del paquete del Modelo de Presentación.....	44
Figura 1.35 Elementos de la página por defecto .....	44
Figura 1.36 Página con elementos añadidos.....	45
Figura 1.37 Enlaces de atributos en la capa de presentación .....	45
Figura 1.38 Controlador de atributo .....	46
Figura 1.39 Especificación del nombre del controlador .....	46
Figura 1.40 Paso para especificar una expresión regular.....	47
Figura 1.41: Modelo de Presentación. ....	48
Figura 1.42: Modelo de Facade. ....	48
Figura 1.43: Modelo de Datos.....	49
Figura 1.44: Generación de Código, Script y Archivos. ....	50
Figura 1.45: Archivo de autenticación a la Base de Datos.....	51
Figura 1.46: Valores de autenticación.....	51
Figura 1.47: Carga del script para la Base de Datos. ....	52
Figura 1.48: Log de la ejecución del script.....	52
Figura 1.49: Ejecución de la aplicación.....	52
Figura 2.1: Diagrama de Clases Colegio Segré.....	100
Figura 2.2 PIM para SAE Segré.....	110
Figura 2.3 PSM de Presentación de SAE Segré .....	112
Figura 2.4 PSM de Negocio de SAE Segré .....	113
Figura 2.5 PSM de Datos de SAE Segré .....	114
Figura 3.1: Caso de Prueba 1 .....	120
Figura 3.2: Caso de Prueba 2.....	122
Figura 3.3: Caso de Prueba 3.....	125
Figura 3.4: Caso de Prueba 4.....	126
Figura 3.5: Caso de Prueba 5.....	128
Figura 3.6: Caso de Prueba 6.....	129
Figura 3.7: Caso de Prueba 7.....	131

Figura 3.8: Caso de Prueba 8.....	133
Figura 3.9: Caso de Prueba 9.....	135
Figura 3.10: Caso de Prueba 10.....	136
Figura 3.11: Caso de Prueba 11.....	138
Figura 3.12: Caso de Prueba 12.....	140
Figura 3.13: Caso de Prueba 13.....	142
Figura 3.144: Caso de Prueba 14.....	143
Figura 3.15: Caso de Prueba 15.....	144

## Índice de Tablas

Tabla 1.1: Criterios de comparación a herramientas MDD .....	26
Tabla 1.2: Valores para la evaluación de criterios .....	26
Tabla 1.3: Valores para la evaluación de criterios .....	29
Tabla 2.1: Descripción del entorno de Operación .....	57
Tabla 2.2: Requerimientos de <i>Administración de Información Estudiantil</i> . .....	59
Tabla 2.3: Requerimientos de <i>Matriculación</i> . .....	60
Tabla 2.4: Requerimientos de <i>Administración de Información de Profesor</i> . .....	61
Tabla 2.5: Requerimientos de <i>Administración de Información de Cursos</i> . .....	62
Tabla 2.6: Requerimientos de <i>Administración de Información de Materias</i> . .....	62
Tabla 2.7: Requerimientos de <i>Registro de Evaluaciones</i> . .....	63
Tabla 2.8: Requerimientos de Registro de Evaluaciones. ....	63
Tabla 2.9: Requerimientos de <i>Listado de Estudiantes por Curso</i> . .....	64
Tabla 2.10: Requerimientos de <i>Listado de Notas de los Estudiantes por Materia</i> . .....	64
Tabla 2.11: Requerimientos de <i>Listado de Notas de Materias por Estudiante</i> . ..	65
Tabla 2.12: Requerimientos de <i>Listado de Profesores por Materia</i> . .....	65
Tabla 2.2 Especificación de SW necesario.....	71
Tabla 2.13: Caso de Uso <i>Insertar Estudiante</i> . .....	76
Tabla 2.14: Caso de Uso <i>Buscar Estudiante</i> . .....	77
Tabla 2.15: Caso de Uso <i>Modificar Estudiante</i> . .....	78
Tabla 2.16: Caso de Uso <i>Modificar Estudiante</i> . .....	79
Tabla 2.17: Caso de Uso <i>Insertar Profesor</i> . .....	80
Tabla 2.18: Caso de Uso <i>Buscar Profesor</i> . .....	81
Tabla 2.19: Caso de Uso <i>Modificar Profesor</i> . .....	82
Tabla 2.20: Caso de Uso <i>Eliminar Profesor</i> . .....	83
Tabla 2.21: Caso de Uso <i>Insertar Curso</i> . .....	84
Tabla 2.22: Caso de Uso <i>Buscar Curso</i> . .....	85
Tabla 2.23: Caso de Uso <i>Modificar Curso</i> . .....	86
Tabla 2.24: Caso de Uso <i>Eliminar Curso</i> . .....	87
Tabla 2.25: Caso de Uso <i>Insertar Materia</i> . .....	88
Tabla 2.26: Caso de Uso <i>Buscar Materia</i> . .....	89

Tabla 2.27: Caso de Uso <i>Modificar Materia</i> .....	90
Tabla 2.28: Caso de Uso <i>Matricular Estudiante</i> .....	91
Tabla 2.29: Caso de Uso <i>Registrar Evaluación Por Materia</i> .....	92
Tabla 2.30: Caso de Uso <i>Registrar Evaluación por Estudiante</i> .....	93
Tabla 2.31: Caso de Uso <i>Listar Estudiantes por Curso</i> .....	94
Tabla 2.32: Caso de Uso <i>Listar Evaluaciones de Estudiantes por Materia</i> .....	95
Tabla 2.33: Caso de Uso <i>Listar Evaluaciones de Materias por Estudiante</i> .....	96
Tabla 2.34: Caso de Uso <i>Listar Profesores por Materia</i> .....	97
Tabla 2.35: Descripción <i>Enumeración Especialidad</i> .....	101
Tabla 2.36: Descripción <i>Enumeración Especialidad</i> .....	102
Tabla 2.37: Descripción <i>Estructura Representante</i> .....	103
Tabla 2.38: Descripción <i>Estructura Dirección</i> .....	103
Tabla 2.39: Descripción <i>Clase Estudiante</i> .....	104
Tabla 2.40: Descripción <i>Clase Curso</i> .....	105
Tabla 2.41: Descripción <i>Clase Materia</i> .....	105
Tabla 2.42: Descripción <i>Clase Profesor</i> .....	106
Tabla 2.43: Descripción <i>Relación se_matricula_en</i> .....	107
Tabla 2.44: Descripción <i>Relación es_evaluado</i> .....	108
Tabla 2.45: Descripción <i>Relación dicta</i> .....	109
Tabla 2.46: Descripción <i>Relación se_dicta</i> .....	109
Tabla 2.47: Descripción para la validación de campos.....	116
Tabla 2.48: Expresiones Regulares para validar campos.....	117
Tabla 3.1: Caso de Prueba Administrar Curso – caso válido.....	120
Tabla 3.2: Caso de Prueba Administrar Estudiante – caso válido.....	122
Tabla 3.3: Caso de Prueba Administrar Estudiante – caso erróneo.....	124
Tabla 3.4: Caso de Prueba Administrar Materia – caso válido.....	126
Tabla 3.5: Caso de Prueba Administrar Materia – caso erróneo.....	127
Tabla 3.6: Caso de Prueba Administrar Matrícula – caso válido.....	129
Tabla 3.7: Caso de Prueba Administrar Matrícula – caso erróneo.....	130
Tabla 3.8: Caso de Prueba Administrar Profesor – caso válido.....	132
Tabla 3.9: Caso de Prueba Administrar Profesor – caso erróneo.....	134
Tabla 3.10: Caso de Prueba Administrar AsignarMaterias – caso válido.....	136
Tabla 3.11: Caso de Prueba Administrar AsignarMaterias – caso erróneo.....	137

Tabla 3.12: Caso de Prueba Administrar RegistroEvaluacionPorEstudiante – caso válido.....	139
Tabla 3.13: Caso de Prueba Administrar RegistroEvaluacionPorEstudiante – caso erróneo.....	141
Tabla 3.14: Caso de Prueba Administrar RegistroEvaluacionPorMateria – caso válido .....	143
Tabla 3.18: Caso de Prueba ListarProfesoresPorMateria – caso válido.....	144

## **RESUMEN**

El desarrollo del Sistema de Administración Estudiantil para el colegio Segré, se basa en el Desarrollo Dirigido por Modelos (MDD). La importancia del paradigma en SAE Segré es la separación de parte funcional y la implementación, para lo cual se utilizará una herramienta que permita la aplicación de los conceptos MDD.

La base fundamental es el diagrama de clases, que para MDD es identificado como PIM (Modelo Independiente de la Plataforma), este modelo será transformado por otros modelos que especifican la presentación, negocio y datos y son identificados como PSM (Modelos Específicos de la Plataforma), finalmente estos modelos son transformados en código.

Los modelos antes de la transformación y el código generado pueden ser modificados para que el sistema cumpla los requerimientos solicitados, esto permite a los desarrolladores concentrarse en la ingeniería y simplificar la etapa de codificación.

Para el desarrollo del sistema, se utiliza la metodología Orientada a Objetos adaptada a MDD, ya que la base fundamental del paradigma y de la metodología son los modelos.

A continuación se describe el contenido de los capítulos de este documento:

Capítulo 1: se describe la organización del Colegio Segré, planteamiento del problema, alcance y exclusiones para el desarrollo del Sistema De Administración. Además comprende el marco teórico de MDD, donde especifica el proceso de los modelos, metodología, herramientas, criterios de evaluación para selección de herramientas MDD y los fundamentos principales de la herramienta seleccionada para el desarrollo del sistema.



Capítulo 2: contiene la Recolección de Datos, donde consta la Introducción, Requerimientos Funcionales del Sistema y Requisitos de Interfaz. Además se especifica el Análisis, donde se detalla los Casos de Uso y el diagrama de Clases, que para MDD es el PIM. En el Diseño Detallado se aplica el PIM en la herramienta MDD y se obtienen los PSM's.

Capítulo 3: se especifica el uso de los PSM's para la obtención del código. También se detallan los cambios realizados en el código y la creación de la Base de Datos para el Sistema de Administración Estudiantil.

Capítulo 4: se detallan las conclusiones obtenidas en el desarrollo del sistema y las recomendaciones dadas para futuros trabajos con MDD.

## **PRESENTACIÓN**

El desarrollo de este Sistema surge de la necesidad del personal administrativo, ya que no contaban con una aplicación que les permita automatizar los procesos de registro de información, control de notas y matriculación.

El Sistema de Administración Estudiantil para el colegio Segré, se basa en el Desarrollo Dirigido por Modelos (MDD). El paradigma aplicado en este Sistema, permite cumplir con todos los requerimientos solicitados, aumento de productividad, lo que implica una reducción de costo y tiempo en el desarrollo y el mantenimiento.

El Sistema depende de los modelos, en los cuales se identifican las Capas de Presentación, Lógica, Negocio y Datos, estos modelos son conocidos como PIM, PSM's y Código.

El sistema permite el registro de información de los estudiantes, ejecución del proceso de matriculación y control de notas, permitiendo al personal administrativo y docente registrar los datos estudiantiles.

## **CAPÍTULO 1:**

### **DEFINICIÓN DEL PROBLEMA**

#### **1.1. INVESTIGACIÓN BIBLIOGRÁFICA**

El objetivo principal de este apartado es tener una idea general de las referencias más importantes y que contienen los conceptos más claros que ayudarían al desarrollo de esta tesis. Esto no quiere decir que las referencias detalladas serán las únicas tomadas en cuenta.

##### **Detalle Bibliográfico Más Importante**

- **Object Management Group (OMG). Desarrollo Dirigido por Modelos. Guía año 2003**

Especifica una propuesta que proporciona mecanismos para mejorar la calidad del software, facilitando la portabilidad, interoperabilidad y re-usabilidad, donde promueven el uso intensivo de modelos en el proceso de desarrollo de software. OMG especifica que la participación de los desarrolladores es construir los modelos de los sistemas software utilizando primitivas de alto nivel de abstracción. Estos modelos son transformados hasta obtener el código fuente del sistema software final. Además se especifica que la base fundamental de un Desarrollo Dirigido Por Modelos es el uso de UML (Unified Modeling Language) desde la versión del año 1995.

- **Colin Atkinson and Thomas Kühne. Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, September/October 2003.**

Especifican sobre el concepto de Desarrollo Dirigido por Modelos, proponiendo una serie de elementos que permitirán la construcción de los sistemas. En resumen, indica que deben especificarse

- Meta-modelos de los modelos,
  - Sintaxis de los modelos y
  - Transformaciones de los modelos a código fuente.
- **Bran Selic. The Pragmatics of Model-Driven Development. *IEEE Software*, September/October 2003.**

Pone de relevancia el nivel de abstracción que supone el desarrollo de software basado en modelos conceptuales respecto a los métodos tradicionales basados en lenguajes de programación. Los desarrolladores se centran en describir el dominio del sistema que desean automatizar en lugar de lidiar con cuestiones particulares de los lenguajes de implementación. El autor especifica que los métodos y herramientas que soporten el Desarrollo Dirigido Por Modelos permitan las transformaciones automáticas a partir de los modelos.

- **Acta Desarrollo de Software Dirigido por Modelos<sup>1</sup>**

Este documento permite comprender los fundamentos de MDD, analizar e identificar los problemas y soluciones existentes, además define líneas de trabajo e investigación futuras y comparte experiencias a nivel académico e industrial en la aplicación de técnicas y herramientas basadas en MDD, todo ello en el ámbito de la ingeniería del software.

- **Manual OptimalJ**

El documento explica sobre el manejo de la herramienta, con ejemplos claros y sencillos. Además muestra links que permitirá visualizar ejemplos que pueden ser visualizados de forma real.

---

<sup>1</sup> [www.dsic.upv.es/workshops/dsdm04/files/Actas-DSDM04.pdf](http://www.dsic.upv.es/workshops/dsdm04/files/Actas-DSDM04.pdf)

## **1.2. DEFINICIÓN DEL PROYECTO**

### **1.2.1. DESCRIPCIÓN DE LA ORGANIZACIÓN**

Emilio Segré es una institución educativa, que identifica claramente la misión, principios fundamentales y objetivos de la institución. Los siguientes puntos especificados de misión, principios fundamentales y organigrama de la institución son referencias obtenidas de la dirección de la unidad educativa Emilio Segré.

#### **1.2.1.1. Misión**

El colegio Emilio Segré tiene como misión transmitir, difundir, transferir y asimilar el conocimiento para contribuir al desarrollo sostenido y sustentable de nuestro país; como resultado de este esfuerzo institucional, los estudiantes se podrán defender en otras instituciones como institutos, universidades, empresas, etc.

Además sin descuidar la formación cívica, esta formación es muy importante ya que, se enseña el respeto por nuestra patria, donde se enseña el porque de los símbolos patrios, el respeto a nuestra gente y nuestras raíces.

#### **1.2.1.2. Principios Fundamentales**

El colegio Emilio Segré es una institución laica y democrática, que garantiza la libertad de pensamiento, expresión y culto de todos sus integrantes, sin discriminación alguna.

Además estimula el respeto de los valores inherentes a la persona y, en particular, la libertad, la igualdad y el cumplimiento de las leyes y normas vigentes de la ley de educación.

### 1.2.1.3. Organigrama de la institución.

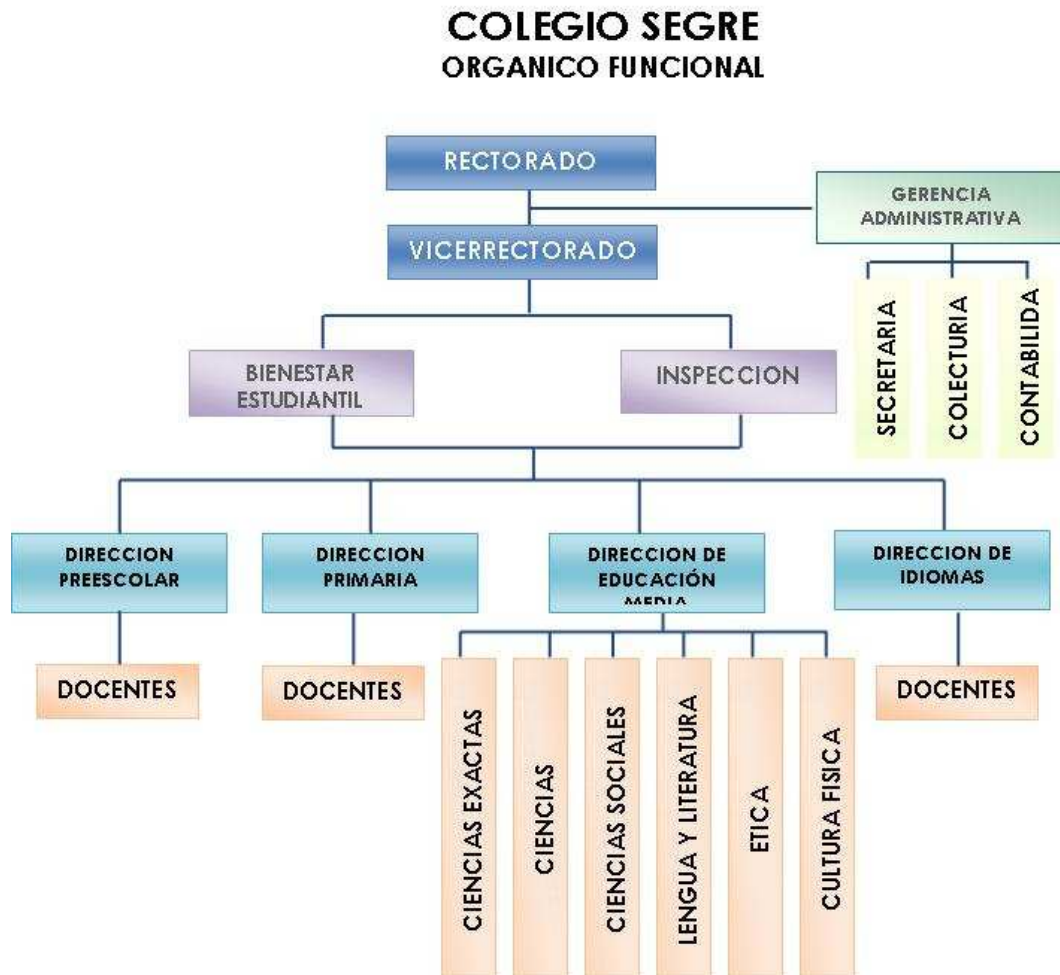


Figura 1.1 Organigrama de la Institución.

### 1.2.2. PLANTEAMIENTO DEL PROBLEMA

- El colegio Segré no tiene procesos adecuados, ni un sistema que permita manejar el registro de información estudiantil, ejecución del proceso de matrículas y control de notas.
- El registro de información estudiantil es un proceso que almacena toda la información de los estudiantes que van a pertenecer a la institución, durante el año lectivo, esta información es registrada cuando el estudiante es

matriculado por primera vez, luego de esto, la información se mantendrá como registro histórico del colegio.

- El proceso de matrículas, es un proceso en el que se registra el año que va a cursar el estudiante, pero cada año a cursar tiene materias específicas que deberá tomar.
- Control de notas, es un proceso en el cual se despliega la información sobre las notas registradas de la evaluación a los estudiantes durante el período escolar. Este proceso permitirá a los estudiantes llevar un control sobre su rendimiento académico.

### **1.2.3. ALCANCE**

El Sistema de Administración Estudiantil constará de:

- Registro de información estudiantil, este proceso se encargará de almacenar toda la información que corresponde al estudiante. Mediante este proceso se podrá identificar al estudiante mediante sus datos personales, a pesar de que cuando el estudiante sea registrado por primera vez en el sistema se lo identificará mediante un código único que se le asignará en la institución. Este proceso también se encarga de asignar un código al estudiante nuevo en la institución. En caso de que el estudiante haya sido registrado anteriormente en el sistema, se mantendrá el código ya asignado.
- Ejecución del proceso de matriculación, con este proceso el estudiante que pertenecerá a la institución, estará legalmente registrado en el año lectivo correspondiente. Dentro de este proceso se especificará en que año lectivo se encuentra el estudiante y posteriormente se especificaran las materias que el estudiante tomará, ya que, esto está definido por el Ministerio de Educación y el colegio se rige a las normas y leyes del Ministerio. Además dentro de este proceso se especifica un solo horario para todos los estudiantes, ya que es un colegio matutino.
- Control de evaluaciones, dentro de este proceso, se permite el registro de las evaluaciones hechas a los estudiantes por los profesores, este registro es

realizado únicamente por la secretaria. Una vez registradas las evaluaciones de cada estudiante de todas las materias, los estudiantes podrán llevar un control de sus reportes académicos.

Para que las evaluaciones sean registradas en el sistema, éstas deben ser proporcionadas por el profesor de cada materia y enviadas a la secretaria mediante una lista de registro académico de todos los estudiantes de un curso.

#### **1.2.4. EXCLUSIONES**

- El sistema no hará un control de asistencia estudiantil, a pesar de es información necesaria de requisito de matriculación al próximo año escolar.
- El sistema no realizará control de horarios para los profesores, ya que este proceso lo realiza el comité de profesores de la institución.
- El sistema no realizará una gestión de horarios sobre las aulas, ya que estos se mantienen durante todos los años y depende de la cantidad de estudiantes matriculados en el período escolar, en caso de que se requiera cambiar la disponibilidad de las aulas, será evaluado por el comité de profesores.

#### **1.2.5. DEFINICIONES**

**Estudiante:** usuario del sistema al cual va dirigido, una persona común se vuelve en este tipo de usuario cuando se inicia el proceso de registro dentro del sistema.

**Secretaria:** usuario que manejará el sistema, la cual está autorizada a registrar toda la información estudiantil, proceso de matriculación y registro de notas.

**Año lectivo:** período en el cual se mantienen las actividades académicas, este período está basado en las leyes y normas del Ministerio de Educación.



**Horario:** listado de actividades, sea de los alumnos, aulas o profesores dentro de un período de tiempo.

**Colegio matutino:** institución secundaria, donde sus actividades académicas se las realiza en la mañana.

**Control de sus reportes académicos:** proceso en el cual se puede llevar una revisión continua de las calificaciones que son resultado de las evaluaciones echas por los profesores.

**Control de asistencia estudiantil:** proceso en el cual se lleva registro de la asistencia y puntualidad a las actividades académicas de los estudiantes.

**Control de horarios:** proceso el cual se organiza el horario de los estudiantes, profesores o aulas.

**Listado de registro académico:** plantilla en el cual constan todos los nombres de los estudiantes de una materia específica, donde se registra las notas, que son resultado de la evaluación realizada por los profesores.

**Comité de profesores:** grupo de profesores de la institución, que se encarga de resolver asuntos internos de la institución.

**Gestión de horarios:** proceso el cual se encarga de administrar la organización de los horarios de los profesores, estudiantes o aulas.

**Año escolar:** los años escolares constan de seis niveles, donde se identifica a un grupo de estudiantes en que nivel se encuentra ubicado, de primer a sexto curso.

**Período escolar:** los períodos es equivalente al año lectivo.

## **1.3. DESARROLLO DIRIGIDO POR MODELOS**

### **1.3.1. DEFINICIÓN DE MDD (MODEL DRIVEN – DEVELOPMENT ‘DESARROLLO DIRIGIDO POR MODELOS’)**

Es una aproximación al desarrollo de software basado en el modelado del sistema y su generación a partir de los modelos. Al ser únicamente una aproximación, sólo proporciona una estrategia general a seguir en el desarrollo de software, pero MDD no define ninguno de los siguientes puntos:

- Técnicas a utilizar,
- Fases del proceso,
- Algún tipo de guía metodológica.

De esta manera MDD proporciona un mejoramiento y optimización de los procesos de desarrollo de software, pero MDD se concentra en los siguientes objetivos:

- Análisis conceptual de las técnicas de modelado de software.
- Estudio, elaboración y formalización de lenguajes para crear modelos de software.
- Catalogación y formalización de mecanismos de transformación entre modelos.
- Definición y formalización de lenguajes para describir transformaciones.
- Implementación de herramientas de software automáticas.

### **1.3.2. ASPECTOS EN LOS QUE MDD MEJORA EL PROCESO<sup>2</sup>**

Para el desarrollo de una aplicación es muy importante tomar en cuenta los aspectos en los que va a mejorar y optimizar los procesos. En el caso de MDD

---

<sup>2</sup> <http://www.lifia.info.unlp.edu.ar/es/platero.htm>

se considera que contribuye a mejorar el proceso de desarrollo de software en los siguientes aspectos:

- **Productividad:** El desarrollo de software basado en modelos se ha comprobado que mejora la productividad global del proyecto, ya que una de las etapas del desarrollo de un sistema es el diseño y con una buena abstracción de modelamiento se aplican los mismos modelos para desarrollo de software, permitiendo ahorrar el tiempo de codificación y especialmente la etapa de mantenimiento, ya que introducir modificaciones sobre los modelos es mucho más rápido y fácil que hacerlo sobre el código.
- **Portabilidad:** la transformación de los modelos permite la independencia de las tecnologías. Este es un aspecto muy importante ya que, mientras transcurre el tiempo, se presentan nuevas tecnologías y los sistemas deben adaptarse a las tecnologías nuevas. MDD facilita la portabilidad ya que, una vez obtenidos los modelos se los puede llevar a una herramienta que cumpla con las tecnologías modernas.
- **Re-uso:** una vez que se han realizado los modelos correctamente y con toda la características de los requerimientos, es posible usar estos modelos para una parte o completamente para desarrollar nuevos sistemas.
- **Interoperabilidad:** el desarrollo de software de los últimos años exige a que los sistemas tengan la característica de interoperabilidad; **MDD** favorece la interoperabilidad entre sistemas heterogéneos, ya que permite la especificación de las conexiones entre las distintas tecnologías.

### 1.3.3. PARADIGMA DE MDD

MDD se basa en dos ejes principales:

- MDD se concentra en la separación entre la especificación de la funcionalidad esencial del sistema y la implementación de la funcionalidad del sistema usando plataformas tecnológicas que permitan la transformación de los modelos al sistema.

Para ello, el MDD identifica dos tipos principales de modelos:

- PIM (Modelo Independiente De La Plataforma 'Modelo Independiente De La Plataforma')
  - PSM (Platform Specific Model 'Modelo Especifico De La Plataforma')
- Un PIM es transformado en uno o más PSM's, lo que quiere decir que MDD permite especificar para cada plataforma un PSM específico. Es decir el principio básico de MDD es la transformación entre modelos, de esta manera los modelos pasan de ser entidades contemplativas a entidades productivas.

### 1.3.4. MODELOS

#### 1.3.4.1. PIM (Platform Independent Model 'Modelo Independiente De La Plataforma')

Es un modelo del sistema desde el punto de vista independiente de la plataforma, el cual podrá ser aplicado en diferentes plataformas. Este modelo representa la estructura, funcionalidad y restricciones del sistema, siendo la base para todo el proceso de desarrollo y el único que debe ser creado íntegramente por el desarrollador.

#### **1.3.4.2. PSM (Platform Specific Model ‘Modelo Específico De La Plataforma’)**

Son modelos del sistema desde el punto de vista específico de la plataforma, es decir, con detalles específicos de la plataforma elegida para implementar el sistema. Este modelo está más cercano al código que el PIM, pero puede incluir detalle dependiendo de su propósito.

Lo que muestra el PSM, son tres conceptos fundamentales:

- Un conjunto de conceptos técnicos que representan las diferentes formas o partes que componen un sistema, y los servicios que provee.
- Conceptos que explican los diferentes elementos que provee una plataforma para implementar una aplicación en un sistema.
- Los requerimientos de conexión y uso de las partes que integran la plataforma y la conexión de las aplicaciones a dicha plataforma.

#### **1.3.4.3. Implementación específica de la plataforma o Código**

Se trata del código del sistema a desarrollar. Este código en su totalidad o en parte, será generado automáticamente mediante reglas de transformación.

#### **1.3.5. ELEMENTOS PARA MDD**

La OMG ha definido lenguajes de modelado para especificar PIM's y PSM's. El más conocido y usado es el UML (Unified Modeling Language ‘Lenguaje de Modelamiento Unificado’), acompañado por el OCL que es un lenguaje de consulta para UML. UML es un lenguaje muy expresivo que permite la creación de distintos modelos del sistema desde diferentes puntos de vista y diferentes niveles de abstracción. UML provee adicionalmente mecanismos de extensión llamados “perfiles” que nos permiten definir lenguajes derivados de UML.

Lenguajes definidos mediante perfiles 'profiles' son adecuados para crear modelos en dominios específicos. Los perfiles generalmente se utilizan para definir PSM's.

#### 1.3.5.1. Técnicas Para MDD<sup>3</sup> <sup>4</sup>

Para desarrollar y ejecutar MDD, existen tres técnicas principales:

- MDA
- Factorías de Software
- DSM (Domain Specific Modeling 'Desarrollo Especifico del Domino')

Para desarrollar un sistema basado en el Desarrollo Dirigido en Modelos se necesitan lenguajes que permitan la abstracción en un mayor nivel. Estos lenguajes son denominados DSL (Domain Specific Language 'Lenguajes De Modelado o Lenguajes Especificos del Dominio').

En la Figura 1.2 se indica el proceso que se realiza desde la construcción de un DSL hasta llegar al lenguaje de máquina, para que exista este proceso son necesarios compiladores diferentes.

Para las conversiones se parte de un *DSL* y este lenguaje de modelado es procesado por un *compilador* que lo convierte en Lenguaje de Programación. A continuación este lenguaje de programación es procesado por otro compilador que lo convierte en *Lenguaje de Máquina*. La parte fundamental de MDD es el Lenguaje de Modelado.

---

<sup>3</sup> [www.dsic.upv.es/workshops/dsdm04/files/01-Munoz.ppt](http://www.dsic.upv.es/workshops/dsdm04/files/01-Munoz.ppt)

<sup>4</sup> <http://cuartageneracion.blogspot.com/search/label/MDA>

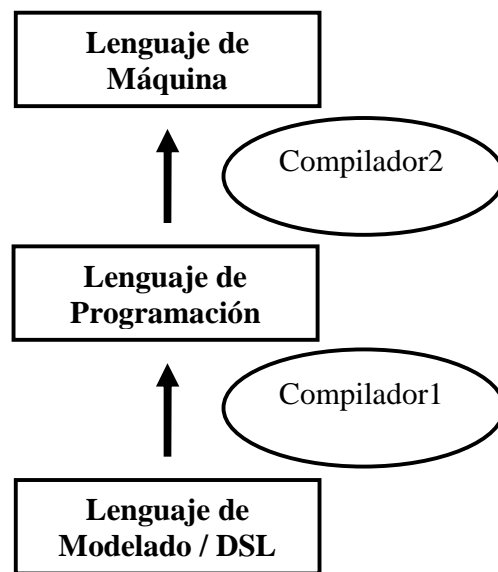


Figura 1.2: Proceso para el desarrollo de un DSL hasta el lenguaje de máquina.

#### 1.3.5.1.1. MDA<sup>5</sup>

Es un 'framework' de desarrollo de sistemas, donde las herramientas se basan en los modelos del sistema para guiar todo el proceso de construcción del software. MDA se concentra en los procesos más importante de un proyecto de software que van desde su análisis y diseño, implementación y mantenimiento hasta la integración en caso de ser necesario con otros sistemas.

MDA<sup>6</sup> se basa en tres objetivos principales para el desarrollo de un sistema de software:

- Definir un modelo independiente
- Definir modelos de plataformas específicas
- Generar el código de la aplicación a partir de los modelos.

<sup>5</sup> <http://www.lcc.uma.es/~av/MDD-MDA>

<sup>6</sup> [es.wikipedia.org/wiki/Model\\_Driven\\_Architecture](http://es.wikipedia.org/wiki/Model_Driven_Architecture)

Para llegar a los objetivos señalados es importante especificar que, se debe partir de los modelos que permiten la abstracción de un software, estos modelos son especificaciones pertenecientes a la OMG.

Esta organización, establece dos capas de abstracción para el desarrollo de software:

- Modelo independiente de la plataforma PIM
- Modelo específico de la plataforma PSM

Cada uno de los modelos pueden ser aplicados en lenguajes diferentes que pueden ser L1 y L2, y las transformaciones PIM-PSM y PSM-código necesitan herramientas de transformación. Las transformaciones son expresadas mediante un lenguaje de transformación.

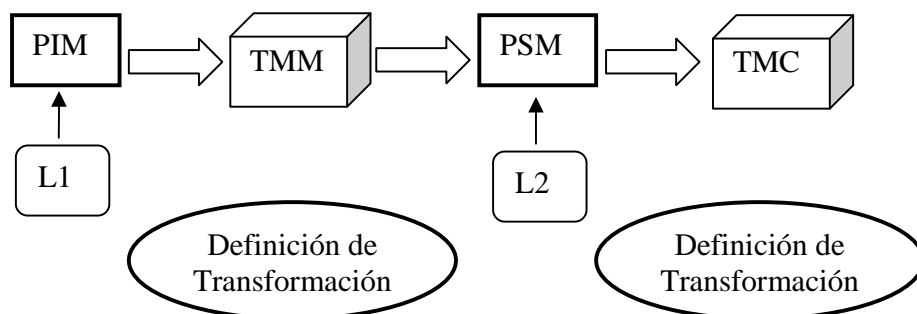


Figura 1.3: Procesos de la transformación de modelos a modelos y modelos a código

Para poder cumplir el ciclo de transformación de modelos se han propuesto estándares que permiten una buena aplicación de MDA. La OMG ha propuesto y desarrollado los siguientes estándares: UML, MOF, OCL (Object Constraint Language), XMI (XML Metadata Interchange 'XML de Intercambio de Metadatos'), QVT (Query/Views/Transformations).

Estos estándares son un conjunto de lenguajes que permiten el diseño y modelado, lo cual ayudará a desarrollar los modelos iniciales para la transformación de modelos a modelos y modelos a código.



a) UML<sup>7</sup>

Es un lenguaje que permite especificar, construir, visualizar y documentar los artefactos o componentes de un sistema. Lo que proporciona el artefacto es la información como resultado de un proceso de desarrollo de software.

La meta de UML es convertirse en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo.

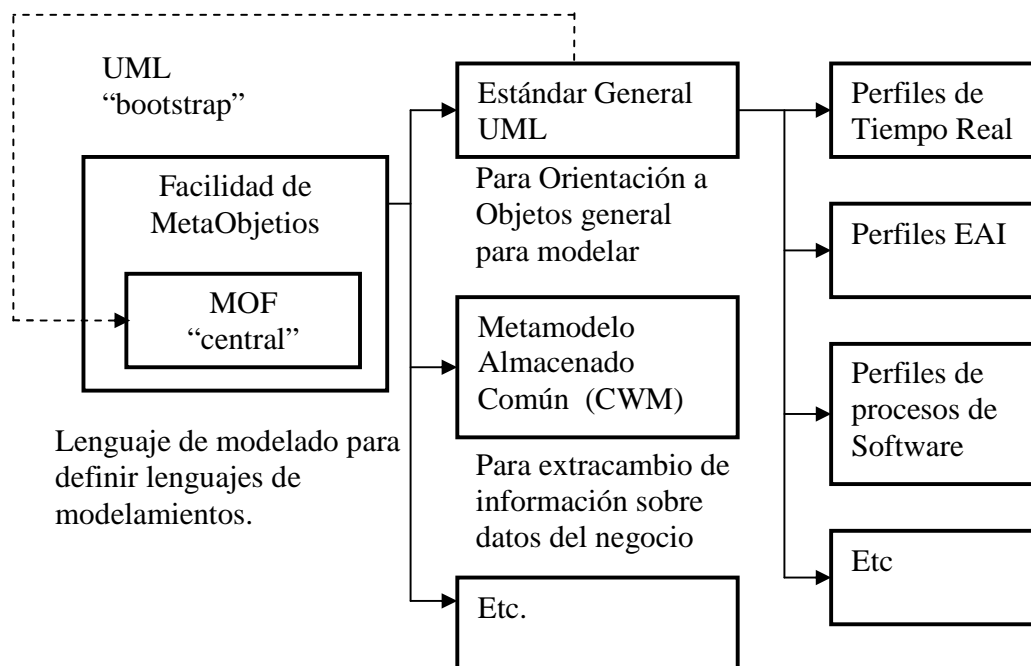


Figura 1.4: Niveles de UML, tutorial de UML

<sup>7</sup> <http://www.monografias.com/trabajos5/insof/insof.shtml>

b) *MOF*<sup>8</sup>

Es un lenguaje de modelado que fue creado a causa de las exigencias técnicas de diseño que se iban incrementando al desarrollar nuevas aplicaciones. A causa de estas necesidades se desarrollan lenguajes que puedan interpretar mediante modelos el ámbito del negocio, especialmente que se pueda interpretar lo que son los PIM's o PSM's. MOF es un lenguaje que permite describir lenguajes de modelado. Un lenguaje de modelado puede usar MOF, para definir formalmente la sintaxis abstracta de su conjunto de constructores de modelos, es decir, lo que permite este lenguaje es definir los constructores formalmente.

Además un metamodelo también especifica semántica informal, la cual es interpretada por el lenguaje natural. La combinación de definiciones formales e informales se llama metamodelo MOF o modelo MOF.

MOF es un estándar hermano de UML y es uno de los elementos en los que se basa MDA incluso mucho más que UML y sus perfiles, ya que hasta los perfiles son definidos vía MOF.

MOF toma prestados de UML los constructores del modelo de clases orientado a objetos y los presenta como la norma para describir la sintaxis abstracta de los constructores de modelado para definir la sintaxis abstracta de un metamodelo. Por lo tanto, los metamodelos MOF son como los diagramas de clases UML, y en consecuencia se pueden usar herramientas de diseño UML para crearlos.

Con MOF, se modela un constructor de modelado como una clase y las propiedades del constructor como los atributos de la clase, modelando además las relaciones entre constructores como asociaciones.

---

<sup>8</sup> [www.omg.org/mda/specs.htm](http://www.omg.org/mda/specs.htm)

c) *OCL*

Permite expresar restricciones semánticas a los metamodelos. Es un complemento ideal para MOF, ya que MOF no puede añadir estas restricciones.

El OCL de la Figura 1.5 asegura de que una tabla al heredar de otra herede también las columnas de sus superclases.

```
-- Una Tabla hereda las columnas de su superClases.

context Tabla inv:
  superclass.columna->forAll
    (superClassColumna | self.columna->incluyes
      (superClassColumna) )
```

Figura 1.5: Redefinición de subclase

d) *XMI*<sup>9</sup>

Es un estándar basado en XML que fue creado con el objetivo de ejecutar el intercambio de modelos. XMI llegó a ser popular después de que la especificación MOF fuese escrita y se presentó como un medio excelente para las herramientas que necesitaban de un estándar de intercambio de modelos.

La arquitectura de XMI permite poder representar metamodelos y modelos MOF en ficheros XML y por lo tanto simplifica la comunicación entre aplicaciones que los usen. La importancia de XMI es que potencia la reutilización de objetos y componentes. Los metamodelos son representados mediante DTDs o Esquemas XML, mientras que los modelos son ficheros XML que se adaptan a un tipo determinado.

XMI proporciona un conjunto de reglas de producción para transformar metamodelos basados en MOF a DTD's XML y para transformar modelos basados en metamodelos MOF en archivos XML.

<sup>9</sup> [albertocc.tripod.com/pdf/TecnologiasMDA.pdf](http://albertocc.tripod.com/pdf/TecnologiasMDA.pdf)

Las reglas XML para producir DTD's y documentos XML permiten no solo generar DTD's desde un metamodelo MOF automáticamente, sino que también podrían ser usadas para generar código que importe modelos desde y hacia repositorios de metadatos.

e) *QVT*<sup>10</sup>

El principal problema sobre MDA es la transformación entre modelos, este problema se ha definido, ya que no existe un lenguaje estándar que realice las transformaciones. Sin embargo en la creación de lenguajes, la OMG definió que las propuestas de los lenguajes estándares tienen que ser compatibles con: UML, MOF, CWM, SPEM. Además el lenguaje debe cumplir con lo siguiente:

- realizar transformaciones entre modelos de metamodelos definidos bajo MOF
- permitir la creación de la vista de un metamodelo
- permitir la flexibilidad en su implementación.

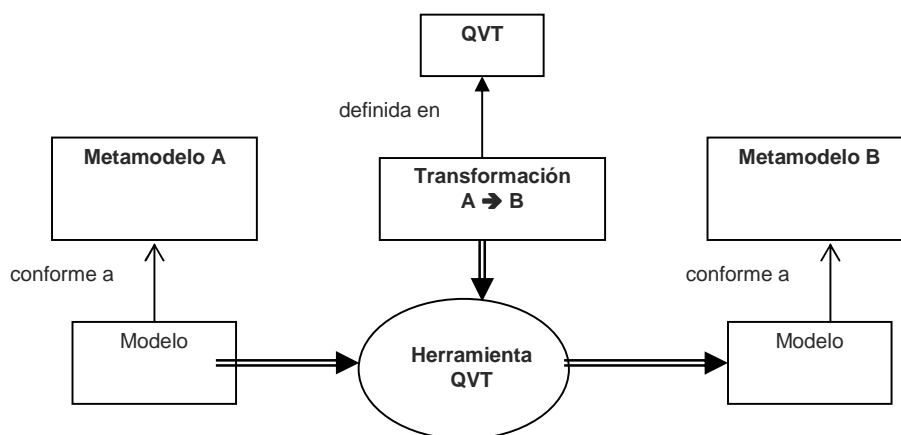


Figura 1.6: Proceso de transformación con QVT

<sup>10</sup> [www.omg.org/mda/specs.htm](http://www.omg.org/mda/specs.htm)

#### *1.3.5.1.2. Factorías de Software*

La factoría de software o fábrica de software es una línea de producto software cuyos componentes de producción forman un IDE configurado para permitir el desarrollo rápido de los miembros de la familia de productos. Es decir, son elementos de software instalados en un entorno de desarrollo para favorecer el desarrollo de aplicaciones concretas para un dominio de forma productiva y generando código de calidad.

Un ejemplo se lo puede definir dentro de una aplicación de comercio electrónico, la factoría de software se la define dentro de un conjunto de elementos software que son procesos, patrones, frameworks, DSL y herramientas.

Las factorías de software se basan en dos elementos principales:

- Esquema de la factoría
- Plantilla de la factoría

El Esquema De La Factoría se basa en tabular y describir toda la información necesaria para producir una aplicación de una familia de línea de producto, es decir, establece todos los artefactos necesarios y las relaciones entre ellos, y distingue partes fijas de partes variables. Estos artefactos son procesos, patrones, frameworks, DSL y herramientas que constituyen la línea de producto software.

La Plantilla de la Factoría está formada por el código y metadatos que implementan los artefactos. Una plantilla debe instalarse y configurarse en un IDE para crear aplicaciones concretas.

#### *1.3.5.1.3. DSM*

Desarrollo Específico del Dominio o Desarrollo de Software Específico del Dominio plantea elevar el nivel de abstracción por encima de los lenguajes de

programación, de modo que sea posible especificar la solución mediante conceptos del dominio, siendo el código final generado automáticamente a partir de esas especificaciones de alto nivel. Los expertos en un dominio crean los lenguajes específicos del dominio (DSL) y sus generadores de código, y los desarrolladores los usan para especifican una solución de alto nivel, de una forma más productiva que escribiendo el código en un lenguaje de programación.

### **1.3.6. APLICACIÓN DE MDD PARA LOS SISTEMAS DE SOFTWARE.**

Es muy importante contar con una metodología que permita definir de forma ordenada los procesos para llegar al sistema final. La metodología adecuada y que respalda los conceptos fundamentales de MDD es O-O Method “Metodología Orientada a Objetos”<sup>11</sup>, esta metodología se basa en separar la definición abstracta de un sistema de información “espacio del problema”, de sus implementaciones concretas “espacio de la solución” y como se especifica en el punto 1.3.3. , MDD concuerda mucho con el concepto de la metodología Orientada a Objetos. Por este motivo el desarrollo del Sistema De Administración Estudiantil será basado en la metodología mencionada.

#### **1.3.6.1. Metodología Orientada a Objetos Para MDD<sup>12</sup>**

Como se explicó en el apartado 1.3.6. , las metodologías son los mecanismos que se utilizan para cumplir con una etapa. Se realizará una breve explicación sobre los mecanismos para el desarrollo del sistema.

---

<sup>11</sup> <http://www.sistedes.es/sistedes/pdf/2007/JISBD-07-giner-ubiquitous.pdf>

<sup>12</sup> <http://www.dsic.upv.es/workshops/dsdm04/files/08-Molina.pdf>

## **Requerimientos**

Para MDD, la etapa de Requerimientos no es muy indispensable, sin embargo dependería del proyecto si es necesaria la ejecución de la captura de Requerimientos.

Para la mayoría de metodologías en esta fase depende de las empresas, desarrolladores o usuarios para usar estándares, plantillas de documentación u otros mecanismos que sirvan para la documentación de los requerimientos. Para el caso del desarrollo del sistema de administración estudiantil para el colegio Segré se basará en el estándar IEEE Std 830:1998, ya que es un estándar que se acopla a las necesidades del proyecto a desarrollarse y se concentra específicamente en los requerimientos esenciales del proyecto y permitirá al cliente describir de manera precisa que es lo que necesita y al desarrollador comprender la necesidad del cliente en el sistema de administración estudiantil. La plantilla de este estándar es mostrada en el *ANEXO I* de este proyecto.

## **Análisis**<sup>13</sup>

Para esta fase, en que caso de haber realizado la etapa de Requerimientos se recibe como entrada los documentos realizados y como resultado de la etapa de *Análisis* se obtendrá un PIM, en caso de no haber realizado los Requerimientos se limitará al entendimiento del negocio y del problema para la creación del PIM. Este Modelo Independiente de la Plataforma según la *Metodología Orientada a Objetos* es el diagrama de clases, que es uno de los documentos entregables de esta etapa.

Las actividades para esta etapa son los siguientes:

---

<sup>13</sup> <http://www.cs.ualberta.ca/~pfiguero/soo/metod/requerimientos.html>

### **Actividades técnicas**

- Identificar Casos de Uso del sistema
- Desarrollar el diagrama de casos de uso identificados
- Detallar casos de uso
- Identificar Tipos de Datos, Clases, Atributos, Operaciones y Relaciones.
- Desarrollar el modelo de Diagrama de Clases.
- Detallar Tipos de Datos, Clases, Atributos, Operaciones y Relaciones.
- Validar Modelos

### **Documentos Entregables**

- Diagrama de casos de uso
- Detalle de casos de uso
- Modelo de Diagrama de Clases
- Detalle de
  1. Tipos de Datos
  2. Clases
    - a. Atributos
    - b. Operaciones
  3. Relaciones

Es importante contar con un documento que describa los casos de uso y la interacción con el usuario y el sistema. Para documentar los casos de uso, se utilizará la propuesta hecha por la *Universidad Nacional de Río Cuarto de Argentina*. El tipo de documento que proponen es detallado y claro, que permite describir de manera adecuada cada uno de los casos de uso y que basado en estos, se crea el diagrama de clases. Esta plantilla es mostrada en el *ANEXO II*<sup>14</sup> de este proyecto.

---

<sup>14</sup> <http://dc.exa.unrc.edu.ar/nuevodic/materias/proyecto/Repositorio%20de%20archivos/1215097517/1189440714/Plantillas%20CU%20Genericos2007.pdf>



Igual que para los Casos de Uso, para las Clases es necesario contar con una plantilla que permita documentar las características de las Clases. Esta plantilla es mostrada en el *ANEXO III*<sup>15</sup>

### ***Diseño Detallado***

En esta etapa se recibe como entrada el resultado del Análisis, el cual se compilará en la herramienta seleccionada y se obtendrá como resultado uno o más PSM's. Las actividades que se deben cumplir en esta etapa son:

#### **Actividad Técnica**

- Adaptación y Aplicación del PIM a la herramienta seleccionada.
- Compilación del PIM.
- Generación de los PSM's

#### **Documento Entregable**

- PIM aplicado a la herramienta.
- PSM's generados.
  - Especificación de campos adicionales a nivel de presentación.
  - Especificación de controladores para validaciones.

Será necesario que los modelos generados puedan ser visualizados, para un seguimiento y entendimiento de la aplicación.

### ***Codificación***

Se genera el código en la herramienta seleccionada como el resultado de la transformación del modelo o modelos PSM's. Las actividades a cumplir son las siguientes:

---

<sup>15</sup> <http://www.slideshare.net/karla.romero/ciclo-de-produccion-web-y-multimedia/>

**Actividad técnica:**

- Generación de código
- Creación de Base de Datos

**Documento entregable:**

- Descripción de cambios en código generado.
- Código fuente

***Pruebas***

Se realiza una verificación de la funcionalidad de los módulos del sistema.

**Actividad técnica:**

- Prueba de módulos

***Despliegue***

Se realiza un procedimiento de instalación y muestra al cliente final. En caso de que el desarrollador y el usuario estén de acuerdo se realizarán correcciones en el Análisis y se ejecutará el proceso nuevamente.

Una vez que se ha explicado cada una de las fases de la metodología, se presenta en la Figura 1.7 el ciclo de vida para MDD el ciclo de vida para el desarrollo de software, donde se pueden visualizar las entradas y resultados de cada una de las fases de la metodología descrita.

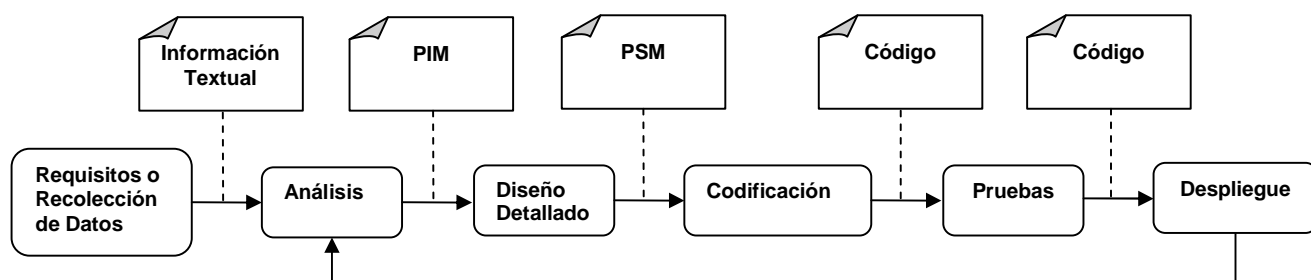


Figura 1.7: Ciclo de vida para MDD

### 1.3.6.2. Criterios de Evaluación para Herramientas MDD

Para la selección de una herramienta adecuada, es importante tomar en cuenta ciertos parámetros y criterios <sup>16</sup> que permitan identificarla. Los criterios tomados en cuenta son los siguientes:

ID	Propiedad	Id	Propiedad
P01	Soporte para PIM	P10	Soporte para consistencia incremental
P02	Soporte para PSM	P11	Soporte de transformaciones de modelos (PIM-PIM, PSM-PSM, etc.)
P03	Permite varias implementaciones	P12	Trazabilidad
P04	Integración de Modelos	P13	Soporte del ciclo de vida de desarrollo.
P05	Interoperabilidad	P14	Uso de estándares
P06	Definición de transformaciones	P15	Control y refinamiento de transformaciones.
P07	Verificación de modelos	P16	Calidad del código generado.

<sup>16</sup> <http://www.dsic.upv.es/workshops/dsdm04/files/09-Garcia.pdf>

P08	Expresividad de los modelos.	P17	Herramienta de soporte.
P09	Uso de patrones		

Tabla 1.1: Criterios de comparación a herramientas MDD

Para realizar un análisis y valoración en cada uno de los criterios, se tomarán valores de la siguiente manera:

Puntaje	Descripción
0	Nulo
1	Mínimo
2	Medio
3	Bueno
4	Excelente

Tabla 1.2: Valores para la evaluación de criterios

Hay que tomar en cuenta que cada criterio valorado dependerá mucho del proyecto a realizarse, en otras palabras, será considerado cómo la herramienta que solucionará el *Planteamiento Del Problema* y el *Alcance* especificados en el punto 1.2.2. y 1.2.3. .

#### 1.3.6.2.1. Herramientas a ser evaluadas

En este apartado se dará un breve resumen de las herramientas más estables según los expertos en MDD y difundidas en el desarrollo dirigido por modelos:

##### a) *MOFScript*

MOFScript es una herramienta “*open source*” para Eclipse que permite generar texto a partir de modelos basados en Ecore (subconjunto de MOF).

Una transformación tiene como entrada un metamodelo y está compuesta de una o más reglas. Cada regla se define sobre un elemento del metamodelo. Las reglas pueden tener argumentos, incluso devolver valores. La regla principal “*main*” es el punto de entrada de la transformación.

*b) MTF (Model Transformation Framework)*

MTF es una propuesta del IBM como respuesta al MOF/QVT RFP de OMG. IBM desarrolla MTF como un prototipo que implementa algunos conceptos de QVT. MTF está constituido por un conjunto de plug-ins que se integran a Eclipse o a las plataformas de desarrollo de IBM.

*c) OptimalJ 17*

Es un entorno de desarrollo para aplicaciones J2EE, construido por Compuware Corporation. OptimalJ permite generar aplicaciones a partir de modelos visuales, basándose en el uso de patrones. Proporciona una manera de escribir aplicaciones Java distribuidas basadas en J2EE.

Proporcionan un editor de modelos de dominio para construir modelos de dominio. Este modelo de alto nivel está basado en objetos y contiene la estructura de la información manejada en la aplicación, así como las relaciones entre las diferentes estructuras de datos. El modelo de dominio contiene las clases de dominio, sus atributos, asociaciones, métodos y reglas de negocio. A partir del modelo de dominio, es posible generar modelos de presentación, lógico y de datos de la aplicación a través de la generación de patrones en línea con el estándar J2EE de Sun.

Estos patrones están constituidos por un conjunto de plantillas de diseño e implementación y se utilizan para generar el código de las aplicaciones. Con OptimalJ se pueden crear aplicaciones utilizando el rango completo de

---

<sup>17</sup> <http://www.compuware.com/products/optimalj/>

componentes J2EE (Session and Entity Enterprise Java Beans (EJB), Java Server Pages (JSP) y tablas de datos de aplicación).

d) *ArcStyler*<sup>18</sup>

Es una herramienta de Interactive Objects. Se define como una aproximación compatible con MDD, abierta a distintas plataformas. Utiliza MOF para soportar los estándares UML, XMI Y JMI.

Esta herramienta permite la creación, edición y mantenimiento de *cartuchos* MDA (MDA-Cartridge). Los cartuchos MDA contienen la tecnología y mecanismos para la transformación automática de modelos a infraestructuras de bajo nivel. Incluye herramientas relacionadas con el modelado del negocio y el modelado de requisitos.

Por muchos especialistas, esta herramienta es una de las completas pero cuestiona la dificultad para llevar a cabo la integración de los modelos.

1.3.6.2.2. *Evaluación de las herramientas descritas*

Una vez realizada la descripción de las herramientas se realiza la evaluación basada en los criterios para identificar cual sería la más adecuada para el desarrollo del sistema.

ID	MOFScript	MFT	OptimaJ	ArcStyler
P01	4	2	4	4
P02	2	2	4	0
P03	2	1	0	3
P04	3	3	4	2
P05	4	3	4	4
P06	2	3	2	4

<sup>18</sup> <http://www.codagen.com/products/architect/default.htm>

P07	1	1	3	2
P08	0	0	2	4
P09	1	1	4	3
P10	2	1	4	3
P11	1	0	2	2
P12	3	3	3	1
P13	3	2	3	4
P14	3	3	4	4
P15	1	0	2	4
P16	3	3	3	3
P17	3	2	3	3
Total	38	31	51	50

Tabla 1.3: Valores para la evaluación de criterios

#### 1.3.6.2.3. Conclusión de las herramientas evaluadas

Como se puede visualizar en el total de la Tabla 1.3, según la evaluación a las herramientas, existen dos que ayudarían a la solución del proyecto, estas son OptimalJ y ArcStyler. Sin embargo el resultado todavía no ayuda a definir la herramienta adecuada, ya que la diferencia de valores entre éstas no es muy amplia, por lo que se hará un análisis entre las dos, tomando en cuenta los principales conceptos de la Orientación a Objetos con el Desarrollo Dirigido por Modelos.

#### **Análisis entre OptimalJ y ArcStyler**

Las valoraciones realizadas en la Tabla 1.3, OptimalJ y ArcStyler permiten la aplicación de PIM's con un excelente soporte, pero es diferente en el soporte para PSM's, ArcStyler no permite la identificación de las capas de presentación, negocio y datos; esto quiere decir que ArcStyler llega al código a partir del

modelo PIM, pero no significa que la herramienta ArcStyler no ejecute la conversión de PIM a PSM's, conversión es interna y no puede ser visualizada, lo cual hace complicado el seguimiento de las transformación en los modelos, por lo que el concepto de trazabilidad es más alto en OptimalJ.

Además hay que tomar en cuenta que el mantenimiento de la aplicación será más sencillo con OptimaJ, ya que el acceso a los PSM's permitirá realizar cambios que faciliten obtener lo requerido por el cliente.

Finalmente, si bien es cierto que la creación de varios PIM's permitirá un mejor entendimiento para la herramienta sobre lo que se desea desarrollar como es el caso de ArcStyler, esto también es una gran desventaja, ya que hace complicado el manejo de la herramienta y significaría demora en el tiempo de desarrollo.

Por lo tanto, según las evaluaciones realizadas y con el análisis realizado, la herramienta más adecuada para el desarrollo del sistema es OptimalJ.

### **1.3.7. FUNDAMENTOS DE LA HERRAMIENTA SELECCIONADA**

#### **1.3.7.1. Creación de un Proyecto en OptimalJ**

Para el desarrollo de los proyectos en OptimalJ existen tres tipos de modelos con se aprecia en la Figura 1.8:

- Modelo del Dominio: Este modelo relaciona conceptos y términos en UML que identifican los elementos de un sistema.
- Modelo de la Arquitectura: Este modelo permite visualizar elementos arquitectónicos que permiten la funcionalidad y ejecución del sistema.
- Modelo de la Aplicación: Contiene los modelos PSM's.



Los PIM's serán aplicados en el Modelo del Dominio.

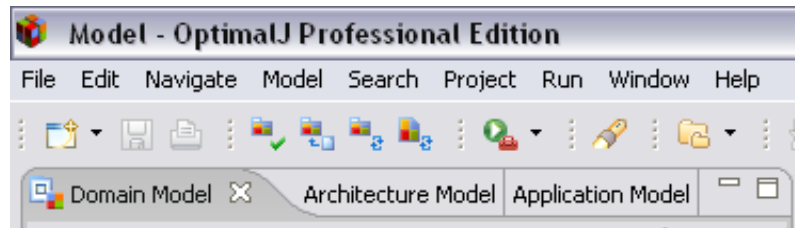


Figura 1.8 : Modelo del Dominio.

Será necesario inicialmente crear un nuevo proyecto como se indica en la Figura 1.9. File -> New -> Modeling Project.

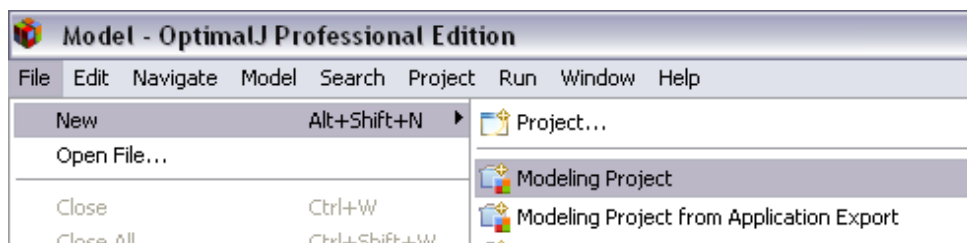


Figura 1.9 : Modelo del Dominio.

A continuación solicitará especificar el Nombre del Proyecto y si es necesario se cambiará la dirección donde se guardarán los archivos del proyecto. Luego se ejecutará **Next**.

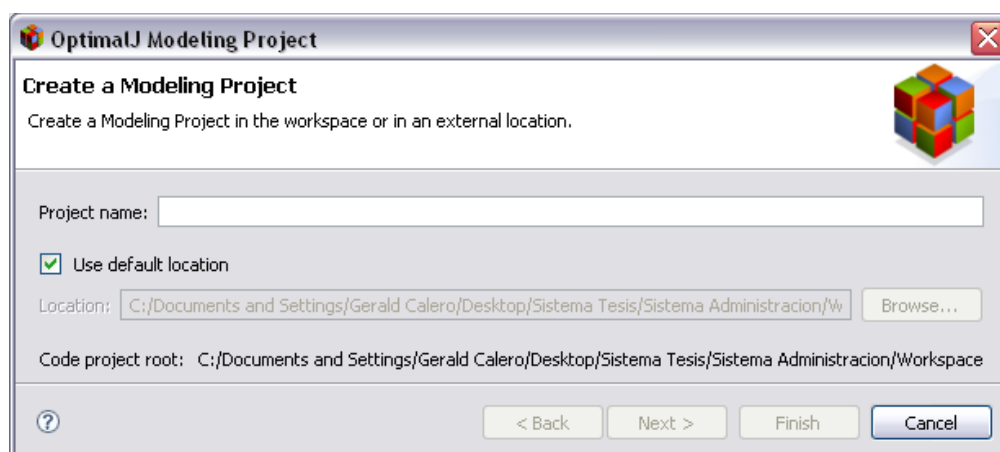


Figura 1.10: Especifica el nombre del proyecto y localización.

En el siguiente paso se especificará un nombre para el paquete y el ID del proyecto, que permitirán identificar al proyecto de otros.

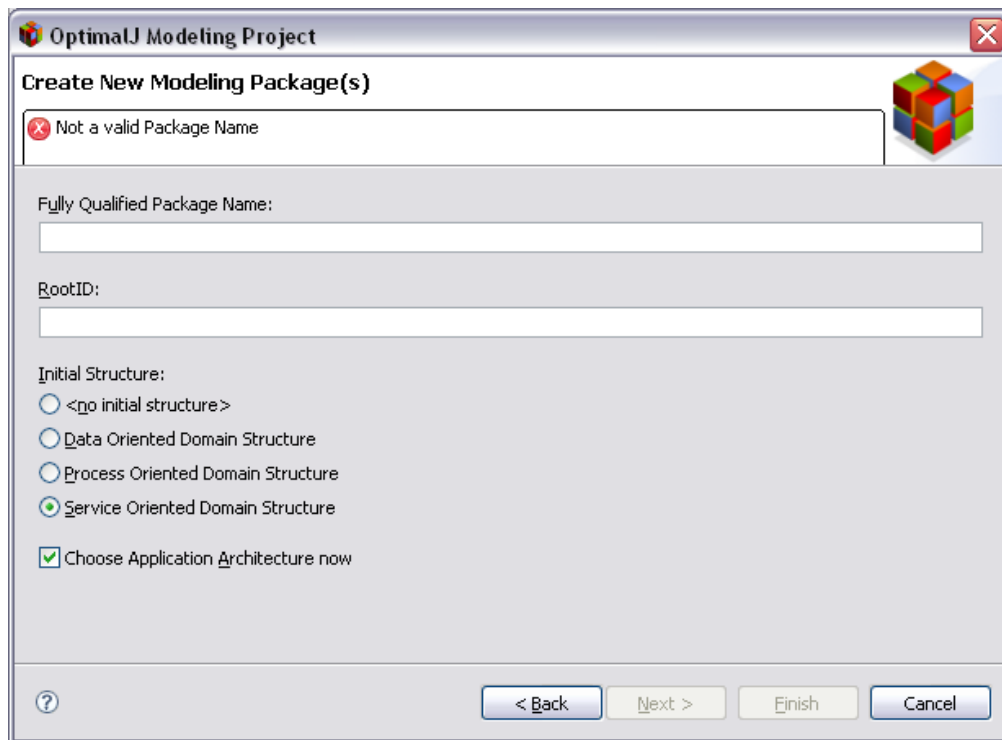


Figura 1.11: Especifica el nombre del paquete e ID del proyecto.

Una vez especificados los datos anteriores, se podrán ver los paquetes donde se almacenarán los modelos. Como se puede observar en la Figura 1.12, en el Modelo de Dominio, se tiene el paquete **class** y **service**. En el paquete **class** se creará el PIM y en el **service** se especificará el flujo de acceso a datos y los atributos que serán usados en la aplicación.

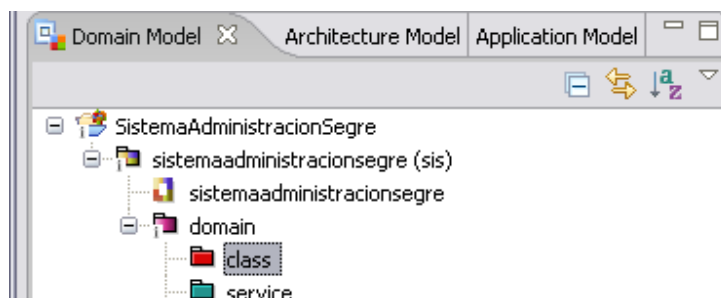


Figura 1.12: Estructura del Modelo de Dominio.

Al ejecutar doble clic sobre el paquete **class**, se abre el ambiente del modelo de dominio. Aquí se podrá crear el diagrama de clases que es el PIM para el Desarrollo Dirigido por Modelos. Como se puede observar en la Figura 1.13, se tiene los elementos para crear el modelo.

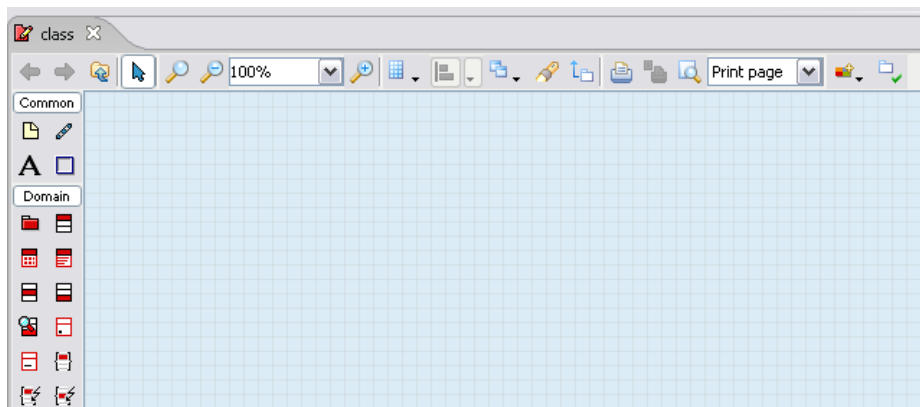


Figura 1.13: Ambiente del paquete *class*.

### 1.3.7.2. Elementos para el Modelo del Dominio.

#### 1.3.7.2.1. Tipo Enumeración

Especifica un listado de valores. Para agregar este elemento será necesario ejecutar el siguiente procedimiento:

Seleccionar el elemento mostrado en la Figura 1.14, y ejecutar clic sobre el ambiente del modelo de clases.

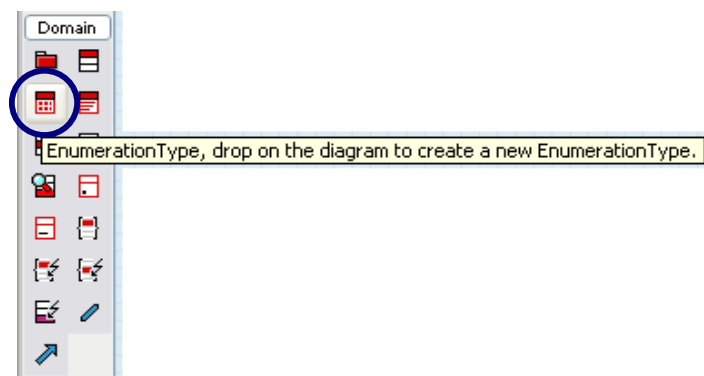


Figura 1.14: Tipo Enumeración.

A continuación se especificará los valores de la Enumeración. El primer paso será especificar el nombre de la Enumeración.



Figura 1.15: Especificación del nombre de la Enumeración.

El siguiente paso es especificar el Módulo en el que se creará la enumeración, en este caso será en el modulo del proyecto creado.

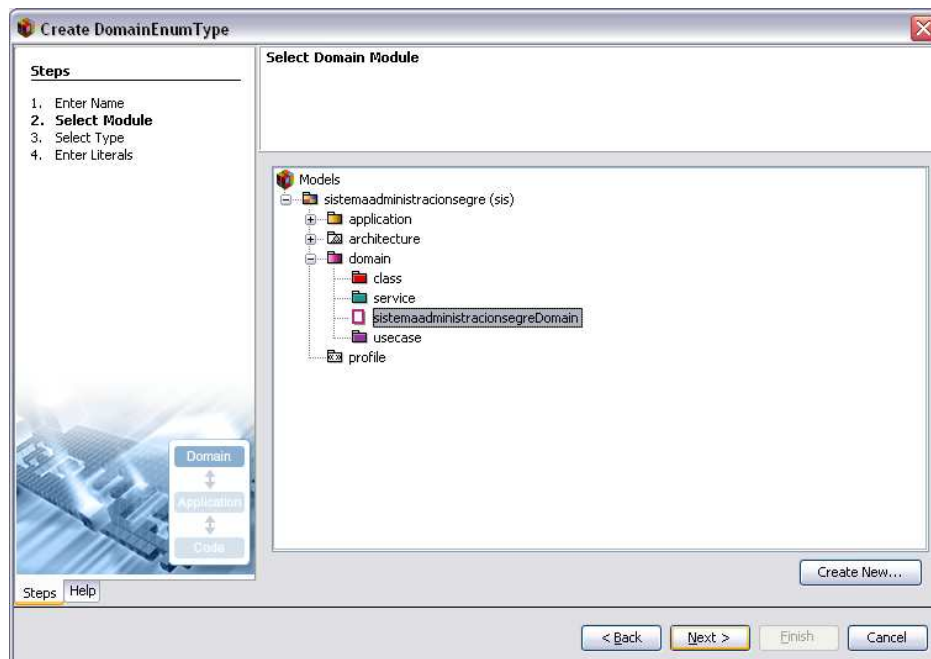


Figura 1.16: Selección del módulo.

En el tercer paso se especificará el tipo de dato que será desplegado en la enumeración.

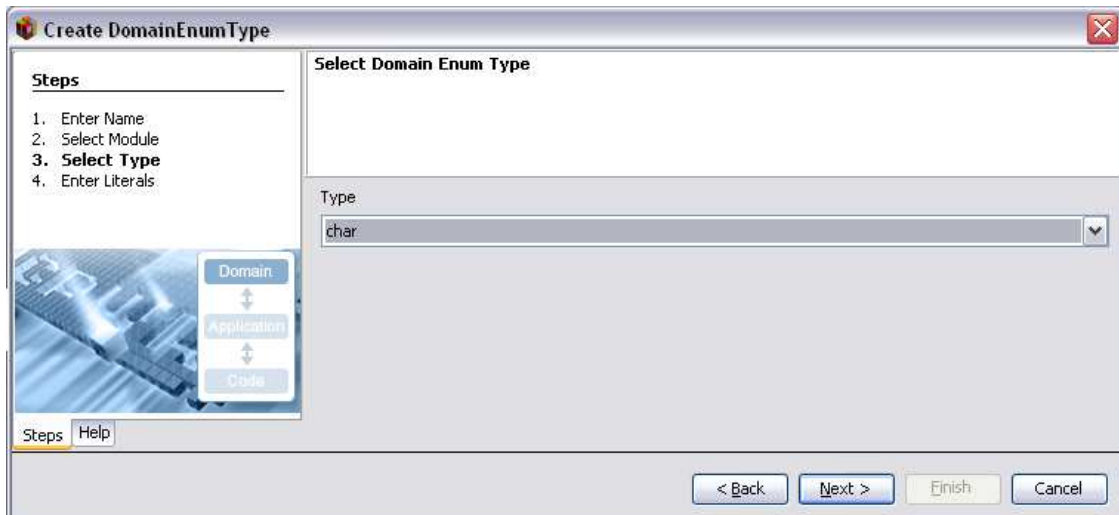


Figura 1.17: Selección del tipo de dato.

Finalmente se especifica los valores que serán desplegados en la Enumeración.

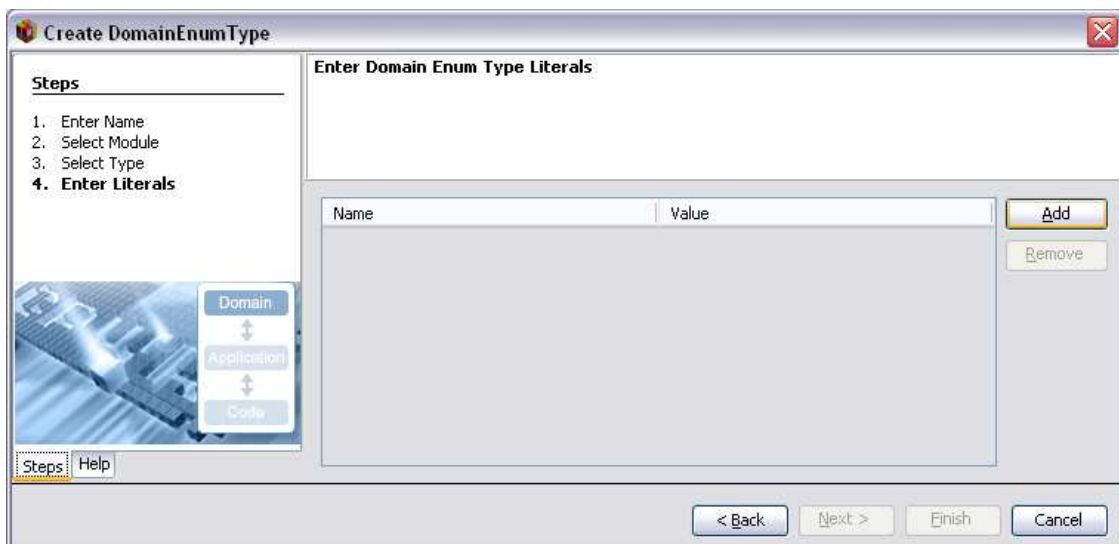


Figura 1.18: Ingreso de valores en la Enumeración.

### 1.3.7.2.2. Tipo Estructura.

Para la creación de estructuras, será necesario el siguiente procedimiento:  
Especificar Structure Type como indica la Figura 1.19.

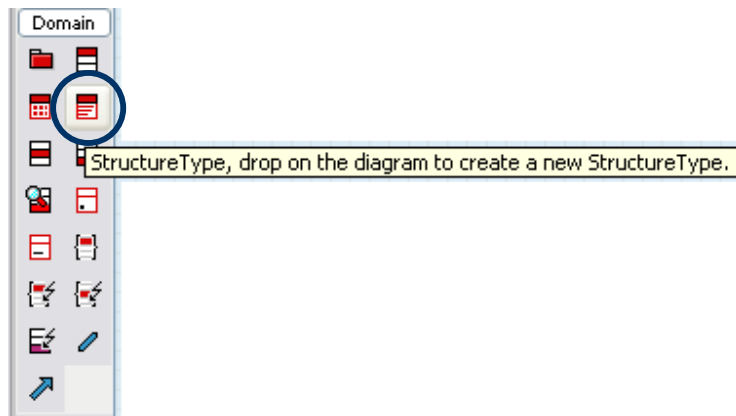


Figura 1.19: Tipo Estructura.

A continuación se especificará el nombre de la estructura, después de ejecutar el clic en el ambiente del modelo.

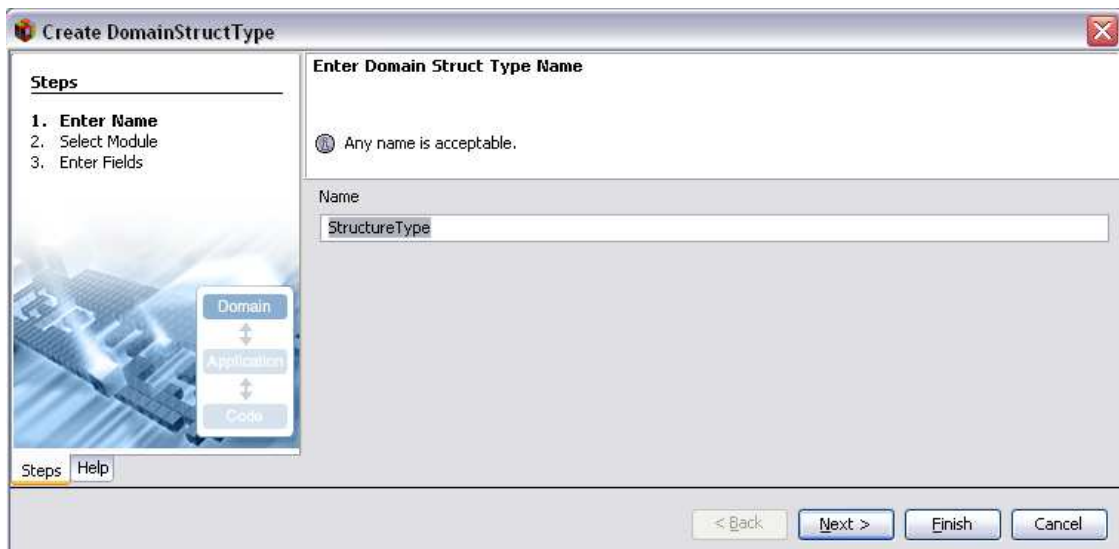


Figura 1.20: Especificación del nombre de la Estructura.

Se especificará en el segundo paso el módulo en el que se crea la Estructura, en este caso será en el módulo del proyecto creado.

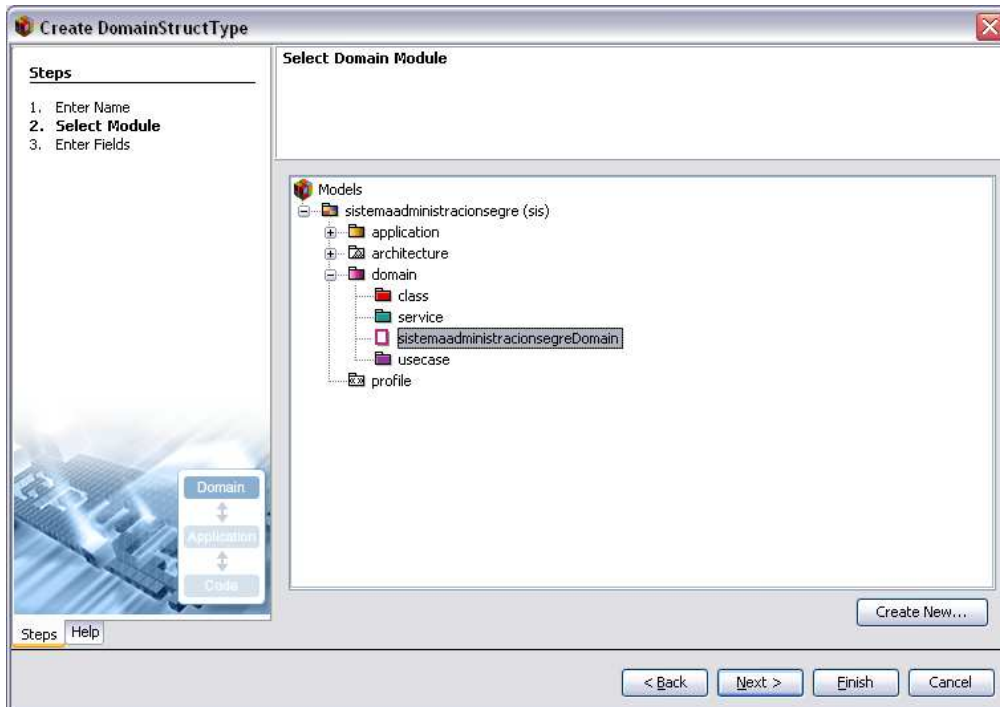


Figura 1.21: Selección del módulo.

Finalmente se añaden los atributos de la estructura y se especifica el tipo de dato de cada atributo.



Figura 1.22: Ingreso de atributos en la Estructura.

### 1.3.7.2.3. Clases

Para especificar una Clase en el modelo, se ejecutará un clic sobre el icono especificado en la Figura 1.23.

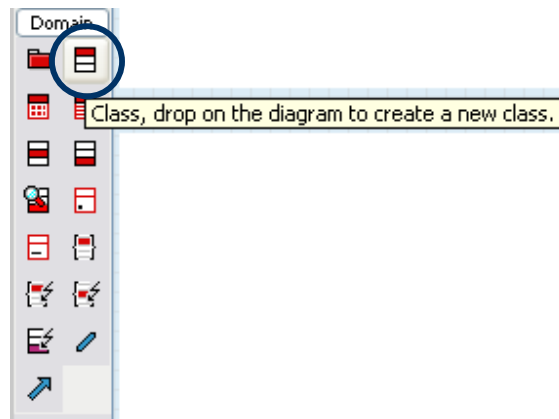


Figura 1.23: Clases.

En el primer paso se especificará el nombre de la Clase.

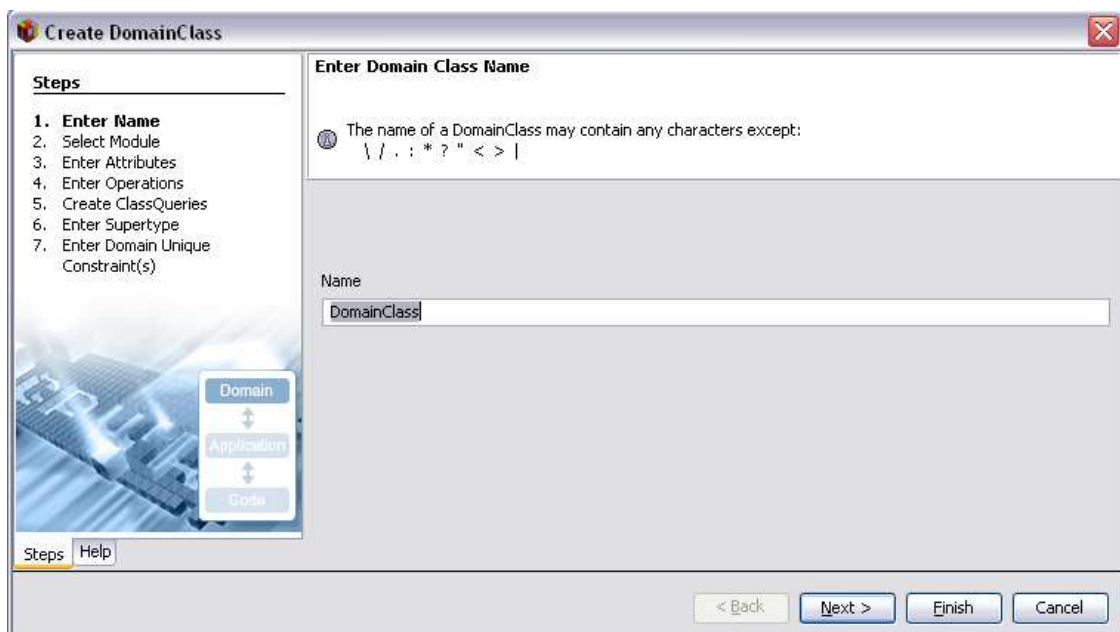


Figura 1.24: Especificación del nombre de la Clase.



En el segundo paso se especificará el módulo del proyecto en el que se creará la clase. En este caso se especificará el módulo del proyecto creado.

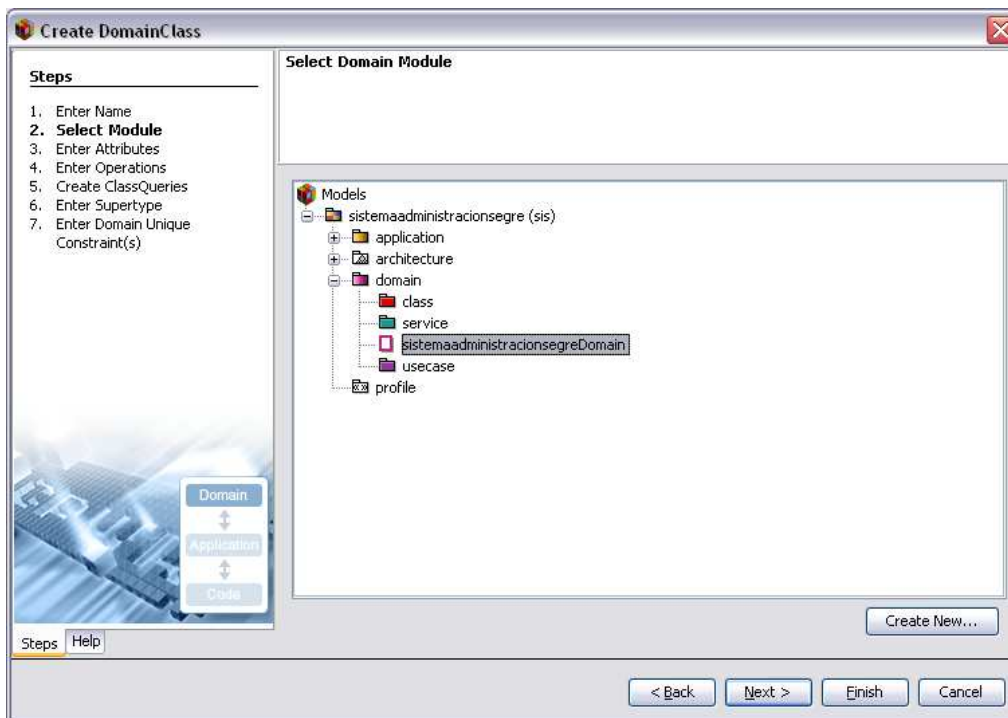


Figura 1.25: Selección del módulo.

En el tercer paso se especificarán los atributos de la clase, con el tipo de dato.

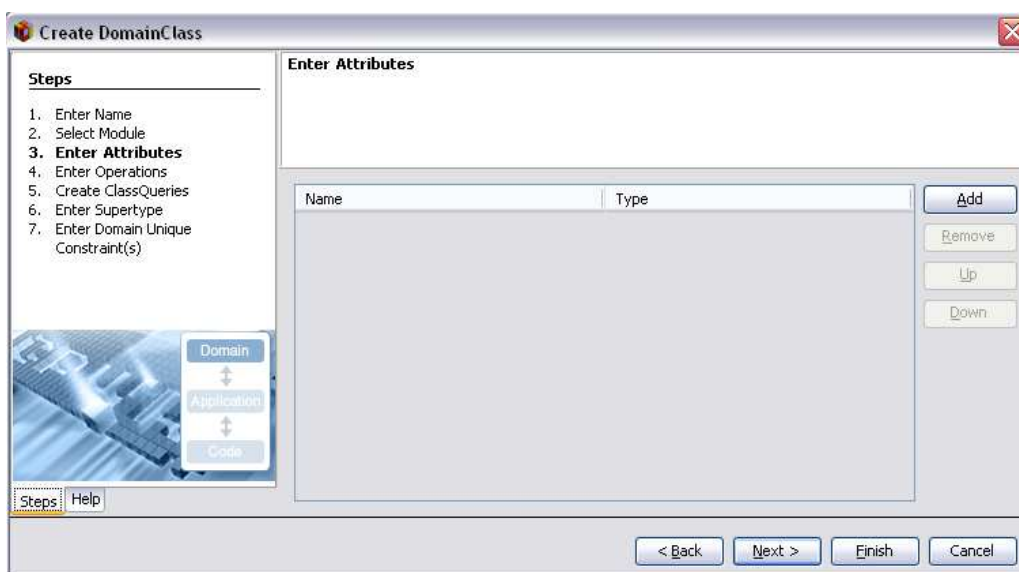


Figura 1.26: Ingreso de atributos en la Clase.

Finalmente se especificará el o los atributos únicos. En el recuadro superior se crearán atributos primarios y en el segundo recuadro se relacionará el atributo de la clase con el creado en el recuadro superior.

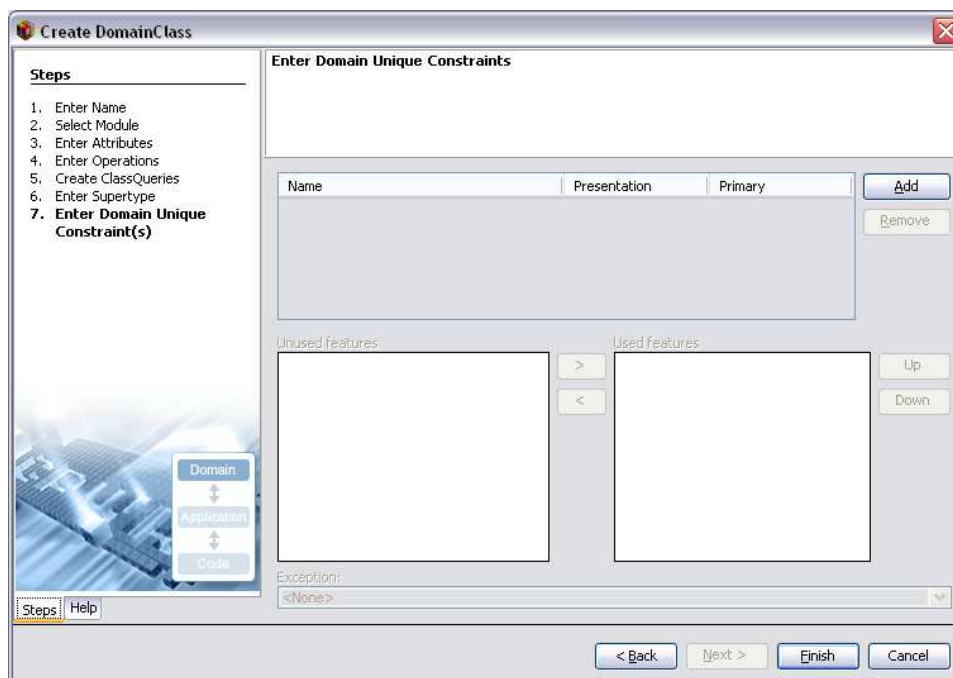


Figura 1.27: Especificación de valores únicos.

#### 1.3.7.2.4. Relaciones

Para la creación de relaciones, será necesario ejecutar un clic en el icono especificado en la Figura 1.28.

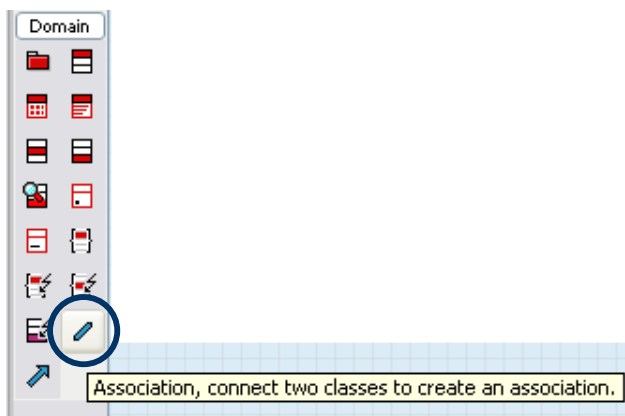


Figura 1.28: Relaciones.

A continuación se especificará con el puntero del mouse las clases que tendrán relación. Luego se ejecutará el wizard que permite especificar la navegabilidad, agregación y multiplicidad, de igual manera sucederá en la otra parte de la relación.

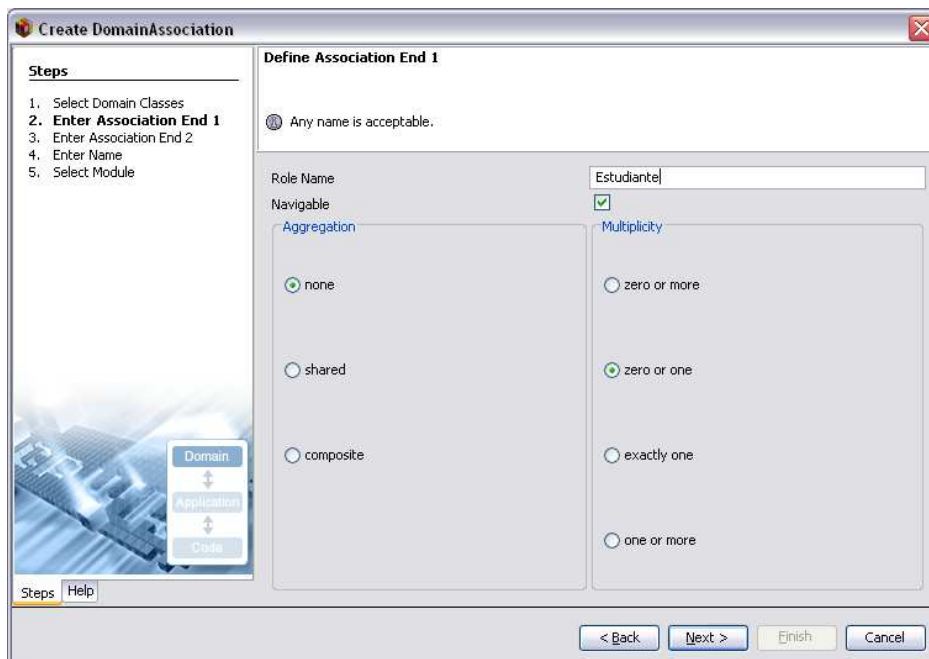


Figura 1.29: Especificación de *Agregaciones y Multiplicidad*.

El siguiente paso será especificar el nombre de la relación.

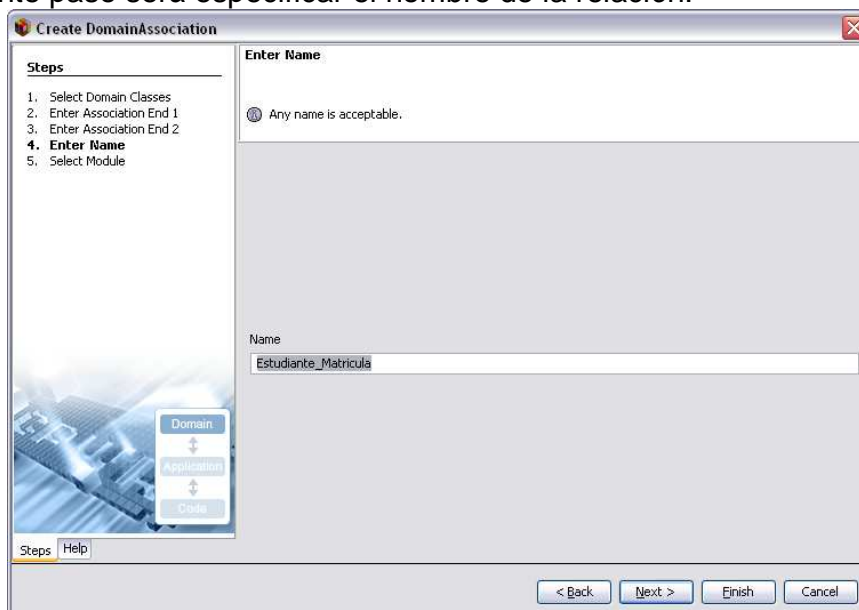


Figura 1.30: Especificación del nombre de la Relación.

Finalmente se especificará el módulo del sistema donde se creará este elemento. Para este caso se creará en el módulo del sistema creado.

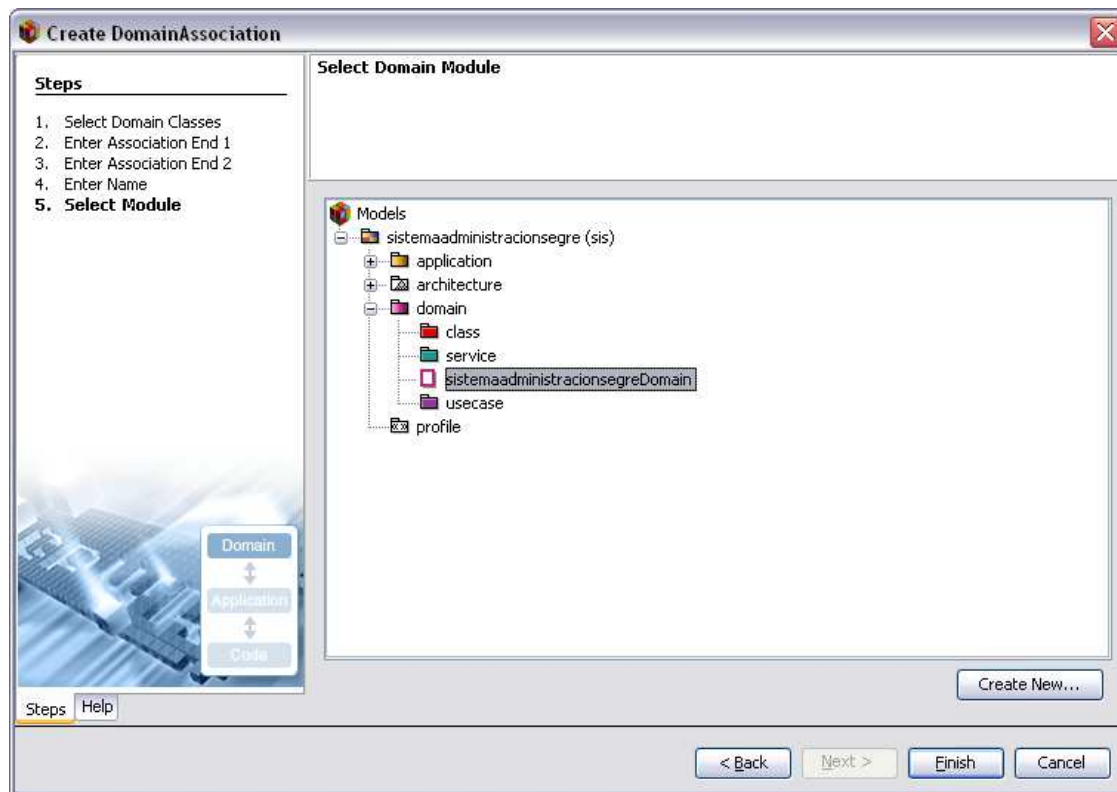


Figura 1.31: Selección del módulo.

### 1.3.7.3. Creación de los Modelos Específicos.

La compilación del PIM mostrará una correcta coherencia con los elementos creados. Si el modelo es correcto se crearán los elementos en el paquete **service** y los PSM's en el paquete del **Modelo de Aplicación**. Como se muestra en la Figura 1.32, este paquete contiene los siguientes PSM's:

- Modelo de Presentación: Contiene los componentes de interfaz para una aplicación Web.
- Modelo del Negocio: Contiene los componentes que especifican las reglas del negocio.

- 📁 **Modelo de Datos:** Contiene los componentes que forman un diagrama relacional y que representan una base de datos.

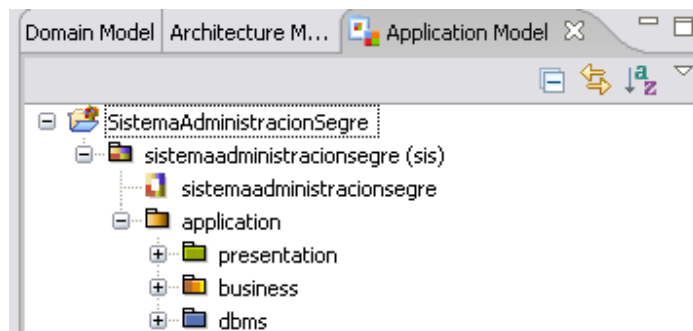


Figura 1.32: Paquetes del Modelo de Aplicación.

Para obtener los Modelos Especificos se ejecutará *Model -> Update All Models*.

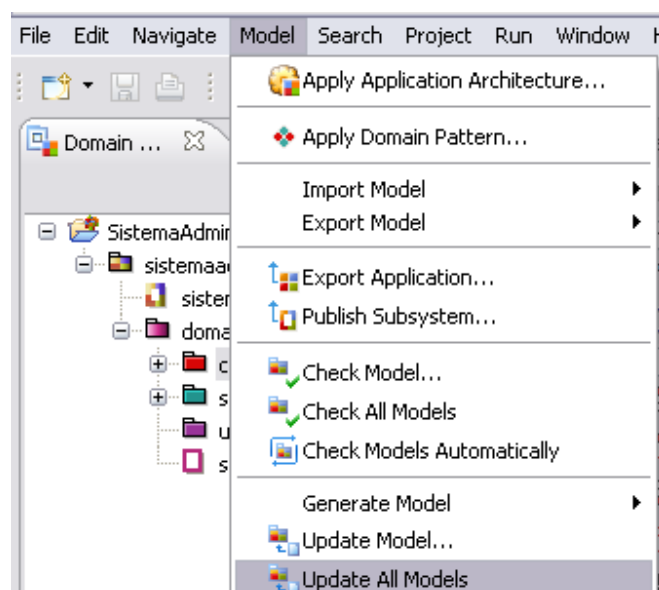


Figura 1.33: Actualización de todos los modelos.

#### 1.3.7.3.1. Modelo de Presentación

Como se observa en la Figura 1.34, en el **Modelo De Presentación** (📁 presentation), se crean todos los elementos Web para el sistema.

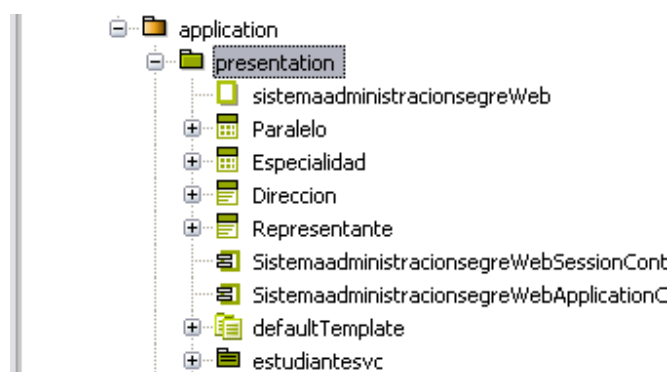


Figura 1.34: Elementos del paquete del Modelo de Presentación.

a) *Elementos capa presentación*

**WEBComponent** 'Componente Web'

Contiene los controles que permiten el manejo del sistema para el usuario.

**WEBPage** 'Paginas Web'

Elementos que son accesibles mediante un navegador web mostrando controles que interactúan con la información solicitada por el usuario.

b) *Aumentar campos en la capa de presentación*

Por defecto *OptimalJ* crea los elementos en la capa de presentación con un formato determinado. En ocasiones el desarrollador necesita mostrar información adicional que el formato de la herramienta no crea. En el caso de la Figura 1.35, se puede observar un listado de solo tres notas, pero es necesaria conocer a que corresponde esta información, por ejemplo a que Materia.

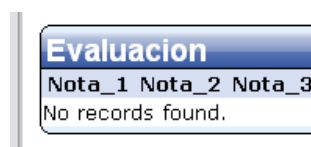


Figura 1.35 Elementos de la página por defecto

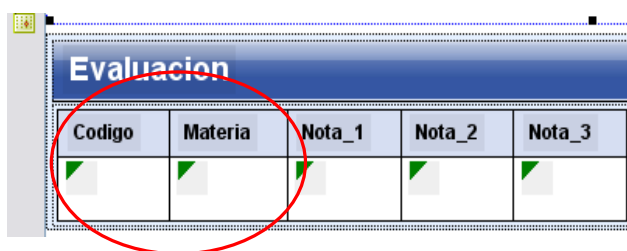
El proceso para aumentar campos en un Componente específico será el siguiente:

1) Expandir un WEBComponent.

Expandir los elementos de una WEBPage y doble clic para visualizar el diseño.

2) Aumentar campos necesarios.

Para el caso de la Figura 1.35 se aumentará dos columnas en la tabla, labels 'etiquetas' que despliegan títulos de 'Código', 'Materia' y los valores correspondientes Figura 1.36.



Evaluación				
Codigo	Materia	Nota_1	Nota_2	Nota_3

Figura 1.36 Página con elementos añadidos

3) Especificar el valor a desplegar de los labels

En la propiedad bindsToPath se enlazará a la clase que contiene los atributos que se necesita que sean desplegados, Figura 1.37.

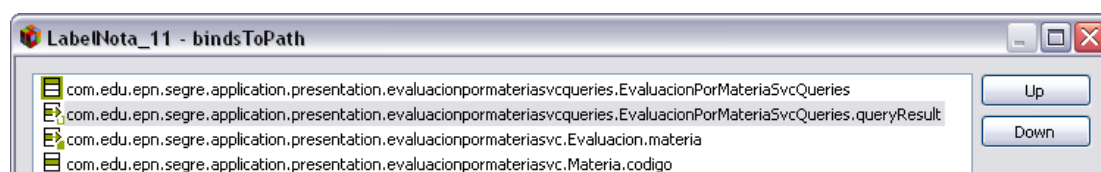


Figura 1.37 Enlaces de atributos en la capa de presentación

c) *Validación de campos*

Para la validación de campos, OptimalJ utiliza el mecanismo estándar de las *Expresiones Regulares*. Estas validaciones se realizarán en la *Capa De*

*Presentación.* Para aplicar un ejemplo sobre este mecanismo, se controlará en un campo que puede ingresar únicamente cadena de caracteres, es decir, no se podrá ingresar ningún tipo de valores como numéricos o símbolos.

Sobre el paquete del modelo de presentación ejecutar clic derecho New -> WEBDataAttributeConstraint.

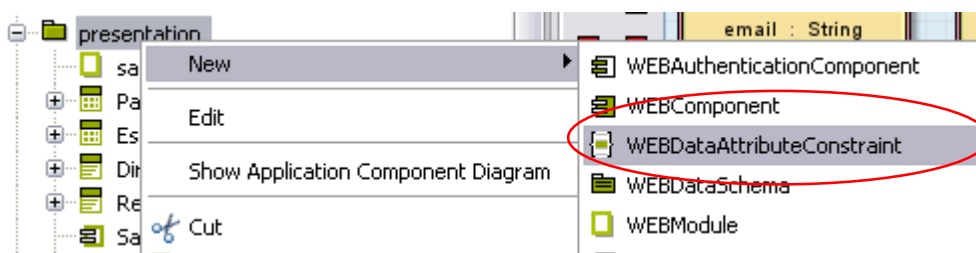


Figura 1.38 Controlador de atributo

Se desplegará un asistente que solicita información para la creación del elemento que permite la validación. En el primer paso solicita un nombre.

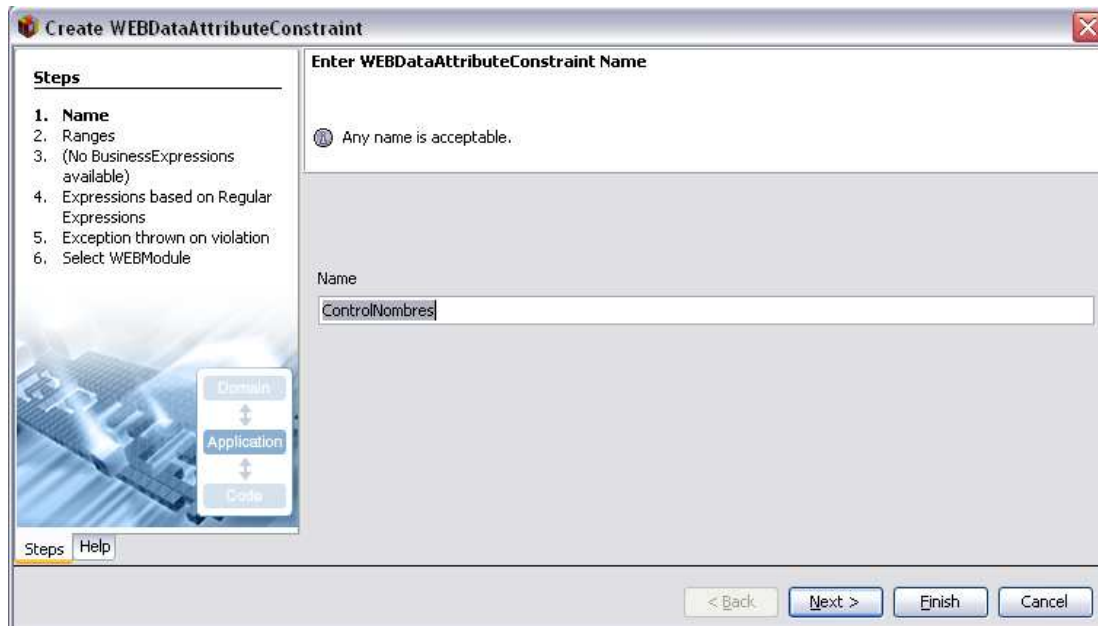


Figura 1.39 Especificación del nombre del controlador



A continuación se seleccionará *Siguiente* hasta el paso cuatro y clic sobre el botón *Añadir*. Se ingresará el *Nombre* y el cuerpo de la *Expresión Regular*. Por último clic en *Finalizar*.

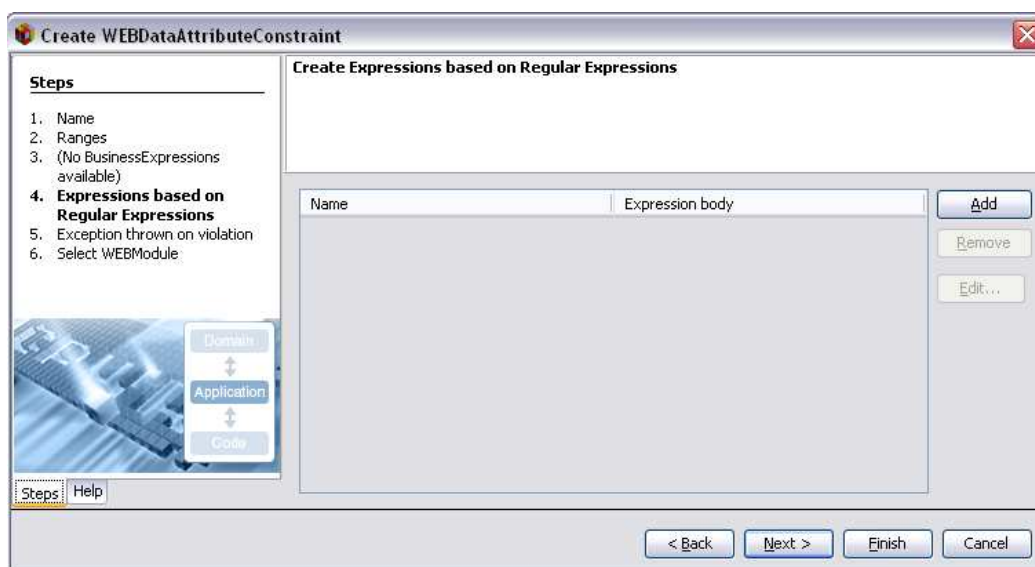


Figura 1.40 Paso para especificar una expresión regular

#### d) *Cambios en Textos de Presentación*

Después de las transformaciones, la herramienta despliega los mensajes, labels 'etiquetas' y textos de los controles en el idioma inglés. Estos textos pueden ser modificados en el archivo *nombreModuloWebResources.properties*, ubicado en el directorio *nombreModuloWebModuleCode/WEB-INF/classes/com/edu/sae/application/presentation*.

Para visualizar el **Modelo De Presentación** se ejecuta doble clic sobre el paquete.

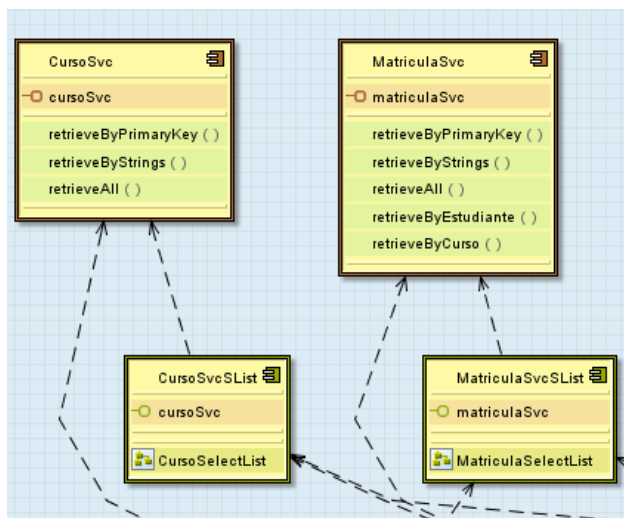



Figura 1.41: Modelo de Presentación.

### 1.3.7.3.2. Modelo del Negocio

El **Modelo Del Negocio** (  ) contiene la lógica del negocio, este es distribuido en los siguientes paquetes:




-  **Facade.-** Contiene los elementos necesarios que proporciona un mecanismo para relacionar la capa de presentación y la de negocio. Para visualizar el modelo, se ejecutará doble clic sobre el paquete.



Figura 1.42: Modelo de Facade.

-  Logic 'Lógico'.- Contiene los componentes lógicos del negocio. OptimalJ por defecto usa *EJBEntityComponents* para generar "Entity Beans" que permite recuperar y manipular datos.
-  Common 'Común'.- Contiene las estructuras de datos que son compartidos por los modelos de aplicación.

### 1.3.7.3.3. Modelo de Datos

Este modelo contiene el diagrama relacional de una base de datos. Para visualizar el diagrama se ejecutará doble clic sobre este paquete.

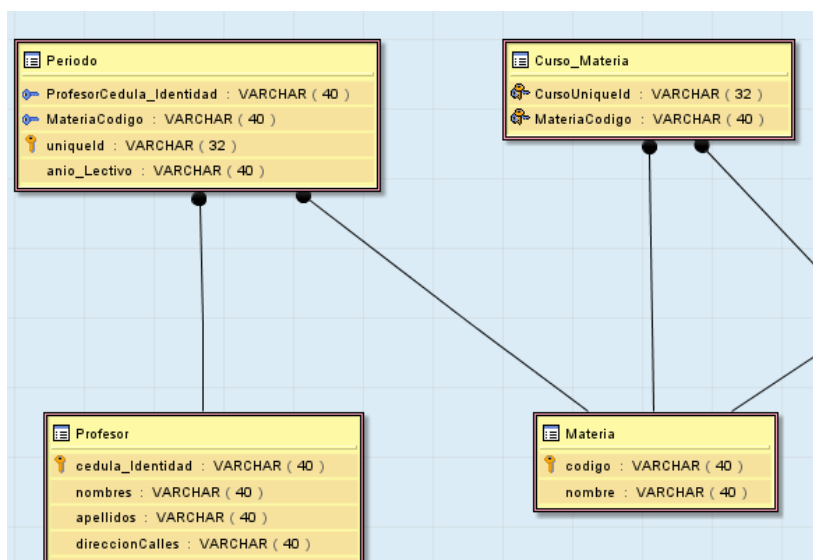


Figura 1.43: Modelo de Datos.

### 1.3.7.4. Generación de Código

Para la generación automática de código en OptimalJ, se la ejecutará en el menú:

Model -> Generate All -> Code

Model -> Generate All -> Build Script

Model -> Generate All -> Archives

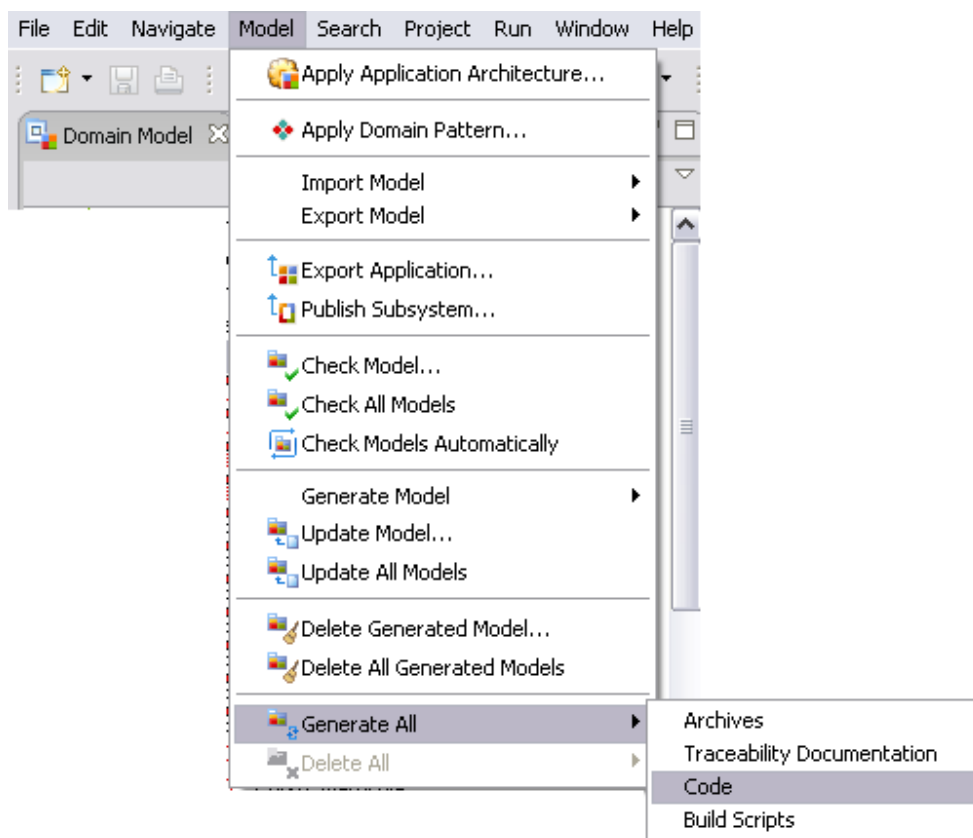


Figura 1.44: Generación de Código, Script y Archivos.

### 1.3.7.5. Creación de Base de Datos

Después de la generación de código, será necesario ejecutar el *script* generado para la base de datos. OptimalJ por defecto trabaja con un motor de Base de Datos llamado Solid.

Para ejecutar el script, se realizará el siguiente proceso:

1. Ejecutar Solid for OptimalJ.
2. En el Modelo de Código, expandir el paquete *nombredelproyectoDbmsModuleCode*. En este paquete se visualiza el archivo *Solid\_MetaNombredelproyectoDbms.sqm*, el cual contiene los datos que permiten la conexión con la base de datos.
3. Ejecutar doble clic sobre el archivo con extensión SQM.

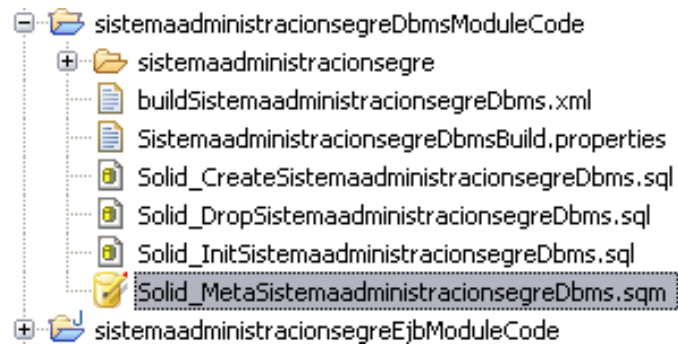


Figura 1.45: Archivo de autenticación a la Base de Datos.

A continuación se aceptarán los valores por defecto que se muestran. Estos valores permitirán la conexión con la base de datos para el sistema.

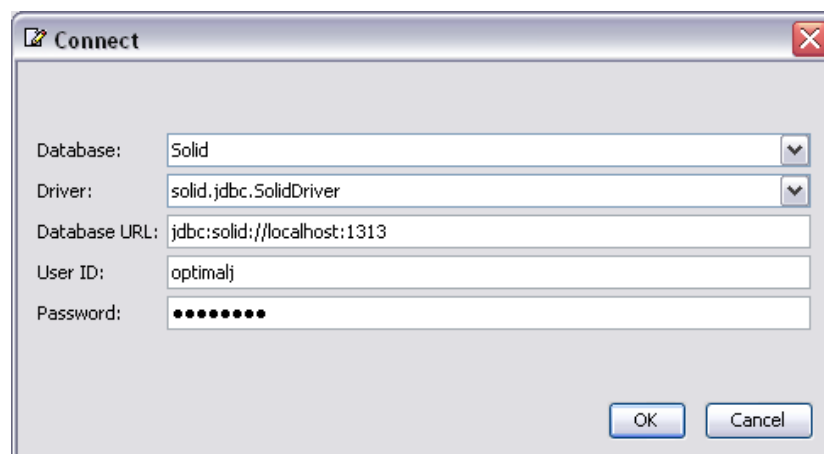
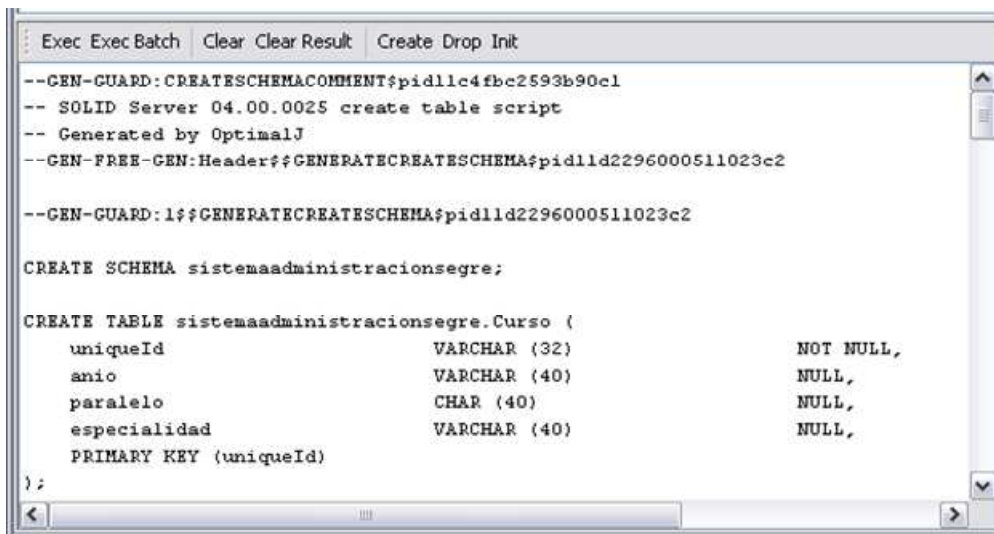


Figura 1.46: Valores de autenticación.

Si los datos solicitados son correctos, se visualizará la consola SQL de Solid. Se ejecutará **Create** para cargar el **script** que fue creado conjuntamente con el código generado.



```

Exec Exec Batch Clear Clear Result Create Drop Init
--GEN-GUARD: CREATESCHEMACOMMENT$pid11c4fbc2593b90c1
-- SOLID Server 04.00.0025 create table script
-- Generated by OptimalJ
--GEN-FREE-GEN: Header$$GENERATECREATESCHEMA$pid11d2296000511023c2
--GEN-GUARD: 1$$GENERATECREATESCHEMA$pid11d2296000511023c2

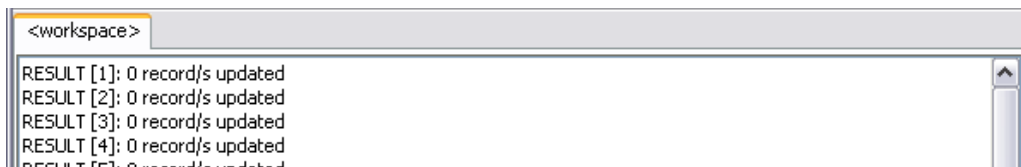
CREATE SCHEMA sistemaadministracionsegre;

CREATE TABLE sistemaadministracionsegre.Curso (
  uniqueId          VARCHAR (32)          NOT NULL,
  anio              VARCHAR (40)          NULL,
  paralelo          CHAR (40)            NULL,
  especialidad      VARCHAR (40)          NULL,
  PRIMARY KEY (uniqueId)
);

```

Figura 1.47: Carga del script para la Base de Datos.

Finalmente, para que el script sea aplicado, será necesario ejecutar **Exec Batch**. Luego de ejecutar el script, en la parte superior de la consola se podrá visualizar el log que desplegará del resultado de la ejecución.



```

<workspace>
RESULT [1]: 0 record/s updated
RESULT [2]: 0 record/s updated
RESULT [3]: 0 record/s updated
RESULT [4]: 0 record/s updated
RESULT [5]: 0 record/s updated

```

Figura 1.48: Log de la ejecución del script.

### 1.3.7.6. Inicialización de la aplicación

Para inicializar la aplicación y entrar en la etapa de pruebas se ejecutará en el menú principal, *Test -> Strat Application Server*.



Figura 1.49: Ejecución de la aplicación.

## **CAPÍTULO 2:**

### **ANÁLISIS Y DISEÑO DEL SISTEMA**

#### **2.1. RECOLECCIÓN DE DATOS**

##### **Desarrollo del Documento de Especificación de Requerimientos**

En esta fase se desarrolla el Documento de Especificación de Requerimientos, para lo cual se aplicará lo descrito en el punto *1.3.6.1. . ANÁLISIS*.



Escuela Politécnica Nacional

Ingeniería de Sistemas Informáticos y  
de Computación

---

# Sistema de Administración de Información Estudiantil para el Colegio Segré

---

Documento de Especificación de  
Requerimientos

Versión 1.0

Desarrollada por:  
Gerardo A. Calero R.  
Aprobada por:  
Ing. Raúl Córdova  
Fecha:



## Historial de Revisiones

---

Nombre	Fecha	Descripción de la modificación	Versión

## **2.1.1. INTRODUCCIÓN**

---

### **2.1.1.1. Ámbito**

El SAE para el colegio Segré, trabajará únicamente para la administración de información estudiantil, proceso crítico para el normal desempeño de las actividades dentro de la Institución. Esta información será manipulada por el personal administrativo y también consultada por los estudiantes del colegio. En consecuencia todos los datos que se manejen en el sistema a desarrollarse, serán relacionados con el estudiante y todo lo referente a la parte académica.

### **2.1.1.2. Definición del sistema**

La funcionalidad del sistema será registrar toda la información del estudiante, permitiendo ejecutar procesos que son necesarios para realizar un seguimiento académico del estudiante.

El sistema debe permitir registrar los cursos y las materias. Además se encargará de ejecutar el proceso de matriculación, el cual consiste en asignar el año lectivo que cursará un alumno.

Luego el sistema debe registrar las evaluaciones que se realizan trimestralmente, además se debe especificar una nota de supletorio y una final que deberá contener el promedio de las notas registradas anteriormente.

El sistema también debe permitir almacenar información personal de los profesores; por otro lado se les asignará las materias que dictará cada uno.

Toda esta información será administrada por una secretaria, y los estudiantes podrán realizar consultas sobre la información registrada, especialmente sobre sus evaluaciones durante el periodo lectivo actual.

### 2.1.1.3. Objetivos Generales

- Agilizar los procesos de administración de la información de los estudiantes.
- Facilitar al personal encargado, el registro de la información estudiantil.
- Obtener un sistema que permita desplegar información de otros períodos lectivos.
- Permitir al estudiante o a su representante hacer un seguimiento sobre la evaluación del período cursado.

### 2.1.1.4. Entorno De Operación

El entorno en el cual va a operar el sistema consta de los siguientes elementos:

Software	Versión	Descripción
JBoss	4.0	Servidor de aplicaciones J2EE de código abierto implementado en Java.
Sun Java SDK	1.5	Máquina virtual independiente del sistema operativo para leer el código compilado.
Internet Explorer	6 ó más	Navegador web que permite la visualización del sistema.
Windows XP Home Edition	2002	Permite la administración eficaz de sus recursos.

Tabla 2.1: Descripción del entorno de Operación

Por políticas de seguridad establecidas en el Colegio Segré, se debe manejar de manera confidencial la descripción del Hardware donde será implantado el Sistema de Administración Estudiantil.

#### 2.1.1.5. Convenciones

- Se utilizará el tipo de letra *Cursiva* para determinar que es una palabra que se especifica en el glosario del presente Proyecto de Titulación.
- Se utilizará **Negrita** en una palabra diferente a un título, para especificar que es una característica que tiene valor.

Ejemplo -> **Modelo:** DELL Inspiron 6400

- Para especificar una nueva funcionalidad se la separará con un salto de línea. Cada sección está precedida por un salto a una nueva página.

#### 2.1.1.6. Material De Referencia

- AMESCUA SECO, A., GARCÍA SÁNCHEZ, L., MARTÍNEZ FERNÁNDEZ, P. y DÍAZ PÉREZ, P., "Ingeniería del Software de Gestión: Análisis y Diseño de Aplicaciones", 1995.
- <http://www.getec.etsit.upm.es/docencia/gproyectos/planificacion/cvida.htm>. CICLO DE VIDA DE SOFTWARE.
- <http://www.biblioteca.co.cr/pdf/unidad12-4.pdf>. MODELO CASCADA

## 2.1.2. REQUERIMIENTO FUNCIONALES DEL SISTEMA DE ADMINISTRACIÓN ESTUDIANTIL

- 1.- Administración de la Información del Estudiante.
- 2.- Matriculación.
- 3.- Administración Información del Profesor.
- 4.- Administración de Cursos.
- 5.- Administración de Materias.
- 6.- Registro de Evaluaciones por Materia.
- 7.- Registro de Evaluaciones por Estudiante.
- 8.- Listado de Estudiantes por Curso.
- 9.- Listado de Evaluaciones de los Estudiantes por Materia.
- 10.- Listado de Evaluaciones de Materias por Estudiante
- 11.- Listado de Profesores por Materia

### 2.1.2.1. Funcionalidad Del Sistema # 1: Administración de Información del Estudiante

Descripción	
El sistema deberá permitir <i>Administrar</i> los datos personales del estudiante.	
Req Num	Requerimiento
Req (01)	El sistema deberá permitir <i>Crear</i> un nuevo estudiante. Un número único será generado por el sistema y a continuación se especificarán los datos personales del estudiante.
Req (02)	El sistema deberá permitir <i>Actualizar</i> los datos del estudiante. El único campo que no podrá ser modificado es el Número Único.
Req (03)	El sistema deberá permitir <i>Buscar y Listar</i> a todos los estudiantes registrados en el sistema, especificando el número único o apellidos del estudiante. Una vez ejecutado este requerimiento, deberá permitir un enlace con el requerimiento <i>Req(02)</i> .

Tabla 2.2: Requerimientos de *Administración de Información Estudiantil*.

### 2.1.2.2. Funcionalidad Del Sistema # 2: *Matriculación*

Descripción	
El sistema deberá permitir <i>Ejecutar</i> el proceso de matrícula para el estudiante, el cual especificará el año lectivo a cursar.	
Precondición	
El sistema deberá tener registrado al estudiante.	
Tratamiento de errores	
No permitirá guardar la matrícula, si no se ha seleccionado a un estudiante.	
Req Num	Requerimiento
Req (04):	El sistema deberá permitir <i>Crear</i> una nueva matrícula.
Req (05):	El sistema deberá permitir <i>Especificar</i> en que año lectivo el estudiante será matriculado.

Tabla 2.3: Requerimientos de *Matriculación*.

### 2.1.2.3. Funcionalidad Del Sistema # 3: **Administración de Información de Profesor**

Descripción	
El sistema deberá permitir <i>Administrar</i> los datos personales del profesor.	
Tratamiento de errores	
No permitirá guardar el registro de un Profesor sin antes especificar la Cédula de Identidad.	
Req Num	Requerimiento
Req (06)	El sistema deberá permitir <i>Crear</i> un nuevo profesor, se

	especificará el número de cédula de identidad, nombres, apellidos, fecha de nacimiento, dirección, ciudad, provincia, e-mail y teléfono.
Req (07)	El sistema deberá permitir <i>Actualizar</i> los datos del profesor. El único campo que no podrá ser modificado es el Número de Cédula de Identidad.
Req (08)	El sistema deberá permitir <i>Buscar</i> y <i>Listar</i> a los profesores registrados en el sistema, mediante el número de cédula ó apellidos del profesor. Una vez que se ha ejecutado este requerimiento, deberá permitir un enlace con el requerimiento <i>Req (07)</i> .

Tabla 2.4: Requerimientos de *Administración de Información de Profesor*.

#### 2.1.2.4. Funcionalidad Del Sistema # 4: Administración de Información de Cursos

Descripción	
El sistema deberá permitir <i>Administrar</i> cursos. La creación de los cursos no estará limitada por el sistema, además no se controlará la disponibilidad de aulas ni horarios.	
Poscondición	
El cupo para la especialidad y el paralelo será limitado, ya que solo existen cuatro paralelos y cinco especialidades.	
Req Num	Requerimiento
Req (09)	El sistema deberá permitir <i>Crear</i> un nuevo curso, especificando el año lectivo, paralelo y especialidad.
Req (10)	El sistema deberá permitir <i>Actualizar</i> los datos del curso.
Req (11)	El sistema deberá permitir <i>Realizar</i> una búsqueda de los cursos mediante el año lectivo, especialidad o paralelo. Una vez que se

	haya ejecutado este requerimiento, el sistema deberá permitir un enlace con el Req (10).
--	--

Tabla 2.5: Requerimientos de *Administración de Información de Cursos*.

### 2.1.2.5. Funcionalidad Del Sistema # 5: Administración de Información de Materias

Descripción	
El sistema deberá permitir <i>Administrar</i> información de materias. El sistema no controlará la disponibilidad de horarios, ni de profesores.	
Precondición	
El sistema deberá tener registrado los cursos.	
Tratamiento de errores	
No permitirá guardar la materia sin antes especificar el curso.	
Req Num	Requerimiento
Req (12)	El sistema deberá permitir <i>Crear</i> una nueva materia, que especificará el código y nombre de la misma.
Req (13)	El sistema deberá permitir <i>Actualizar</i> los datos de la materia.
Req (14)	El sistema deberá permitir <i>Buscar</i> las materias, mediante el código o el nombre de la misma. Una vez que se haya ejecutado este requerimiento, el sistema deberá permitir un enlace con el Req (13).

Tabla 2.6: Requerimientos de *Administración de Información de Materias*.



### 2.1.2.6. Funcionalidad Del Sistema # 6: Registro de Evaluaciones por Materia

Descripción	
El sistema deberá permitir <i>Administrar</i> las evaluaciones trimestrales de los estudiantes por materia, así como también las de supletorio y las del promedio general.	
Poscondición	
Para la asignación de evaluaciones, se deberá seleccionar la materia y posteriormente el estudiante.	
Req Num	Requerimiento
Req (15)	El sistema deberá permitir <i>Guardar</i> las evaluaciones del estudiante de cada materia, especificando el tipo a registrar.

Tabla 2.7: Requerimientos de *Registro de Evaluaciones*.

### 2.1.2.7. Funcionalidad Del Sistema # 7: Registro de Evaluaciones por Estudiante

Descripción	
El sistema deberá permitir <i>Administrar</i> las evaluaciones trimestrales de las materias por estudiante.	
Poscondición	
Para la asignación de evaluaciones, se deberá seleccionar el estudiante y posteriormente la materia.	
Req Num	Requerimiento
Req (15)	El sistema deberá permitir <i>Guardar</i> las evaluaciones de la materia de cada estudiante, especificando el tipo a registrar.

Tabla 2.8: Requerimientos de *Registro de Evaluaciones*.

### 2.1.2.8. Funcionalidad Del Sistema # 8: Listado de Estudiantes por Curso

Descripción	
El sistema deberá permitir <i>Mostrar</i> un listado de estudiantes de un curso determinado.	
Req Num	Requerimiento
Req (16)	El sistema deberá permitir <i>Listar</i> los estudiantes de un curso, especificando el curso, especialidad y paralelo.

Tabla 2.9: Requerimientos de *Listado de Estudiantes por Curso*.

### 2.1.2.9. Funcionalidad Del Sistema # 9: Listado de Notas de los Estudiantes por Materia

Descripción	
El sistema deberá permitir <i>Mostrar</i> un listado de los estudiantes con el resultado de las evaluaciones por materia.	
Precondición	
Se deberá especificar el curso, paralelo, especialidad y materia.	
Req Num	Requerimiento
Req (17)	El sistema deberá permitir <i>Desplegar</i> un listado de los estudiantes con las respectivas notas de cada materia.

Tabla 2.10: Requerimientos de *Listado de Notas de los Estudiantes por Materia*.

### 2.1.2.10. Funcionalidad Del Sistema # 10: Listado de Notas de Materias por Estudiante

Descripción	
El sistema deberá permitir <i>Mostrar</i> un listado de las materias con el resultado de las evaluaciones de cada uno de los estudiantes.	
Precondición	
Se especificará el Número Único o apellidos del estudiante.	
Req Num	Requerimiento
Req (18)	El sistema deberá permitir <i>Desplegar</i> un listado de las materias con las notas de cada estudiante.

Tabla 2.11: Requerimientos de *Listado de Notas de Materias por Estudiante*.

### 2.1.2.11. Funcionalidad Del Sistema # 11: Listado de Profesores por Materia

Descripción	
El sistema deberá permitir <i>Visualizar</i> un listado de los profesores por materia.	
Precondición	
Se deberá especificar el código o el nombre de la materia.	
Req Num	Requerimiento
Req (19)	El sistema deberá permitir <i>Visualizar</i> un listado de profesores por materia.

Tabla 2.12: Requerimientos de *Listado de Profesores por Materia*.

### 2.1.3. REQUISITOS DE INTERFAZ

---

#### 2.1.3.1. Interfaces De Usuario

##### 2.1.3.1.1. *Interface Inicial*

Cada una de las funcionalidades especificadas en la sección 2, deberán estar listadas en la interface inicial. Para facilidad de los usuarios cada funcionalidad se mostrará en forma de links “vínculos”, que serán identificados con un nombre que describa su uso en el Sistema. Esta es una característica que se maneja en los Sistemas Web.

#### **Administración de la Información del Estudiante:**

Esta funcionalidad permitirá al sistema *Crear* un nuevo estudiante, solicitando información como: nombres, apellidos, fecha de nacimiento, foto y dirección donde constará la ciudad, provincia, e-mail y teléfono, además deberá incluir los datos del representante, que son número de cédula, nombres, apellidos, dirección en el cuál también especificará la ciudad, provincia, e-mail y teléfono.

Para realizar la consulta, el sistema deberá mostrar un método de búsqueda, donde se especificará el número único o los apellidos del estudiante.

Para realizar la actualización de datos, previamente se deberá buscar al estudiante que se desea modificar, luego se cambiarán los datos desplegados a excepción del número único, que será un campo que se encuentra bloqueado para evitar los cambios.

**Matriculación:**

Primero se especificará un nuevo registro, a continuación se ingresará el período en el cuál se va a ejecutar la matrícula. Finalmente se realizará la búsqueda de un estudiante registrado.

**Administración de Información del Profesor:**

En esta funcionalidad el sistema permitirá crear un nuevo profesor, solicitando información como son: número de cédula, nombres, apellidos, fecha de nacimiento, dirección, ciudad, provincia, e-mail y teléfono.

Para realizar la búsqueda, el sistema deberá mostrar un método de consulta, donde se especificará el número de cédula o los apellidos del profesor.

La actualización de datos, se realizará después de haber echo la búsqueda de un determinado profesor y ejecutar un método de actualización, a continuación se podrán cambiar los datos desplegados a excepción del número de cédula, que será un campo bloqueado.

**Registro de Cursos:**

Permitirá registrar cursos ejecutando el método de creación de un nuevo curso, donde se especificará el año, especialidad y paralelo. Para la especificación de especialidad y paralelo se lo realizará mediante valores ya establecidos, estos son:

*Especialidades*

- Físico Matemático
- Contabilidad
- Electrónica
- Informática

- Ciclo Básico

#### *Paralelos*

- A
- B
- C
- D

Para realizar la búsqueda se lo hará mediante un método de consulta donde se especificará el año lectivo.

La actualización se realizará especificando el curso, luego de haber ejecutado el método de búsqueda y podrá ser modificado la especialidad y/o el paralelo.

#### **Registro de Materias:**

Permitirá registrar materias ejecutando el método de creación de un nuevo curso, donde se especificará el código y el nombre de la materia.

Para realizar la búsqueda, el sistema deberá mostrar un método de consulta, donde se especificará el código o el nombre de la materia.

La actualización de datos, se ejecutará después de haber echo la búsqueda de una materia y de ejecutar un método de actualización, luego se podrán cambiar los datos de la materia.

#### **Registro de Evaluaciones por Materia:**

Se ejecutará el método de creación de un nuevo registro, donde se ingresará el tipo de evaluación, el valor, la materia y el estudiante. Para especificar la materia

y el estudiante se realizará una búsqueda mediante el código o el nombre de la materia y el número único o los apellidos del estudiante.

#### **Registro de Evaluaciones por Estudiante:**

Se ejecutará el método de creación de un nuevo registro, donde se ingresará el tipo de nota, el valor, la materia y el estudiante. Para especificar la materia y el estudiante se realizará una búsqueda mediante el código o los apellidos del estudiante y el código o nombre de la materia.

#### **Listado de Estudiantes por Curso:**

Para generar el reporte se especificará el año, la especialidad y el paralelo, como resultado se desplegará un listado de todos los estudiantes matriculados en el curso.

#### **Listado de Evaluaciones de los Estudiantes por Materia:**

Para generar el reporte se especificará el año, la especialidad, el paralelo y el código o nombre de la materia, como resultado se desplegará un listado de todos los estudiantes registrados en la materia con sus respectivas evaluaciones.

#### **Listado de Evaluaciones de Materias por Estudiante:**

Para generar el reporte se especificará el año, la especialidad, el paralelo y el número único o los apellidos del estudiante, como resultado se desplegará un listado de materias con cada una de las evaluaciones registradas para el estudiante especificado.

### Listado de Profesores por Materia:

Para generar el reporte se especificará el período y el código o el nombre de la materia, como resultado se desplegará un listado de todos los profesores que dicten la materia especificada.

#### 2.1.3.1.2. Acceso a las Funcionalidades

El acceso a cada una de las funcionalidades dependerá de los privilegios que posea el usuario para la Administración de Información. Solo la secretaria del colegio es quien administra la información y tiene acceso a todas las funcionalidades del sistema. Además los estudiantes del colegio tendrán acceso al sistema, únicamente a los listados de sus propios datos.

#### 2.1.3.1.3. Despliegue de Errores

En caso de que exista un error en los accesos o administración de información, se mostrará en la parte inferior de la pantalla un mensaje de error en el proceso acompañado de una pequeña descripción para la solución de éste.

### 2.1.3.2. Interfaces De Software

Para la correcta funcionalidad del sistema, será necesario utilizar las siguientes Herramientas de Software:

Software	Descripción
Windows XP	Sistema Operativo en el que se desarrollará la aplicación.
jdk-1_5_0 para Windows	Java Development Kit permitirá la ejecución de las clases para la funcionalidad del sistema.



Solid for OptimalJ PE 4.3 built on Eclipse	Repositorio donde se almacenarán los datos que envíe la aplicación.
Internet Explorer (recomendado)	Permite el acceso o la visualización de la aplicación.

Tabla 2.2 Especificación de SW necesario.

## 2.2. ANÁLISIS

### 2.2.1. CASOS DE USO

#### 2.2.1.1. Diagrama de Casos de Uso

Para el desarrollo del Sistema de Administración de Información Estudiantil para el Colegio Segré se obtuvo los siguientes paquetes.

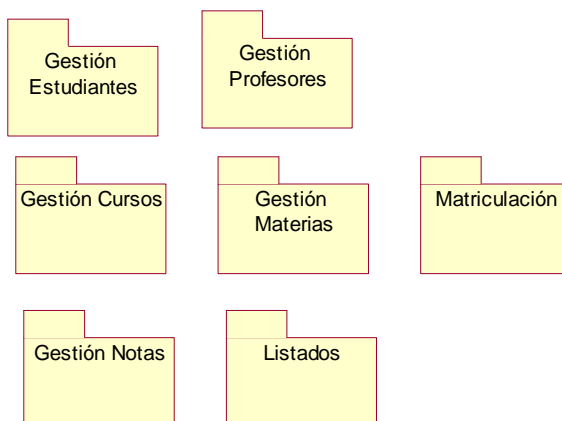


Figura 2.1: Paquetes para el Sistema de Administración

El paquete Gestión Estudiantes describe el siguiente modelo de Casos de Uso.

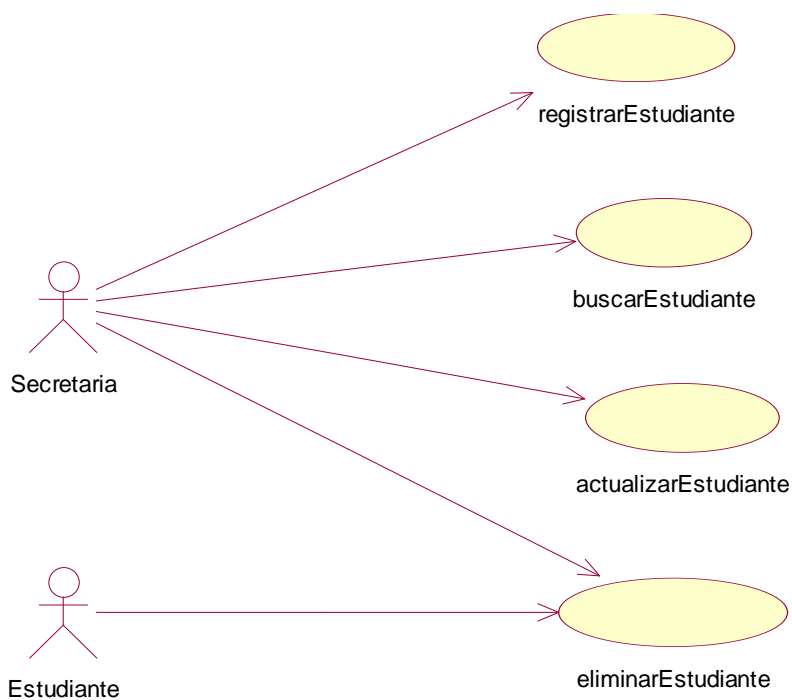


Figura 2.2: Paquete *Gestión Estudiantes*

El paquete Gestión Profesores describe el siguiente modelo de Casos de Uso.

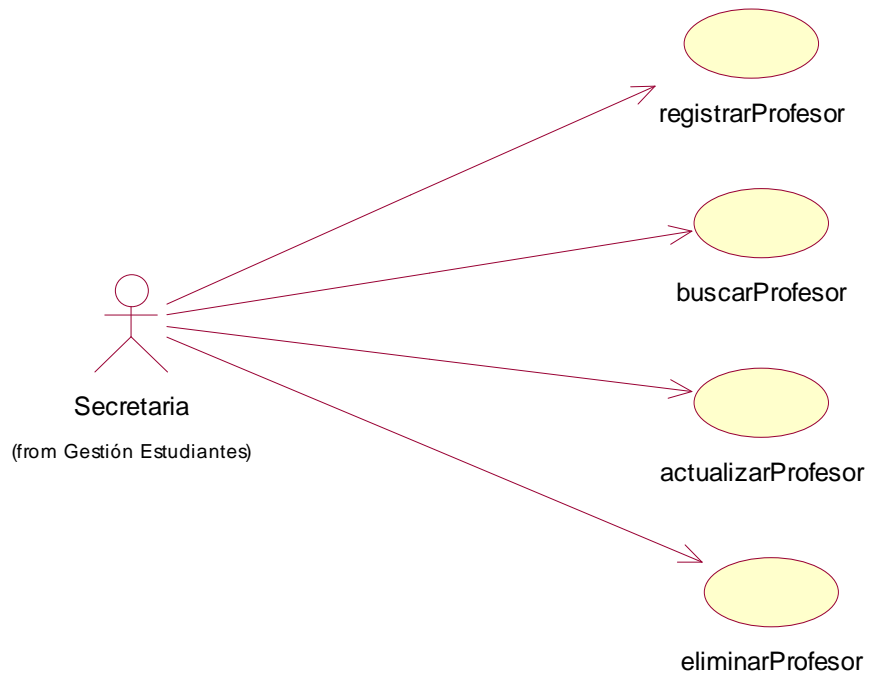


Figura 2.3: Paquete *Gestión Profesores*

El paquete Gestión Cursos describe el siguiente modelo de Casos de Uso.

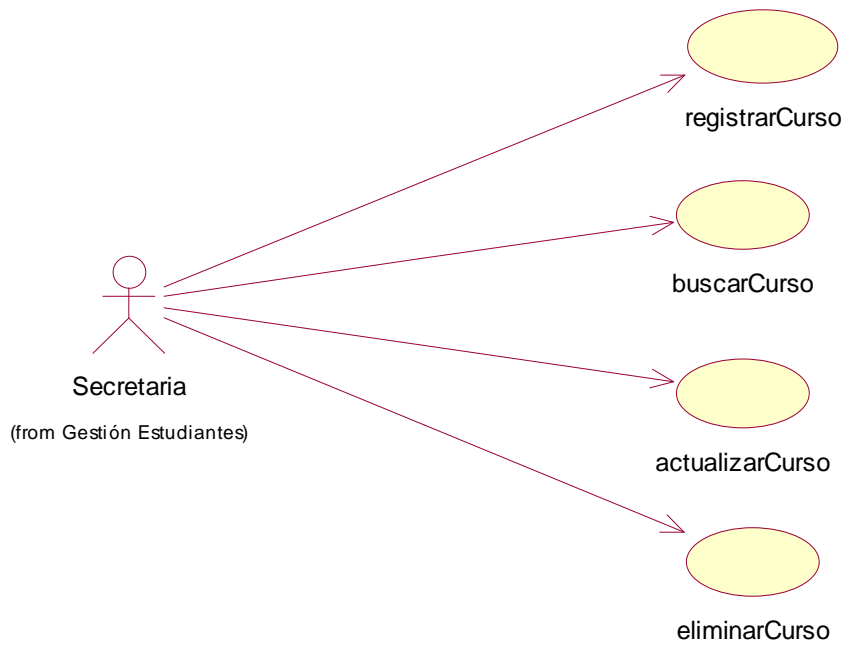


Figura 2.4: Paquete *Gestión Cursos*

El paquete Gestión Materias describe el siguiente modelo de Casos de Uso.

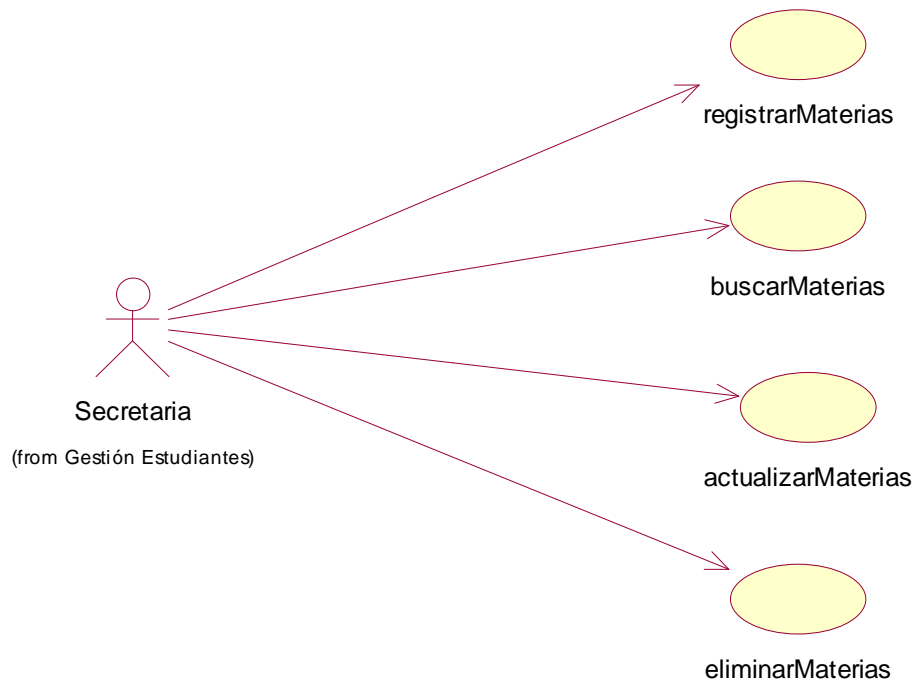


Figura 2.5: Paquete *Gestión Materias*

El paquete Gestión Evaluaciones describe el siguiente modelo de Casos de Uso.

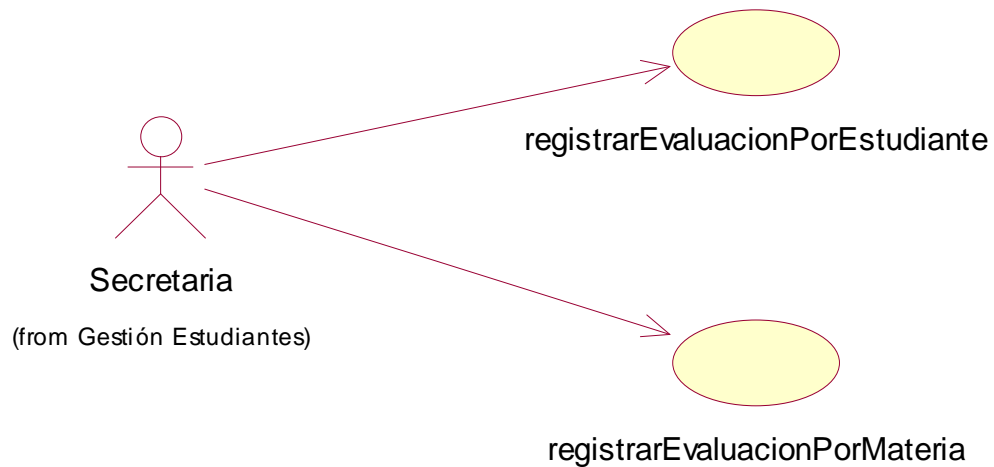


Figura 2.6: Paquete *Gestión Evaluaciones*

El paquete Matriculación describe el siguiente modelo de Casos de Uso.



Figura 2.7: Paquete *Matriculación*

El paquete Listados describe el siguiente modelo de Casos de Uso.

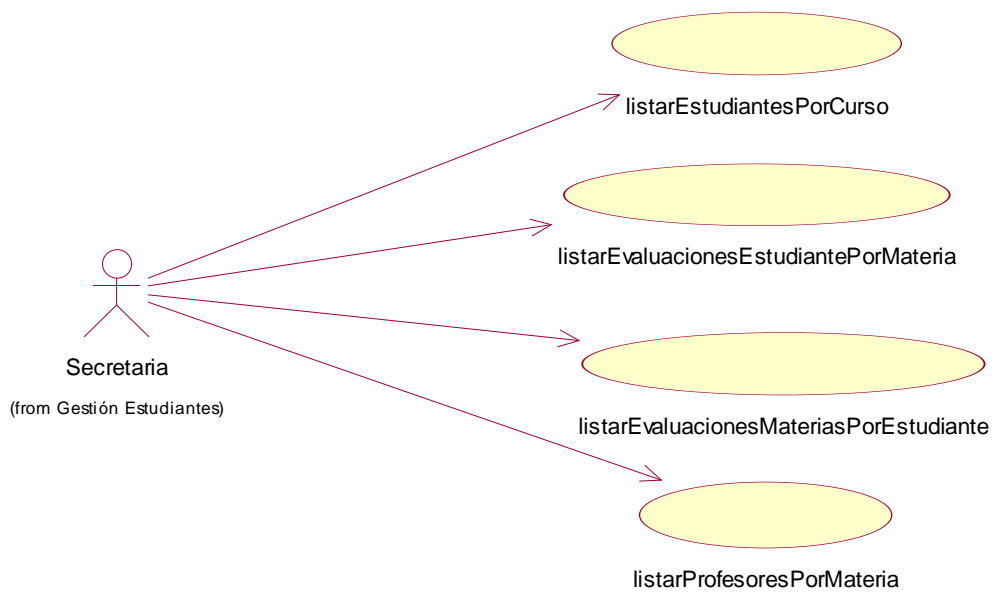


Figura 2.8: Paquete *Listados*

## 2.2.1.2. Detalle de Casos de Uso

### 2.2.1.2.1. Paquete Gestión Estudiante

#### Insertar Estudiante

<b>Nombre</b>	
Insertar <b>Estudiante</b>	
<b>Pre-condición</b>	
Existe un <b>Estudiante</b> a ser ingresado.	
<b>Post-condición</b>	
El <b>Estudiante</b> queda registrado en el sistema.	
<b>Descripción</b>	
Realiza la inserción de un <b>Estudiante</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Ingresar datos estudiante: <b>nombres, apellidos, fecha de nacimiento, foto, dirección (calles, ciudad, provincia, e-mail y número telefónico)</b>, datos representante: <b>nombre, apellidos, dirección (calles, ciudad, provincia, e-mail y número telefónico)</b>.</li> <li>2. Ejecuta función <i>Guardar</i></li> </ol>	<ol style="list-style-type: none"> <li>3. Verifica corrección de los atributos ingresados.               <ol style="list-style-type: none"> <li>3.1. El sistema informa que al menos uno de los atributos ingresado no es correcto.</li> </ol> </li> <li>4. Realiza el alta del <b>Estudiante</b>.</li> </ol>

Tabla 2.13: Caso de Uso *Insertar Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

**Buscar Estudiante**

<b>Nombre</b>	
Buscar <b>Estudiante</b> .	
<b>Pre-condición</b>	
Existe un <b>Estudiante</b> a ser buscado en el sistema.	
<b>Pos-condición</b>	
El <b>Estudiante</b> es encontrado, o no existía el <b>Estudiante</b> en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de un <b>Estudiante</b> en el sistema, controla que exista el <b>Estudiante</b> en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Selecciona el criterio de búsqueda, por <b>número único</b> o <b>apellidos</b>.</li> <li>2. Ingresa atributos adecuados al criterio de búsqueda <b>número único</b> o <b>apellidos</b>.</li>   <li>5. Confirma que el <b>Estudiante</b> encontrado es el buscado. <ol style="list-style-type: none"> <li>5.1. El <b>Estudiante</b> buscado no esta registrado en el sistema.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza la búsqueda por <i>atributos</i> según criterio de búsqueda, <b>número único</b> o <b>apellidos</b>.</li> <li>4. Despliega el <b>Estudiante</b> especificado según el criterio de búsqueda, por <b>número único</b> o <b>apellidos</b>.</li> </ol>

Tabla 2.14: Caso de Uso *Buscar Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

Modificar **Estudiante**

<b>Nombre</b>	
Modificar <b>Estudiante</b> .	
<b>Pre-condición</b>	
Hay <i>atributos</i> de un <b>Estudiante</b> que deben ser modificados.	
<b>Post-condición</b>	
El <b>Estudiante</b> modificado queda registrado en el sistema, o el <b>Estudiante</b> a modificar no existía en el sistema.	
<b>Descripción</b>	
Realiza la modificación de los <i>atributos</i> de un <b>Estudiante</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
<p>1. Ejecuta caso de uso <i>buscar Estudiante</i>.</p> <p>4. Modifica los <i>atributos</i> del <b>Estudiante</b>.</p>	<p>2. Incluye (<b>buscar[Estudiante]</b>).</p> <p>3. Muestra el <b>número único, nombres y apellidos</b> del <b>Estudiante</b> seleccionado.</p> <p>5. Verifica corrección de <i>atributos</i> modificados. 5.1. El sistema informa que al menos uno de los <i>atributos</i> ingresado no es correcto.</p> <p>6. Realiza la actualización del <b>Estudiante</b>.</p>

Tabla 2.15: Caso de Uso *Modificar Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.



Eliminar **Estudiante**

<b>Nombre</b>	
Eliminar <b>Estudiante</b> .	
<b>Pre-condición</b>	
Existe un <b>Estudiante</b> a ser eliminado.	
<b>Post-condición</b>	
El <b>Estudiante</b> fue eliminado del sistema, o el <b>Estudiante</b> a eliminar no existía en el sistema.	
<b>Descripción</b>	
Realiza la eliminación de un <b>Estudiante</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecuta caso de uso <i>buscar <b>Estudiante</b></i> .	2. Include( <b>buscar[Estudiante]</b> ).
4. Confirma la eliminación del <b>Estudiante</b> .	3. Muestra el <b>número único, nombres y apellidos</b> del <b>Estudiante</b> seleccionado.
	5. Realiza la eliminación del <b>Estudiante</b> .

Tabla 2.16: Caso de Uso *Modificar Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

## 2.2.1.2.2. Paquete Gestión Profesor

Insertar Profesor

<b>Nombre</b>	
Insertar <b>Profesor</b> .	
<b>Pre-condición</b>	
Existe un <b>Profesor</b> a ser ingresado.	
<b>Post-condición</b>	
El <b>Profesor</b> queda registrado en el sistema.	
<b>Descripción</b>	
Realiza la inserción de un <b>Profesor</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Ingresar datos de <b>Profesor: cédula, nombres, apellidos, fecha de nacimiento, foto y dirección (calles, ciudad, provincia, e-mail y número telefónico).</b></li> <li>2. Ejecuta función <i>Guardar</i></li> </ol>	<ol style="list-style-type: none"> <li>3. Verifica corrección de los atributos ingresados. <ol style="list-style-type: none"> <li>3.1. El sistema informa que al menos uno de los atributos ingresado no es correcto.</li> </ol> </li> <li>4. Realiza el alta del <b>Profesor</b>.</li> </ol>

Tabla 2.17: Caso de Uso *Insertar Profesor*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

**Buscar Profesor**

<b>Nombre</b>	
Búsqueda de <b>Profesor</b>	
<b>Pre-condición</b>	
Existe un <b>Profesor</b> a ser buscado en el sistema.	
Pos-condición	
El <b>Profesor</b> es encontrado, o no existía el <b>Profesor</b> en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de un <b>Profesor</b> en el sistema, controla que exista el <b>Profesor</b> en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Selecciona el criterio de búsqueda, por <b>número de cédula</b> o <b>apellidos</b>.</li> <li>2. Ingresa atributos adecuados al criterio de búsqueda <b>número de cédula</b> o <b>apellidos</b>.</li> <li>5. Confirma que el <b>Profesor</b> encontrado es el buscado. <ol style="list-style-type: none"> <li>5.1. El <b>Profesor</b> buscado no esta registrado en el sistema.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza la búsqueda por <i>atributos</i> según criterio de búsqueda, <b>número cédula</b> o <b>apellidos</b>.</li> <li>4. Despliega el <b>Profesor</b> especificado según el criterio de búsqueda, por <b>número cédula</b> o <b>apellidos</b>.</li> </ol>

Tabla 2.18: Caso de Uso *Buscar Profesor*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

Modificar Profesor

<b>Nombre</b>	
Modificar <b>Profesor</b> .	
<b>Pre-condición</b>	
Hay <i>atributos</i> de un <b>Profesor</b> que deben ser modificados.	
<b>Post-condición</b>	
El <b>Profesor</b> modificado queda registrado en el sistema, o el <b>Profesor</b> a modificar no existía en el sistema.	
<b>Descripción</b>	
Realiza la modificación de los <i>atributos</i> de un <b>Profesor</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecuta caso de uso <i>buscar Profesor</i> .  4. Modifica los <i>atributos</i> del <b>Profesor</b> .	2. Incluye ( <b>buscar[Profesor]</b> ). 3. Muestra el <b>número de cédula, nombres y apellidos</b> del <b>Profesor</b> seleccionado.  5. Verifica corrección de <i>atributos</i> modificados. 5.1. El sistema informa que al menos uno de los <i>atributos</i> ingresados no es correcto. 6. Realiza la actualización del <b>Profesor</b> .

Tabla 2.19: Caso de Uso *Modificar Profesor*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

Eliminar **Profesor**

<b>Nombre</b>	
Eliminar <b>Profesor</b> .	
<b>Pre-condición</b>	
Existe un <b>Profesor</b> a ser eliminado.	
<b>Post-condición</b>	
El <b>Profesor</b> fue eliminado del sistema, o el <b>Profesor</b> a eliminar no existía en el sistema.	
<b>Descripción</b>	
Realiza la eliminación de un <b>Profesor</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecuta caso de uso <i>buscar Profesor</i> .	2. Include( <b>buscar[Profesor]</b> ).
4. Confirma la eliminación del <b>Profesor</b> . <input type="checkbox"/>	3. Muestra el <b>número de cédula, nombres y apellidos</b> del <b>Profesor</b> seleccionado.
	5. Realiza la eliminación del <b>Profesor</b> .

Tabla 2.20: Caso de Uso *Eliminar Profesor*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

## 2.2.1.2.3. Paquete Gestión Curso

Insertar Curso

<b>Nombre</b>	
Insertar <b>Curso</b> .	
<b>Pre-condición</b>	
Existe un <b>Curso</b> a ser ingresado.	
<b>Post-condición</b>	
El <b>Curso</b> queda registrado en el sistema, o el <b>Curso</b> ya estaba registrado en el sistema.	
<b>Descripción</b>	
Realiza la inserción de un <b>Curso</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ingresa los <b>año, paralelo</b> y <b>especialidad</b> del <b>Curso</b> .	2. Verifica corrección de <b>año, paralelo</b> y <b>especialidad</b> ingresados. 2.1. El sistema informa que al menos uno de los <i>atributos</i> ingresado no es correcto.  3. Realiza el alta del <b>Curso</b> .

Tabla 2.21: Caso de Uso *Insertar Curso*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

Buscar Curso

<b>Nombre</b>	
Buscar <b>Curso</b>	
<b>Pre-condición</b>	
Existe un <b>Curso</b> a ser buscado en el sistema.	
<b>Pos-condición</b>	
El <b>Curso</b> es encontrado, o no existía el <b>Curso</b> en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de un <b>Curso</b> en el sistema, controla que exista el <b>Curso</b> en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Selecciona el criterio de búsqueda, por <b>año</b> y <b>paralelo</b>.</li> <li>2. Ingresa atributos adecuados al criterio de búsqueda <b>año</b> y <b>paralelo</b>.</li> <li>5. Confirma que el <b>Curso</b> encontrado es el buscado. <ol style="list-style-type: none"> <li>5.1. El <b>Curso</b> buscado no esta registrado en el sistema.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza la búsqueda por <i>atributos</i> según criterio de búsqueda, <b>año</b> y <b>paralelo</b>.</li> <li>4. Despliega el <b>Curso</b> especificado según el criterio de búsqueda, por <b>año</b> y <b>paralelo</b>.</li> </ol>

Tabla 2.22: Caso de Uso *Buscar Curso*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

Modificar Curso

<b>Nombre</b>	
Modificar <b>Curso</b> .	
<b>Pre-condición</b>	
Hay un <b>año, paralelo o especialidad</b> de un <b>Curso</b> que deben ser modificados.	
<b>Post-condición</b>	
El <b>Curso</b> modificado queda registrado en el sistema, o el <b>Curso</b> a modificar no existía en el sistema.	
<b>Descripción</b>	
Realiza la modificación de los <b>año, paralelo o especialidad</b> de un <b>Curso</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecuta caso de uso <i>buscar Curso</i> .	2. Incluye ( <b>buscar[Curso]</b> ).
4. Modifica los <i>atributos</i> del <b>Curso</b> .	3. Muestra <b>año, paralelo y especialidad</b> del <b>Curso</b> seleccionado.
	5. Verifica corrección de <i>atributos</i> modificados. 5.1. El sistema informa que al menos uno de los <i>atributos</i> ingresados no es correcto.
	6. Realiza la actualización del <b>Curso</b> .

Tabla 2.23: Caso de Uso *Modificar Curso*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.



Eliminar **Curso**

<b>Nombre</b>	
Eliminación de <b>Curso</b> .	
<b>Pre-condición</b>	
Existe un <b>Curso</b> a ser eliminado.	
<b>Post-condición</b>	
El <b>Curso</b> fue eliminado del sistema, o el <b>Curso</b> a eliminar no existía en el sistema.	
<b>Descripción</b>	
Realiza la eliminación de un <b>Curso</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecuta caso de uso <i>buscar <b>Curso</b></i> .  4. Confirma la eliminación del <b>Curso</b> . <input type="checkbox"/>	2. Include( <b>buscar[Curso]</b> ). 3. Muestra el <b>año</b> , <b>paralelo</b> y <b>especialidad</b> del <b>Curso</b> seleccionado.  5. Realiza la eliminación del <b>Curso</b> .

Tabla 2.24: Caso de Uso *Eliminar Curso*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

## 2.2.1.2.4. Paquete Gestión Materia

Insertar Materia

<b>Nombre</b>	
Insertar <b>Materia</b> .	
<b>Pre-condición</b>	
Existe una <b>Materia</b> a ser ingresado.	
<b>Post-condición</b>	
La <b>Materia</b> queda registrada en el sistema, o la <b>Materia</b> ya estaba registrada en el sistema.	
<b>Descripción</b>	
Realiza la inserción de una <b>Materia</b> , controlando la existencia del elemento en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Ingresar los <b>código, nombre</b> de la <b>Materia</b>.</li> <li>2. Asocia la <b>Materia</b> a <i>Curso</i>. <ol style="list-style-type: none"> <li>2.1. Ejecuta caso de uso Buscar <i>Curso</i>.</li> <li>2.2. Seleccionar <i>Curso</i>.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Verifica existencia por <b>código</b>. <ol style="list-style-type: none"> <li>3.1. El sistema informa de la existencia de la <b>Materia</b> identificado con <b>código</b>.</li> </ol> </li> <li>4. Verifica corrección de <b>código</b> y <b>nombre</b> ingresados. <ol style="list-style-type: none"> <li>4.1. El sistema informa que al menos uno de los <i>atributos</i> ingresados no es correcto.</li> </ol> </li> <li>5. Verifica la asociación con <i>Curso</i>. <ol style="list-style-type: none"> <li>5.1. El sistema informa que <i>la materia no ha sido asociada con el Curso</i>.</li> </ol> </li> <li>6. Realiza el alta de <b>Materia</b>.</li> </ol>

Tabla 2.25: Caso de Uso *Insertar Materia*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

Buscar **Materia**

<b>Nombre</b>	
Búscar <b>Materia</b>	
<b>Pre-condición</b>	
Existe una <b>Materia</b> a ser buscado en el sistema.	
<b>Pos-condición</b>	
La <b>Materia</b> es encontrada, o no existía la <b>Materia</b> en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de una <b>Materia</b> en el sistema, controla que exista la <b>Materia</b> en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Selecciona <b>código</b> o <b>nombre</b> como criterio de búsqueda.</li> <li>2. Ingresa <b>código</b> o <b>nombre</b>, adecuados al criterio de búsqueda.</li> <li>4. Confirma que la <b>Materia</b> encontrada es la buscada. <ol style="list-style-type: none"> <li>4.1. El <b>Materia</b> buscada no esta registrada en el sistema.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza la búsqueda por <b>código</b> o <b>nombre</b> según el criterio de búsqueda.</li> </ol>

Tabla 2.26: Caso de Uso *Buscar Materia*.

Nota: En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

Modificar **Materia**

<b>Nombre</b>	
Modificar <b>Materia</b> .	
<b>Pre-condición</b>	
Hay <b>nombres</b> de una <b>Materia</b> que deben ser modificados.	
<b>Post-condición</b>	
La <b>Materia</b> modificada queda registrada en el sistema, o la <b>Materia</b> a modificar no existía en el sistema.	
<b>Descripción</b>	
Realiza la modificación de los <b>nombres</b> de una <b>Materia</b> , controlando el cumplimiento de la relación con <i>Materia</i> .	
<b>ACTOR</b>	<b>SISTEMA</b>
<p>1. Ejecuta caso de uso <i>buscar <b>Materia</b></i>.</p> <p>4. Modifica el <b>año, paralelo y especialidad</b> de la <b>Materia</b>.</p>	<p>2. Incluye (<b>buscar[Materia]</b>).</p> <p>3. Muestra el <b>año, paralelo y especialidad</b> de la <b>Materia</b> seleccionada.</p> <p>5. Verifica corrección de <b>años, paralelo y especialidad</b> ingresados. 5.1. El sistema informa que al menos uno de los <b>años, paralelos o especialidades</b> ingresado no es correcto.</p> <p>6. Verifica que <b>Materia</b> tenga relación con <i>Curso</i>. 6.1. El sistema informa la existencia de la relación con <i>Curso</i>.</p> <p>7. Realiza la actualización de la <b>Materia</b>.</p>

Tabla 2.27: Caso de Uso *Modificar Materia*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

2.2.1.2.5. *Paquete Matricular Estudiante***Matricular Estudiante**

<b>Nombre</b>	
Matricular Estudiante.	
<b>Pre-condición</b>	
Existe un <b>Estudiante</b> registrado en el sistema.	
<b>Post-condición</b>	
La <b>Matricula</b> queda registrada en el sistema.	
<b>Descripción</b>	
Realiza el registro de una <b>Matricula</b> en cada período estudiantil.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ingresar <b>año</b> lectivo para <b>Matricular Estudiante</b> .	2. Verifica corrección de <b>año</b> lectivo ingresado. 2.1.El sistema informa que el <b>año</b> lectivo ingresado no es correcto.  3. Ejecuta caso de uso <b>Buscar Estudiante</b> .  4. Seleccionar un Estudiante como resultado de <b>Buscar Estudiante</b> .  5. Realiza el alta de la <b>Matricula</b> .

Tabla 2.28: Caso de Uso *Matricular Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

2.2.1.2.6. *Paquete Evaluación***Registrar Evaluación por Materia**

<b>Nombre</b>	
Registrar <b>Evaluación por Materia</b> .	
<b>Pre-condición</b>	
Existen <b>Estudiantes Matriculados</b> a ser <b>Evaluados</b> en una <b>Materia</b> .	
<b>Post-condición</b>	
La <b>Evaluación</b> echa al <b>Estudiante</b> queda registrado en el sistema, por cada <b>Materia</b> .	
<b>Descripción</b>	
Realiza el registro de las <b>Evaluaciones</b> echas a cada una de las <b>Materias</b> a cada uno de los <b>Estudiantes</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ejecutar caso de uso <b>Buscar Materia</b> .  3. Ingresa las <b>Evaluaciones</b> realizadas para cada <b>Estudiante</b> .	2. Despliega un listado de los <b>Estudiantes Matriculados</b> en la <b>Materia</b> .  4. Verifica corrección de las <b>Evaluaciones</b> ingresadas. 4.1.El sistema informa que al menos una de las <b>Evaluaciones</b> ingresadas no es correcta.  5. Realiza e registro de todas las <b>Evaluaciones</b> .

Tabla 2.29: Caso de Uso *Registrar Evaluación Por Materia*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

**Registrar Evaluación por Estudiante**

<b>Nombre</b>	
Registrar <b>Evaluación por Estudiante</b> .	
<b>Pre-condición</b>	
Existe un <b>Estudiante Matriculado</b> a ser <b>Evaluado</b> en varias <b>Materias</b> .	
<b>Post-condición</b>	
Las <b>Evaluaciones</b> echas al <b>Estudiante</b> queda registrado en el sistema, por varias <b>Materias</b> .	
<b>Descripción</b>	
Realiza el registro de las <b>Evaluaciones</b> echas a cada <b>Estudiante</b> por <b>Materia</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
<p>1. Ejecutar caso de uso <b>Buscar Estudiante</b>.</p> <p>3. Ingresa las <b>Evaluaciones</b> realizadas por cada <b>Materia</b>.</p>	<p>2. Despliega un listado de las <b>Materias</b> que el <b>Estudiante</b> toma en un periodo lectivo.</p> <p>4. Verifica corrección de las <b>Evaluaciones</b> ingresadas. 4.1.El sistema informa que al menos una de las <b>Evaluaciones</b> ingresadas no es correcta.</p> <p>5. Realiza e registro de todas las <b>Evaluaciones</b>.</p>

Tabla 2.30: Caso de Uso *Registrar Evaluación por Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

## 2.2.1.2.7. Paquete Listar Información Estudiantil

**Listar Estudiantes por Curso.**

<b>Nombre</b>	
Listar Estudiantes por Curso	
<b>Pre-condición</b>	
Existe un grupo de <b>Estudiantes</b> a ser buscado en el sistema de un <b>Curso</b> y <b>período</b> lectivo determinado.	
<b>Pos-condición</b>	
El grupo de <b>Estudiantes</b> es desplegado en una <b>Lista</b> , o no existía el grupo en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de un grupo de <b>Estudiantes</b> y <b>Listándolos</b> en el sistema, especificando un <b>período</b> y un <b>Curso</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ingresar el <b>período, año lectivo y especialidad</b> .	2. Realiza la búsqueda por <b>período, año lectivo y especialidad</b> . 3. Despliega un listado con los datos especificados.

Tabla 2.31: Caso de Uso *Listar Estudiantes por Curso*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.



**Listar Evaluaciones de los Estudiantes por Materia.**

<b>Nombre</b>	
Listar Evaluaciones de Estudiantes por Materia.	
<b>Pre-condición</b>	
Existe un grupo de <b>Estudiantes</b> a ser buscado con las <b>Evaluaciones</b> de cada <b>Materia</b> en el sistema, especificando la <b>Materia</b> y <b>período</b> lectivo.	
<b>Pos-condición</b>	
El grupo de <b>Estudiantes</b> es desplegado en una <b>Lista</b> con cada una de sus <b>Evaluaciones</b> en cada <b>Materia</b> .	
<b>Descripción</b>	
Realiza la búsqueda de un grupo de <b>Estudiantes</b> y <b>Listándolos</b> en el sistema, especificando un <b>período</b> y un <b>Curso</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ingresa el <b>código de materia</b> y el <b>período lectivo</b> .	2. Realiza la búsqueda por <b>código de materia</b> y el <b>período lectivo</b> .  3. Despliega un listado con los datos especificados.

Tabla 2.32: Caso de Uso *Listar Evaluaciones de Estudiantes por Materia*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

### Listar Evaluaciones de Materias por Estudiante

<b>Nombre</b>	
Listar Evaluaciones de Materias por Estudiante	
<b>Pre-condición</b>	
Existe un grupo de <b>Evaluaciones</b> a ser buscado en el sistema de una <b>Materia</b> y <b>Estudiante</b> en un período determinado.	
<b>Pos-condición</b>	
El grupo de <b>Evaluaciones</b> por <b>Materia</b> es desplegado en una <b>Lista</b> en el sistema.	
<b>Descripción</b>	
Realiza la búsqueda de un grupo de <b>Evaluaciones</b> de <b>Materias</b> por <b>Estudiante</b> y <b>Listándolos</b> en el sistema, especificando un <b>período</b> , <b>código de materia</b> y <b>Estudiante</b> .	
<b>ACTOR</b>	<b>SISTEMA</b>
<ol style="list-style-type: none"> <li>1. Ingresa el <b>período</b> y <b>código</b> de <b>Materia</b>.</li> <li>2. Especifica el <b>Estudiante</b> el cual se va a ingresar las <b>Evaluaciones</b>.</li> </ol>	<ol style="list-style-type: none"> <li>3. Realiza el listado con los datos especificados.</li> </ol>

Tabla 2.33: Caso de Uso *Listar Evaluaciones de Materias por Estudiante*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

**Listar Profesores por Materia**

<b>Nombre</b>	
Listar Profesores por Materia	
<b>Pre-condición</b>	
Existe un grupo de <b>Profesores</b> a ser buscado en el sistema.	
<b>Pos-condición</b>	
El grupo de <b>Profesores</b> es desplegado según la <b>Materia</b> .	
<b>Descripción</b>	
Realiza un listado de <b>Profesores</b> por <b>Materia</b> en el sistema.	
<b>ACTOR</b>	<b>SISTEMA</b>
1. Ingresa <b>código</b> de <b>Materia</b> . <input type="checkbox"/>	2. Realiza la búsqueda por <b>código</b> de <b>Materia</b> .  3. Despliega un <b>Listado</b> de <b>Profesores</b> por <b>Materia</b> .

Tabla 2.34: Caso de Uso *Listar Profesores por Materia*.

**Nota:** En cualquier momento el usuario puede cancelar la ejecución del caso del uso.

## 2.2.2. CLASES

### 2.2.2.1. Diagrama de Clases

Es necesario determinar una nomenclatura para cada una de las Clases, atributos, operaciones y relaciones.

#### **Nomenclatura**

Para los tipos de datos, al tipo enumeración se establece que la primera letra del nombre sea mayúscula. Si el nombre de la *enumeración* está compuesto por dos o más palabras, serán separadas por un guión bajo “\_” y no contendrá ningún símbolo ortográfico. La primera letra de cada palabra será mayúscula.

Para la especificación de las estructuras, la primera letra del nombre será mayúscula. Si el nombre de la *estructura* está compuesto por dos o más palabras, serán separadas por un guión bajo “\_” y no contendrá ningún símbolo ortográfico. La primera letra de cada nombre tendrá letra mayúscula.

Para las clases se determinará con un nombre que identifique el motivo de la clase en la solución del problema, la primera letra del nombre será mayúscula.

Si el nombre de la clase está compuesto por dos o más palabras, serán separadas por un guión bajo “\_” y no contendrá ningún símbolo ortográfico.

Para los atributos se establece que tendrá la primera letra minúscula. Si está compuesto por dos o más palabras, éstas serán separadas por un guión bajo “\_” y la primera letra de la segunda o mayor palabra será mayúscula, no tendrá ningún símbolo ortográfico.

Para las relaciones, el nombre contendrá únicamente letras minúsculas. En caso de contener más de dos palabras, estas serán separadas por un guión bajo “\_”. Ninguna palabra contendrá un símbolo ortográfico.

## Elementos

Según el detalle de los requerimientos y los casos de uso, para el desarrollo del sistema, serán necesarios los siguientes elementos:

### Tipos de Datos

- Enumeración
  - Especialidad
  - Paralelo
- Estructura
  - Representante
  - Direccion

### Clases

- Estudiante
- Curso
- Materia
- Profesor

### Relaciones

- se\_matricula\_en: entre las clases Estudiante y Curso.
- es\_evaluado: entre las clases Estudiante y Materia.
- dicta: entre las clases Profesor y Periodo.
- contiene: entre las clases Curso y Materia.

Para el desarrollo del Sistema de Administración de Información Estudiantil para el Colegio Segré se obtuvo el siguiente Diagrama de Clases:

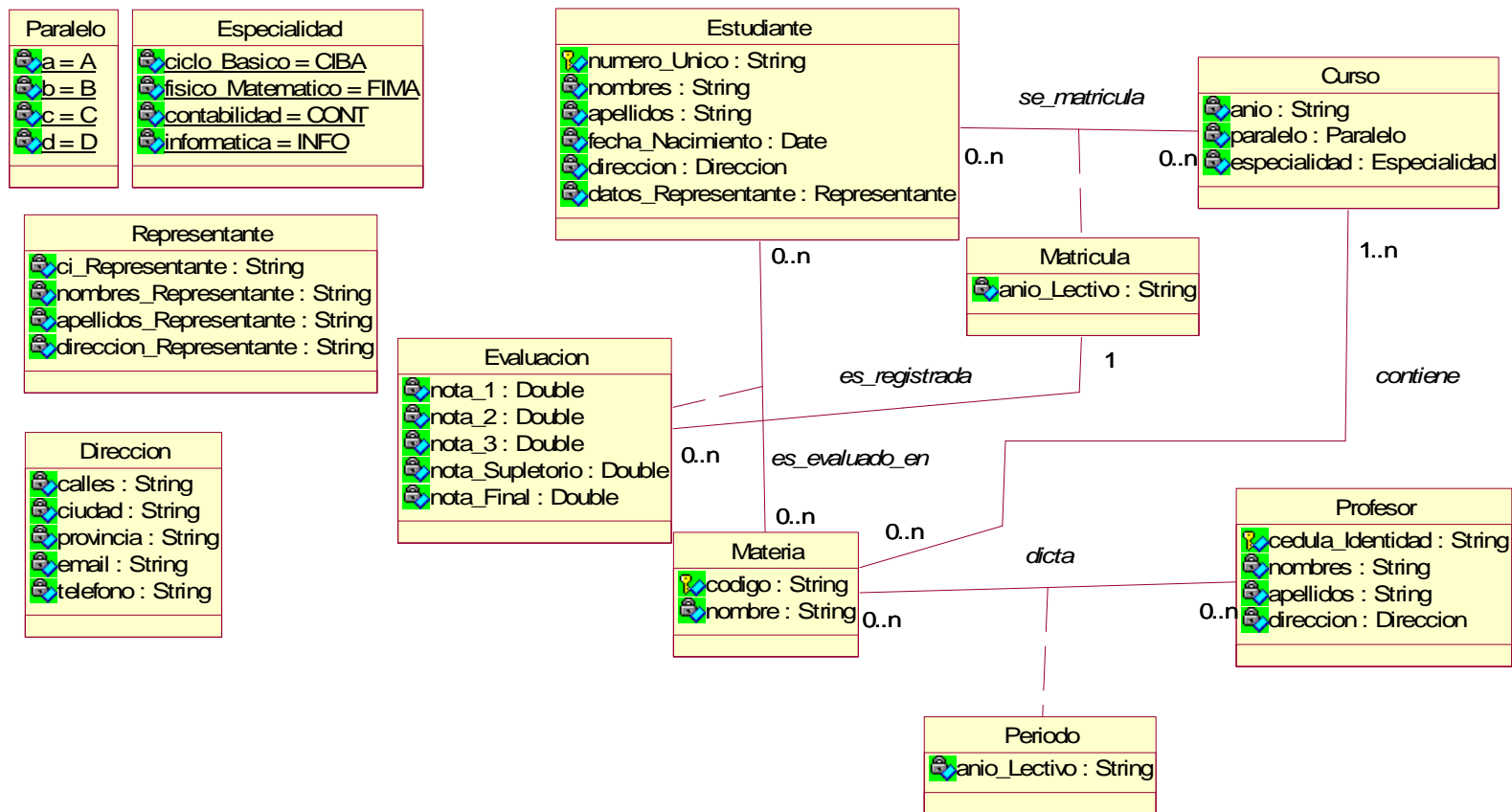


Figura 2.1: Diagrama de Clases Colegio Segré

### 2.2.2.2. Detalle de Objetos

#### 2.2.2.2.1. Tipos de Datos

##### a) Enumeración

### Especialidad

Nombre		
Especialidad		
Descripción		
Especifica las especialidades del colegio y son diferenciadas en ciclo básico y ciclo diversificado. Los niveles de primer a tercer curso serán registrados en el sistema como <i>Ciclo Básico</i> .		
Atributos	Valor	Descripción
ciclo_Basico	CIBAS	Enumeración que especifica, el nivel de <i>Ciclo Básico</i> .
físico_Matematico	FIMA	Enumeración que especifica la especialidad de <i>Físico Matemático</i> .
Contabilidad	CONTAB	Enumeración que especifica la especialidad de <i>Contabilidad</i> .
Informatica	INFORM	Enumeración que especifica la especialidad <i>Informática</i> .

Tabla 2.35: Descripción *Enumeración Especialidad*.

## Paralelo

Nombre		
Paralelo		
Descripción		
Especifica el paralelo del año.		
Atributos	Valor	Descripción
A	A	Enumeración que especifica el paralelo A.
B	B	Enumeración que especifica el paralelo B.
C	C	Enumeración que especifica el paralelo C.
D	D	Enumeración que especifica el paralelo D.

Tabla 2.36: Descripción *Enumeración Especialidad*.

### b) Estructura

## Representante

Nombre			
Representante			
Descripción			
Contiene los atributos de la información de los representantes.			
Atributos	Tipo	Tamaño	Descripción
ci_Representante	String	10	Contiene el número de cédula del representante.
nombres_Representante	String	20	Contiene los nombres del representante.
apellidos_Representante	String	20	Contiene los apellidos del



			representante.
direccion_Representante	Direccion	149	Contiene la información necesaria de localización del representante.

Tabla 2.37: Descripción *Estructura Representante*.

## Direccion

Nombre			
Direccion			
Descripción			
Contiene únicamente datos de dirección.			
Atributos	Tipo	Tamaño	Descripción
Calles	String	50	Contiene una dirección.
ciudad	String	20	Contiene el nombre de la ciudad en la que es ubicada la dirección.
provincia	String	20	Contiene el nombre de la provincia de la ciudad que fue registrada anteriormente.
telefono	String	9	Contiene el número telefónico en el cual se puede localizar a una persona.
Email	String	50	Contiene la dirección de correo electrónica.

Tabla 2.38: Descripción *Estructura Dirección*.

## 2.2.2.2.2. Clases

**Estudiante**

<b>Nombre</b>			
Estudiante			
<b>Descripción</b>			
Contiene la información de cada uno de los estudiantes que todavía no se han matriculado en el colegio. No contiene información sobre la parte académica o año que cursa o cursó en el colegio.			
<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
numero_Unico	String	7	Contiene una cadena alfanumérica que es única para los estudiantes y lo identifica en el sistema.
nombres	String	20	Contiene los nombres del estudiante.
apellidos	String	20	Contiene los apellidos del estudiante.
Fecha_Nacimiento		10	Contiene la fecha de nacimiento del estudiante.
direccion	String	30	Contiene la dirección donde vive el estudiante.
Datos_Representante	Representante_ Estudiante	199	Contiene la información del representante del estudiante. Este tiene un conjunto de campos que son necesarios para localizar al representante.
Foto	Imagen		Contiene una imagen frontal del Estudiante.

Tabla 2.39: Descripción Clase Estudiante.

**Curso**

<b>Nombre</b>			
Curso			
<b>Descripción</b>			
Contiene el año en el que se encuentran los estudiantes especificando la especialidad y paralelo. No contiene información de horario, contenido o alcance del curso.			
<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
Anio	String	3	Atributo que especifica el nivel del curso.
paralelo	Paralelo	1	Atributo que especifica el paralelo del nivel del curso
especialidad	Especialidad	4	Atributo que especifica la especialidad del paralelo.

Tabla 2.40: Descripción *Clase Curso*.**Materia**

<b>Nombre</b>			
Materia			
<b>Descripción</b>			
Contiene un código identificador y nombre de la materia.			
<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo	String	5	Atributo que contiene el código identificador de la materia.
nombre	String	25	Atributo que contiene el nombre de la materia.

Tabla 2.41: Descripción *Clase Materia*.

## Profesor

Nombre			
Profesor			
Descripción			
Clase que contiene la información de los profesores.			
Atributos	Tipo	Tamaño	Descripción
cedula_Identidad	String	10	Atributo que contiene la información del número de cédula del profesor.
nombres	String	20	Atributo que contiene la información de los nombres del profesor.
apellidos	String	20	Atributo que contiene la información de los apellidos del profesor.
direccion	Direccion	149	Atributo que contiene la información necesaria de domicilio del profesor.

Tabla 2.42: Descripción Clase *Profesor*.

### 2.2.2.2.3. Relaciones

#### ***se\_matricula\_en***

Nombre
<i>se_matricula_en</i>
Descripción
Relación formada al especificar que el <i>ESTUDIANTE</i> pertenece a un <i>CURSO</i> . Este proceso es llamado <i>MATRICULA</i> y es formado por una <i>Asociación De Clases</i> "Association Class".

<b>Clases Relacionadas</b>	Estudiante – Curso			
<b>Dirección</b>	Bidireccional			
<b>Cardinalidad</b>	<i>Estudiante:</i> 0..n <i>Curso:</i> 0..1			
<b>Asociación</b>	Matricula			
	<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
	anio_Lectivo	String	9	Atributo que contiene el período lectivo de matrícula del estudiante.

Tabla 2.43: Descripción *Relación se\_matricula\_en*.***es\_evaluado***

<b>Nombre</b>	
<i>es_evaluado</i>	
<b>Descripción</b>	
Relación formada para la obtención y asignación de información académica. Un <i>ESTUDIANTE</i> toma un conjunto de <i>MATERIAS</i> que son evaluadas. Esta relación almacena valores que son el resultado de evaluaciones echas a cada materia. La relación forma una Asociación de Clases llamada <i>EVALUACION</i> .	
<b>Clases Relacionadas</b>	Estudiante – Materia
<b>Dirección</b>	Bidireccional
<b>Cardinalidad</b>	<i>Estudiante:</i> 0..n <i>Curso:</i> 0..n

Asociación	Evaluacion			
	Atributos	Tipo	Tamaño	Descripción
	nota1	Float		Atributo que especifica la primera evaluación.
	nota2	Float		Atributo que especifica la segunda evaluación.
	nota3	Float		Atributo que especifica la tercera evaluación.
	nota_Suplet orio	Float		Atributo que especifica la evaluación de supletorio.
	nota_Final	Float		Atributo que especifica la evaluación final.

Tabla 2.44: Descripción *Relación es\_evaluado*.***dicta***

Nombre	
<i>Dicta</i>	
Descripción	
Relación para asignar <i>PROFESORES</i> a una <i>MATERIA</i> . Permitirá tener un historial en cada uno de los periodos lectivos. La relación forma una Asociación de Clase llamada <i>PERIODO</i> .	
Clases Relacionadas	Profesor – Materia
Dirección	Bidireccional
Cardinalidad	<i>Estudiante:</i> 0..n <i>Curso:</i> 0..n

<b>Asociación</b>	Periodo			
	<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
	anio_Lectivo	String	9	Atributo que contiene el período lectivo en el que se designó al <i>PROFESOR</i> la <i>MATERIA</i> .

Tabla 2.45: Descripción *Relación dicta*.***se\_dicta***

<b>Nombre</b>				
<i>se_dicta</i>				
<b>Descripción</b>				
Relación para asignar <i>MATERIAS</i> a una <i>CURSO</i> .				
<b>Clases Relacionadas</b>	Curso – Materia			
<b>Dirección</b>	Bidireccional			
<b>Cardinalidad</b>	<i>Estudiante:</i> 0..n <i>Curso:</i> 1..n			
<b>Asociación</b>	Periodo			
	<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
	anio_Lectivo	String	9	Atributo que contiene el período lectivo en el que se designó al <i>PROFESOR</i> la <i>MATERIA</i> .

Tabla 2.46: Descripción *Relación se\_dicta*.





### **2.3.2. PSM'S GENERADOS**

Para la obtención de los PSM's generados es necesario desarrollar lo especificado en el punto 1.3.7.1. , el resultado se muestra en la Figura 2.3, Figura 2.4 y Figura 2.5 PSM de Datos de SAE Segré:

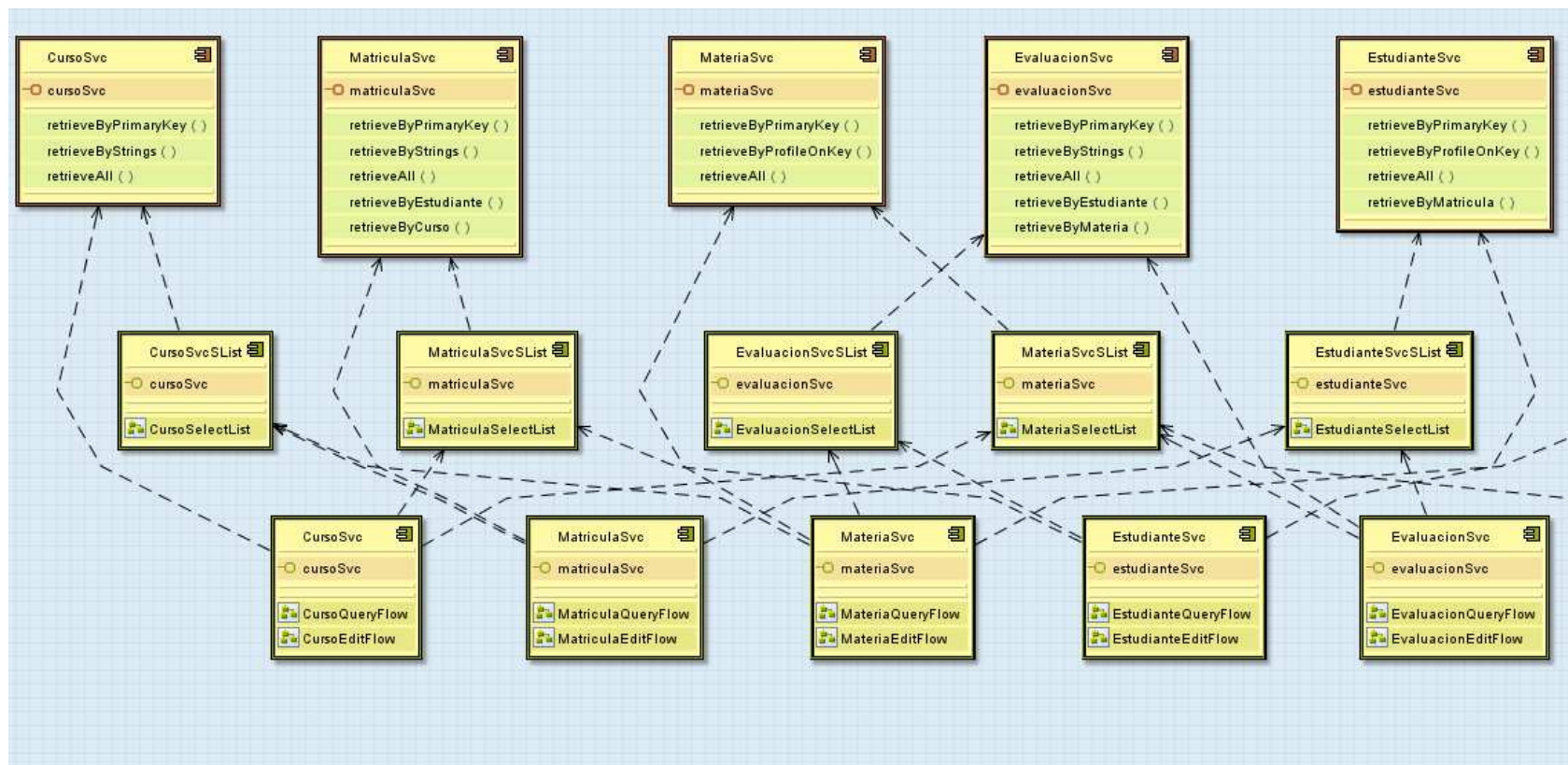


Figura 2.3 PSM de Presentación de SAE Segré

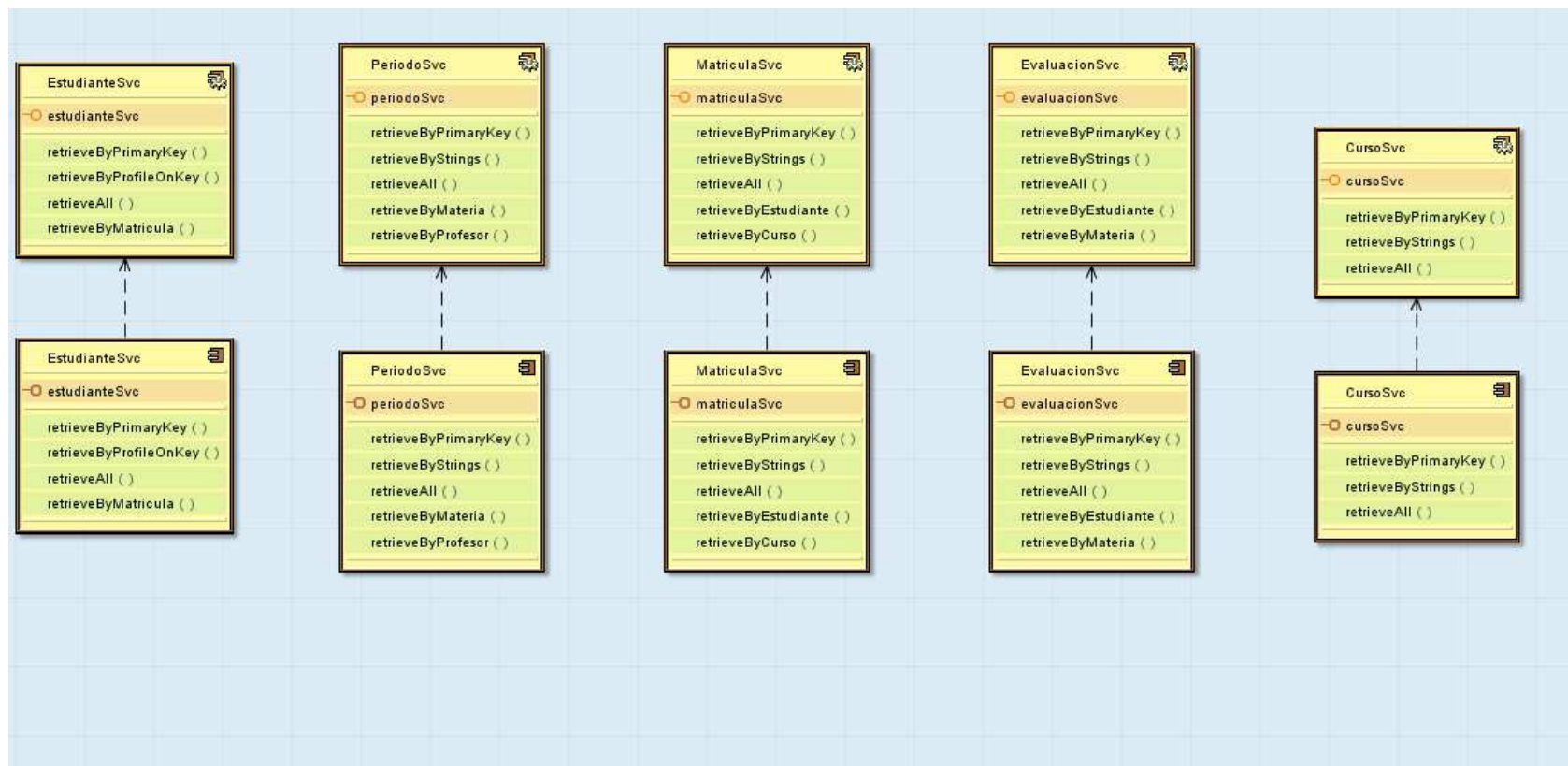


Figura 2.4 PSM de Negocio de SAE Segr

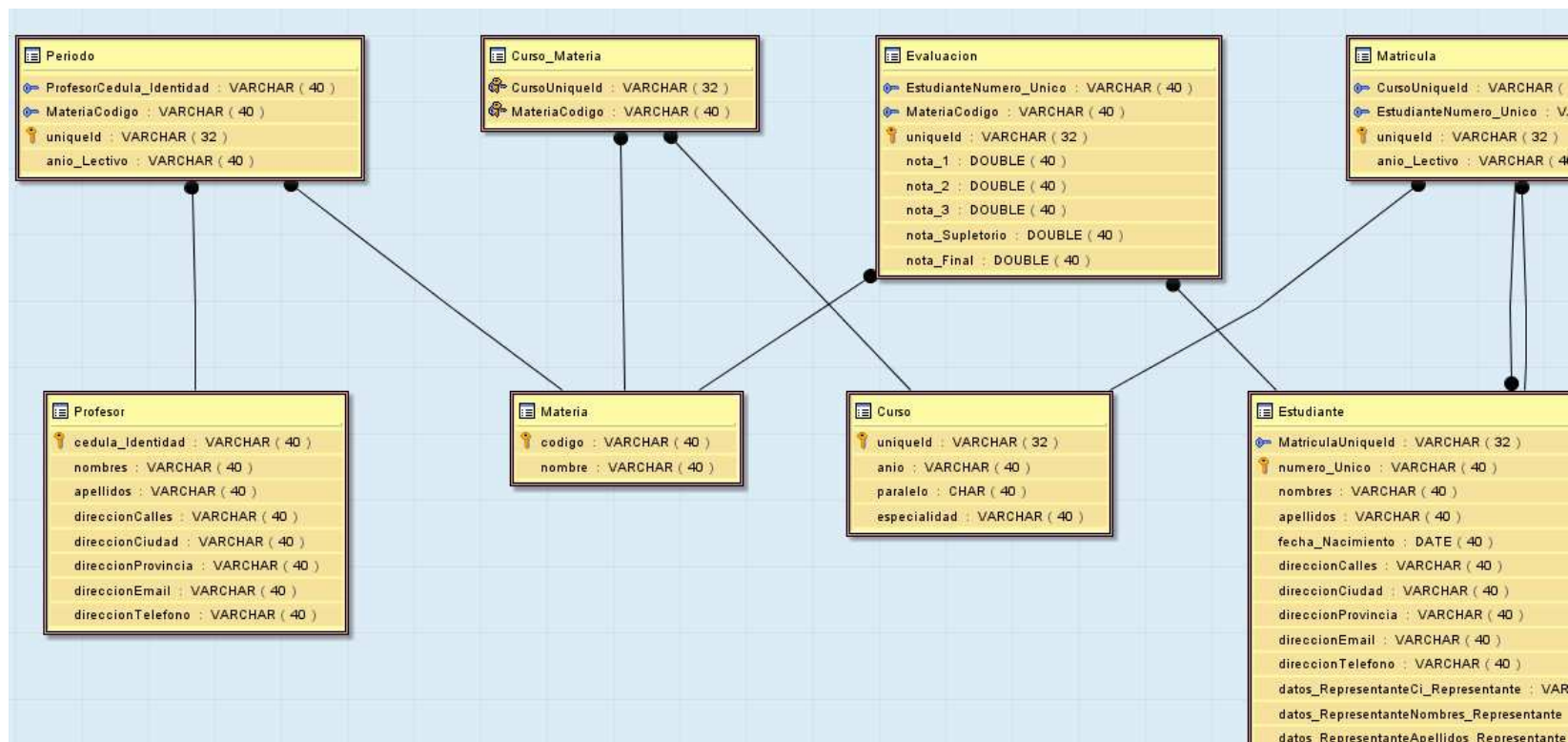


Figura 2.5 PSM de Datos de SAE Segre

### 2.3.2.1. Campos adicionales

Para aplicar *Campos Adicionales* en Evaluaciones es necesario desarrollar el apartado b). Será necesario los campos adicionales en:

#### 2.3.2.1.1. Evaluación Por Estudiante

Se deberá especificar el código y nombre de la materia a la que es evaluada, ya que el formato de transformación de modelos muestra únicamente las evaluaciones.

Los valores aplicados en la propiedad *bindsToPath* son:

#### Código de Materia

```
com.edu.sae.application.presentation.evaluacionsvqueries.EvaluacionSvcQueries
com.edu.sae.application.presentation.evaluacionsvqueries.EvaluacionSvcQueries.queryResult
com.edu.sae.application.presentation.evaluacionsvc.Evaluacion.materia
com.edu.sae.application.presentation.evaluacionsvc.Materia.codigo
```

Figura 2.6 PSM de Datos de SAE Segré

#### Nombre de Materia

```
com.edu.sae.application.presentation.evaluacionsvqueries.EvaluacionSvcQueries
com.edu.sae.application.presentation.evaluacionsvqueries.EvaluacionSvcQueries.queryResult
com.edu.sae.application.presentation.evaluacionsvc.Evaluacion.materia
com.edu.sae.application.presentation.evaluacionsvc.Materia.nombre
```

Figura 2.7 PSM de Datos de SAE Segré

#### 2.3.2.1.2. Evaluación Por Materia

Se deberá especificar el código, nombres y apellidos del estudiante que es evaluado, ya que el formato de transformación de modelos muestra únicamente las evaluaciones.

### 2.3.2.2. Validación de campos

Las validaciones se realizarán en los siguientes campos:

Atributo	Descripción	Ejemplo
nombres	Es necesario que se ingrese dos nombres, la primera letra de cada nombre deberán ser mayúscula y separados por un espacio.	Juan Carlos
apellidos	Es necesario que se ingrese dos apellidos, la primera letra de cada apellido deberán ser mayúscula y separados por un espacio.	Perez Gonzales
números telefónicos	Es necesario que se ingrese el número telefónico completo, es decir, que incluya los códigos de celular o provincial. Además que cada tres dígitos sea separados por un guión '-'. -	099-876-555
email	Se deberá controlar la sintaxis de las direcciones de correos electrónicos	gcalero@hotmail.com

Tabla 2.47: Descripción para la validación de campos

## Procedimiento

Para la validación de los campos especificados anteriormente se desarrollará las especificaciones detalladas en el apartado c) con las siguientes características.

Campo a validar	Expresión Regular
Cédula de Identidad	
Nombres y Apellidos (Estudiantes y Profesores)	^[A-Z][a-z]* [A-Z][a-z]*\$
Número telefónico	^0[1-9]{2,2}-[1-9]{3,3}-[1-9]{3,3}\$
Email	^([a-zA-Z0-9_\. \-])+\@((([a-zA-Z0-9\-.])+\.)+([a-zA-Z0-9]{2,4})+)\$

Tabla 2.48: Expresiones Regulares para validar campos

## CAPÍTULO 3:

### CONSTRUCCIÓN Y PRUEBAS DEL SISTEMA

#### 3.1. CODIFICACIÓN

Para generar el código del Sistema de Administración Estudiantil, se ejecutará el apartado 1.3.7.3.

##### 3.1.1. DESCRIPCIÓN DE CAMBIOS EN EL CÓDIGO

###### 3.1.1.1. Función `init()` de `EstudianteForm.java`

**Paquete:** `e pn.edu.sistema.application.presentation.estudiantesvc`

Esta función inicializa los campos de las formas que estén relacionadas con Estudiante.

El cambio realizado consiste en tomar una parte de la clave generada por la aplicación y reducirla a tamaño 7 y concatenarla con una cadena “CS”, con se especifica en el apartado 2.2.3.3.

```
if (keyGenerator!=null) {
    // numero_Unico has auto-keygeneration of type: OPTIMALJ
    //setNumero_Unico(keyGenerator.getUniqueKey("Estudiante").toString
    ());
    // CAMBIO REALIZADO
    String CS = new String ("CS-");
    setNumero_Unico(CS.concat((keyGenerator.getUniqueKey("Estudiante")
    .toString()).substring(0, 7)));
}
```

###### 3.1.1.2. Función `calcularNotaFinal()` de `EvaluationBean.java`

**Paquete:** `e pn.edu.sistema.application.business.logic`

La declaración de esta función fue creada en el modelo, pero no especifica ningún tipo de operación.

El código que será implementado es el siguiente:

```
returnValue=Double.valueOf(this.getNota1().doubleValue()+
this.getNota2().doubleValue()+this.getNota3().doubleValue()+this.getNot
a_Supletorio().doubleValue());
```



La función será invocada en la siguiente clase y función:

```
EvaluacionUpdateObject en setNota_Final()
```

A continuación se ejecuta Build Script, el archivo buildSistema.xml y los Archives.

### 3.1.2. CREACIÓN DE LA BASE DE DATOS

Para la creación de la Base de Datos se ejecutará lo expuesto en el apartado 1.3.7.4.

## 3.2. PRUEBAS

### Caso Prueba 1

<b>Prueba Administrar Curso – caso válido</b>	
<b>Nombre de Proceso:</b>	Registrar Curso
<b>Descripción de Proceso:</b>	Registro de nuevos cursos.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Curso.
<b>Entradas:</b>	Año: 1ro Paralelo: a Especialidad: ciclo_basico
<b>Pre – condiciones:</b>	Selección de función: Nuevo
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar Curso</i> . 2.- Seleccionar <i>Nuevo</i> . 3.- Ingresar los valores indicados en las <i>entradas</i> . 4.- Seleccionar <i>Actualizar</i> . 5.- Seleccionar <i>Guardar</i> .
<b>Resultado:</b>	1.- Seleccionar <i>Administrar Curso</i> . 2.- Seleccionar <i>Buscar</i> .


	Se visualiza el <i>Curso</i> con los datos de entrada. 
<b>Observaciones:</b>	Luego de ejecutar el procedimiento en el paso 4, se puede visualizar una grilla con el nuevo <i>Curso</i> .

Tabla 3.1: Caso de Prueba Administrar Curso – caso válido

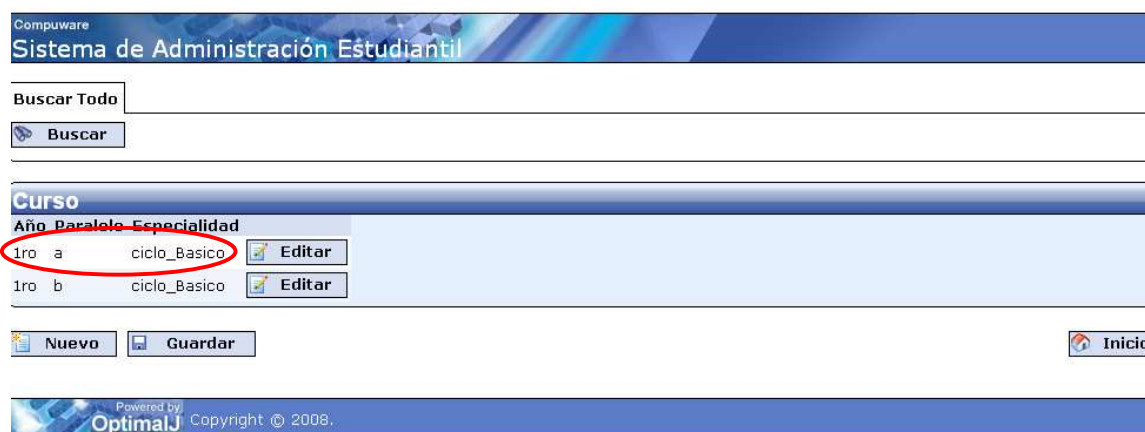


Figura 3.1: Caso de Prueba 1

## Caso Prueba 2

<b>Prueba Administrar Estudiante – caso válido</b>	
<b>Nombre de Proceso:</b>	Registrar Estudiante
<b>Descripción de Proceso:</b>	Registro de nuevos estudiantes.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Estudiante.
<b>Entradas:</b>	<p>Nombres: Carlos Alejandro  Apellidos: Guzman Romero  Fecha Nacimiento: 19-jun-2000  Calles: Florida y Yumbos 534  Ciudad: Quito  Provincia: Pichincha  Email: carlos@hotmail.com  Teléfono: 023-391-055  Cédula_Representante: 1717658210  Nombres_Representante: Jorge Alejandro  Apellidos_Representante: Guzman Castro  Calles_Representante: Florida y Yumbos 534  Ciudad_Representante: Quito  Provincia_Representante: Pichincha  Email_Representante: jorge@hotmail.com  Teléfono_Representante: 023-391-055  Foto: C:\Documents and Settings\Personal\Escritorio\foto carnet cole.jpg</p> 

<b>Pre – condiciones:</b>	Selección de función: Nuevo												
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1.- Menú inicial, seleccionar <i>Administrar Estudiante</i>.</li> <li>2.- Seleccionar <i>Nuevo</i>.</li> <li>3.- Ingresar los valores indicados en las <i>entradas</i>.</li> <li>4.- Seleccionar <i>Actualizar</i>.</li> <li>5.- Seleccionar <i>Guardar</i>.</li> </ol>												
<b>Resultado:</b>	<ol style="list-style-type: none"> <li>1.- Seleccionar <i>Administrar Estudiante</i>.</li> <li>2.- Seleccionar <i>Buscar Todos</i>.</li> </ol> <p>Se visualiza el <i>Estudiante</i> con los datos de entrada de nombres y apellidos, incluido el <i>Número Único</i> que es dado por el sistema.</p> <table border="1"> <thead> <tr> <th colspan="4">Estudiante</th> </tr> <tr> <th>Numero_Unico</th> <th>Nombres</th> <th>Apellidos</th> <th>Fecha_Nacimiento</th> </tr> </thead> <tbody> <tr> <td>CS-c9046ea</td> <td>Carlos Alejandro</td> <td>Guzman Romero</td> <td>19-jun-2000</td> </tr> </tbody> </table>	Estudiante				Numero_Unico	Nombres	Apellidos	Fecha_Nacimiento	CS-c9046ea	Carlos Alejandro	Guzman Romero	19-jun-2000
Estudiante													
Numero_Unico	Nombres	Apellidos	Fecha_Nacimiento										
CS-c9046ea	Carlos Alejandro	Guzman Romero	19-jun-2000										
<b>Observaciones:</b>	Luego de ejecutar el procedimiento en el paso 4, se puede visualizar una grilla con el nuevo <i>Estudiante</i> .												

Tabla 3.2: Caso de Prueba Administrar Estudiante – caso válido

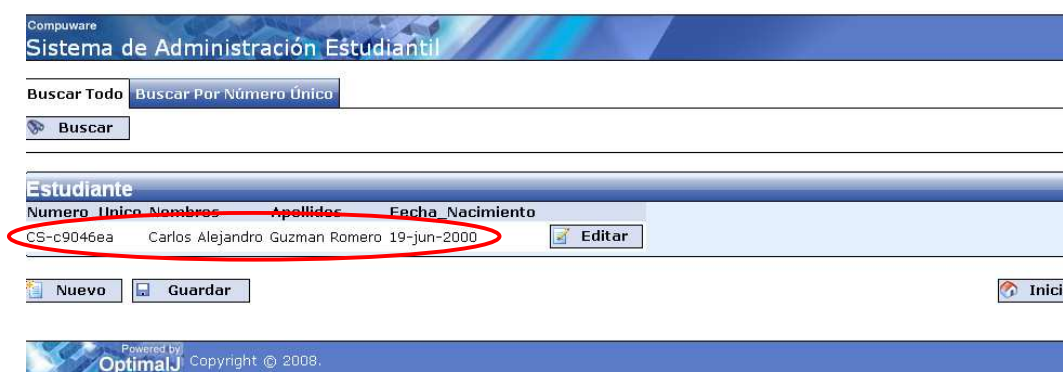


Figura 3.2: Caso de Prueba 2

## Caso Prueba 3

<b>Prueba Administrar Estudiante – caso erróneo</b>	
<b>Nombre de Proceso:</b>	Registrar Estudiante
<b>Descripción de Proceso:</b>	Registro de nuevos estudiantes.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Estudiante.
<b>Entradas:</b>	<p> Nombres: 3duardo Alejandro  Apellidos: Guzman Romero  Fecha Nacimiento: 19-jun-2000  Calles: Florida y Yumbos 534  Ciudad: Quito  Provincia: Pichincha  Email: carlos  Teléfono: 02ee91055  Cédula_Representante: 1717658210  Nombres_Representante: 9orge Alejandro  Apellidos_Representante: Guzmán Castro  Calles_Representante: Florida y Yumbos 534  Ciudad_Representante: Quito  Provincia_Representante: Pichincha  Email_Representante: jorge@hotmail.com  Teléfono_Representante: 023391055  Foto: C:\Documents and Settings\Personal\Escritorio\foto carnet cole.jpg </p> 

<b>Pre – condiciones:</b>	Selección de función: Nuevo
<b>Procedimiento:</b>	<p>1.- Menú inicial, seleccionar <i>Administrar Estudiante</i>.</p> <p>2.- Seleccionar <i>Nuevo</i>.</p> <p>3.- Ingresar los valores indicados en las <i>entradas</i>.</p> <p>4.- Seleccionar <i>Actualizar</i>.</p> <p>5.- Seleccionar <i>Guardar</i>.</p>
<b>Resultado:</b>	<p>Error de aplicación</p> <p>Debería corregir el(los) erro(es) antes de continuar:</p> <ul style="list-style-type: none"> <li>• Nombres es incorrecto</li> <li>• Email es incorrecto</li> <li>• Teléfono es incorrecto</li> <li>• Nombres_Representante es incorrecto</li> </ul> <p><b>Error de Aplicación</b></p> <p>Debería corregir el(los) error(es) antes de continuar:</p> <ul style="list-style-type: none"> <li>• nombres es incorrecto.</li> <li>• email es incorrecto.</li> <li>• telefono es incorrecto.</li> <li>• nombres_Representante es incorrecto.</li> </ul>
<b>Observaciones:</b>	Los campos mostrados en el resultado no tiene el formato indicado.

Tabla 3.3: Caso de Prueba Administrar Estudiante – caso erróneo

Provincia: Pichincha

Email: carlos ejm: (estudiante@server.com)

Telefono: 02ee91055 ejm: (099-123-456)

Ci\_Representante: 1717658210 ejm: (1234567890)

Nombres\_Representante: 9orge Alejandro ejm: (Juan Carlos)

Apellidos\_Representante: Guzman Castro ejm: (Perez Ramos)

Calles: Florida y Yumbos 534


Ciudad: Quito

Provincia: Pichincha

Email: jorge@hotmail.com ejm: (representante@server.com)

Telefono: 092-123-124 ejm: (099-123-456)

Foto: C:\Documents and Settings\Gerald Calero\E: Examinar...



Eliminar Actualizar Cancelar Inicio

**Error de Aplicación**

Debería corregir el(los) error(es) antes de continuar:

- nombres es incorrecto.
- email es incorrecto.
- telefono es incorrecto.
- nombres\_Representante es incorrecto.

Figura 3.3: Caso de Prueba 3

## Caso Prueba 4

Prueba Administrar Materia – caso válido	
<b>Nombre de Proceso:</b>	Registrar Materia
<b>Descripción de Proceso:</b>	Registro de nuevas materias.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Materia.
<b>Entradas:</b>	Código: ccnn1 Nombre: Ciencias Naturales 1 Ejecutar Crear o Añadir <i>Materia</i> Si Crear Año: 1ro Paralelo: c


	<p>Especialidad: ciclo_basico</p> <p>Si Añadir</p> <p>Ejecutar <i>BuscarTodo</i></p> <p>Seleccionar 1ro a ciclo_Basico</p>
<b>Pre – condiciones:</b>	Selección de función: Nuevo
<b>Procedimiento:</b>	<p>1.- Menú inicial, seleccionar <i>Administrar Materia</i>.</p> <p>2.- Seleccionar <i>Nuevo</i>.</p> <p>3.- Ingresar los valores indicados en las <i>entradas</i>.</p> <p>4.- Seleccionar <i>Actualizar</i>.</p> <p>5.- Seleccionar <i>Guardar</i>.</p> 
<b>Resultado:</b>	<p>1.- Seleccionar <i>Administrar Materia</i>.</p> <p>2.- Seleccionar <i>BuscarTodo</i>.</p> <p>Se visualiza la <i>Materia</i> con los datos de entrada.</p>
<b>Observaciones:</b>	Luego de ejecutar el procedimiento en el paso 4, se puede visualizar una grilla con la nueva <i>Materia</i> .

Tabla 3.4: Caso de Prueba Administrar Materia – caso válido



Figura 3.4: Caso de Prueba 4



## Caso Prueba 5

Prueba Administrar Materia – caso erróneo	
<b>Nombre de Proceso:</b>	Registrar Materia
<b>Descripción de Proceso:</b>	Registro de nuevas materias.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Materia.
<b>Entradas:</b>	Código: CCNN1 Nombre: Ciencias Naturales 1
<b>Pre – condiciones:</b>	Selección de función: Nuevo
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar Materia</i> . 2.- Seleccionar <i>Nuevo</i> . 3.- Ingresar los valores indicados en las <i>entradas</i> . 4.- Seleccionar <i>Actualizar</i> . 5.- Seleccionar <i>Guardar</i> .
<b>Resultado:</b>	Error de aplicación Debería corregir los siguientes datos: <ul style="list-style-type: none"> <li>• Asociación <i>Materia</i> y <i>Curso</i> es mandatorio</li> </ul> <b>Error de Aplicación</b> Debería corregir el(los) error(es) antes de continuar: <ul style="list-style-type: none"> <li>• Asociación <i>Materia.curso</i> es mandatorio</li> </ul>
<b>Observaciones:</b>	Significa que es necesario crear o añadir un <i>Curso</i> para crear la <i>Materia</i> .

Tabla 3.5: Caso de Prueba Administrar Materia – caso erróneo

Compuware  
Sistema de Administración Estudiantil

**Materia**

Código \*

Nombre

**Curso**

Año	Paralelo	Especialidad
No records found.		

**Error de Aplicación**

Debería corregir el(los) error(es) antes de continuar:

- Asociación Materia.curso es obligatorio

Powered by  
OptimalJ Copyright © 2008.

Figura 3.5: Caso de Prueba 5

## Caso Prueba 6

Prueba Administrar Matrícula – caso válido	
<b>Nombre de Proceso:</b>	Registrar Matrícula
<b>Descripción de Proceso:</b>	Registro de nuevas materias.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Matrícula.
<b>Entradas:</b>	Periodo_Lectivo: 2007-2008  Estudiante: Numero_Único Nombres Apellidos CS-293iejd2 Carlos Alejandro Guzmán Romero  Curso: Año Paralelo Especialidad 1ro a ciclo_basico

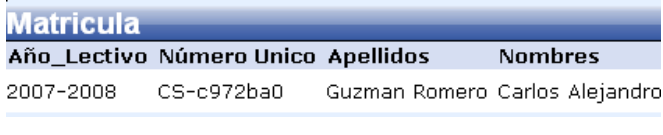
<b>Pre – condiciones:</b>	Tener registros de <i>Estudiantes</i> . Tener registros de <i>Cursos</i> . Selección de función: Nuevo
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar Matricula</i> . 2.- Seleccionar <i>Nuevo</i> . 3.- Ingresar los valores indicados en las <i>entradas</i> . 4.- Seleccionar <i>Actualizar</i> . 5.- Seleccionar <i>Guardar</i> .  
<b>Resultado:</b>	Se ha ejecutado el registro de la <i>Matricula</i> .
<b>Observaciones:</b>	

Tabla 3.6: Caso de Prueba Administrar Matrícula – caso válido

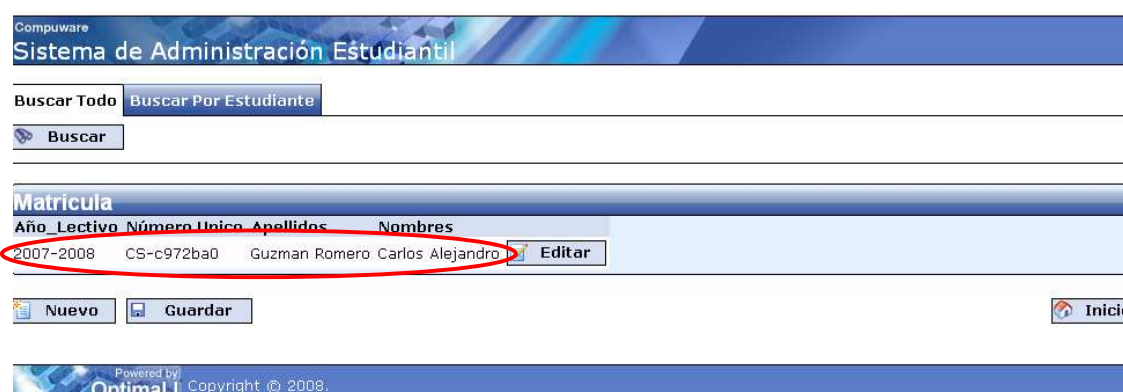


Figura 3.6: Caso de Prueba 6

### Caso Prueba 7

Prueba Administrar Matrícula – caso erróneo	
<b>Nombre de Proceso:</b>	Registrar Matrícula
<b>Descripción de Proceso:</b>	Registro de nuevas materias.

<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Matrícula.
<b>Entradas:</b>	Periodo_Lectivo: 2007-2008  Estudiante: Numero_Único    Nombres                    Apellidos CS293iej2    Carlos Alejandro    Guzmán Romero
<b>Pre – condiciones:</b>	Tener registros de <i>Estudiantes</i> . Tener registros de <i>Cursos</i> . Selección de función: Nuevo
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar Matrícula</i> . 2.- Seleccionar <i>Nuevo</i> . 3.- Ingresar los valores indicados en las <i>entradas</i> . 4.- Seleccionar <i>Actualizar</i> .
<b>Resultado:</b>	Error de aplicación Debería corregir los siguientes datos: <ul style="list-style-type: none"> <li>• Asociación <i>Matrícula</i> y <i>Curso</i> es necesaria.</li> </ul> <b>Error de Aplicación</b> Debería corregir el(los) error(es) antes de continuar: <ul style="list-style-type: none"> <li>• Asociación Matrícula.curso es mandatorio</li> </ul>
<b>Observaciones:</b>	Significa que es necesario añadir un <i>Curso</i> para crear la <i>Matrícula</i> .

Tabla 3.7: Caso de Prueba Administrar Matrícula – caso erróneo

Compuware  
Sistema de Administración Estudiantil

---

**Matricula**

Año\_Lectivo

---

**Curso \***

Año  
Paralelo <Not Set>  
Especialidad <Not Set>

---

**Estudiante \***

Numero\_Unico \* CS-c972ba0  
Nombres Carlos Alejandro  
Apellidos Guzman Romero

---

**Error de Aplicación**

Debería corregir el(los) error(es) antes de continuar:

- Asociación Matricula.curso es mandatorio

---

Powered by  
**OptimalJ** Copyright © 2008.

Figura 3.7: Caso de Prueba 7

## Caso Prueba 8

Prueba Administrar Profesor – caso válido	
<b>Nombre de Proceso:</b>	Registrar Profesor
<b>Descripción de Proceso:</b>	Registro de nuevos profesores.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Profesor.
<b>Entradas:</b>	Cédula: 1717456310 Nombres: Gonzalo Raul Apellidos: Acosta Zurita Calles: Prensa y Florida 534 Ciudad: Quito Provincia: Pichincha Email: gonzalo@hotmail.com


	<p>Teléfono: 023-344-058</p> <p>Foto: C:\Documents and Settings\Personal\Escritorio\foto carnet prof.jpg</p> 						
<b>Pre – condiciones:</b>	Selección de función: Nuevo						
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1.- Menú inicial, seleccionar <i>Administrar Profesor</i>.</li> <li>2.- Seleccionar <i>Nuevo</i>.</li> <li>3.- Ingresar los valores indicados en las <i>entradas</i>.</li> <li>4.- Seleccionar <i>Actualizar</i>.</li> <li>5.- Seleccionar <i>Guardar</i>.</li> </ol>						
<b>Resultado:</b>	<ol style="list-style-type: none"> <li>1.- Seleccionar <i>Administrar Profesor</i>.</li> <li>2.- Seleccionar <i>Buscar Todos</i>.</li> </ol> <p>Se visualiza el <i>Profesor</i> con los datos de entrada de cédula, nombres y apellidos.</p> <table border="1"> <thead> <tr> <th colspan="2">Profesor</th> </tr> <tr> <th>Cedula_Identidad</th> <th>Nombres Apellidos</th> </tr> </thead> <tbody> <tr> <td>1717456310</td> <td>Gonzalo Raul Acosta Zurita</td> </tr> </tbody> </table>	Profesor		Cedula_Identidad	Nombres Apellidos	1717456310	Gonzalo Raul Acosta Zurita
Profesor							
Cedula_Identidad	Nombres Apellidos						
1717456310	Gonzalo Raul Acosta Zurita						
<b>Observaciones:</b>	Luego de ejecutar el procedimiento en el paso 4, se puede visualizar una grilla con el nuevo <i>Profesor</i> .						

Tabla 3.8: Caso de Prueba Administrar Profesor – caso válido

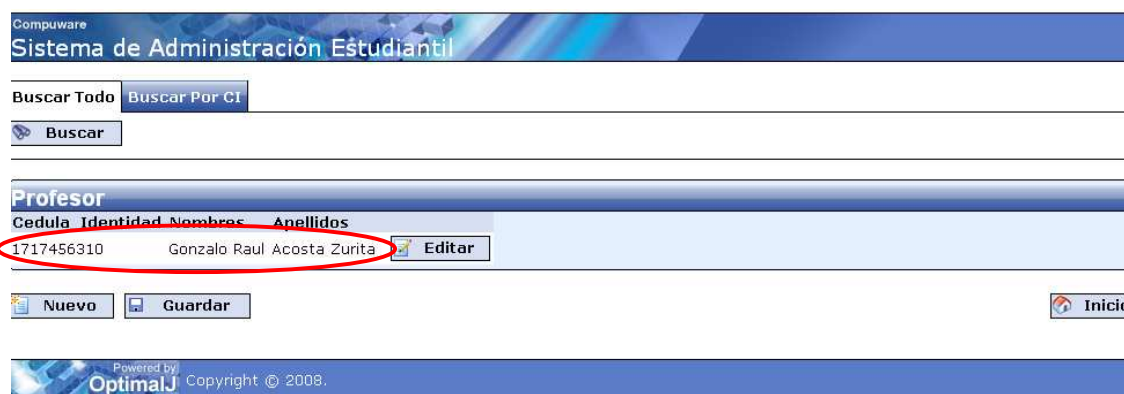


Figura 3.8: Caso de Prueba 8

## Caso Prueba 9

Prueba Administrar Profesor – caso erróneo	
<b>Nombre de Proceso:</b>	Registrar Profesor
<b>Descripción de Proceso:</b>	Registro de nuevos profesores.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar Profesor.
<b>Entradas:</b>	Cédula: 171745610 Nombres: Go4zalo Raul Apellidos: Guzman Romero Calles: Florida y Machala 534 Ciudad: Qui7o Provincia: Pichincha Email: gonzalo Teléfono: 02ee91055 Foto: C:\Documents and Settings\Personal\Escritorio\foto carnet prof.jpg

<b>Pre – condiciones:</b>	Selección de función: Nuevo
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1.- Menú inicial, seleccionar <i>Administrar Profesor</i>.</li> <li>2.- Seleccionar <i>Nuevo</i>.</li> <li>3.- Ingresar los valores indicados en las <i>entradas</i>.</li> <li>4.- Seleccionar <i>Actualizar</i>.</li> </ol>
<b>Resultado:</b>	<p>Error de aplicación</p> <p>Debería corregir el(los) error(es) antes de continuar:</p> <ul style="list-style-type: none"> <li>• Cédula Identidad es incorrecto</li> <li>• Nombres es incorrecto</li> <li>• Email es incorrecto</li> <li>• Teléfono es incorrecto</li> </ul> <p><b>Error de Aplicación</b></p> <p>Debería corregir el(los) error(es) antes de continuar:</p> <ul style="list-style-type: none"> <li>• cedula_Identidad es incorrecto.</li> <li>• nombres es incorrecto.</li> <li>• email es incorrecto.</li> <li>• telefono es incorrecto.</li> </ul>
<b>Observaciones:</b>	Los campos mostrados en el resultado no tiene el formato indicado.

Tabla 3.9: Caso de Prueba Administrar Profesor – caso erróneo



Compuware  
Sistema de Administración Estudiantil

**Profesor**

Cedula_Identidad *	171745610	ejm: (1234567890)	• cedula_Identidad es incorrecto.
Nombres	Go4zalo Raul	ejm: (Juan Carlos)	• nombres es incorrecto.
Apellidos	Acosta Zurita	ejm: (Perez Ramos)	
Calles	Prensa y Florida 534		
Ciudad	Quito		
Provincia	Pichincha		
Email	gonzalo	ejm: (profesor@server.com)	• email es incorrecto.
Telefono	02ee91055	ejm: (099-123-456)	• telefono es incorrecto.
Foto		Examinar...	

Eliminar Actualizar Cancelar Inicio

**Error de Aplicación**

Debería corregir el(los) error(es) antes de continuar:

- cedula\_Identidad es incorrecto.
- nombres es incorrecto.
- email es incorrecto.
- telefono es incorrecto.

Figura 3.9: Caso de Prueba 9

## Caso Prueba 10

Prueba Administrar AsignarMaterias – caso válido	
<b>Nombre de Proceso:</b>	Asignar Materias
<b>Descripción de Proceso:</b>	Asignar materias a profesores.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar AsignarMaterias.
<b>Entradas:</b>	Periodo_Lectivo: 2007-2008  Materia: Codigo            Nombre CCNN1            Ciencias Naturales 1

	<p>Profesor:</p> <table> <tr> <td>Cédula</td> <td>Nombres</td> <td>Apellidos</td> </tr> <tr> <td>1717456310</td> <td>Gonzalo Raúl</td> <td>Acosta Zurita</td> </tr> </table>	Cédula	Nombres	Apellidos	1717456310	Gonzalo Raúl	Acosta Zurita						
Cédula	Nombres	Apellidos											
1717456310	Gonzalo Raúl	Acosta Zurita											
<b>Pre – condiciones:</b>	<p>Tener registros de <i>Profesores</i>.</p> <p>Tener registros de <i>Materias</i>.</p> <p>Selección de función: Nuevo</p>												
<b>Procedimiento:</b>	<p>1.- Menú inicial, seleccionar <i>Administrar AsignarMaterias</i>.</p> <p>2.- Seleccionar <i>Nuevo</i>.</p> <p>3.- Ingresar los valores indicados en las <i>entradas</i>.</p> <p>4.- Seleccionar <i>Actualizar</i>.</p> <p>5.- Seleccionar <i>Guardar</i>.</p>												
<b>Resultado:</b>	<p>Se ha ejecutado el registro de <i>AsignarMaterias</i>.</p> <table border="1"> <thead> <tr> <th colspan="4">Periodo</th> </tr> <tr> <th>Año_Lectivo</th> <th>Cédula de Identidad</th> <th>Apellidos</th> <th>Nombres</th> </tr> </thead> <tbody> <tr> <td>2007-2008</td> <td>1717456105</td> <td>Acosta Zurita</td> <td>Gonzalo Raul</td> </tr> </tbody> </table>	Periodo				Año_Lectivo	Cédula de Identidad	Apellidos	Nombres	2007-2008	1717456105	Acosta Zurita	Gonzalo Raul
Periodo													
Año_Lectivo	Cédula de Identidad	Apellidos	Nombres										
2007-2008	1717456105	Acosta Zurita	Gonzalo Raul										
<b>Observaciones:</b>													

Tabla 3.10: Caso de Prueba Administrar AsignarMaterias – caso válido

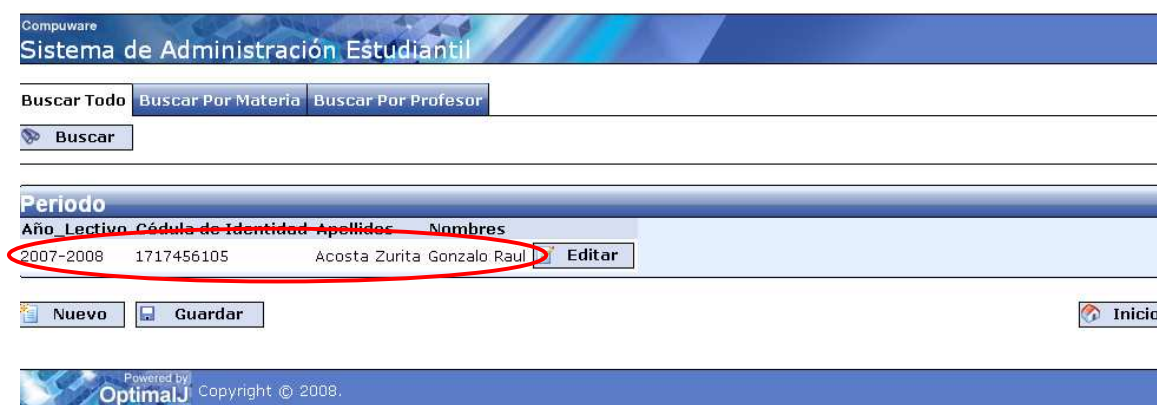


Figura 3.10: Caso de Prueba 10

## Caso Prueba 11

Prueba Administrar AsignarMaterias – caso erróneo	
<b>Nombre de Proceso:</b>	Registrar Materias
<b>Descripción de Proceso:</b>	Asignar materias a profesores.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar AsignarMaterias.
<b>Entradas:</b>	Periodo_Lectivo: 2007-2008  Profesor: Cédula            Nombres            Apellidos 1717456310    Gonzalo Raúl    Acosta Zurita
<b>Pre – condiciones:</b>	Tener registros de <i>Profesores</i> . Tener registros de <i>Materias</i> . Selección de función: Nuevo
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar AsignarMaterias</i> . 2.- Seleccionar <i>Nuevo</i> . 3.- Ingresar los valores indicados en las <i>entradas</i> . 4.- Seleccionar <i>Actualizar</i> .
<b>Resultado:</b>	Error de aplicación Debería corregir los siguientes datos: <ul style="list-style-type: none"> <li>• Asociación <i>Periodo</i> y <i>Materia</i> es necesaria.</li> </ul> <b>Error de Aplicación</b> Debería corregir el(los) error(es) antes de continuar: <ul style="list-style-type: none"> <li>• Asociación Periodo.materia es mandatorio</li> </ul>
<b>Observaciones:</b>	Significa que es necesario añadir una <i>Materia</i> para Asignar Materias a los <i>Profesores</i> .

Tabla 3.11: Caso de Prueba Administrar AsignarMaterias – caso erróneo

Compuware  
Sistema de Administración Estudiantil

**Periodo**  
Año\_Lectivo: 2007-2008

**Materia \***  
Seleccionar  
Código \*  
Nombre

**Profesor \***  
Seleccionar  
Cedula\_Identidad \*: 1717456105  
Nombres: Gonzalo Raul  
Apellidos: Acosta Zurita

Eliminar Actualizar Cancelar Inicio

**error de Aplicación**  
debería corregir el(los) error(es) antes de continuar:

- Asociación Periodo.materia es mandatorio

Powered by OptimalJ Copyright © 2008.

Figura 3.11: Caso de Prueba 11

## Caso Prueba 12

Prueba Administrar RegistroEvaluacionesPorEstudiante – caso válido	
<b>Nombre de Proceso:</b>	Registrar Evaluaciones Por Estudiante
<b>Descripción de Proceso:</b>	Registrar evaluaciones de los estudiantes.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar RegistroEvaluacionesPorEstudiante.
<b>Entradas:</b>	Numero_Único: CSa6934ds Periodo_Lectivo: 2007-2008  Materia: Codigo            Nombre CCNN1            Ciencias Naturales 1

	Evaluación: Nota1: 9																												
<b>Pre – condiciones:</b>	Tener registros de <i>Estudiantes</i> . El <i>Estudiante</i> tenga registro de <i>Matricula</i> .																												
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1.- Menú inicial, seleccionar <i>Administrar RegistroNotasPorEstudiante</i>.</li> <li>2.- Ingresar Numero_Único y Periodo_Lectivo especificados en <i>entradas</i>.</li> <li>3.- Ejecutar <i>Buscar</i>.</li> <li>4.- Ejecutar <i>Seleccionar</i> en el elemento desplegado.</li> <li>5.- Ejecutar <i>Nuevo</i> en <i>Evaluación</i>.</li> <li>6.- Ejecutar <i>Seleccionar</i> en <i>Materia</i>.</li> <li>7.- Ejecutar <i>Buscar</i>.</li> <li>8.- Seleccionar la <i>Materia</i> especificada en la <i>entrada</i>.</li> <li>9.- Ingresar el valor de la <i>Evaluación</i> especificada en la <i>entrada</i>.</li> <li>10.- Ejecutar <i>Actualizar</i>.</li> <li>11.- Ejecutar <i>Actualizar</i>.</li> <li>12.- Ejecutar <i>Guardar</i>.</li> </ol>																												
<b>Resultado:</b>	<p>Se puede visualiza las <i>Materias</i> y <i>Evaluaciones</i> registradas.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="7">Materia: Ciencias Naturales 1</th> </tr> <tr> <th colspan="7">Evaluacion</th> </tr> <tr> <th>Código</th> <th>Apellidos</th> <th>Nombres</th> <th>Nota_1</th> <th>Nota_2</th> <th>Nota_3</th> <th>Nota_Final</th> </tr> </thead> <tbody> <tr> <td>9530e</td> <td>Calero</td> <td>Romero Gerardo</td> <td>Alberto</td> <td>9,0</td> <td></td> <td>3,0</td> </tr> </tbody> </table>	Materia: Ciencias Naturales 1							Evaluacion							Código	Apellidos	Nombres	Nota_1	Nota_2	Nota_3	Nota_Final	9530e	Calero	Romero Gerardo	Alberto	9,0		3,0
Materia: Ciencias Naturales 1																													
Evaluacion																													
Código	Apellidos	Nombres	Nota_1	Nota_2	Nota_3	Nota_Final																							
9530e	Calero	Romero Gerardo	Alberto	9,0		3,0																							
<b>Observaciones:</b>	Luego de ingresar las Evaluaciones, el sistema se encarga de actualizar la Nota_Final.																												

Tabla 3.12: Caso de Prueba Administrar RegistroEvaluacionPorEstudiante – caso válido

Compuware  
Sistema de Administración Estudiantil

Buscar Por Materia

Código

---

Materia: Ciencias Naturales 1  
**Evaluación**

Código	Apellidos	Nombres	Nota_1	Nota_2	Nota_3	Nota_Supletorio	Nota_Final	
0530e	Calero	Romero Gerardo Alberto	9,0				3,0	<input type="button" value="Editar"/>

Powered by OptimalJ Copyright © 2008.

Figura 3.12: Caso de Prueba 12

### Caso Prueba 13

Prueba Administrar RegistroEvaluacionesPorEstudiante – caso erróneo	
<b>Nombre de Proceso:</b>	Registrar Evaluaciones Por Estudiante
<b>Descripción de Proceso:</b>	Registrar evaluaciones de los estudiantes.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar RegistroEvaluacionesPorEstudiante.
<b>Entradas:</b>	Numero_Único: CSa6934ds Periodo_Lectivo: 2007-2008  Evaluación: Nota1: 8
<b>Pre – condiciones:</b>	Tener registros de <i>Estudiantes</i> . El <i>Estudiante</i> tenga registro de <i>Matricula</i> .
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar RegistroNotasPorEstudiante</i> . 2.- Ingresar Numero_Único y Periodo_Lectivo especificados en <i>entradas</i> .

	<p>3.- Ejecutar <i>Buscar</i>.</p> <p>4.- Ejecutar <i>Seleccionar</i> en el elemento desplegado.</p> <p>5.- Ejecutar <i>Nuevo</i> en <i>Evaluación</i>.</p> <p>6.- Ingresar el valor de la <i>Evaluación</i> especificada en la <i>entrada</i>.</p> <p>7.- Ejecutar <i>Actualizar</i>.</p>
<b>Resultado:</b>	<p>Error de aplicación</p> <p>Debería corregir los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Asociación <i>Evaluación</i> y <i>Materia</i> es necesaria.</li> </ul> <hr/> <p><b>Error de Aplicación</b></p> <p>Debería corregir el(los) error(es) antes de continuar:</p> <ul style="list-style-type: none"> <li>• Asociación Evaluacion.materia es mandatorio</li> </ul>
<b>Observaciones:</b>	<p>Significa que es necesario añadir una <i>Materia</i> para crear la <i>Evaluación</i>.</p>

Tabla 3.13: Caso de Prueba Administrar RegistroEvaluacionPorEstudiante – caso erróneo

Evaluación	
Vota_1	9,0
Vota_2	
Vota_3	
Vota_Supletorio	
Vota_Final	

---

Estudiante *	
<input type="button" value="Seleccionar"/>	
Numero_Unico *	CS-c972ba0
Nombres	Carlos Alejandro
Apellidos	Guzman Romero

---

Materia *	
<input type="button" value="Seleccionar"/>	
Código *	
Nombre	

**error de Aplicación**

Debería corregir el(los) error(es) antes de continuar:

- Asociación Evaluacion.materia es mandatorio

Powered by OptimalJ Copyright © 2008.

Figura 3.13: Caso de Prueba 13

### Caso Prueba 14

Prueba Administrar RegistroEvaluacionesPorMateria – caso válido	
<b>Nombre de Proceso:</b>	Registrar Evaluaciones Por Materia
<b>Descripción de Proceso:</b>	Registrar evaluaciones de los estudiantes por materia.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará Administrar RegistroEvaluacionesPorMateria.
<b>Entradas:</b>	Periodo_Lectivo: 2007-2008 Código : CCNN1  Estudiante: Numero_Único Nombres Apellidos CS293iejd2 Carlos Alejandro Guzmán Romero



	Evaluación: Nota2: 8																										
<b>Pre – condiciones:</b>	Tener registros de <i>Estudiantes</i> . El <i>Estudiante</i> tenga registro de <i>Matricula</i> .																										
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>Administrar RegistroNotasPorMateria</i> . 2.- Ingresar <i>Periodo_Lectivo</i> y <i>Código</i> especificados en <i>entradas</i> . 3.- Ejecutar <i>Buscar</i> . 4.- Ejecutar la función <i>Editar</i> en el <i>Estudiante</i> con los datos especificados en la <i>entrada</i> . 5.- Ejecutar <i>Actualizar</i> . 6.- Ejecutar <i>Guardar</i> .																										
<b>Resultado:</b>	Al ejecutar el procedimiento 5, se puede visualizar la actualización de la Nota2.  <table border="1"> <tr> <td colspan="6">Estudiante: Calero Romero Gerardo Alberto</td> </tr> <tr> <td colspan="6"><b>Evaluacion</b></td> </tr> <tr> <td><b>Código</b></td> <td><b>Materia</b></td> <td><b>Nota_1</b></td> <td><b>Nota_2</b></td> <td><b>Nota_3</b></td> <td><b>Nota_Supletorio</b></td> <td><b>Nota_Final</b></td> </tr> <tr> <td>ccnn1</td> <td>Ciencias Naturales 1</td> <td>9,0</td> <td></td> <td></td> <td></td> <td>3,0</td> </tr> </table>	Estudiante: Calero Romero Gerardo Alberto						<b>Evaluacion</b>						<b>Código</b>	<b>Materia</b>	<b>Nota_1</b>	<b>Nota_2</b>	<b>Nota_3</b>	<b>Nota_Supletorio</b>	<b>Nota_Final</b>	ccnn1	Ciencias Naturales 1	9,0				3,0
Estudiante: Calero Romero Gerardo Alberto																											
<b>Evaluacion</b>																											
<b>Código</b>	<b>Materia</b>	<b>Nota_1</b>	<b>Nota_2</b>	<b>Nota_3</b>	<b>Nota_Supletorio</b>	<b>Nota_Final</b>																					
ccnn1	Ciencias Naturales 1	9,0				3,0																					
<b>Observaciones:</b>	Luego de ingresar las Evaluaciones, el sistema se encarga de actualizar la Nota_Final.																										

Tabla 3.14: Caso de Prueba Administrar RegistroEvaluacionPorMateria – caso válido

Compuware  
Sistema de Administración Estudiantil

Buscar Por Estudiante

Numero\_Unico:

---

Estudiante: Calero Romero Gerardo Alberto

**Evaluacion**

Código	Materia	Nota_1	Nota_2	Nota_3	Nota_Supletorio	Nota_Final
ccnn1	Ciencias Naturales 1	9,0				3,0

Powered by OptimalJ Copyright © 2008.

Figura 3.144: Caso de Prueba 14

## Caso Prueba 15

Prueba ListarProfesoresPorMateria – caso válido	
<b>Nombre de Proceso:</b>	Listar Profesores Por Materia
<b>Descripción de Proceso:</b>	Desplegar un listado todos los profesores que dicten una Materia.
<b>Escenario:</b>	En la pantalla inicial el usuario seleccionará <i>ListarProfesoresPorMateria</i> .
<b>Entradas:</b>	Código: CCNN1
<b>Pre – condiciones:</b>	Tener registros de <i>Profesores</i> . El <i>Profesor</i> tenga asignada <i>Materias</i> .
<b>Procedimiento:</b>	1.- Menú inicial, seleccionar <i>ListarProfesoresPorMateria</i> . 2.- Ingresar los valores especificados en <i>entradas</i> . 3.- Ejecutar <i>Buscar</i> .
<b>Resultado:</b>	Se visualiza un listado de todos los <i>Profesores</i> de una <i>Materia</i> determinada.
<b>Observaciones:</b>	

Tabla 3.18: Caso de Prueba ListarProfesoresPorMateria – caso válido

Compuware  
Sistema de Administración Estudiantil

Buscar Por Materia

Código

---

Materia: Ciencias Naturales 1

**Evaluación**

Código	Apellidos	Nombres	Nota_1	Nota_2	Nota_3	Nota_Supletorio	Nota_Final
9530e	Calero Romero	Gerardo Alberto	9,0				3,0

Powered by OptimalJ Copyright © 2008.

Figura 3.15: Caso de Prueba 15

## **CAPÍTULO 4:**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **4.1. CONCLUSIONES**

- Con MDD, se logra minimizar el tiempo de codificación ya que enfatiza mucho más en la etapa de diseño que es la base de la ingeniería.
- A pesar de que no es aconsejable añadir requerimientos en la etapa de pruebas, MDD permite realizarla ahorrando recursos, ya que introducir mejoras en los modelos es más sencillo, que hacerlo en la codificación.
- La creación del PIM depende de la abstracción de los desarrolladores, sin embargo la transformación a PSM's y a código, están sujetos a los patrones de transformaciones propios de cada herramienta MDD.
- Una buena interpretación de los requerimientos en la creación del PIM, permite obtener los modelos con menor susceptibilidad a cambios, sin embargo esto no libera de realizar modificaciones en los modelos para el cumplimiento de determinados requerimientos específicos.
- Para el cumplimiento de los requerimientos, es importante contar con herramientas que permitan el mantenimiento de los modelos y código generado.

## 4.2. RECOMENDACIONES

- Es recomendable profundizar el estudio de todos los elementos UML para crear un PIM completo, que permita obtener el sistema solicitado por el usuario.
- Es recomendable realizar estudios sobre la aplicación de metodologías tradicionales en MDD.
- Es recomendable realizar un análisis dirigido a encontrar nuevos parámetros para la evaluación de herramientas MDD.
- Es recomendable investigar toda la funcionalidad de OptimalJ, para aprovechar al máximo las propiedades para la creación de sistemas.
- Es recomendable, para una óptima transformación de modelos y código, realizar un análisis de nombres para cada componente a ser creado en las herramientas MDD.

## BIBLIOGRAFÍA

- JACOBSON I., BOOCH G., RUMBAUGH J., “El proceso unificado de desarrollo de software”, pearson education, edicion 1, 2000.
- KEPPLER A, WARMER J, BAST WIM, “MDA explained: the model driven architecture – practice and promise”, paperback, 2003
- FRANKEL David, “model driven architecture: applying mda to enterprise computing”, paperback, 2003.
- MANUALES DE ARCSTYLER. Interactive objects. [www.arcstyler.com](http://www.arcstyler.com) año: 2006.
- MANUALS DE OPTIMALJ. Compuware. [www.compuware.com/optimalj](http://www.compuware.com/optimalj) año:2006
- OMG: guia de MDD [ww.omg.org/mdd](http://ww.omg.org/mdd) año: 1997 – 2005 (Copyright).
- META OBJECT FACILITY (MOF). Año: 1997 – 2005 (Copyright) [www.omg.org/technology/documents/modelingspeccatalog.htm#M](http://www.omg.org/technology/documents/modelingspeccatalog.htm#M)
- OBJECT CONSTRAINT LANGUAGE (OCL). Año 2005 [www.csci.csusb.edu/dic/samples/ocl.html](http://www.csci.csusb.edu/dic/samples/ocl.html)

**ANEXOS.**

# **Anexo I**

## **Plantilla Documento de Requerimientos**

<< Logo de la(s) empresa(s)  
Involucrado(s) en el proyecto>>

<< Nombre de la(s) empresa(s) >>

---

<< Nombre del Proyecto >>

---

## Documento de Especificación de Requerimientos

Versión 1.0

Desarrollada por:

Aprobada por:

Fecha:



## Historial de Revisiones

---

Nombre	Fecha	Descripción de la modificación	Versión

# 1. INTRODUCCIÓN

---

## 1.1. Ámbito

<< Definir el ámbito de desarrollo del software. >>

## 1.2. Definición del sistema

<< Realizar una breve descripción del sistema de gestión de noticias – qué han decidido implementar – >>

## 1.3. Objetivos Generales

<< Aquí deben incluirse los objetivos generales del software a desarrollar. >>

## 1.4. Entorno De Operación

<< Realizar una descripción del entorno en el cual va a operar el software, incluyendo la plataforma software, el sistema operativo, las versiones y el modo de acceso. >>

## 1.5. Convenciones

<< Determinar cualquier estándar o convención tipográfica a utilizar dentro de este documento, tales como negritas, itálica o subrayado, para indicar algún concepto de especial significancia. >>

## 1.6. Material De Referencia

<< Describir todo el material que se ha utilizado para determinar los objetivos a alcanzar durante la asignatura; servicios similares, páginas web, hemerotecas, libros de referencia, etc. >>

## 2. REQUERIMIENTO FUNCIONALES DEL SISTEMA

---

<< En esta sección se organizan los requisitos funcionales del software. Estos deben estar organizados por las características principales del sistema. >>

<< Listar el conjunto de características principales que definen el sistema, o las funciones que se ejecutan dentro del sistema. Una descripción mas detallada de estas características o funcionalidades debe ser proporcionada en la sección 3. >>

### 2.1. Funcionalidad Del Sistema # 1

#### 2.1.1 Descripción

<< Proporcionar una breve descripción de la funcionalidad que se desea desarrollar. >>

#### 2.1.2 Requisitos Funcionales

<< Listar todos los requisitos funcionales que están asociados con la funcionalidad descrita en este apartado. Debe incluirse las precondiciones necesarias para la ejecución, las poscondiciones (o como modifica la funcionalidad la entrada), como responder a entradas invalidas o errores. Estos requisitos funcionales deben ser concisos, completos, no ambiguos y verificables.

Cada requisito funcional debe estar identificado por un número clave o etiqueta significativa: >>

Req-1:

Req-2:

### 2.2. Funcionalidad Del Sistema # 2

### 3. REQUISITOS DE INTERFAZ

---

#### 3.1. Interfaces De Usuario

<< Describir las características de las interfaces entre el usuario y el sistema. Esto puede incluir acceso a las funcionalidades, mensajes de error, etc. >>

#### 3.2. Interfaces De Software

<< Describir las conexiones de la aplicación con otros componentes software (nombre y versión) que se utilicen para la ejecución del sistema, incluyendo bases de datos, sistemas operativos, programas de conexión específicos (Putty, XTerm, ...) o compiladores, especificando para cada uno de ellos la importancia que tiene en el producto software final. >>

# **Anexo II**

## **Plantilla Documento Casos de Uso**

### Plantilla 1: Inserción <<elemento>>

Parámetros:

- Elemento: ítem a ser insertado. Un ítem esta compuesto por los atributos que lo definen. Esto es: atributo clave y atributos.
- Atributo clave: las propiedades que identifican al elemento unívocamente.
- Atributos: propiedades que componen el elemento.
- Reglas de negocio (r1,...,rn).

Nombre: Inserción de <<elemento>>.

Pre-condición: existe un <<elemento>> a ser ingresado.

Post-condición: el <<elemento>>, queda registrado en el sistema, o el <<elemento>> ya estaba registrado en el sistema.

Descripción: realiza la inserción de un <<elemento>>, controlando la existencia del elemento en el sistema y el cumplimiento de las reglas del negocio (r1,...,rn) asociadas al << elemento >>.

ACTOR	SISTEMA
<p>6. Ingresar &lt;&lt; atributos clave&gt;&gt; del &lt;&lt; elemento &gt;&gt;.</p> <p>3. Ingresar el resto de los &lt;&lt; atributos &gt;&gt; del &lt;&lt; elemento &gt;&gt;.</p>	<p>2. Verifica la existencia por &lt;&lt; atributos clave&gt;&gt;.</p> <p>4. Verifica corrección de los &lt;&lt;atributos&gt;&gt; ingresados.</p> <p>5. Verifica reglas de negocio &lt;&lt;r1,..rn&gt;&gt;, asociadas al caso de uso.</p> <p>6. Realiza el alta del &lt;&lt; elemento &gt;&gt;.</p>

2.1 El sistema informa de la existencia del << elemento >> identificado con << atributos clave >>.

4.1 El sistema informa que al menos uno de los << atributos >> ingresado no es correcto.

5.1 El sistema informa las reglas **ri** que no se verifican (con  $1 < i < n$ ).

Nota: En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

Plantilla 2: Modificación de <<elemento>>

Parámetros:

- Elemento: ítem a modificar. Un ítem esta compuesto por los atributos que lo definen. Esto es: atributo clave y atributos.
- Atributo clave: las propiedades que identifican al elemento unívocamente.
- Atributos: propiedades que componen el elemento.
- Reglas de negocio ( $r_1, \dots, r_n$ ).

Nombre : Modificación de <<elemento>>.

Pre-condición: hay << atributos >> de un <<elemento>> a ser modificado.

Post-condición: el <<elemento>> modificado queda registrado en el sistema, o el <<elemento>> a modificar no existía en el sistema.

Descripción: realiza la modificación de los << atributos >> de un <<elemento>>, controlando el cumplimiento de las reglas del negocio ( $r_1, \dots, r_n$ ) asociadas al << elemento >>.

ACTOR	SISTEMA
<p>3. Modifica los &lt;&lt; atributos &gt;&gt; del &lt;&lt; elemento &gt;&gt;.</p>	<p>1. Incluye (<b>buscar[elemento]</b>).</p> <p>2. Muestra los &lt;&lt; atributos &gt;&gt; del &lt;&lt; elemento &gt;&gt; seleccionado.</p> <p>4. Verifica corrección de &lt;&lt; atributos &gt;&gt; modificados.</p> <p>5. Verifica reglas de negocio &lt;&lt; <math>r_1, \dots, r_n</math> &gt;&gt; asociadas al caso de uso.</p> <p>6. Realiza la actualización del &lt;&lt; elemento &gt;&gt;.</p>

4.1 El sistema informa que al menos uno de los << atributos >> ingresado no es correcto.

5.1 El sistema informa las reglas  $r_i$  que no se verifican (con  $1 < i < n$ ).

Nota: En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

### Plantilla 3: Eliminación de <<elemento>>

Parámetros:

- Elemento: ítem a eliminar del sistema. Un ítem esta compuesto por los atributos que lo definen. Esto es: atributo clave y atributos.
- Atributo clave: las propiedades que identifican al elemento unívocamente.
- Atributos: propiedades que componen el elemento.
- Reglas de negocio ( $r_1, \dots, r_n$ ).

Nombre: Eliminación de <<elemento>>.

Pre-condición: existe un <<elemento>> a ser eliminado.

Post-condición: el <<elemento>> fue eliminado del sistema, o el <<elemento>> a eliminar no existía en el sistema.

Descripción: realiza la eliminación de un <<elemento>>, controlando la existencia del elemento en el sistema y el cumplimiento de las reglas del negocio ( $r_1, \dots, r_n$ ) asociadas al << elemento >>.

ACTOR	SISTEMA
<p>3. Confirma la eliminación del &lt;&lt; elemento &gt;&gt;.</p>	<p>1. Include(<b>buscar[elemento]</b>).</p> <p>2. Muestra los &lt;&lt; atributos &gt;&gt; del &lt;&lt; elemento &gt;&gt;seleccionado.</p> <p>4. Verifica reglas de negocio &lt;&lt; <math>r_1, \dots, r_n</math> &gt;&gt; asociadas al caso de uso.</p> <p>Realiza la eliminación del &lt;&lt; elemento &gt;&gt;.</p>



4.1 El sistema informa las reglas  $r_i$  que no se verifican (con  $1 < i < n$ ).

Nota: En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

### Plantilla 3: Búsqueda de <<elemento>>

Parámetros:

- Elemento: objeto a buscar en el sistema. Un objeto esta conformado por los atributos que lo definen: atributo/s clave y atributos.
- Atributo/s clave: las propiedades que identifican unívocamente al objeto.
- Atributos: propiedades que componen el elemento.
- Criterio de búsqueda: atributos por los cuales se desea realizar la búsqueda.

Nombre: Búsqueda de <<elemento>>.

Pre-condición: existe un <<elemento>> a ser buscado en el sistema.

Post-condición: el <<elemento>> fue encontrado, o no existía el <<elemento>> en el sistema.

Descripción: realiza la búsqueda de un <<elemento>> en el sistema, controla que existencia el << elemento >> en el sistema.

ACTOR	SISTEMA
<p>3. Selecciona el &lt;&lt; criterio de búsqueda &gt;&gt;.</p> <p>4. Ingresar &lt;&lt; atributos &gt;&gt; adecuados al &lt;&lt; criterio de búsqueda &gt;&gt;.</p> <p>4. Confirma que el &lt;&lt; elemento &gt;&gt; encontrado es el buscado.</p>	<p>5. Realiza la búsqueda por &lt;&lt; atributos &gt;&gt; según &lt;&lt; criterio de búsqueda &gt;&gt;.</p>

4.1 El << elemento >> buscado no está registrado en el sistema.

Nota: En cualquier momento el usuario puede cancelar la ejecución del caso de uso.

# **Anexo III**

## **Plantilla Documento Clases y Recomendaciones**

## Clases

<b>Nombre</b>			
<< Clase - Estructura >>			
<b>Descripción</b>			
<< Descripción de la clase >>			
<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
<<atributo1>>	<<tipo atributo1>>	<<tamaño atributo1>>	<< Descripción del atributo >>

## Relaciones

<b>Nombre</b>				
<< Nombre de la relación >>				
<b>Descripción</b>				
<< Descripción de la clase >>				
<b>Clases Relacionadas</b>	<< Clases relacionadas >>			
<b>Dirección</b>	<< Tipo de direccionalidad de la relación >>			
<b>Cardinalidad</b>	<< Especificar la cardinalidad en las clases de la relación >>			
<b>Asociación</b>	<< En caso de existir una asociación, especificar el nombre >>			
	<b>Atributos</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
	<<nombre atributo1>>	<<tipo atributo1>>	<<tamaño atributo1 >>	<<descripción atributo1>>