

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



DESARROLLO DE UN SISTEMA PARA GENERAR LA AUTOMATIZACIÓN  
DE LOS PROCESOS ACADÉMICOS Y ADMINISTRATIVOS VINCULADOS A  
LA I Y II ETAPA DE EDUCACIÓN BÁSICA DENTRO DE UNA UNIDAD  
EDUCATIVA

GIMENEZ ANDARCIA ENDRIKSON JOSÉ

C. I.: 13.689.716

TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA  
OPTAR AL TÍTULO DE INGENIERO EN COMPUTACIÓN

PUERTO LA CRUZ, FEBRERO DE 2009

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



DESARROLLO DE UN SISTEMA PARA GENERAR LA AUTOMATIZACIÓN  
DE LOS PROCESOS ACADÉMICOS Y ADMINISTRATIVOS VINCULADOS A  
LA I Y II ETAPA DE EDUCACIÓN BÁSICA DENTRO DE UNA UNIDAD  
EDUCATIVA

---

M.Sc. María Gerardino  
(Jurado Principal)

---

Ing. Pedro Dorta  
(Jurado Principal)

---

Ing. Yulitza Mujica  
(Asesor Académico)

## **ARTÍCULO 44**

De acuerdo con el artículo 44 del Reglamento de Trabajo de Grado: “Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario.”

## **DEDICATORIA**

Al Padre, Hijo y Espíritu Santo. A mi Padre Celestial, Señor y Creador de mi vida. A Jesucristo mi Salvador, quien dio su vida hasta derramar su sangre y morir por mí en la cruz. Al Espíritu Santo, mi Amigo Fiel, quién nunca me dejó solo en este largo trayecto, revelándome todo lo que debía hacer para continuar el camino.

A mis padres, Juan Bautista Gimenez y Carmen Luisa de Gimenez, quienes han sido incomparable apoyo y máxima inspiración.

A mis Apóstoles y Pastores, Alberto Cabeza y Noemí Luna porque fueron abrigo y fortaleza.

A los jóvenes del Centro Cristiano Piedra Viva, porque ellos siempre me inspiraron en terminar esta carrera y hacerlo con excelencia.

## **AGRADECIMIENTO**

Gracias Dios por estar siempre presente en mi vida, porque aunque estuve alejado un tiempo de Ti, reconocí la verdad de ser parte principal en mí existir y me diste la “Victoria”. Te amo Dios.

Gracias a mis bendecidos padres Carmen Luisa y Juan Gimenez por su paciencia hacia mí, en esperar el tiempo para culminar este proyecto. Fueron ustedes quienes me enseñaron a luchar. Gracias. Los amo incondicionalmente.

Gracias Pastores Alberto Cabeza y Noemí Luna, por darme refugio en su hogar, por creer en mí, por apoyarme constantemente y alentarme en los momentos más difíciles. Gracias por sus oraciones.

Gracias a mi hermano Franklin Gimenez por regalarme la computadora en el momento preciso y cuando más la necesitaba en mi carrera; debido a ese regalo pude terminar a tiempo. A mis demás hermanos Jaider, Hecmer, Noreglis, Ogdeglis y Migdeglis Gimenez por estar siempre pendiente de mi y extender su apoyo fuertemente para que no me sintiera solo.

Gracias Mariana y Josué, mis hermanitos preciosos que Dios me regaló, por orar constantemente por mí y estar al pendiente sobre mi presentación.

Gracias a mi mejor amigo y hermano Luis Martínez por soportarme cuando estaba a punto de abandonar, gracias por tus palabras alentadoras, tu apoyo incondicional, por llorar conmigo y además muchísimas gracias por orar tan insistentemente creyendo que Dios haría el trabajo. Gracias, “papito”. Estoy seguro de que Dios tiene todo lo mejor para ti.

Gracias a mi preciosa novia Patricia Díaz por entenderme en los momentos más difíciles, gracias por saber esperar, gracias por tus oraciones y constante credibilidad en el proyecto de vida. Gracias por creer en mí, además de hacerte parte de mi proyecto al decir que era también tuyo. Gracias por regalarme momentos de alegrías cuando pesaba mucho cansancio. Gracias por prestarme tu computador portátil.

Gracias a mi asesora la Ingeniero Yulitza Mujica por ser tan excelente guiadora hasta la culminación de mi proyecto, su asesoramiento fue impecable. Gracias por atenderme cuando no era su obligación pues estaba de permiso postnatal. Dios bendiga a su bebita y declaro una gran bendición en su vida.

Gracias a la gran familia del Centro Cristiano Piedra Viva por sus oraciones, ya que ellas arrojaron esta “Gran Victoria”.

Gracias al Pastor Luis Rodríguez por regalarme el monitor que necesitaba para mi computador.

Gracias a la familia de mi novia, a María, Robert y la abuela Inés por su apoyo y oraciones.

Gracias a la Ingeniero Yris Ochoa por apoyarme durante y después de nuestra carrera universitaria. Tú eres una excelente persona. Gracias. También a las Ingenieras Bethsy e Yrma quienes me hicieron pasar momentos muy divertidos en los tiempos de clases dentro y fuera de las aulas. Ustedes son buenas amigas.

Gracias Profesor José Luis Bastardo, usted demostró ser un excelente amigo.

Y a todos los que siempre me apoyaron, simplemente, GRACIAS.

# ÍNDICE

ARTÍCULO 44 .....	III
DEDICATORIA .....	IV
AGRADECIMIENTO .....	V
ÍNDICE .....	VII
ÍNDICE DE FIGURAS.....	XIII
RESUMEN.....	XIX
INTRODUCCIÓN .....	XX
CAPÍTULO 1 .....	21
PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS .....	21
1.1. Planteamiento del Problema.....	21
1.2. Objetivos .....	25
1.2.1. Objetivo General .....	25
1.2.2. Objetivos Específicos.....	25
CAPÍTULO 2 .....	26
ANTECEDENTES Y MARCO TEÓRICO.....	26
2.1. Antecedentes .....	26
2.2. Marco Teórico.....	27
2.2.1. Software .....	27
2.2.2. Ingeniería de Software. ....	27
2.2.3. Programación orientada a objetos .....	27
2.2.3.1. Características de lo Orientado a Objetos .....	28
2.2.3.2. Fundamentos de lo Orientado a Objetos .....	29
2.2.4. Proceso Unificado .....	30
2.2.4.1. Fases del Proceso Unificado .....	31
2.2.4.2. Flujos de trabajo del Proceso Unificado .....	31
2.2.5. Lenguaje Unificado de Modelado (UML, Unified Modeling Language)..	32

2.2.5.1. Diagramas de UML.....	33
2.2.6. Base de Datos.....	35
2.2.6.1. Sistema Manejador de Base de Datos (SMBD).....	36
2.2.6.2. Base de Datos Relacional.....	37
2.2.6.3. Lenguajes de Consulta en el SMBD.....	37
2.2.7. Administración.....	37
2.2.7.1. Proceso Administrativo.....	38
2.2.8. Sistema Automatizado.....	38
CAPÍTULO 3.....	40
FASE DE INICIO.....	40
3.1. Introducción.....	40
3.2. Captura de Requisitos.....	40
3.2.1. Lista Inicial de Requisitos.....	40
3.3. Modelo de Dominio.....	43
3.4. Glosario de Términos.....	43
3.5. Identificación y Descripción de Riesgos.....	45
3.5.1. Lista de Riesgos Críticos.....	45
3.5.1.1. Riesgo: Procesar Alumnos.....	45
3.5.1.2. Riesgo: Procesar Empleados.....	46
3.5.1.3. Riesgo: Configurar Sistema.....	46
3.5.1.4. Riesgo: Constancias y Reportes.....	47
3.5.1.5. Riesgos: Ingresos y Egresos.....	47
3.6. Modelo de Casos de Uso.....	47
3.6.1. Actores.....	48
3.6.1.1. Identificación de Actores.....	48
3.6.2. Casos de Uso.....	49
3.6.2.1. Identificación de Casos de Uso.....	49
3.6.2.2. Diagrama de Caso de Uso.....	50
3.6.2.3. Caso de Uso Procesar Alumnos.....	52



3.6.2.4. Caso de Uso Procesar Empleados .....	54
3.6.2.5. Caso de Uso Ingresos y Egresos .....	56
3.6.2.6. Caso de Uso Constancias y Reportes .....	58
3.6.2.7. Caso de Uso Mostrar Ayuda .....	59
3.6.2.8. Caso de Uso Calificaciones.....	60
3.6.2.9. Caso de Uso Configurar Sistema .....	61
3.7. Análisis.....	63
3.7.1. Análisis de la Arquitectura.....	64
3.7.2. Modelo de Análisis .....	64
3.7.2.1. Paquete del Análisis del Caso de Uso Procesar Alumnos .....	65
3.7.2.2. Paquete del Análisis del Caso de Uso Procesar Empleados .....	65
3.7.2.3. Paquete del Análisis del Caso de Uso Calificaciones .....	66
3.7.2.4. Paquete del Análisis del Caso de Uso Mostrar Ayuda.....	66
3.7.2.5. Paquete del Análisis del Caso de Uso Ingresos y Egresos.....	66
3.7.2.6. Paquete del Análisis del Caso de Uso Constancias y Reportes .....	67
3.7.2.7. Paquete del análisis del Caso de Uso Configurar Sistema.....	67
3.7.3. Diagrama de Clases de Análisis.....	68
3.7.3.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Alumnos	68
3.7.4. Diagrama de Colaboración.....	70
3.7.4.1. Diagrama de Colaboración del Caso de Uso Procesar Alumnos .....	70
3.8. Diseño .....	72
3.8.1. Modelo de Diseño .....	72
3.8.2. Subsistemas de Diseño.....	73
3.8.2.1. Subsistema de Diseño de Gestión de Procesar Alumnos. ....	74
3.8.2.2. Subsistema de Diseño de Gestión de Procesar Empleados.....	74
3.8.2.3. Subsistema de Diseño de Gestión de Ingresos y Egresos .....	75
3.8.2.4. Subsistema de Diseño de Gestión de Mostrar Ayuda .....	75
3.8.2.5. Subsistema de Diseño de Gestión de Constancias y Reportes.....	75
3.8.2.6. Subsistema de Diseño de Gestión de Calificaciones .....	76

3.8.2.7. Subsistema de Diseño de Gestión de Configurar Sistema .....	76
3.8.3. Diagrama de Clases del Diseño.....	77
3.8.3.1. Realización del Caso de Uso Procesar Alumnos .....	77
3.8.4. Prototipos de Interfaz de Usuario.....	79
3.8.4.1. Prototipo de Interfaz Principal .....	80
3.8.4.2. Prototipo de Interfaz de Gestión .....	80
3.8.5. Nodos y Configuraciones de Red.....	81
3.9. Planificación de las Sigüientes Fases .....	82
3.10. Resumen de la Fase de Inicio.....	83
CAPÍTULO 4.....	84
FASE DE ELABORACIÓN .....	84
4.1. Introducción .....	84
4.2. Modelo de Casos de Uso.....	84
4.3. Análisis.....	84
4.3.1. Análisis de la Arquitectura.....	85
4.3.2. Modelo de Análisis .....	85
4.3.3. Diagrama de Clase de Análisis .....	85
4.3.3.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Empleados .....	85
4.3.3.2. Diagrama de Clases de Análisis del Caso de Uso Ingresos y Egresos .....	87
4.3.3.3. Diagrama de Clases de Análisis del Caso de Uso Constancias y Reportes .....	88
4.3.3.4. Diagrama de Clases de Análisis del Caso de Uso Mostrar Ayuda ....	90
4.3.3.5. Diagrama de Clases de Análisis del Caso de Uso Calificaciones.....	91
4.3.3.6. Diagrama de Clases de Análisis del Caso de Uso Configurar Sistema .....	92
4.3.4. Diagrama de Colaboración.....	93
4.3.4.1. Diagrama de Colaboración del Caso de Uso Procesar Empleados ....	93

4.3.4.2. Diagrama de Colaboración del Caso de Uso Ingresos y Egresos .....	95
4.3.4.3. Diagrama de Colaboración del Caso de Uso Constancias y Reportes	96
4.3.4.4. Diagrama de Colaboración del Caso de Uso Calificaciones.....	98
4.3.4.5. Diagrama de Colaboración del Caso de Uso Mostrar Ayuda .....	99
4.3.4.6. Diagrama de Colaboración del Caso de Uso Configurar Sistema ...	100
4.4. Diseño .....	101
4.4.1. Subsistemas de Diseño.....	101
4.4.2. Dependencias y Capas del Diseño .....	102
4.4.3. Diagrama de Clases del Diseño.....	103
4.4.3.1. Realización del Caso de Uso Procesar Empleados .....	104
4.4.3.2. Realización del Caso de Uso Ingresos y Egresos.....	105
4.4.3.3. Realización del caso Constancias y Reportes .....	107
4.4.3.4. Realización del Caso de Uso Calificaciones .....	110
4.4.3.5. Realización del Caso de Uso Configurar Sistema.....	111
4.4.3.6. Realización del Caso de Uso Mostrar Ayuda .....	113
4.4.4. Diseño de la Base de Datos.....	114
4.4.4.1. Descripción de Tablas .....	115
4.5. Implementación.....	121
4.5.1. Implementación del Caso de Uso Procesar Alumnos. ....	121
4.5.2. Diagrama del Modelo Relacional.....	185
4.6. Planificación de las Sigüientes Fases.....	191
4.7. Resumen de la Fase de Elaboración.....	192
CAPÍTULO 5 .....	193
FASE DE CONSTRUCCIÓN .....	193
5.1. Introducción .....	193
5.2. Implementación.....	193
5.2.1. Implementación del Caso de Uso Procesar Empleados .....	194
5.2.2. Implementación del Caso de Uso Ingresos y Egresos .....	209
5.2.3. Implementación del Caso de Uso Calificaciones.....	230

5.2.4. Implementación del Caso de Uso Mostrar Ayuda .....	240
5.2.5. Implementación del Caso de Uso Configurar Sistema .....	243
5.2.6. Implementación del Caso de Uso Constancias y Reportes. ....	258
5.3. Prueba.....	283
5.3.1. Prueba del Caso de Uso Procesar Alumnos .....	283
5.2.2. Prueba del Caso de Uso Ingresos y Egresos .....	285
5.4. Planificación de las Sigüientes Fases.....	287
5.5. Resumen de la Fase de Construcción.....	287
CAPÍTULO 6 .....	289
FASE DE TRANSICIÓN .....	289
6.1. Introducción .....	289
6.2. Resumen de la Fase de Transición.....	289
CAPÍTULO 7 .....	290
MANUAL DE USUARIO .....	290
7.1. Introducción .....	290
7.2. Interfaz Principal.....	290
7.2.1. Menú Configurar Sistema .....	290
7.2.2. Menú Datos Personales .....	293
7.2.3. Menú Calificaciones .....	303
7.2.4. Menú Ingresos y Egresos. ....	306
7.2.5. Menú Constancias y Reportes.....	310
7.2.6. Menú Ayuda.....	321
CONCLUSIONES .....	323
RECOMENDACIONES.....	324
BIBLIOGRAFÍA .....	325

## ÍNDICE DE FIGURAS

Figura 3.1. Modelo de Dominio del Sistema Actual.....	43
Figura 3.2. Modelo de Casos de Uso .....	51
Figura 3.3. Paquetes del Análisis .....	64
Figura 3.4. Paquete del Análisis del Caso de Uso Procesar Alumnos .....	65
Figura 3.5. Paquete del Análisis del Caso de Uso Procesar Empleados .....	65
Figura 3.6. Paquete del Análisis del Caso de Uso Calificaciones.....	66
Figura 3.7. Paquete del Análisis del Caso de Uso Mostrar Ayuda. ....	66
Figura 3.8. Paquete del Análisis del Caso de Uso Ingresos y Egresos .....	67
Figura 3.9. Paquete del Análisis del Caso de Uso Constancias y Reportes.....	67
Figura 3.10. Paquete del Análisis del Caso de Uso Configurar Sistema. ....	68
Figura 3.11. Diagrama de Clases de Análisis Procesar Alumnos.....	68
Figura 3.12. Diagrama de Colaboración del Caso de Uso Procesar Alumnos.....	72
Figura 3.13. Subsistemas de Diseño. ....	73
Figura 3.14. Subsistemas de Diseño de Gestión de Procesar Alumnos.....	74
Figura 3.15. Subsistemas de Diseño de Gestión de Procesar Empleados.....	74
Figura 3.16. Subsistemas de Diseño de Gestión de Ingresos y Egresos. ....	75
Figura 3.17. Subsistemas de Diseño de Gestión de Mostrar Ayuda. ....	75
Figura 3.18. Subsistemas de Diseño de Gestión de Constancias y Reportes.....	76
Figura 3.19. Subsistemas de Diseño de Gestión de Calificaciones.....	76
Figura 3.20. Subsistemas de Diseño de Gestión de Configurar Sistema. ....	77
Figura 3.21. Diagrama de Clases del Diseño del Caso de Uso Procesar Alumnos.....	78
Figura 3.22. Diagrama de Secuencia del Caso de Uso Procesar Alumnos. ....	78
Figura 3.23. Prototipo de Interfaz Principal.....	80
Figura 3.23. Prototipo de Interfaz de Gestión.....	81
Figura 3.24. Identificación de Nodos y Configuraciones de Red. ....	82
Figura 4.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Empleados...	86

Figura 4.2. Diagrama de Clases de Análisis del Caso de Uso Ingresos y Egresos. ....	88
Figura 4.3. Diagrama de Clases de Análisis del Caso de Uso Constancias y Reportes. 89	
Figura 4.4. Diagrama de Clases de Análisis del Caso de Uso Mostrar Ayuda. ....	90
Figura 4.5. Diagrama de Clases de Análisis del Caso de Uso Calificaciones. ....	91
Figura 4.6. Diagrama de Clases de Análisis del Caso de Uso Configurar Sistema. ...	92
Figura 4.7. Diagrama de Colaboración del Caso de Uso Procesar Empleados.....	94
Figura 4.8. Diagrama de Colaboración del Caso de Uso Ingresos y Egresos.....	96
Figura 4.9. Diagrama de Colaboración del Caso de Uso Constancias y Reportes.....	97
Figura 4.10. Diagrama de Colaboración del Caso de Uso Calificaciones. ....	98
Figura 4.11. Diagrama de Colaboración del Caso de Uso Mostrar Ayuda.....	99
Figura 4.12. Diagrama de Colaboración del Caso de Uso Configurar Sistema. ....	100
Figura 4.13. Subsistema de Servicio Gestores .....	102
Figura 4.14. Dependencias y Capas de Diseño. ....	102
Figura 4.15. Diagrama de Clases del Diseño del Caso de Uso Procesar Empleados	104
Figura 4.16. Diagrama de Secuencia del Caso de Uso Procesar Empleados .....	105
Figura 4.17. Diagrama de Clases del Diseño del Caso de Uso Ingresos y Egresos..	106
Figura 4.18. Diagrama de Secuencia del Caso de Uso Ingresos y Egresos. ....	107
Figura 4.19. Diagrama de Clases del Diseño del Caso de Uso Constancias y Reportes. 108	
Figura 4.20. Diagrama de Secuencia del Caso de Uso Constancias y Reportes. ....	109
Figura 4.21. Diagrama de Clases del Diseño del Caso de Uso Calificaciones .....	110
Figura 4.22. Diagrama de Secuencia del Caso de Uso Constancias y Reportes. ....	111
Figura 4.23. Diagrama de Clases del Diseño del Caso de Uso Configurar Sistema.	112
Figura 4.24. Diagrama de Secuencia del Caso de Uso Configurar Sistema. ....	113
Figura 4.25. Diagrama de Clases del Diseño del Caso de Uso Mostrar Ayuda.....	113
Figura 4.26. Diagrama de Secuencia del Caso de Uso Mostrar Ayuda. ....	114
Figura 4.27. Pantalla de Ingresar Alumno .....	122
Figura 4.28. Pantalla de Ingresar datos de Padres y Representantes. ....	123

Figura 4.29. Pantalla de Ingresar Pagos por concepto de inscripción.....	123
Figura 4.30. Pantalla de Consultar Alumno .....	124
Figura 4.31. Pantalla de Retirar Alumno .....	124
Figura 4.32. Pantalla de Modificar Alumno.....	125
Figura 4.33. Diagrama del Modelo Relacional de la Base de Datos de SAPSEE. ...	185
Figura 5.1. Pantalla de Ingresar Empleado. ....	194
Figura 5.2. Pantalla para Consultar Empleado.....	195
Figura 5.3. Pantalla para Modificar Empleado .....	196
Figura 5.4. Pantalla para Retirar Empleado. ....	196
Figura 5.5. Pantalla de Ingreso de Pago de Matrícula.....	209
Figura 5.6. Pantalla de Consultar Matrícula. ....	221
Figura 5.7. Pantalla de Ingresar Pago de Nómina.....	222
Figura 5.8. Pantalla para Consultar Pagos de Nómina.....	230
Figura 5.9. Pantalla para Ingresar Calificaciones.....	231
Figura 5.10. Pantalla para Ingresar Calificaciones Históricas. ....	239
Figura 5.11. Pantalla para Consultar Calificaciones. ....	239
Figura 5.12. Pantalla Acerca de SAPSEE.....	240
Figura 5.13. Pantalla para Iniciar Sistema .....	244
Figura 5.14. Pantalla para Ingresar Directivos del Plantel.....	244
Figura 5.15. Pantalla de Configurar Asignaturas.....	250
Figura 5.16. Pantalla de Configurar Documentos.....	253
Figura 5.17. Pantalla para generar Constancias de Inscripción. ....	258
Figura 5.18. Pantalla para generar Constancias de Estudios.....	259
Figura 5.19. Pantalla para generar Boletín Informativo.....	259
Figura 5.20. Pantalla para generar el Reporte de Alumnos por Grado .....	260
Figura 5.21. Pantalla para generar el Reporte de Alumnos por Sexo .....	260
Figura 5.22. Pantalla para generar el Reporte de Alumnos por Año Escolar. ....	261
Figura 5.23. Pantalla para generar el Reporte de Ingresos Diarios.....	261
Figura 5.24. Pantalla para generar el Reporte de Ingresos Mensuales.....	262

Figura 5.25. Pantalla para generar el Reporte de Ingresos Anuales.....	262
Figura 5.26. Pantalla para generar el Reporte de Egresos Diarios.....	263
Figura 5.27. Pantalla para generar el Reporte de Egresos Mensuales. ....	263
Figura 5.28. Pantalla para generar Reportes de Egresos Anuales.....	264
Figura 5.29. Pantalla para generar Reporte de los Empleados del Plantel.....	264
Figura 5.30. Clases de Equivalencia del Caso de Uso Procesar Alumnos.....	284
Figura 5.31. Prueba de Caja Negra del Caso de Uso Procesar Alumnos.....	285
Figura 5.32. Clases de Equivalencia del Caso de Uso Ingresos y Egresos.....	286
Figura 5.33. Prueba de Caja Negra del Caso de Uso Ingresos y Egresos. ....	287
Figura 7.1. Interfaz Principal del Sistema SAPSEE. ....	290
Figura 7.2. Menú para Configurar el Sistema. ....	291
Figura 7.3. Interfaz para Iniciar el Sistema. ....	291
Figura 7.4. Interfaz para Configurar Documentos. ....	292
Figura 7.5. Interfaz para Configurar Asignaturas. ....	293
Figura 7.6. Menú Datos Personales.....	294
Figura 7.7. Submenú Inscribir Alumno.....	294
Figura 7.8. Interfaz para Ingresar Alumnos Nuevos. ....	295
Figura 7.9. Interfaz para Ingresar datos de los Padres y Representantes. ....	295
Figura 7.10. Submenú Consultar Alumno. ....	296
Figura 7.11. Interfaz para Consultar los datos de los alumnos. ....	296
Figura 7.12. Submenú Retirar Alumno. ....	297
Figura 7.13. Interfaz para Retirar Alumnos del plantel. ....	297
Figura 7.14. Submenú Regulares. ....	298
Figura 7.15. Interfaz para Inscribir Alumnos Regulares.....	298
Figura 7.16. Submenú Ingresar Empleado.....	299
Figura 7.17. Interfaz para Ingresar Empleados. ....	299
Figura 7.18. Submenú Consultar Empleado. ....	300
Figura 7.19. Interfaz para Consultar los datos de los Alumnos. ....	300
Figura 7.20. Submenú Retirar Empleado.....	301



Figura 7.21. Interfaz para Retirar Empleados del Plantel. ....	301
Figura 7.22. Submenú Actualizar Empleado .....	302
Figura 7.23. Interfaz para Actualizar los datos de los Empleados. ....	302
Figura 7.24. Menú Calificaciones .....	303
Figura 7.25. Interfaz para Ingresar las Calificaciones de los alumnos.....	304
Figura 7.26. Interfaz para Ingresar Calificaciones Históricas.....	304
Figura 7.27. Interfaz para Consultar Calificaciones.....	305
Figura 7.28. Menú Ingresos y Egresos.....	306
Figura 7.29. Submenú Matrícula.....	306
Figura 7.30. Interfaz para Ingresar Pagos de Matrícula. ....	307
Figura 7.31. Interfaz para Consultar pagos de Matrícula.....	308
Figura 7.32. Interfaz para Consultar Deudas Pendientes. ....	308
Figura 7.33. Submenú Ingresos y Egresos.....	309
Figura 7.34. Interfaz para Registrar Pagos de Nómina. ....	309
Figura 7.35. Interfaz para la Consulta de Pagos de Nómina. ....	310
Figura 7.36. Menú Constancias y Reporte.....	311
Figura 7.37. Submenú Constancias.....	311
Figura 7.38. Interfaz para generar Constancias de Inscripción.....	311
Figura 7.39. Interfaz para generar Constancias de Estudios. ....	312
Figura 7.40. Interfaz para generar Boletines Informativos. ....	313
Figura 7.41. Submenú Reportes.....	314
Figura 7.42. Interfaz para generar Reportes de Alumnos por Grado.....	315
Figura 7.43. Interfaz para generar Reportes de Alumnos por Sexo.....	315
Figura 7.44. Interfaz para generar Reportes de Alumnos por Año Escolar.....	316
Figura 7.45. Interfaz para generar Reportes de Empleados del Plantel. ....	317
Figura 7.46. Interfaz para generar Reportes de Ingresos Diarios.....	317
Figura 7.47. Interfaz para generar Reportes de Ingresos Mensuales. ....	318
Figura 7.48. Interfaz para generar Reportes de Ingresos Anuales. ....	319
Figura 7.49. Interfaz para generar Reportes de Egresos Diarios.....	320

Figura 7.50. Interfaz para generar Reportes de Egresos Mensuales. ....	321
Figura 7.51. Interfaz para generar Reportes de Egresos Anuales. ....	321
Figura 7.52. Menú Ayuda. ....	322

## **RESUMEN**

En la Primera y Segunda Etapa de Educación en Unidades Educativas, se realiza de forma manual el manejo de los datos y el procesamiento de la información, generando entonces un completo desorden en el momento de ser solicitada. Por tal motivo ha surgido la necesidad de automatizar dichas actividades por medio del desarrollo de un sistema (software) que permita la optimización de los datos de los alumnos, empleados, ingresos y egresos, procesamiento de constancias y reportes. Se utilizó una base de datos relacional porque ofrece un excelente rendimiento en la obtención y almacenamiento de datos, a través de consultas por medio del lenguaje SQL. Para desarrollar la aplicación se utilizó el Proceso Unificado de desarrollo de software, aplicado por medio de las fases de Inicio, Elaboración y Construcción, con su respectivo modelo de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación, modelo de prueba e integración del sistema. Se obtuvo una arquitectura robusta que lleva por nombre SAPSEE (Sistema de Automatización Primera y Segunda Etapa de Educación). Permite optimizar las actividades primarias del plantel como: inscripción de alumnos, ingreso de empleados, procesamiento de calificaciones, ingresos y egresos de dinero y la emisión de constancias y reportes.

## INTRODUCCIÓN

Actualmente cualquier empresa o institución cuenta con equipos de computación para realizar sus actividades académicas y administrativas, sin embargo no cuenta con aplicaciones óptimas que les ayude a agilizar con certeza las tareas diarias.

La educación no es ajena a esta necesidad debido a que los constantes crecimientos en la cantidad de alumnos y los cambios generados en el proceso educativo, demandan del personal que allí labora una completa capacidad para realizar las tareas que un plantel educativo genera.

Por consiguiente se hace necesario implementar un sistema de automatización que facilite el desempeño de las actividades y procesos que son ejecutados durante los periodos escolares dentro de una Unidad Educativa.

El siguiente trabajo de investigación define el desarrollo y ejecución de un sistema de automatización en las actividades que se realizan dentro de una Unidad Educativa, utilizando nuevas herramientas tecnológicas y aplicando el Proceso Unificado de Desarrollo de Software conjuntamente con el Lenguaje Unificado de Modelado (UML).

# **CAPÍTULO 1**

## **PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS**

### **1.1. Planteamiento del Problema**

La automatización le ofrece a las instituciones una serie de soluciones a los problemas vinculados a las actividades contempladas dentro del logro de metas, que le permitan alcanzar el éxito, pero sobre todo el ahorro de recursos y materia prima. Dichas actividades se desarrollan dentro de dependencias, departamentos, cubículos y áreas que conjuntamente con el recurso humano logran en gran medida los objetivos planteados.

Dentro de las herramientas usadas para la automatización se encuentra el desarrollo de software, el cual es combinado con la ingeniería de software para solventar en gran medida los procesos institucionales, entre los que se hallan los administrativos, financieros, educativos, etc., que combinados con herramientas tecnológicas como el computador se reduce tiempo, pérdidas, y se producen ganancias y el alcance de los objetivos programados inicialmente por las empresas.

Aunado a eso se presentan las bases de datos que manejan grandes volúmenes de información, almacenando de forma permanente los datos informáticos evitando la inconsistencia en la información, logrando que se agilicen los procesos, se logre la optimización para la eficiencia de la producción de bienes y servicios.

La Unidad Educativa Anthony de Mello es un plantel privado, el cual está ubicado en la ciudad de Barcelona del estado Anzoátegui. Fue fundada en septiembre del 2000, cuando comenzó sus actividades al recibir alumnos de Preescolar y en años posteriores se fueron sumando alumnos de la I y II Etapa de Educación.

Dicho plantel educativo posee equipos de computación que intentan agilizar los procesos tanto académicos como administrativos, sin embargo existe inconsistencia de datos, repetición de información, un sistema de información débil, pérdida de tiempo en el momento de entregar constancias y reportes, entre otras cosas.

En fin los beneficios que ofrecen dichos equipos no son aprovechados en su totalidad por el plantel.

Actualmente los procesos se realizan de la siguiente manera:

1. La inscripción de los alumnos tanto nuevos como regulares se realiza manualmente, lo que trae como consecuencia que los datos de los estudiantes a veces no se guarden correctamente, existiendo así la posibilidad de no encontrar algunos datos cuando son requeridos.
2. La entrega de documentos se realiza por medio de hojas transcritas en un procesador de palabras (Microsoft Word), permitiendo que en muchas ocasiones se pierda material de oficina (hoja de papel para impresión) dado que se necesita actualizar las fechas, los nombres de los alumnos, entre otros datos que en algunos casos es muy pesado sobre todo cuando se promueven alumnos de sexto grado a séptimo grado, lo cual implica colocar dato por dato de cada alumno.
3. El control de pago se realiza de forma manual por medio de recibos teniendo que buscar recibo por recibo, en el momento que los representantes soliciten pagar deudas viejas o necesiten saber fechas de pagos anteriores; y por otro lado el administrador del colegio debe hacer este proceso paso por paso para realizar cálculos de montos que luego son

llevados a una hoja de Excel dispuesta para ello, en donde muchas veces se pierde información o se redunda en la misma dado que el control es llevado mensualmente.

4. Las entradas y salidas de dinero son registradas en un cuaderno de contabilidad teniendo que escribir constantemente la fecha, el concepto por el cual entra o sale dinero y la cantidad en bolívares, para luego tener un saldo general de los mismos, este proceso se registra mensualmente.
5. Toda esta información es llevada a un archivo vertical que no es más que un amontonamiento de papeles, recibos, facturas, documentos, constancias y reportes que en un tiempo dado aumenta la pérdida de información, retraso en el desarrollo de las actividades y sobre todo desgastamiento de los recursos de la institución.

El tiempo usado para realizar dichos procesos es extremadamente grande, es decir, se usa demasiado tiempo para realizar las tareas, se maneja un gran volumen de información como datos personales, constancias, pago de mensualidades, entrada y salida de dinero, entre otras cosas; existe retraso en el momento de entregar y recibir información necesaria para el desenvolvimiento eficaz de los procesos educativos y administrativos dentro de la Unidad Educativa Anthony de Mello.

De este modo se hace necesaria la creación e implementación de un sistema (software), que facilite el desempeño de las actividades y procesos que son ejecutados para el mejoramiento de los recursos existentes en la institución y así obtener mayor productividad al reducir el tiempo de forma significativa.

Por todo lo antes expuesto se propone realizar un Sistema para generar la automatización de los procesos académicos y administrativos vinculados a la I y II

etapa de educación básica dentro de la Unidad Educativa Anthony de Mello. El resultado final es lograr la optimización de los procesos, lograr la eficiencia y productividad del personal administrativo del plantel, para comprimir el tiempo usado en las tareas, además del reducir el tiempo que deben esperar tantos los alumnos, representantes, personal docente como el mismo personal administrativo en recibir alguna solicitud de algún servicio.

Al implantar este sistema se mejoraría el rendimiento, se reducirá el tiempo en las actividades, se evitará la inconsistencia de datos, se manejará de forma efectiva la información, se obtendrá una base de datos real y se logrará el éxito dentro del departamento administrativo y académico de la Unidad Educativa Anthony de Mello.

El presente proyecto muestra el desarrollo de un Sistema para gestionar la automatización de los procesos administrativos y académicos, y para ello se desarrollará mediante la utilización de la metodología y los lineamientos del Proceso Unificado de Desarrollo de Software, y cumpliendo con las necesidades de la Unidad Educativa Anthony de Mello.



## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Desarrollar un sistema para generar la automatización de los procesos académicos y administrativos vinculados a la I y II Etapa de Educación Básica dentro de una Unidad Educativa.

### **1.2.2. Objetivos Específicos**

1. Determinar los requerimientos del sistema para la obtención de los principales procesos de automatización.
2. Analizar la estructura base para la automatización del sistema.
3. Diseñar los módulos con su respectiva interfaz y las bases de datos del sistema para el almacenamiento, organización y control de la información.
4. Codificar los módulos del sistema para la obtención del software.
5. Realizar las pruebas de unidad y de integración a los módulos del sistema.
6. Elaborar el manual de usuario del sistema.

## **CAPÍTULO 2**

### **ANTECEDENTES Y MARCO TEÓRICO**

#### **2.1. Antecedentes**

1. “Desarrollo de un software que permita el control de las actividades asociadas a una Unidad Educativa Industrial, en las modalidades: Educación Básica (III Etapa), Diversificada y Profesional”. Presentado por English Josefina Rodríguez González. Permite la simplificación de los procesos de inscripción, evaluación y caja, manteniendo la consistencia de los datos y generando eficientemente reportes y constancias.
  
2. “Desarrollo de un sistema para la automatización del proceso de admisión y control de estudios en un Instituto Universitario”, presentado por Adrián Eduardo Moya Guzmán. Genera la automatización del sistema de registro y control de calificaciones del alumno universitario, información docente y el proceso de inscripción. Este sistema automatizará las tareas de elaboración del registro de notas, al llevar el historial de calificaciones de los estudiantes, guardados en una base de datos, facilitando la búsqueda y preparación de este documento, así como de constancias de notas, de inscripción, entre otros documentos.
  
3. “Desarrollo de un Sistema Automatizado para el Control de Estudios del Área de Post-Grado del Núcleo de Anzoátegui de la Universidad de Oriente”, presentado por Thamara Isabel Rojas B. en el año 2002. Se aplicó el Lenguaje Unificado de Modelado (UML) y el modelado de datos relacional. Genera la automatización del sistema de control de estudios del área de Postgrado para un óptimo manejo de toda la información creada.

## **2.2. Marco Teórico**

### **2.2.1. Software**

Es un conjunto de programas de explotación que el fabricante del computador entrega junto con la máquina o ya implementado en él. Un producto de software es un producto diseñado para un usuario. Es un soporte lógico que engloba tanto los sistemas operativos, lenguajes de programación como a los utilitarios, datos y documentos. Cada uno de estos elementos compone una configuración que se crea como parte del proceso de la ingeniería de software. El software es la esencia de multimedia.

### **2.2.2. Ingeniería de Software.**

La Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos". Es una forma de ingeniería que aplica los principios de la ciencia de los computadores y matemáticas para conseguir la solución a los problemas de software de forma efectiva y económica. Abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos, que facilitan controlar el proceso de desarrollo de software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

### **2.2.3. Programación orientada a objetos**

OOP por sus siglas en inglés de OBJECT ORIENTED PROGRAMMING, como paradigma, es una forma de pensar, una filosofía, de la cual surge una cultura

nueva que incorpora técnicas y metodologías diferentes. Pero estas técnicas y metodologías, y la cultura misma, provienen del paradigma, no lo hacen.

La OOP como paradigma es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose por sí mismo, y comunicándose con los demás por medio de mensajes.

La Programación Orientada a Objetos desde el punto de vista computacional es un método de implementación en el cuál los programas son organizados como grupos cooperativos de objetos, cada uno de los cuales representa una instancia de alguna clase, y estas clases, todas son miembros de una jerarquía de clases unidas vía relaciones de herencia.

En resumen la OOP permite la estructuración de las aplicaciones en objetos que cooperan, cada objeto es una instancia de alguna cosa y las clases están relacionadas mediante la herencia.

### **2.2.3.1. Características de lo Orientado a Objetos**

Los principios del modelo OO son:

1. **Abstracción:** Es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros.
2. **Encapsulación:** En el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.
3. **Modularidad:** Es la propiedad de un sistema que ha sido descompuesto en un conjunto de módulos coherentes e independientes.

4. Jerarquía o herencia: Es el orden de las abstracciones organizado por niveles.
5. Tipificación: Es la definición precisa de un objeto de tal forma que objetos de diferentes tipos no puedan ser intercambiados o, cuando mucho, puedan intercambiarse de manera muy restringida.
6. Concurrencia: Es la propiedad que distingue un objeto que está activo de uno que no lo está.
7. Persistencia: Es la propiedad de un objeto a través de la cual su existencia trasciende el tiempo (es decir, el objeto continua existiendo después de que su creador ha dejado de existir) y/o el espacio (es decir, la localización del objeto se mueve del espacio de dirección en que fue creado).

### **2.2.3.2. Fundamentos de lo Orientado a Objetos**

El paradigma OO se basa en el concepto de objeto. Un objeto es aquello que tiene estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos).

La estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancia y objeto son intercambiables.

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

Las relaciones entre objetos definen el comportamiento del sistema. Se dice que un objeto es un actor, si su única función es operar sobre otros objetos. El objeto es un servidor si solo es manejado por otros objetos y es un agente si tiene ambas propiedades. Se dice que los objetos actúan entre sí mediante mensajes, es decir, acciones que pide el objeto transmisor que ejecute el objeto receptor. Dependiendo del comportamiento definido para un objeto, éste tomará las acciones para ejecutar o no el mensaje, de manera apropiada.

#### **2.2.4. Proceso Unificado**

El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad.

Se basa en muchos años de experiencia en el uso de la tecnología orientada a objetos en el desarrollo de software de misión crítica en una variedad de industrias por la compañía "Rational", donde confluyen 'los tres amigos' como se llaman a sí mismos o los tres grandes OO: Grady Booch, James Rumbaugh e Ivar Jacobson.

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado.

El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas. El Proceso Unificado es un proceso porque "define quién está haciendo qué, cuándo lo hacer y cómo alcanzar cierto objetivo, en este caso el desarrollo de software".

El proceso unificado está definido por conceptos claves que los sustentan:

1. Componentes e interfaces bien definidas.
2. Utiliza el Lenguaje Unificado de Modelado (UML).
3. Aspectos característicos: dirigido por casos de usos, centrado en la arquitectura y por último iterativo e incremental.

#### **2.2.4.1. Fases del Proceso Unificado**

A continuación se presentan las fases necesarias para la implementación del Proceso Unificado:

1. Fase de Inicio. Primer ciclo de vida del sistema. Se desarrolla una descripción del producto final y se presenta el análisis del negocio; se identifican y priorizan los riesgos más importantes, se planifica la siguiente fase y se estima el proyecto de manera aproximada.
2. Fase de Elaboración. Segundo ciclo de vida del sistema. Se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema; se planifican las actividades y estiman los recursos para terminar el proyecto.
3. Fase de Construcción. Tercer ciclo de vida del sistema. Se desarrolla/construye el producto.
4. Fase de Transición. Cuarto ciclo de vida del sistema. Se proporciona el sistema a la comunidad de usuarios finales.

#### **2.2.4.2. Flujos de trabajo del Proceso Unificado**

Existen algunos flujos de trabajos que los desarrolladores aplican para el desarrollo de software. Los flujos de trabajo son los siguientes:

1. **Requisitos:** Desarrollar un modelo del sistema que se va a construir; identificar los requisitos de dicho sistema. Se obtiene el Modelo de Casos de Uso (MCU) y el Modelo del Dominio (o del Negocio).
2. **Análisis:** Tener una especificación más precisa de los requisitos obtenidos en requisitos y recogidos en el MCU. Se obtiene el Modelo del Análisis (diagramas de colaboración y paquetes/subpaquetes de análisis).
3. **Diseño:** Encontrar la forma del sistema que cumpla con todos los requisitos (encontrar la solución). Se obtiene el Modelo del Diseño (diagramas de secuencia, diagrama de clases y sistemas/subsistemas de diseño).
4. **Implementación:** Realizar la codificación correspondiente al diseño. Se obtiene el Modelo de la Implementación (componentes: clases y subsistemas implementados).
5. **Pruebas:** Realizar la verificación de la implementación; diseñar, implementar casos de prueba y ejecutarlos. Hay que obtener los casos y procedimientos de prueba.

### **2.2.5. Lenguaje Unificado de Modelado (UML, Unified Modeling Language)**

Es una especificación de notación orientada a objetos. Se basa en las anteriores especificaciones BOOCH, RUMBAUGH y COAD-YOURDON.

Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la



arquitectura del proyecto. UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

Un modelo UML esta compuesto por tres clases de bloques de construcción:

1. Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones).
2. Relaciones: relacionan los elementos entre sí.
3. Diagramas: Son colecciones de elementos con sus relaciones.

#### **2.2.5.1. Diagramas de UML**

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones, usualmente representado por un grafo conectado de vértices (elementos) y arcos (relaciones). En concreto, un **diagrama** ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas, los estáticos y los dinámicos:

Diagramas Estáticos:

1. Diagrama de Caso de Uso: El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo,

además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

2. Diagrama de Clases: Muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto. Es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos.
3. Diagrama de Objetos: Muestra las instancias de objetos de las clases. Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
4. Diagramas de Componentes: Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones. Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Se consideran en este tipo de diagramas solo tipos de componentes. Instancias específicas se encuentran en el diagrama de ejecución. Muestran los componentes de un sistema desde un punto de vista estático.
5. Diagrama de Despliegue: Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice. Muestra el despliegue de un sistema desde el punto de vista estático.

6. Diagrama de Colaboración: Es una forma de representar interacción entre objetos, alterna al diagrama de secuencia. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales) y ciclos en la ejecución.

#### Diagramas Dinámicos:

1. Diagrama de Secuencia: Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación. Es un diagrama de interacción que hace énfasis en la ordenación temporal de los mensajes.
2. Diagrama de Estados: Muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos. Es decir tratan la vista dinámica de un sistema.
3. Diagrama de Actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

#### **2.2.6. Base de Datos**

Es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular. Por datos se entiende hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos representa algún aspecto del

mundo real. Las bases de datos proporcionan la infraestructura requerida para los Sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas.

#### **2.2.6.1. Sistema Manejador de Base de Datos (SMBD).**

Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un lenguaje de manipulación de datos (DML: Data Manipulation Language) y un lenguaje de consulta (SQL: Structure Query Language).

El lenguaje de datos (DDL) es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos.

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos. El lenguaje de consulta (SQL) es empleado por el usuario para extraer la información de las bases de datos. Este permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el SELECT. Los programas de aplicación, batch o en línea, en lugar de acceder directamente a los archivos donde residen los datos, invocan al manejador de base de datos especificado por el programa.

La secuencia conceptual de operaciones que ocurren para acceder cierta información que contiene una base de datos es la siguiente:

1. El usuario solicita cierta información contenida en la base de datos.
2. El SMBD intercepta este requerimiento y lo interpreta.
3. El SMBD realiza las operaciones necesarias para acceder y/o actualizar la información solicitada.

#### **2.2.6.2. Base de Datos Relacional.**

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

#### **2.2.6.3. Lenguajes de Consulta en el SMBD.**

Son lenguajes que además de consultar la base de datos permiten definir y modificar la estructura de los datos, así como establecer los criterios de seguridad de los mismos. Entre ellos se encuentra el SQL, basado en el cálculo relacional del dominio. El SQL es una herramienta para organizar, gestionar y recuperar los datos almacenados en una base de datos.

Este lenguaje es esencial para interactuar con una base de datos con modelo relacional. Las peticiones sobre los datos se expresan mediante sentencias, que deben escribirse de acuerdo con unas reglas sintácticas y semánticas de este lenguaje y de esa forma se obtienen los datos necesarios para su manipulación.

#### **2.2.7. Administración.**

Es el conjunto de actos de gestión y disposición de bienes y servicios ejecutados sobre una institución.

### **2.2.7.1. Proceso Administrativo**

Conjunto de operaciones o tareas propias de una entidad con el fin de dirigir, ordenar, disponer y organizar los bienes relacionados a una empresa para lograr su eficacia y desempeño exitoso dentro de sus responsabilidades.

En una institución educativa va relacionado con la realización de tareas como: la entrega de documentos o constancias (de estudio, de conducta) y reportes (boletín informativo) entre otros procesos. También permite organizar el sistema de información que se maneja como: datos de estudiantes, empleados, entre otros.

### **2.2.8. Sistema Automatizado.**

Es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos. Algunos objetivos y beneficios de la automatización son los siguientes:

1. Mejorar la productividad de la empresa, reduciendo los costos de la producción y mejorando la calidad de la misma.
2. Mejorar las condiciones de trabajo, suprimiendo los trabajos penosos e incrementando la productividad. Realizar las operaciones imposibles de controlar intelectual o manualmente.
3. Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.
4. Simplificar el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.

5. Integrar la gestión y la producción.
6. Realizar las operaciones imposibles de controlar intelectual o manualmente

## **CAPÍTULO 3**

### **FASE DE INICIO**

#### **3.1. Introducción**

El Proceso Unificado se desarrolla en diferentes fases: Inicio, Elaboración, Construcción y Transición, siendo la Fase de Inicio la que da comienzo al avance del sistema.

A través de esta fase se establece la viabilidad, creando el análisis del sistema, sin estudiarlo completamente, y buscando el porcentaje de casos de uso necesarios para fundamentarlo, además de describir o esbozar una propuesta de la arquitectura del sistema.

Esta fase consta de diferentes actividades las cuales serán descritas a continuación.

#### **3.2. Captura de Requisitos**

##### **3.2.1. Lista Inicial de Requisitos**

Para determinar la información inicial necesaria en el proceso de desarrollo de software se realizó un acercamiento a los procesos realizados de forma manual, teniendo en cuenta la ejecución de dichos procesos y el tiempo empleado para el mismo, además de los formatos usados para tal fin. Por otro lado se realizó entrevistas no estructuradas al personal administrativo y directivo, los cuales permite la estructuración del glosario de términos para una comprensión más exacta del funcionamiento del sistema actual. Al realizar el contacto con la institución se obtuvo lo siguiente:



1. Manejar de forma efectiva y óptima el proceso de inscripción de alumnos nuevos y de alumnos regulares (alumnos ya inscritos en la institución). Esto implica la entrega por parte de los representantes de los documentos necesarios (Constancia de Buena Conducta, Boletín de Calificaciones, Partida de Nacimiento,...) así como del pago correspondiente, seguidamente se asigna el alumno al grado relacionado con su nivel de estudio (I ó II Etapa), luego la institución, emite una constancia de inscripción. El encargado principal de manejar este proceso es el personal administrativo, el cual se encarga de manejar los pagos por mensualidad, y otros aranceles, emitiendo a su vez recibos de pagos contentivos de dicha información la cual es asentada en el control diario de ingresos.
2. El sistema debe permitir el control de notas obtenidas por parte de los estudiantes, esto implica que debe registrarse las evaluaciones obtenidas al final de cada lapso (tres lapsos) y al final del año por parte de cada estudiante en las diferentes etapas para emitir un boletín informativo. Este proceso es ejecutado por el personal directivo al final de cada lapso.
3. Se debe tener un control en las emisiones de constancias que son solicitadas por los representantes de los alumnos. El representante solicita el documento, cancela el costo de dicha constancia, es procesada por el personal administrativo y luego emitida por el personal directivo, en el caso de que un alumno se retire del plantel debe estar solvente para emitir una constancia de egreso.
4. El sistema debe permitir el control de ingresos y egresos diarios y mensuales sobre todo los gastos realizados por medio de cheques. Cuando el administrador del colegio emite un cheque debe registrar a quien va emitido, la fecha, el motivo y la cantidad en bolívares fuertes, generando a

veces la pérdida de información si no se copia inmediatamente, lo cual arroja un proceso ineficiente; además cuando el director del plantel solicita en cualquier momento la búsqueda de un cheque específico se pierde demasiado tiempo al hacerlo manualmente.

5. Lograr almacenar la información en una base de datos. Esto indica que la información perdurará en el tiempo para su posterior uso.
6. El sistema debe funcionar en un PC. De esta forma se hace óptimo el desempeño de las actividades académicas y administrativas realizadas diariamente en el plantel.

En resumen los procesos candidatos a automatizar serían los siguientes:

1. Proceso de Inscripción de Alumnos.
2. Control de Notas. Procesar las calificaciones de los alumnos.
3. Procesar las solicitudes de constancias (solvencias, constancia de estudio, certificados de notas, carta de conducta,...) y recibos (recibos de pagos, recibos de cobro).
4. Control de Ingresos y Egresos diarios y mensuales.
5. Automatizar el almacenamiento de la información generada debido a las actividades tanto académicas como administrativas.

Todos estos procesos son fundamentales para el desarrollo de las actividades principales dentro de la Unidad Educativa.

Sin embargo no se descarta la necesidad de implementar o incluir nuevos procesos para lograr un mejor desempeño del sistema.

### 3.3. Modelo de Dominio

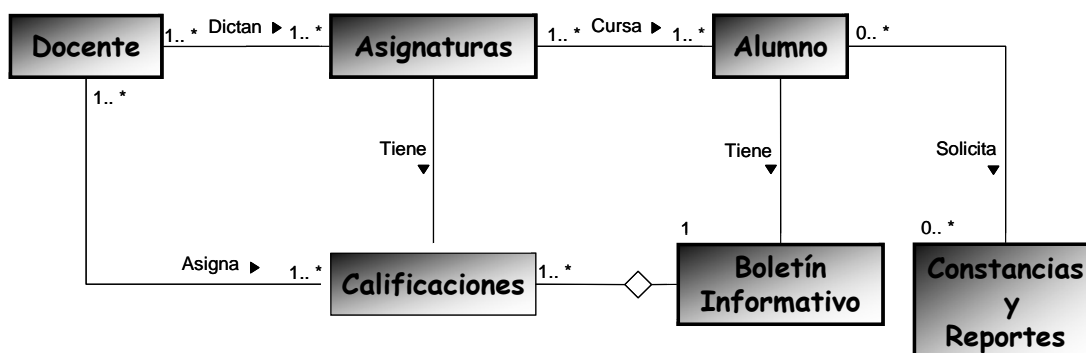


Figura 3.1. Modelo de Dominio del Sistema Actual.

Luego de la captura de requisitos se representa a través del Modelo de Dominio (Figura 3.1) las actividades más importantes del sistema actual, mostrando los eventos o acciones que existen en el entorno. El Sistema propuesto se identificará por medio de actores y el desarrollo de casos de uso. En el modelo actual se observa que los Alumnos de la Unidad Educativa cursan las Asignaturas dictadas por los docentes siendo ellos quienes asignan las calificaciones de los estudiantes, las cuales son registradas en un Boletín Informativo (boletín de calificaciones) para de esta forma llevar el control de las notas obtenidas por los alumnos. Los alumnos pueden hacer peticiones de Constancias y Reportes donde están registradas sus calificaciones, datos personales y otros registros vinculados a su desarrollo escolar.

### 3.4. Glosario de Términos

El glosario de términos ayuda al usuario a identificar el comportamiento del sistema de forma conceptual, descrito en el modelo de dominio.

1. Base de Datos: Conjunto representado por la unión de tablas relacionadas entre sí que contienen toda la información del sistema. Estas son usadas para almacenar información que perdura en el tiempo.
2. Boletines Informativos: Son constancias de calificaciones realizadas a los alumnos durante cada lapso y al finalizar el año escolar.
3. Constancias: Documentos que son solicitados por los representantes, como: constancia de inscripción, boletines, constancia de estudio, entre otros.
4. Gastos: Representa el egreso del dinero por parte de la institución por medio de cheques, tarjetas de débitos o efectivo.
5. Inscripción: Representa un proceso mediante el cual se realiza la inscripción de alumnos nuevos para el año escolar que esté en curso dentro del plantel.
6. Manejador de Ayuda: Proceso externo que gestiona la ayuda del sistema.
7. Manejador de Base de Datos: Es un gestor externo que permite almacenar y manipular la información existente en la base de datos.
8. Pagos: Cancelación de los pagos mensuales por parte de los representantes de los alumnos, realizados cada mes y en efectivo ó a través de cheque.
9. Personal Administrativo: Es la cantidad de personas que laboran en el departamento de Administración, compuesto por el Administrador y el Asistente Administrativo.

10. Personal Directivo: Representa a la las personas encargadas de la Dirección del plantel. Está constituida por la Directora y la Coordinadora Académica.
11. Personal Docente: Cantidad de personas que laboran en el área académica, son los que dictan clases dentro de las aulas de la institución.
12. Reinscripción: Representa la inscripción por parte del personal administrativo de los alumnos ya existentes en el sistema que cursarán un nuevo año escolar, es decir, los alumnos regulares.
13. Recibos: Documentos emitidos por los pagos realizados por los representantes, relacionados con la mensualidad, inscripción, constancias y demás documentación solicitadas por los mismos.

### **3.5. Identificación y Descripción de Riesgos**

Un riesgo es cualquier agente que ejerce influencia sobre el sistema de manera que perturba el desarrollo y construcción del proyecto.

Por lo tanto se realiza una lista para identificarlos, verlos claramente e implementar un plan de contingencia para eliminarlos, buscando la forma de atacarlos a través de ese plan y de identificadores como descripción, prioridad e impacto para ofrecer un nuevo y mejor sistema que satisfaga los requerimientos iniciales.

#### **3.5.1. Lista de Riesgos Críticos**

##### **3.5.1.1. Riesgo: Procesar Alumnos**

1. Impacto: Todo el Sistema.

2. Descripción: El riesgo presente en este proceso va relacionado a la inconsistencia de datos de los alumnos suministrados al sistema, la cual produce mala manipulación de los mismos con la base de datos cuando se realizan consultas a las mismas.
3. Contingencia: Desarrollar módulos que ofrezcan óptima manipulación de los datos de los alumnos en el sistema.

#### **3.5.1.2. Riesgo: Procesar Empleados**

1. Impacto: Todo el Sistema.
2. Descripción: El riesgo presente en este proceso va relacionado con la inconsistencia de datos de los empleados suministrados al sistema, la cual produce mala manipulación de los mismos con la base de datos cuando se realizan consultas a la misma.
3. Contingencia: Desarrollar módulos que ofrezcan óptima manipulación de los datos de los empleados en el sistema.

#### **3.5.1.3. Riesgo: Configurar Sistema**

1. Impacto: Todo el Sistema.
2. Descripción: El problema presente en este proceso está vinculado a los datos de inicio del sistema (RIF, NIT, Código D. E. A, N° Cuenta Bancaria, Dirección, Datos del Director, Coordinador y Otros) y su inclusión a la base de datos para la configuración del sistema.
3. Contingencia: Desarrollar un módulo en donde se pueda introducir, eliminar o modificar información almacenada en la base de datos (RIF, NIT, Código DEA, Datos del Director y Coordinador del Plantel) manejada por el sistema.

#### **3.5.1.4. Riesgo: Constancias y Reportes.**

1. Impacto: Constancias y Reportes.
2. Descripción: El riesgo que se presenta en este proceso está en dirección a la entrega de reportes o constancias las cuales son requeridas por los alumnos.
3. Contingencia: Se deben desarrollar módulos para la preparación de constancias y reportes como son constancias de estudio, constancias de calificaciones, constancias de retiro, solvencias; alumnos por edad, sexo, grado...

#### **3.5.1.5. Riesgos: Ingresos y Egresos.**

1. Impacto: Ingresos y Egresos.
2. Descripción: Este riesgo se relaciona con el almacenamiento errado de la información de los pagos de mensualidades de alumnos además de los pagos a los empleados, entre otros.
3. Contingencia: Desarrollar módulos que permitan el mejor almacenamiento de estos datos para el cálculo de deudas (ingresos) y pagos en general (egresos), que se registrarán en la base de datos para futuro control.

### **3.6. Modelo de Casos de Uso**

El modelo de casos de uso nos permite tener una representación de la captura de requisitos funcionales del sistema de forma que se pueda comunicar fácilmente con el resto de los implicados en el desarrollo del mismo.

Los casos de uso además facilitan la captura de requisitos del software en general, dirigen el proceso de desarrollo en su totalidad permitiendo la identificación

de clases, subsistemas e interfaces, planificar las iteraciones del desarrollo y la integración del sistema.

En fin, nos hace entrega de los requisitos plasmados en forma de diagrama de objetos.

El sistema tendrá por nombre SAPSEE, Sistema de Automatización de la Primera y Segunda Etapa de Educación.

### **3.6.1. Actores**

Un actor es un usuario que se comunica con el sistema mediante el envío y recepción de mensajes hacia y desde el sistema. A continuación se representa la descripción de cada actor del sistema:

#### **3.6.1.1. Identificación de Actores**

1. Personal Administrativo: Está representado por el Administrador de la Unidad Educativa y la Secretaria, este actor manipula toda la información administrativa relacionada con las inscripciones de los alumnos, ingresando los datos del mismo como también los datos del personal docente. Este actor también ingresa los pagos de las mensualidades de los alumnos así como los pagos al personal obrero, docente y administrativo.
2. Personal Directivo: Está compuesto por el Director del Plantel y el Coordinador Académico. Este actor le da curso al ingreso de las calificaciones de los alumnos las cuales son suministradas por el personal docente para la emisión de los boletines informativos. Se encarga de configurar el sistema ingresando los datos necesarios para el desarrollo y



gestión de constancias y reportes, que en resumidas cuentas son los datos del funcionamiento del plantel (RIF, NIT, Código D. E. A, N° Cuenta Bancaria, Dirección, Datos del Director, Coordinador y Otros).

### **3.6.2. Casos de Uso**

Es el modo de capturar los requerimientos y necesidades de los usuarios para su comunicación con el sistema.

Los casos de uso dirigen todo el proceso de desarrollo debido a que la mayoría de las actividades (análisis, diseño) se derivan de ellos. Los casos de uso representan el comportamiento de una parte del sistema desde el punto de vista del usuario.

También se puede decir que es una forma particular en que un actor puede utilizar el sistema.

#### **3.6.2.1. Identificación de Casos de Uso**

1. Caso de Uso Procesar Alumnos: Este caso de uso se encarga de las actividades académicas y administrativas relacionadas al manejo de información de los alumnos del Plantel. Es ejecutado por el actor Personal Administrativo.
2. Caso de Uso Procesar Empleados: Se encarga de las actividades académicas y administrativas relacionadas al manejo de información de los empleados del Plantel. Es ejecutado por el actor Personal Administrativo.
3. Caso de Uso Ingresos y Egresos: Se encarga de las actividades relacionadas con los pagos de matrícula (mensualidades) por parte de los representantes de los alumnos, los pagos emitidos a los docentes, personal

administrativo, directivo, obrero del plantel. Es ejecutado por el actor Personal Administrativo.

4. Constancias y Reportes: Se encarga de todas las actividades relacionadas a la tramitación de constancias y generación de reportes. Es ejecutado por el actor Personal Administrativo y el actor Personal Directivo.
5. Mostrar Ayuda: Se encarga de todas las actividades relacionadas con la ayuda a los actores del sistema: Personal Administrativo y Personal Directivo durante la ejecución del sistema.
6. Calificaciones: Se encarga de las gestiones relacionadas con el ingreso de las calificaciones de los alumnos a la base de datos al finalizar cada lapso académico (cada 3 meses) y por supuesto al final de año para la generación de los boletines informativos.
7. Configurar Sistema: Se encarga de gestionar el ingreso de los datos relacionados con el funcionamiento del plantel: RIF, NIT, Código D. E. A. (número bajo el cual la Unidad Educativa está inscrito en el Ministerio de Educación y Deportes), dirección, nombre del Director, Coordinador, Administrador de la Institución, Secretaria, requeridos para generar: constancias, recibos de pago de matrícula, recibos de pago al personal obrero, administrativo y directivo así como los pagos que se emiten a los proveedores. Por otro lado el nombre de los directivos se usará para generar las certificaciones y otras constancias. También gestiona el ingreso de las asignaturas dictadas y de los documentos emitidos por el plantel. Se ejecuta por el actor Personal Directivo.

#### **3.6.2.2. Diagrama de Caso de Uso.**

El diagrama de casos de uso muestra la interacción de los actores con cada uno de los casos de uso del sistema. El diagrama de caso de uso muestra los casos de uso de un sistema desde un punto de vista estático. Captura la funcionalidad del

sistema, incluida la funcionalidad significativa para la arquitectura. Además define casos de uso que se analizarán con más profundidad en el modelo de análisis. En la figura 3.2 se muestra que el actor Personal Administrativo hace uso de los casos de uso: Procesar Alumnos, Procesar Empleados, Ingresos y Egresos, Constancias y Reportes y Mostrar Ayuda para realizar actividades relacionadas a la administración y academia de la institución. El actor Personal Directivo utiliza los casos de uso: Constancias y Reportes, Mostrar Ayuda, Calificaciones y Configurar Sistema. El actor Manejador de Base de Datos manipula todos los casos de uso menos el caso de uso Mostrar Ayuda.

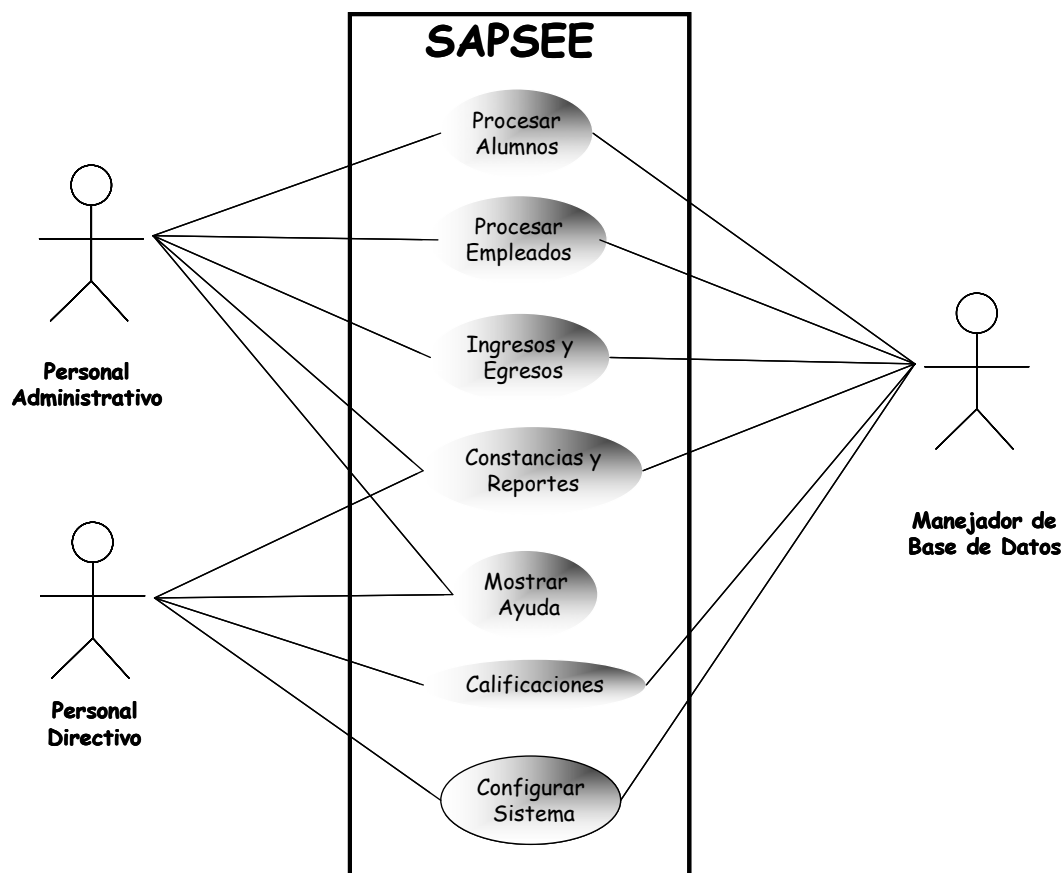


Figura 3.2. Modelo de Casos de Uso

A continuación se describe el comportamiento de los casos de uso en el sistema:

### 3.6.2.3. Caso de Uso Procesar Alumnos

En este caso de uso se integran todas las actividades concernientes al ingreso, retiro, actualización y consulta de de los datos de los estudiantes dentro del sistema.

Actores: Personal Administrativo, Manejador de Base de Datos.

Precondición: ninguna.

Flujo de sucesos:

1. El Personal Administrativo invoca el caso de uso Procesar Alumnos e ingresa los datos del alumno, padres y representantes a través de la interfaz de usuario.
2. El sistema verifica los datos y los almacena mediante el actor Manejador de Base de Datos, en la base de datos.
3. El sistema envía un mensaje solicitando la impresión del recibo de inscripción y de pago correspondiente al pago de mensualidad, el sistema imprime tales documentos si la respuesta es si por parte del usuario.
4. Si el Personal Administrativo desea modificar los datos del alumno debe ingresar la cédula del alumno a través de la interfaz de usuario.
5. Si los datos están registrados con anterioridad, esto es mediante el proceso de inscripción del alumno, el sistema los busca y los presenta en la interfaz de usuario. El Personal Administrativo hace los cambios y actualizaciones necesarias.

6. El sistema almacena los cambios en la base de datos.
7. Si el Personal Administrativo desea consultar los datos del alumno debe ingresar la cédula del alumno a través de la interfaz de usuario.
8. El sistema busca los datos y los presenta. Los datos deben estar registrados con anterioridad en el sistema previamente, esto es mediante el proceso de inscripción del alumno.
9. Si el Personal Administrativo desea retirar al alumno debe ingresar la cédula del alumno a través de la interfaz de usuario. El sistema busca los datos y los presenta en la interfaz de usuario. Deben existir los datos del alumno, previa inscripción del mismo.
10. El Personal Administrativo confirma el retiro del alumno.
11. El sistema verifica si el alumno está solvente (consultando la deuda del estudiante) y emite la constancia de retiro si el usuario así lo solicita.

Caminos alternativos:

1. En el paso 1, el Personal Administrativo puede cancelar el proceso en vez de confirmar los datos.
2. En el paso 2, si la cédula del alumno (cédula escolar) existe, el sistema emite un mensaje de alumno existente en el sistema.
3. En el paso 4, si el Personal Administrativo ingresa una cédula incorrecta de alumno el sistema emite un mensaje y vuelve a solicitar la cédula.
4. En el paso 5, el Personal Administrativo puede cancelar los cambios sin producir ningún efecto en los datos del alumno.

5. En el paso 7, para consulta de alumno, si el Personal Administrativo ingresa una cédula incorrecta de alumno el sistema emite un mensaje y vuelve a solicitar la cédula.
6. En el paso 9, para retirar el alumno, si el Personal Administrativo ingresa una cédula incorrecta de alumno el sistema emite un mensaje y vuelve a solicitar la cédula.
7. En el paso 11, si el alumno no es solvente debe entonces cancelar el saldo deudor y volver al paso 10.

Poscondición: el caso de uso finaliza cuando el usuario cancele el proceso Procesar Alumno.

#### **3.6.2.4. Caso de Uso Procesar Empleados**

En este caso de uso se integran todas las actividades concernientes al ingreso, retiro, actualización y consulta de de los datos de los empleados del plantel.

Actores: Personal Administrativo, Manejador de Base de Datos.

Precondición: ninguna.

Flujo de sucesos:

1. El Personal Administrativo invoca el caso de uso Procesar Empleados e ingresa los datos del empleado a través de la interfaz de usuario.
2. El sistema verifica los datos y los almacena mediante el actor Manejador de Base de Datos, en la base de datos, una vez que el usuario confirma el ingreso del empleado del plantel al sistema.

3. Si el usuario solicita la consulta de los datos del empleado, debe ingresar la cédula del empleado a través de la interfaz.
4. El sistema busca los datos del empleado a consultar y los presenta en la interfaz. Los datos deben estar registrados en el sistema, previa ejecución del paso 2.
5. Si el usuario solicita modificar los datos del empleado, debe ingresar la cédula del empleado a través de la interfaz.
6. El sistema busca los datos del empleado a modificar y los presenta en la interfaz. Los datos deben estar registrados en el sistema, previa ejecución del paso 2.
7. El usuario hace las modificaciones necesarias a los datos y confirma la actualización.
8. El sistema ingresa los cambios realizados a los datos dentro de la base de datos del sistema.
9. Si el usuario desea retirar al empleado del plantel, debe ingresar la cédula del empleado a través de la interfaz. El sistema busca los datos y los presenta en la interfaz. Los datos deben estar registrados en el sistema, previa ejecución del paso 2.
10. El usuario confirma el retiro y el sistema se dirigirá a la base de datos para poner en inactividad al empleado.

Caminos alternativos:

1. En el paso 3, el Personal Administrativo puede cancelar en vez de confirmar los datos.
2. En el paso 4, si el usuario ingresa una cédula de empleado incorrecta, el sistema emite un mensaje solicitando la cédula nuevamente.
3. En el paso 5, si los datos del empleado no existen en el sistema, se emite un mensaje de no existencia del empleado.
4. En el paso 6, si el usuario ingresa una cédula de empleado incorrecta, el sistema emite un mensaje solicitando la cédula nuevamente.
5. En el paso 7, si los datos del empleado no existen en el sistema, se emite un mensaje de no existencia del empleado. Si el usuario no confirma la actualización de los datos del empleado no se realiza ningún cambio dentro de la base de datos y por lo tanto no se cumple el paso 8.
6. En el paso 9, si el usuario ingresa una cédula de empleado incorrecta, el sistema emite un mensaje solicitando la cédula nuevamente.
7. En el paso 10, el usuario puede no confirma el retiro. De hacerlo no se retira al empleado del plantel.

Poscondición: el caso de uso finaliza cuando el usuario cancele el proceso Procesar Empleados.

#### **3.6.2.5. Caso de Uso Ingresos y Egresos**

A este caso de uso se integran las actividades relacionadas con los cobros de matrícula y los pagos emitidos al personal del plantel.

Actores: Personal Administrativo, Manejador de Base de Datos. Precondición: deben existir los datos de los alumnos y de los empleados dentro del sistema, previa ejecución de los casos de uso Procesar Alumnos y Procesar Empleados respectivamente.



#### Flujo de sucesos:

1. El Personal Administrativo invoca el caso de uso Ingresos y Egresos y si solicita el cobro de mensualidad a los alumnos, debe ingresar la cédula del alumno.
2. El sistema busca los datos del empleado y los presenta en la interfaz de usuario.
3. El usuario confirma el pago de mensualidad y el sistema emite un recibo de pago de mensualidad.
4. El sistema almacena la información de cobro en la base de datos.
5. Si el usuario solicita el pago de nómina al empleado, debe ingresar la cédula del empleado a través de la interfaz de usuario.
6. El sistema busca los datos del empleado y los presenta en la interfaz de usuario.
7. El usuario confirma el pago al empleado y el sistema emite un recibo de pago de nómina.
8. El sistema almacena la información del pago en la base de datos.

#### Caminos alternativos:

1. En el paso 1, si el usuario ingresa una cédula de alumno incorrecta, el sistema emite un mensaje solicitando la cédula nuevamente.

2. En el paso 3, si el usuario no confirma la acción entonces no se ejecuta el paso regresando nuevamente al paso 1.
3. En el paso 5, si el usuario ingresa una cédula de empleado incorrecta, el sistema emite un mensaje solicitando la cédula nuevamente.
4. En el paso 7, si el usuario no confirma la acción entonces no se ejecuta el paso regresando nuevamente al paso 5, en donde debe ingresar nuevamente la cédula de un empleado.

Poscondición: el caso de uso finaliza cuando se cancele el proceso.

#### **3.6.2.6. Caso de Uso Constancias y Reportes**

Este caso de uso es usado para la solicitud e impresión de constancias (estudio, conducta,...) y reportes (estudiantes por grado, ingresos diarios...).

Actores: Personal Directivo, Manejador de Base de Datos.

Precondición: deben existir los datos del alumnado y del personal del plantel, previa ejecución del proceso Procesar Alumnos y Procesar Empleados.

Flujo de sucesos:

1. El Personal Directivo invoca el caso de uso Constancias y Reportes para ingresar las solicitudes de constancias o reportes.
2. El sistema solicita la cédula del alumno ó la cédula del empleado y los documentos a imprimir o los reportes a realizar.

3. Finaliza el caso de uso.

Caminos alternativos:

1. En el paso 2, si la cédula del alumno, o la cédula del empleado, no existe el sistema emite un mensaje y solicita nuevamente el ingreso del dato específico.

Poscondición: el caso de uso finaliza cuando se haya hecho la solicitud, impreso los documentos o la realización del reporte dado.

#### **3.6.2.7. Caso de Uso Mostrar Ayuda**

Este caso de uso es para que los usuarios de SAPSEE tengan una ayuda o guía sobre el uso del sistema. La descripción breve de este caso de uso se realiza a continuación:

Actores: Personal Directivo, Personal Administrativo.

Precondición: ninguna.

Flujo de sucesos:

1. El Personal Directivo o Personal Administrativo invocan el caso de uso Mostrar Ayuda.
2. El sistema despliega la guía de ayuda para el uso de SAPSEE.
3. Finaliza el caso de uso.

Poscondición: el caso de uso finaliza cuando se haya usado la guía de SAPSEE o se cancele el proceso.

#### **3.6.2.8. Caso de Uso Calificaciones**

Este caso de uso es usado para el ingreso y la consulta de las calificaciones de los alumnos en el sistema, obtenidas durante un lapso académico y durante el año escolar actual, por lo tanto se tendrá un registro para futuras consultas de notas.

Actores: Personal Directivo, Manejador de Base de Datos.

Precondición: en el sistema deben existir alumnos inscritos para el año escolar en curso, previa ejecución del caso de uso Procesar Alumnos.

Flujo de sucesos:

1. El Personal Directivo invoca el caso de uso Ingresar Calificaciones.
2. El sistema solicita la cédula del alumno.
3. El Personal Directivo ingresa las calificaciones correspondientes a cada una de las asignaturas del grado del alumno.
4. Si el usuario solicita la consulta de las notas del alumno, debe ingresar la cédula del alumno a través de la interfaz, el lapso y el año escolar de estudio
5. El sistema busca las calificaciones del alumno y las presenta en la interfaz de usuario.

6. Finaliza el caso de uso.

Caminos alternativos:

1. En el paso 2, si la cédula del alumno no existe el sistema emite un mensaje y solicita nuevamente el ingreso del dato.
2. En el paso 3, el usuario puede cancelar el ingreso de calificaciones.
3. En el paso 4, si el usuario ingresa una cédula errada el sistema emite un mensaje y solicita la cédula nuevamente.
4. En el paso 5, si el alumno no existe emite un mensaje y regresa al paso 4.

Poscondición: el caso de uso finaliza cuando se hayan ingresado todas las calificaciones en la base de datos del sistema por medio del Manejador de Base de Datos o cuando se cancele el proceso.

### **3.6.2.9. Caso de Uso Configurar Sistema**

Actores: Personal Directivo, Manejador de Base de Datos.

Precondición: ninguna.

En este caso de uso se integran todas las actividades relacionadas a la alineación del sistema para el comienzo de sus tareas académicas y administrativas.

A continuación se realiza una breve descripción de las actividades que realiza este caso de uso, además de su ejecución:

1. Ingresar los datos necesarios para que el sistema opere eficientemente (RIF, Código D. E. A, Dirección, Teléfonos, datos del Director, Coordinador, Administrador del Plantel, Secretaria).
2. Crear los documentos y constancias emitidas por el plantel, además de poder modificarlos, consultarlos y eliminarlos dentro del sistema.
3. Crear las asignaturas que se dictarán durante el año escolar, además de poder modificarlas y de eliminarlas dentro del sistema.

Flujo de sucesos:

1. El Personal Directivo invoca el caso de uso Configurar Sistema para ingresar documentos, asignaturas y los datos del plantel.
2. El Personal Directivo ingresa los datos de inicio de sistema a través de la interfaz, para que el sistema funcione eficientemente
3. El usuario confirma el ingreso de datos de inicio. El sistema incluye los datos en la base de datos.
4. El Personal Administrativo ingresa los datos de los documentos que emite la institución a través de la interfaz.
5. El usuario confirma el ingreso de documentos en el sistema. El sistema incluye los documentos en la base de datos.
6. El Personal Administrativos ingresa las asignaturas que dicta el plantel a través de la interfaz de usuario.

7. El usuario confirma el ingreso de los datos de las asignaturas. El sistema incluye las asignaturas en la base de datos.
8. Finaliza el caso de uso.

Caminos alternativos:

1. En el paso 3, el usuario puede cancelar en vez de confirmar. De hacerlo no se producen cambios en la base de datos.
2. En el paso 5, el Personal Administrativo puede cancelar en vez de confirmar el ingreso de los documentos, sin producir efecto en la base de datos.
3. En el paso 7, el usuario puede cancelar el ingreso de asignaturas dentro del sistema. De hacerlo no se produce ningún cambio en la base de datos.

Poscondición: el caso de uso finaliza cuando se hayan ingresados los datos correspondientes o se cancele el proceso.

### **3.7. Análisis**

En este flujo de trabajo se analizan los requisitos, para refinarlos y estructurarlos en un modelo de objetos que sirvan para un esbozo concreto en el modelo de diseño y es así como se obtiene un modelo de análisis inicial para definir con precisión los casos de uso. Se desarrollará un 10% durante este flujo de trabajo.

El objetivo de hacerlo es conseguir una visión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a

estructurar el sistema entero, incluyendo la arquitectura. Este análisis arrojará toda la información necesaria para la suma de objetos indispensables en la línea base de la arquitectura.

### 3.7.1. Análisis de la Arquitectura

Según el conjunto inicial de casos de uso definidos en la sección anterior por medio de su modelo, se hace un esquema del modelo de análisis a través de la identificación del paquete del análisis tomando como base los requisitos representados en los casos de uso.

Es así entonces como se obtiene una representación del modelo inicial de casos de uso y una arquitectura candidata para el sistema.

### 3.7.2. Modelo de Análisis

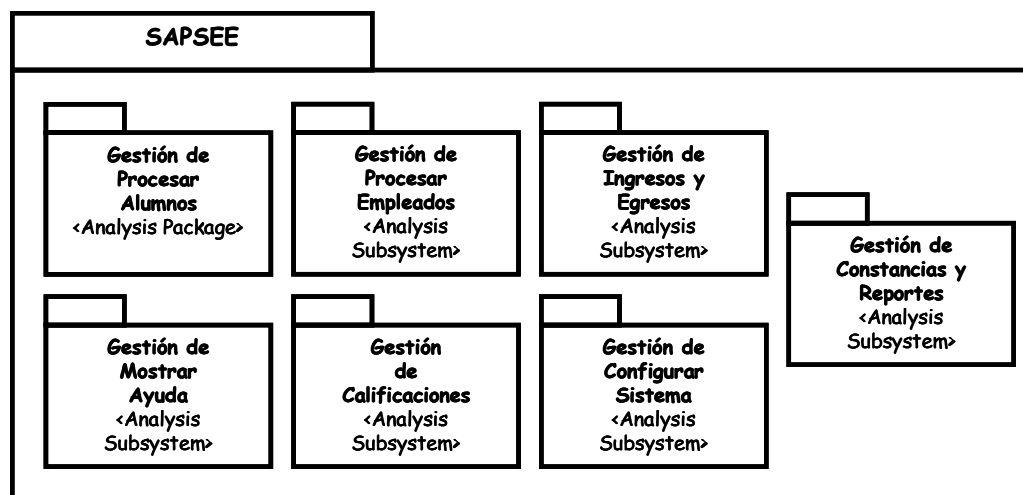


Figura 3.3. Paquetes del Análisis

Los paquetes de análisis proporcionan un medio para organizar el modelo de análisis en objetos más fáciles de manejar. La Figura 3.3 muestra la identificación de



los paquetes del análisis, los cuales permitirán organizar el modelo de análisis en unidades que, básicamente, englobarán los requisitos funcionales del sistema definidos como casos de uso.

### 3.7.2.1. Paquete del Análisis del Caso de Uso Procesar Alumnos

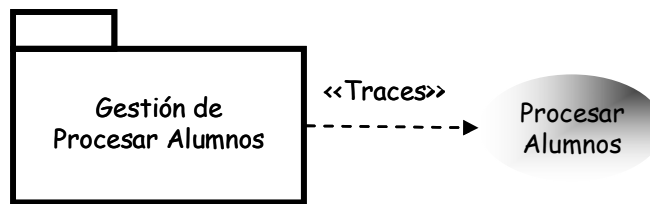


Figura 3.4. Paquete del Análisis del Caso de Uso Procesar Alumnos

En este paquete de análisis se incluyen todas las actividades relacionadas al manejo de información de los alumnos.

Tiene relación directa con el caso de uso Procesar Alumnos (Figura 3.4).

### 3.7.2.2. Paquete del Análisis del Caso de Uso Procesar Empleados

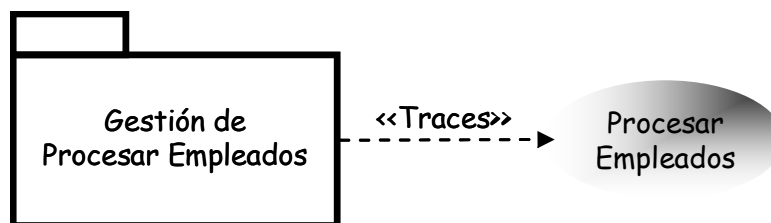


Figura 3.5. Paquete del Análisis del Caso de Uso Procesar Empleados

En este paquete de análisis se incluyen todas las actividades relacionadas al manejo de información de los empleados.

Tiene relación directa con el caso de uso Procesar Empleados (Figura 3.5).

### 3.7.2.3. Paquete del Análisis del Caso de Uso Calificaciones

En este paquete de análisis se incluyen todas las actividades relacionadas con el ingreso o consulta de las calificaciones de los alumnos del plantel. Tiene relación directa con el caso de uso Calificaciones (Figura 3.6).

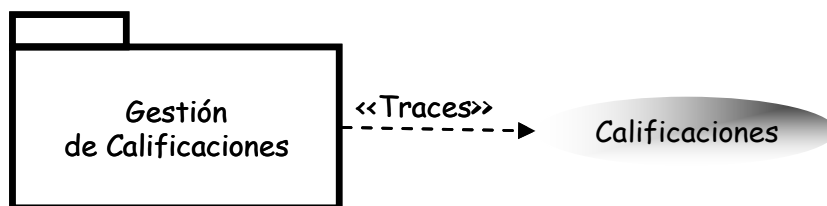


Figura 3.6. Paquete del Análisis del Caso de Uso Calificaciones.

### 3.7.2.4. Paquete del Análisis del Caso de Uso Mostrar Ayuda

En este paquete de análisis se incluyen todas las actividades relacionadas con la ejecución de la ayuda del sistema, cuando cualquier usuario esté en contacto con el mismo. Según la figura 3.7, tiene relación directa con el caso de uso Mostrar Ayuda.

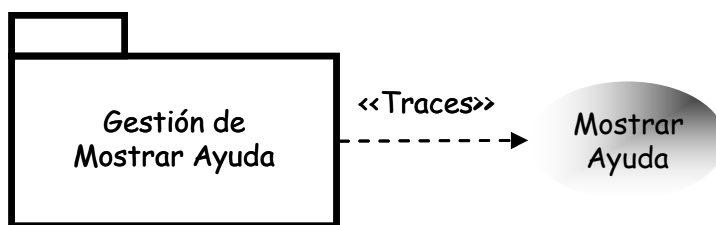


Figura 3.7. Paquete del Análisis del Caso de Uso Mostrar Ayuda.

### 3.7.2.5. Paquete del Análisis del Caso de Uso Ingresos y Egresos.

En este paquete de análisis se incluyen todas las actividades relacionadas con los pagos de matrícula (ingresos) por parte de los representantes de los alumnos (mensualidades), los pagos emitidos (egresos) a todo el personal del plantel.

Tiene relación directa con el caso de uso Ingresos y Egresos (Figura 3.8).

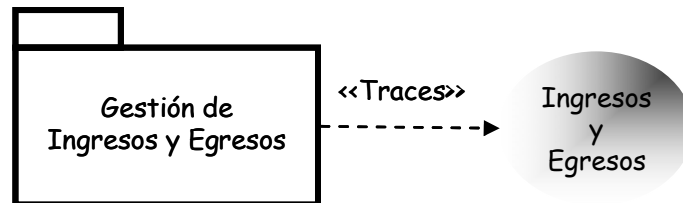


Figura 3.8. Paquete del Análisis del Caso de Uso Ingresos y Egresos

### 3.7.2.6. Paquete del Análisis del Caso de Uso Constancias y Reportes

En este paquete de análisis se incluyen todas las actividades relacionadas con la solicitud e impresión de constancias y reportes requeridos por los alumnos y el personal administrativo y directivo del plantel. Tiene relación directa con el caso de uso Constancias y Reportes (Figura 3.9).



Figura 3.9. Paquete del Análisis del Caso de Uso Constancias y Reportes.

### 3.7.2.7. Paquete del análisis del Caso de Uso Configurar Sistema

Según la figura 3.10, en este paquete de análisis se incluyen todas las actividades relacionadas con la configuración del sistema: como los datos del plantel, del personal directivo y administrativo para el buen funcionamiento durante todo el año escolar.

Tiene relación directa con el caso de uso Configurar Sistema.

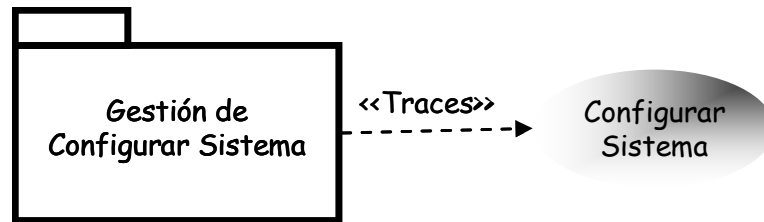


Figura 3.10. Paquete del Análisis del Caso de Uso Configurar Sistema.

### 3.7.3. Diagrama de Clases de Análisis

Durante esta fase, el modelo de análisis se conforma por la línea base de la arquitectura. Se construirá el modelo de análisis y sólo se desarrollará el 10% de los casos de uso definidos.

#### 3.7.3.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Alumnos

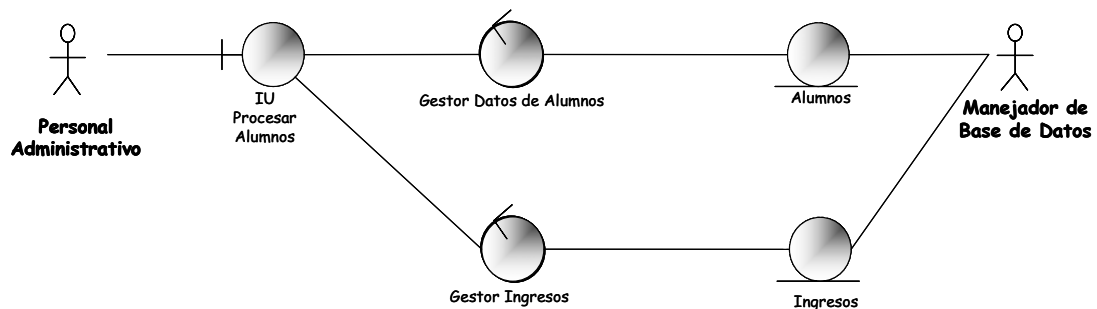


Figura 3.11. Diagrama de Clases de Análisis Procesar Alumnos

El paquete del análisis del caso de uso Procesar Alumnos presenta la realización del caso de uso Procesar Alumno.

Actores: Personal Administrativo, Administrador de Base de Datos.

Clases: IU Procesar Alumnos, Gestor Datos De Alumno, Gestor Ingresos, Alumnos, Ingresos.

Para la realización del caso de uso Procesar Alumnos el actor Personal Administrativo hace uso de la IU Procesar Alumnos para ingresar, actualizar, consultar y retirar los datos de los alumnos. Para ingresar los datos del alumno dentro del sistema (inscripción) la interfaz solicita los datos que deben ser facilitados por el Personal Administrativo. La clase de control Gestor Datos de Alumnos va a la Base de Datos para ingresar los datos al sistema, guardando la información del alumno con la ayuda de la clase de entidad Alumnos. Por medio de la clase de control Gestor de Ingresos se procesa el ingreso de dinero por concepto de inscripción. El sistema emite una constancia de inscripción y recibo de pago. Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de las clases de entidad: Ingresos y Alumnos. Para la consulta de los datos del alumno la interfaz solicita los datos de búsqueda que deben ser facilitados por el Personal Administrativo. La clase de control Gestor Datos de Alumnos va a la base de datos para buscar al alumno con la ayuda de la clase de entidad Alumnos y al conseguir la información la presenta en la interfaz con la ayuda de la clase IU Procesar Alumnos.

Para actualizar los datos del alumno dentro del sistema, la interfaz solicita la cédula del alumno para la actualización que debe ser facilitada por el Personal Administrativo. La clase de control Gestor Datos de Alumnos va a la base de datos para buscar los datos del alumno con la ayuda de la clase de entidad Alumnos y presentarlos en la interfaz por medio de la clase IU Procesar Alumnos. El Personal Administrativo realiza las actualizaciones y los cambios necesarios en los datos del alumno. Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de la clase de entidad Alumnos.

Para el retiro de los datos del alumno dentro del sistema, la interfaz solicita la cédula del alumno a retirar que debe ser facilitado por el Personal Administrativo. La

clase de control Gestor Datos de Alumnos va a la base de datos para buscar los datos del alumno, si los encuentra los presenta en la interfaz por medio de la clase IU Procesar Alumno con la ayuda de la clase de entidad Alumnos. Pero antes por medio de la clase de control Gestor de Ingresos se verifica la solvencia del alumno. Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de las clases de entidad: Ingresos y Alumnos. (Figura 3.11).

#### **3.7.4. Diagrama de Colaboración**

Se utilizarán diagramas de colaboración para detallar los casos de uso que forman parte de la línea base de la arquitectura. Estos diagramas de interacción enfatizan la organización estructural de los objetos que envían y reciben mensajes. Son diagramas que muestran las interacciones organizadas alrededor de instancias y de los enlaces entre ellas. En fin, estos diagramas permiten alcanzar un nuevo nivel de desarrollo para la arquitectura del sistema.

##### **3.7.4.1. Diagrama de Colaboración del Caso de Uso Procesar Alumnos**

La colaboración del Caso de Uso Procesar Alumnos presenta la elaboración de las actividades relacionadas al ingreso, actualización, consulta y retiro de los datos de los alumnos del plantel.

La figura 3.12 muestra que el actor Personal Administrativo ejecuta el caso de uso Procesar Alumnos para ingresar (inscribir) los datos del alumno en el sistema (enlace 1) que activa el objeto IU Procesar Alumnos, éste a su vez solicita al usuario los datos del alumno (enlace 2) una vez confirmada la inscripción (enlace 3) el objeto Gestor Datos de Alumnos es activado para enviar los datos (enlace 4) que serán almacenados en la base de datos por medio del objeto Alumnos (enlace 5). El objeto

Gestor Ingresos es activado para procesar el ingreso de dinero por concepto de pago de matrícula (enlace 6) y a través del objeto Ingresos se envía y se almacena la información de pago en la base de datos (enlace 7 y enlace 8).

Para retirar al alumno del plantel, el actor Personal Administrativo ejecuta el proceso Retirar Alumno (enlace 9), en la interfaz escribe los datos de retiro solicitando su búsqueda (enlace 10). Cuando el actor solicita los datos del alumno el objeto Gestor Datos de Alumnos busca los datos (enlace 11), obteniéndolos por medio del objeto Alumnos (enlace 12) a través de una consulta a la base de datos y luego los muestra en la interfaz (enlace 13), finalmente el usuario confirma el retiro (enlace 14) activando al objeto Gestor Ingresos para calcular la deuda pendiente del alumno (enlace 15), seguidamente se envían los datos de ingreso por retiro (enlace 16) y guardados (enlace 17) en la base de datos por medio del objeto Ingresos. La información de retiro es enviada (enlace 18) a la base de datos y se ejecuta el proceso por medio del objeto Alumnos (enlace 19).

Para consultar los datos del alumno, el actor Personal Administrativo solicita la ejecución del proceso Consultar Alumno (enlace 20), en la interfaz escribe los datos para la consulta (enlace 21). Cuando el actor solicita los datos del alumno el objeto Gestor Datos de Alumnos busca los datos del alumno (enlace 22), los obtiene por medio del objeto Alumnos a través de una consulta a la base de datos (enlace 23) y luego los muestra en la interfaz (enlace 24).

Para modificar los datos del alumno, el actor Personal Administrativo solicita la ejecución del proceso Modificar Alumno (enlace 25), en la interfaz escribe los datos para la modificación (enlace 26). Cuando el actor solicita los datos del alumno el objeto Gestor Datos de Alumnos busca la información (enlace 27), los obtiene por medio del objeto Alumnos a través de una consulta a la base de datos (enlace 28) y luego los muestra en la interfaz (enlace 29), el actor realiza las modificaciones y

cambios necesarios a los datos del alumno en la interfaz, confirma la modificación de los datos (enlace 30) activando al objeto Gestor Datos de Alumnos para enviar los datos modificados (enlace 31) y guardarlos en la base de datos por medio del objeto Alumnos (enlace 32).

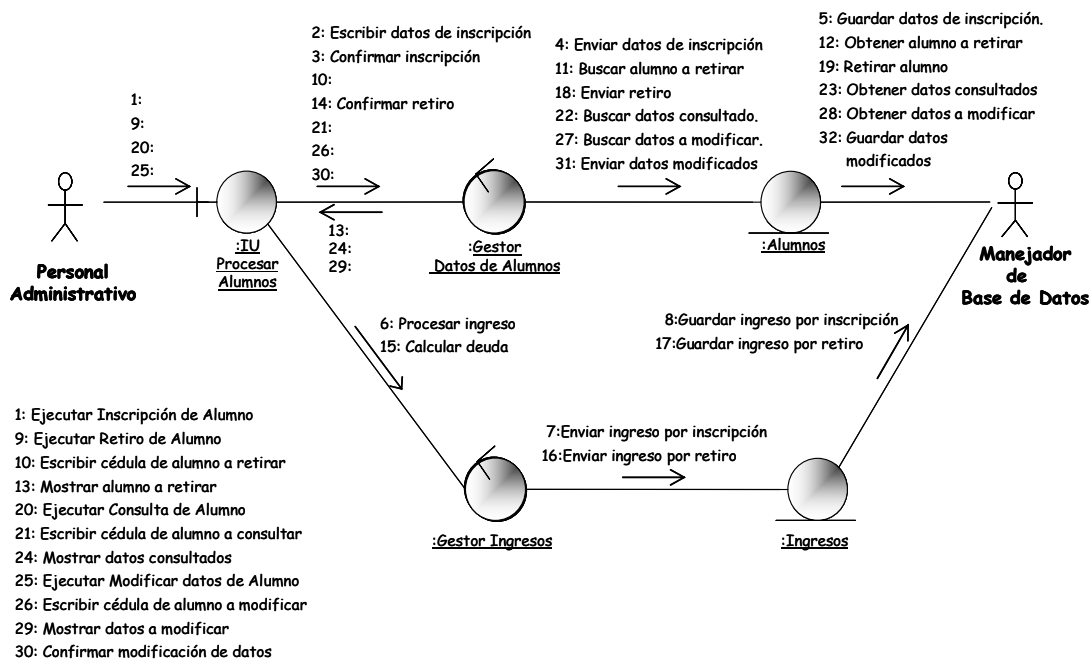


Figura 3.12. Diagrama de Colaboración del Caso de Uso Procesar Alumnos.

### 3.8. Diseño

Este flujo de trabajo permite obtener subsistemas de diseño para lograr una arquitectura más robusta, dando forma al sistema mientras que intenta preservar la estructura definida por el modelo de análisis.

#### 3.8.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso.



Además el modelo de diseño sirve de abstracción en la implementación del sistema y es utilizado como una entrada fundamental en las actividades de implementación.

Se representa por un sistema de diseño que denota el subsistema de diseño para hacerlo más manejable. En este esbozo del proyecto se identifican los subsistemas de desarrollo que permitirán la elaboración de la aplicación y la forma que se verán sus interfaces una vez desarrolladas.

### 3.8.2. Subsistemas de Diseño

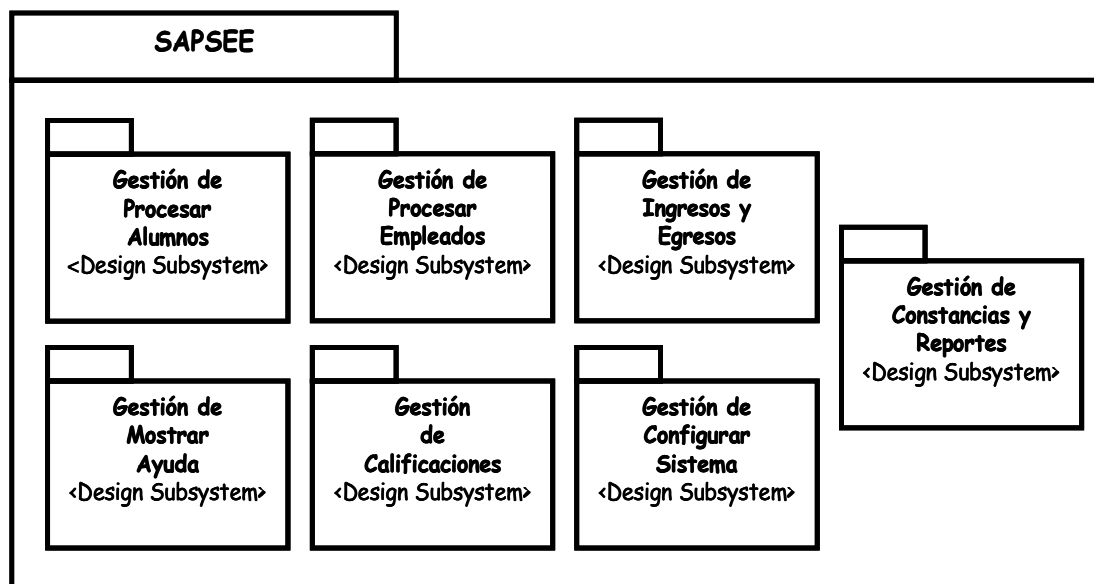


Figura 3.13. Subsistemas de Diseño.

Luego que en el flujo de trabajo anterior se identificaron los paquetes de análisis por medio del análisis inicial (clase del análisis), el paso siguiente es identificar los subsistemas de diseño que tendrán relación directa con cada paquete de análisis y establecer la relación entre ellos (clases del análisis y clases del diseño). (Figura 3.13).

### 3.8.2.1. Subsistema de Diseño de Gestión de Procesar Alumnos.

El Subsistema de Gestión de Procesar Alumnos contendrá la realización de casos de uso relacionados con las actividades directas al manejo de la información de los alumnos.

Tiene relación directa con el Paquete del Análisis del caso de uso Procesar Alumnos (Figura 3.14).

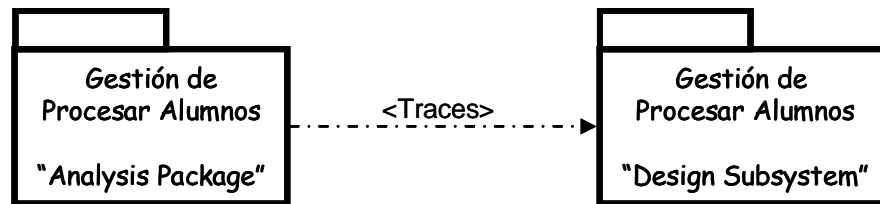


Figura 3.14. Subsistemas de Diseño de Gestión de Procesar Alumnos.

### 3.8.2.2. Subsistema de Diseño de Gestión de Procesar Empleados.

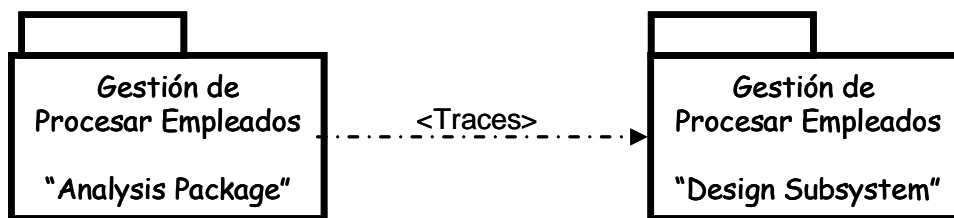


Figura 3.15. Subsistemas de Diseño de Gestión de Procesar Empleados.

El Subsistema de Gestión de Procesar Empleados contendrá la realización de casos de uso relacionados con las actividades directas al manejo de la información de los alumnos

Tiene relación directa con el Paquete del Análisis del caso de uso Procesar Empleados (Figura 3.15).

### 3.8.2.3. Subsistema de Diseño de Gestión de Ingresos y Egresos

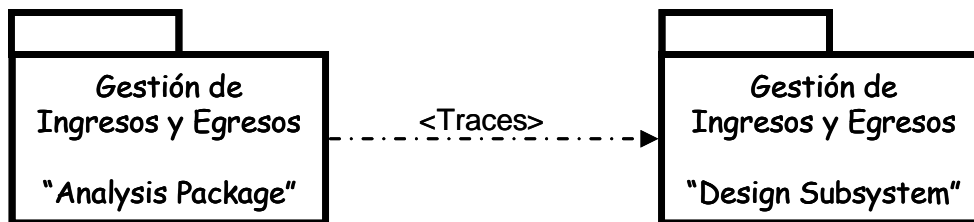


Figura 3.16. Subsistemas de Diseño de Gestión de Ingresos y Egresos.

El Subsistema de Gestión de Ingresos y Egresos vinculará todos los casos de uso relacionados a gestionar los ingresos y egresos en la institución. Tiene relación directa con el Paquete del Análisis del caso de uso Ingresos y Egresos. (Figura 3.16).

### 3.8.2.4. Subsistema de Diseño de Gestión de Mostrar Ayuda

El Subsistema de Diseño de Gestión de Mostrar Ayuda, se relaciona con todas las actividades pertenecientes a la ejecución de la ayuda del sistema, para su correcto uso por parte de los usuarios. Tiene relación directa con el Paquete de Análisis del caso de uso Mostrar Ayuda (Figura 3.17).

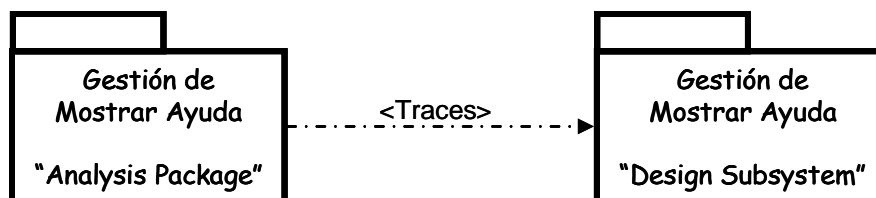


Figura 3.17. Subsistemas de Diseño de Gestión de Mostrar Ayuda.

### 3.8.2.5. Subsistema de Diseño de Gestión de Constancias y Reportes

El Subsistema de Gestión de Constancias y Reportes permite el desarrollo de todos los casos de uso que ayudan a gestionar las actividades relacionadas con el

procesamiento de constancias y reportes en el plantel. Según la figura 3.18 este subsistema tiene relación directa con el Paquete del Análisis del caso de uso Constancias y Reportes.

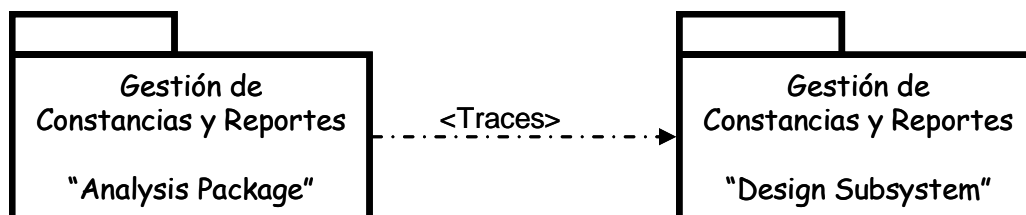


Figura 3.18. Subsistemas de Diseño de Gestión de Constancias y Reportes.

#### 3.8.2.6. Subsistema de Diseño de Gestión de Calificaciones

El Subsistema de Gestión de Calificaciones permite el desarrollo de todos los casos de uso que ayudan a gestionar las actividades relacionadas con el ingreso de calificaciones (obtenidas por los estudiantes durante todo el año escolar) al sistema.

La figura 3.19 muestra que este subsistema tiene relación directa con el Paquete de Análisis del caso de uso Constancias y Reportes.

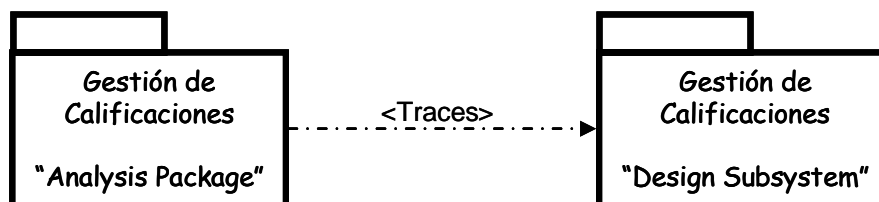


Figura 3.19. Subsistemas de Diseño de Gestión de Calificaciones.

#### 3.8.2.7. Subsistema de Diseño de Gestión de Configurar Sistema

El Subsistema de Diseño de Gestión de Configurar Sistema contiene todos los casos de usos destinados a preparar el sistema con toda la información necesaria para

la ejecución y control del mismo durante su aplicación en el plantel. Tiene relación directa con el Paquete del Análisis del caso de uso Configurar Sistema (Figura 3.20).

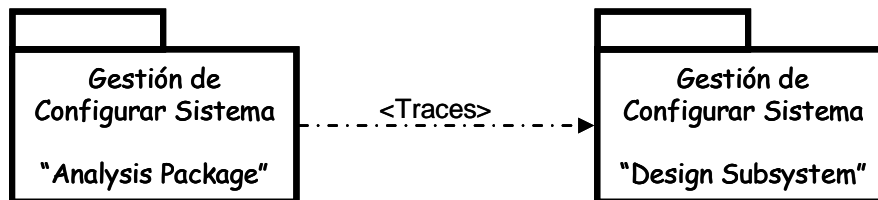


Figura 3.20. Subsistemas de Diseño de Gestión de Configurar Sistema.

### 3.8.3. Diagrama de Clases del Diseño

El diagrama de clases del diseño sirve para visualizar las clases que componen el sistema y cómo se relacionan entre sí. Permiten mostrar lo que el sistema puede hacer, además, de cómo puede ser construido. En esta fase se desarrollará el 10% de los casos de uso.

Para las realizaciones de casos de uso se emplearán diagramas de clases y de interacción para especificar la comunicación entre los objetos del diseño. Se observará de manera más clara el funcionamiento del sistema ya que los diagramas de secuencia hacen énfasis en la ordenación temporal de los mensajes entre los objetos del diseño.

#### 3.8.3.1. Realización del Caso de Uso Procesar Alumnos

La Figura 3.21 muestra las clases del diseño para el caso de uso Procesar Alumnos. La clase ProcesarAlumnos es la interfaz que permite ingresar, retirar, modificar y consultar los datos de los alumnos. La clase GestorAlumnos es la encargada de realizar las operaciones solicitadas por la interfaz. La clase GestorIngresos se encarga de ingresar los datos de los pagos de mensualidad

(matriculación) realizados por los alumnos. Alumnos e Ingresos representan el almacenamiento en el sistema de los datos, tanto de los alumnos como de los pagos.

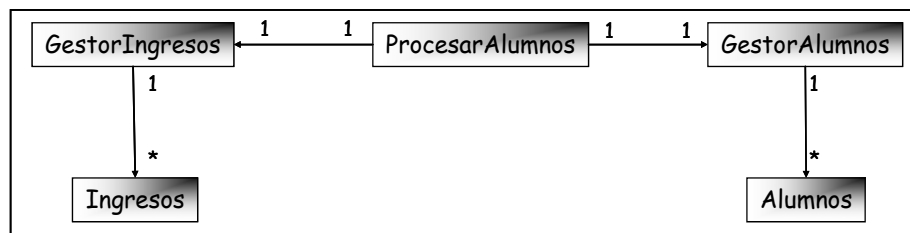


Figura 3.21. Diagrama de Clases del Diseño del Caso de Uso Procesar Alumnos

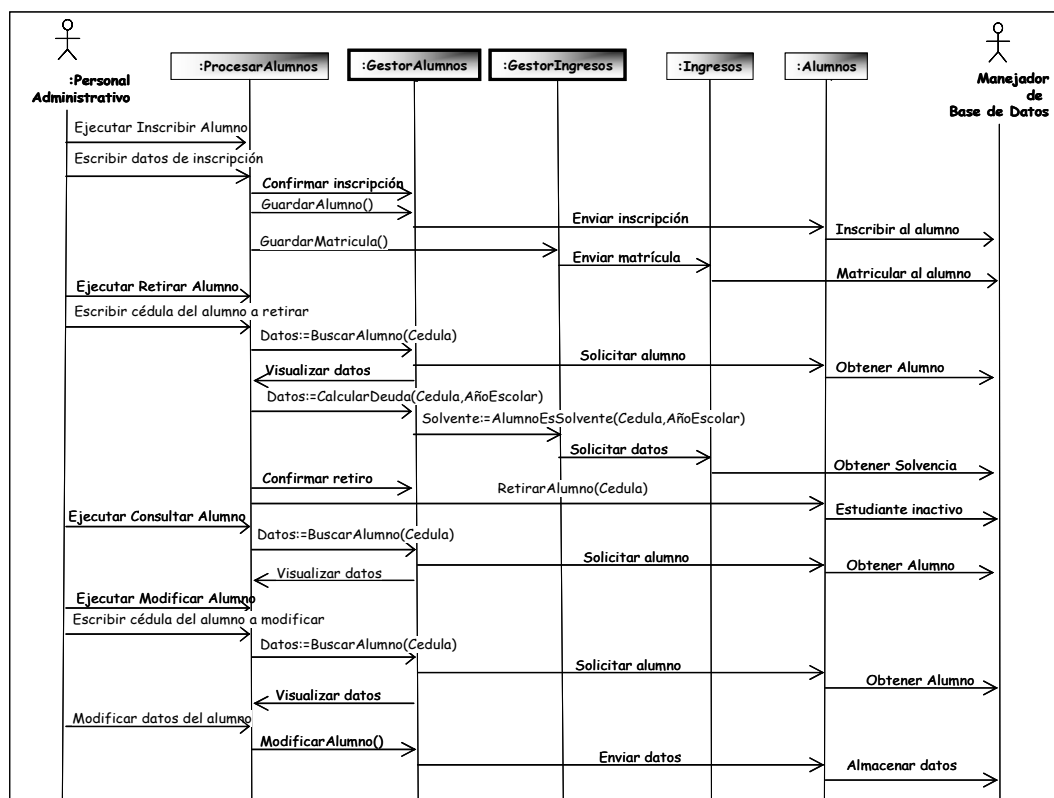


Figura 3.22. Diagrama de Secuencia del Caso de Uso Procesar Alumnos.

La Figura 3.22 muestra el diagrama de secuencia del caso de uso Procesar Alumnos, donde el Personal Administrativo lo ejecuta para ingresar los datos de los alumnos a inscribir por medio de la interfaz ProcesarAlumnos. Al conformar todos

los datos de inscripción la interfaz llama al método `GuardarAlumno()` de la clase `GestorAlumnos`, quien se conecta a la base de datos y almacena la información dada en el sistema. Cuando el usuario confirma el pago de matrícula, la interfaz llama al método `GuardarMatrícula()` de la clase `GestorIngresos`, quien envía los datos para ser almacenados dentro del sistema.

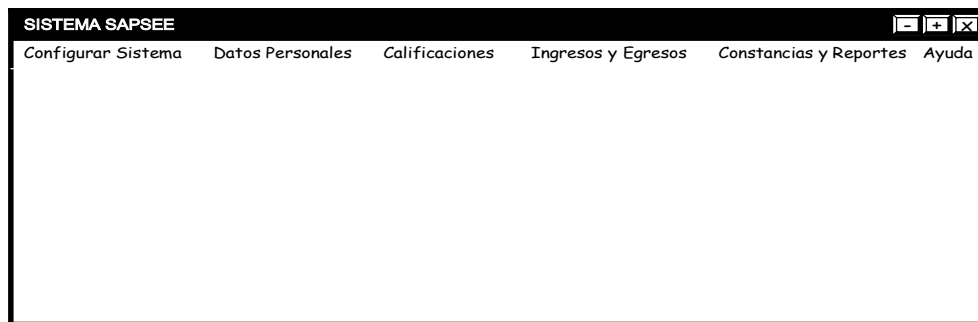
Para retirar los datos del alumno, el Personal Administrativo ingresa a través de la interfaz `ProcesarAlumnos` la cédula del alumno a retirar la cual activa el método `BuscarAlumno()` de la clase `GestorAlumnos`; ella obtiene los datos por consulta a la BDD para que la interfaz los muestre, e inmediatamente se activa el método `CalcularDeuda()` del `GestorAlumnos` quien a su vez activa el método `AlumnoEsSolvente()` para obtener la solvencia o no del alumno, el usuario confirma el retiro y se activa el método `RetirarAlumno()` de la clase `GestorAlumno` para finalmente registrar el retiro en la BDD del sistema. Para consultar los datos de los alumnos, el Personal Administrativo ingresa a través de la interfaz `ProcesarAlumnos` la cédula del alumno a retirar para activar el método `BuscarAlumno()` de la clase `GestorAlumnos`; ella obtiene los datos por consulta a la BDD para que la interfaz los muestre en pantalla. Para modificar los datos de los alumnos, el Personal Administrativo ingresa a través de la interfaz la cédula del alumno a modificar para activar el método `ModificarAlumno()` de la clase `GestorAlumnos`, una vez modificados los datos del alumno se envía dicha información hacia la BDD del sistema.

#### **3.8.4. Prototipos de Interfaz de Usuario**

Es necesario diseñar una interfaz de usuario que permita a los actores del sistema: Personal Administrativo y Personal Directivo, llevar a cabo los casos de uso de manera eficiente. Por tal motivo, el diseño de éstas se hará orientado a facilitar al máximo la ejecución de los casos de uso por parte de los usuarios finales.

Para esto, se procederá a realizar un diseño lógico de las interfaces basado en lo que necesita el usuario para ejecutar el caso de uso, para luego crear el diseño físico de la misma y desarrollar prototipos que ilustren cómo puede utilizar el usuario el sistema para efectuar los casos de uso y así poder determinar la optimización de las actividades académicas y administrativas dentro del plantel. La descripción de los prototipos se hace a continuación:

#### 3.8.4.1. Prototipo de Interfaz Principal



**Figura 3.23. Prototipo de Interfaz Principal**

También es necesario que el sistema posea una interfaz principal cuya responsabilidad sea la de enlazar las diferentes interfaces diseñadas para los casos de uso, es decir, una interfaz desde la cual el usuario pueda invocar los casos de uso haciendo llamadas a las interfaces de cada uno de ellos. La Figura 3.23 muestra que la opción ideal es utilizar una interfaz de múltiples documentos (MDI) que posea un menú principal con las diferentes opciones a las cuales el usuario pueda acceder según el proceso puesto en marcha.

#### 3.8.4.2. Prototipo de Interfaz de Gestión

Cada uno de los casos de uso analizados hasta ahora posee una o más clases de entidad asociadas, que representan objetos manejados por el sistema. El usuario



necesita acceder a la información contenida en los mismos y manipularla. Para ello, la interfaz debe contar con elementos que le permitan realizar tareas de ingreso, búsqueda, actualización y eliminación de los datos correspondientes al objeto en cuestión.

Por este motivo y con la intención de estandarizar las distintas interfaces requeridas se ha diseñado una interfaz modelo que generalice las funciones que deben poseer todas las interfaces relacionadas con los casos de uso estudiados.

Se ha diseñado un prototipo de interfaz que brinda al usuario la posibilidad de manipular la información referente a los objetos, a través de diversos controles y especialmente una barra de herramientas que ofrece las opciones antes mencionadas lo que ofrece un fácil acceso a las operaciones básicas referentes al objeto que se esté manipulando en ese momento. Esta interfaz es muy sencilla ya que se requiere una buena comunicación entre el usuario y el sistema. (Figura 3.23).

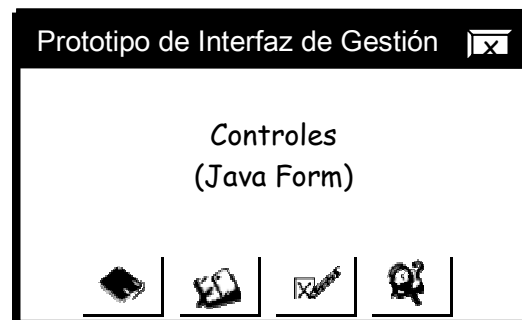
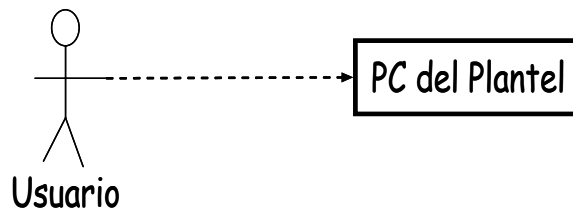


Figura 3.23. Prototipo de Interfaz de Gestión

### 3.8.5. Nodos y Configuraciones de Red.

Como el sistema SAPSEE estará funcionando en un PC y además la institución no cuenta con una red local, la configuración estará diseñada como muestra la Figura 3.24.



**Figura 3.24. Identificación de Nodos y Configuraciones de Red.**

### 3.9. Planificación de las Sigüientes Fases

1. Fase de Elaboración: En la fase de elaboración se especificarán con detalles la mayoría de los casos de uso, se obtiene la línea base de la arquitectura expresada en forma de vistas de algunos modelos del sistema (modelo de casos de uso, modelo de análisis, modelo de diseño) obteniendo una nueva versión de cada uno de esos modelos.
2. Fase de Construcción: Se usarán los requerimientos para el desarrollo, es decir, convirtiéndose en una versión beta, se termina el software con su arquitectura, la línea base de la arquitectura se convierte en un sistema completo, estable y en su contenido están todos los casos de uso acabados para su desarrollo. Se obtienen los modelos completos del sistema.
3. Fase de Transición: Durante esta fase se entregará a los usuarios el software (sistema) terminado para la instalación del sistema, así como la construcción del manual de usuario y la arquitectura final. Serán los usuarios quienes se encarguen de su posterior instalación y ejecución.

### 3.10. Resumen de la Fase de Inicio

Al finalizar la primera fase del proceso unificado se obtuvo lo siguiente:

1. Lista de Requisitos para la construcción del sistema y que dan paso al modelo inicial de los casos de uso.
2. Modelo de Dominio para la comprensión del sistema.
3. Lista de Riesgos Críticos que ayudan a detener las posibles fallas del sistema.
4. Modelo Inicial de Caso de Uso como representación de la captura de los requisitos del sistema.
5. Descripción de la ejecución de cada caso de uso del sistema.
6. Modelo de Análisis identificado por Paquetes del Análisis de cada uno de los casos de uso iniciales. Diagrama de Clases del Análisis y Diagrama de Colaboración de un caso de uso.
7. Modelo de Diseño identificado por Subsistemas de Diseño de cada uno de los Paquetes del Análisis. Diagrama de Clases del Diseño de un caso de uso.
8. El Prototipo de Interfaz de Usuario, como una propuesta de interfaz de gestión e interfaz principal.
9. Planificación de las siguientes fases.

## **CAPÍTULO 4**

### **FASE DE ELABORACIÓN**

#### **4.1. Introducción**

La Fase de Elaboración define la arquitectura base del sistema, estableciendo la solidez para guiar el trabajo durante las fases de Construcción y Transición, es decir, se extiende el análisis de la arquitectura sobre el modelo de casos de uso. Se continúa con la observación y control de los riesgos críticos que aún puedan quedar para el control total del sistema. En esta fase se construye sobre el trabajo de la Fase de Inicio.

El objetivo principal de esta fase es conseguir una línea base de la arquitectura que soporte la especificación inicial del sistema y mitigue los riesgos identificados y descritos en la fase inicial.

Esta fase consta de diferentes actividades las cuales serán descritas a continuación.

#### **4.2. Modelo de Casos de Uso**

En esta fase no se encontraron nuevos casos de usos para el sistema por lo tanto el modelo queda descrito como fue definido en la Fase de Inicio (Figura 3.2).

#### **4.3. Análisis**

En esta fase se analizan los restantes casos de uso para refinarlos, comprender mejor el sistema propuesto y discernir la interferencia de unos con otros.

### **4.3.1. Análisis de la Arquitectura**

En la fase anterior se ha obtenido una arquitectura factible pero se necesita ampliarla para obtener una línea base hacia la arquitectura ejecutable.

### **4.3.2. Modelo de Análisis**

Durante esta fase, el modelo de análisis se conforma por la línea base de la arquitectura.

Se construirá el modelo de análisis sobre el borrador del modelo esbozado en la fase de inicio.

### **4.3.3. Diagrama de Clase de Análisis**

Se realiza el análisis de casos de uso tomando como punto de vista la arquitectura y se obtiene el diagrama de clase de análisis para cada caso de uso restante del sistema

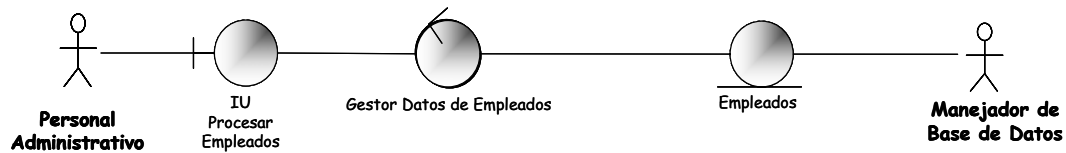
Esto nos permite robustecer el modelo de análisis plasmado anteriormente.

#### **4.3.3.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Empleados**

El paquete del análisis del caso de uso Procesar Empleados presenta la realización del caso de uso Procesar Empleados.

Actores: Personal Administrativo, Administrador de Base de Datos.

Clases: IU Procesar Empleados, Gestor de Datos de Empleados, Empleados.



**Figura 4.1. Diagrama de Clases de Análisis del Caso de Uso Procesar Empleados**

Para la realización del caso de uso Procesar Empleados el actor Personal Administrativo hace uso de la IU Procesar Alumnos para ingresar, actualizar, consultar y retirar los datos de los empleados.

Para ingresar los datos del empleado dentro del sistema la interfaz solicita los datos que deben ser facilitados por el Personal Administrativo. La clase de control Gestor Datos de Empleados va a la base de datos para ingresar los datos al sistema, guardando la información del empleado con la ayuda de la clase de entidad Empleados.

Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de la clase de entidad Alumnos.

Para la consulta de los datos del empleado la interfaz solicita la cédula del empleado que debe ser facilitada por el Personal Administrativo. La clase de control Gestor Datos de Empleados va a la base de datos para buscar al empleado con la ayuda de la clase de entidad Empleados y al conseguir la información la presenta en la interfaz con la ayuda de la clase IU Procesar Empleados.

Para actualizar los datos del empleado dentro del sistema, la interfaz solicita la cédula del empleado para la actualización que debe ser facilitada por el Personal Administrativo. La clase de control Gestor Datos de Empleados va a la base de datos para buscar los datos del empleado con la ayuda de la clase de entidad Empleados y

presentarlos en la interfaz por medio de la clase IU Procesar Empleados. El Personal Administrativo realiza las actualizaciones y los cambios necesarios en los datos. Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de la clase de entidad Empleados.

Para el retiro de los datos del empleado del sistema la interfaz solicita la cédula del empleado a retirar que debe ser facilitada por el Personal Administrativo. La clase de control Gestor Datos de Empleados va a la base de datos para buscar los datos del empleado, si los encuentra los presenta en la interfaz por medio de la clase IU Procesar Empleados con la ayuda de la clase de entidad Empleados. Toda esta información se almacena en la base de datos correspondiente por medio del actor Manejador de Base de Datos y a través de la clase de entidad Empleados (Figura 4.1).

#### **4.3.3.2. Diagrama de Clases de Análisis del Caso de Uso Ingresos y Egresos**

El paquete de análisis del Caso de Uso Ingresos y Egresos presenta la realización del caso de uso Ingresos y Egresos.

Actores: Personal Administrativo, Manejador de Base de Datos.

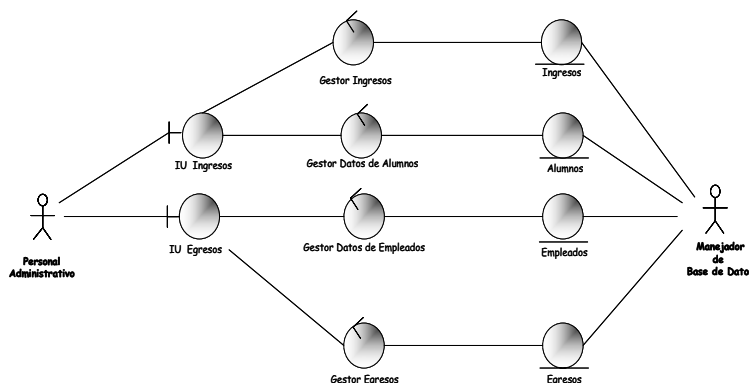
Clases: IU Ingresos, IU Egresos, Gestor Ingresos, Gestor Egresos, Gestor Datos de Alumnos, Gestor Datos de Empleados, Alumnos, Ingresos, Empleados, Egresos.

La figura 4.2 muestra que para la realización del caso de uso Ingresos y Egresos el actor Personal Administrativo hace uso de la IU Ingresos para registrar los pagos de matrícula o mensualidades por parte de los alumnos (ingresos), en su defecto usará la IU Egresos para registrar los pagos emitidos al personal del plantel. El Gestor Ingresos genera los datos necesarios para procesar los pagos de matrícula

de los alumnos. El Gestor Datos de Alumnos precisa los datos del Alumno que cancela la matrícula. El Gestor Egresos registra los pagos realizados al personal del plantel (pagos de nómina) que se ejecutan quincenalmente.

El Gestor Datos de Empleados generará los datos del empleado al cual se le cancelará el sueldo.

Toda la información de pago de matrícula, pago al personal es guardada en la Base de Datos del sistema por medio del Manejador Base de Datos y por medio de las clases de entidad: Alumnos, Empleados, Ingresos y Egresos.



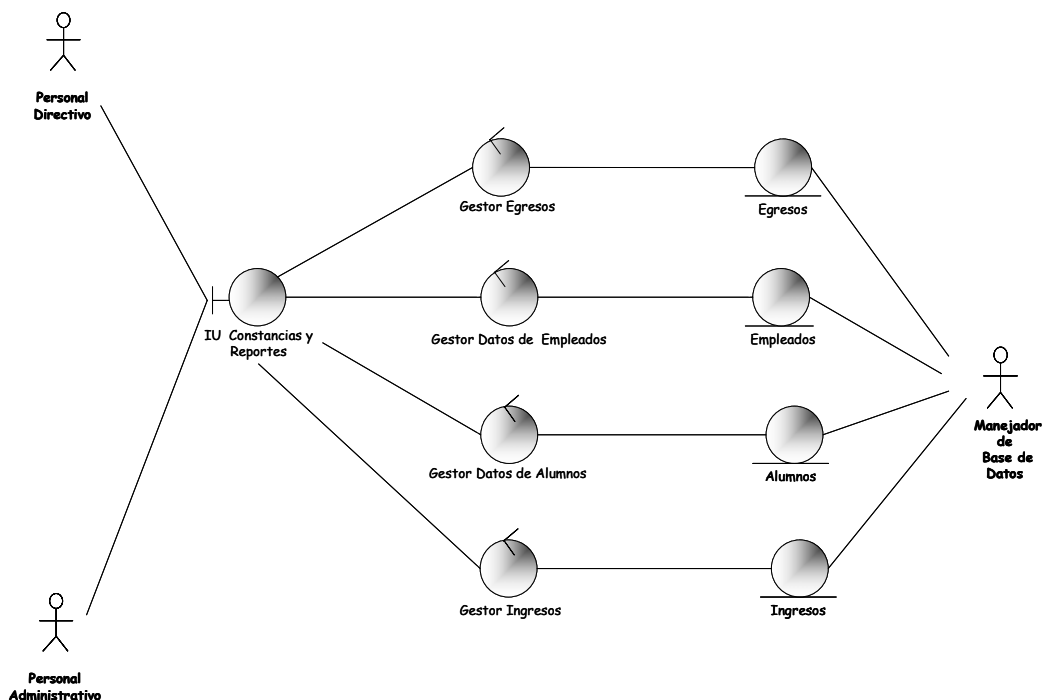
**Figura 4.2. Diagrama de Clases de Análisis del Caso de Uso Ingresos y Egresos.**

#### **4.3.3.3. Diagrama de Clases de Análisis del Caso de Uso Constancias y Reportes**

El paquete de análisis del Caso de Uso Constancias y Reportes presenta la realización del caso de uso Constancias y Reportes.

Actores: Personal Directivo, Personal Administrativo, Manejador de Base de Datos; Clases: IU Constancias y Reportes, Gestor Datos de Alumnos, Gestor Datos de Empleados, Gestor Ingresos, Gestor Egresos, Alumnos, Empleados, Ingresos, Egresos.





**Figura 4.3. Diagrama de Clases de Análisis del Caso de Uso Constancias y Reportes.**

La figura 4.3, muestra que para la realización del caso de uso Constancias y Reportes el Personal Administrativo ó el Personal Directivo hacen uso de la IU Constancias y Reportes para mostrar en pantalla o realizar impresiones de reportes, constancias, como por ejemplo: constancias de estudio, de buena conducta, boletín informativo (boleta de calificaciones), alumnos por edad, sexo, empleados del plantel ingresos diarios, ingresos mensuales,... El Gestor Datos de Alumnos, se encarga de solicitar los datos de los alumnos para emitir por pantalla cualquiera de las constancias: Constancia de Estudio, Constancia de Buena Conducta, Solvencia y Constancia de Inscripción y para mostrar los reportes: Alumnos por Sexo, Alumnos por Grado, Alumnos por Año Escolar, Boletín Informativo, así como también son útiles para su impresión si el usuario así lo requiere. El Gestor Ingresos, se encarga de solicitar los datos de los ingresos registrados en el sistema para mostrar cualquiera de los reportes: Ingresos Anuales, Ingresos Mensuales e Ingresos Diarios, así como también son útiles para su impresión, si el usuario lo requiere.

El Gestor Datos de Empleados, se encarga de solicitar los datos de los empleados registrados en el sistema para mostrar en pantalla o imprimir en un documento el reporte: Empleados del Plantel.

El Gestor Egresos, se encarga de solicitar los datos de los pagos al personal emitidos por el Plantel con el fin de mostrar por pantalla los reportes: Egresos Anuales, Egresos Mensuales y Egresos Diarios, así como también son útiles para su posterior impresión si el usuario lo requiere. Toda la información es procesada en la base de datos del sistema por medio del Manejador de Base de Datos con la ayuda de las clases de entidad: Alumnos, Empleados, Ingresos, Egresos y Calificaciones.

#### 4.3.3.4. Diagrama de Clases de Análisis del Caso de Uso Mostrar Ayuda

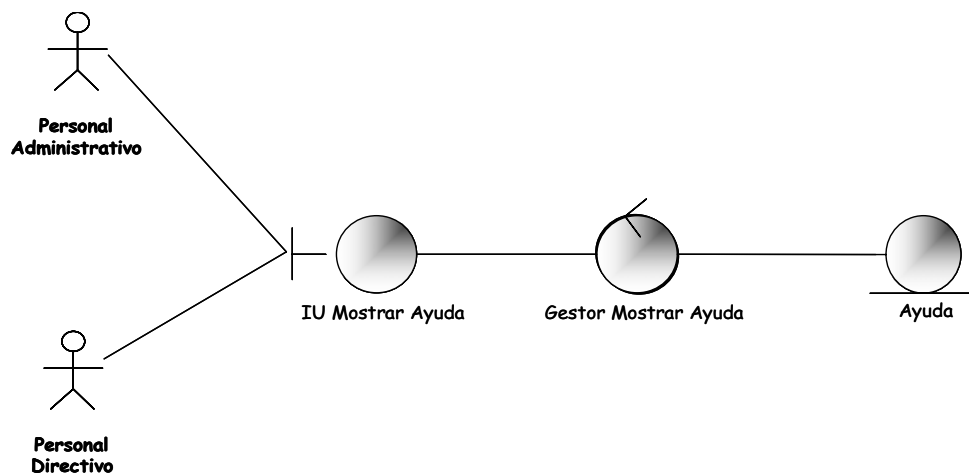


Figura 4.4. Diagrama de Clases de Análisis del Caso de Uso Mostrar Ayuda.

El paquete de análisis del Caso de Uso Mostrar Ayuda presenta la realización del caso de uso Mostrar Ayuda.

Actores: Personal Administrativo, Personal Directivo. Clases: IU Ingresar Mostrar Ayuda, Gestor Mostrar Ayuda, Ayuda.

Para la realización del caso de uso Mostrar Ayuda el Personal Administrativo ó el Personal Directivo hace uso de la IU Mostrar Ayuda para generar la guía de usuario del sistema SAPSEE. Dicha guía les dará las pautas a seguir para el uso del sistema. El Gestor Mostrar Ayuda se encarga de generar la información requerida por el usuario. (Figura 4.4).

#### 4.3.3.5. Diagrama de Clases de Análisis del Caso de Uso Calificaciones

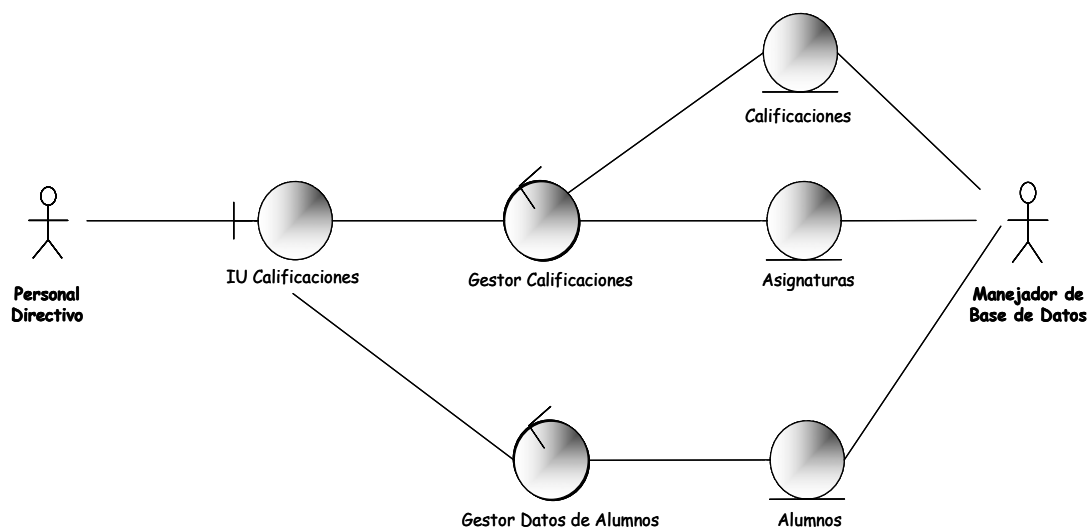


Figura 4.5. Diagrama de Clases de Análisis del Caso de Uso Calificaciones.

El paquete de análisis del Caso de Uso Calificaciones presenta la realización del caso de uso Calificaciones.

Actores: Personal Directivo, Manejador de Base de Datos; Clases: IU Ingresar Calificaciones, Gestor Calificaciones, Gestor Alumnos, Calificaciones, Alumnos, Asignaturas. Para la realización del caso de uso Calificaciones el actor Personal Directivo usa la IU Calificaciones para ingresar las calificaciones obtenidas por los alumnos. Para el ingreso de las calificaciones el Gestor Calificaciones genera la información sobre las asignaturas cursadas por el alumno durante el año escolar en

curso con la ayuda de las clases: Asignaturas y Alumnos. El Gestor Datos de Alumnos solicita a la base de datos del sistema los datos del Alumno al cual se le van a ingresar las calificaciones obtenidas durante el año escolar en curso. La información de las calificaciones es almacenada en la Base de Datos del sistema por medio del actor Manejador de Base de Datos (Figura 4.5).

#### 4.3.3.6. Diagrama de Clases de Análisis del Caso de Uso Configurar Sistema

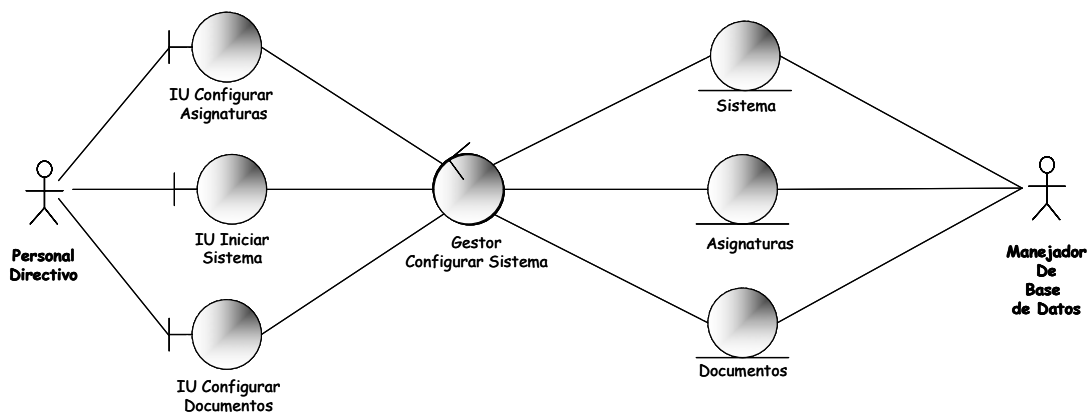


Figura 4.6. Diagrama de Clases de Análisis del Caso de Uso Configurar Sistema.

El paquete de análisis del Caso de Uso Configurar Sistema presenta la realización del caso de uso Configurar Sistema.

Actores: Personal Directivo, Manejador de Base de Datos; Clases: IU Iniciar Sistema, IU Configurar Sistema, IU Configurar Documentos, Gestor Configurar Sistema, Sistema, Asignaturas, Documentos.

Para la realización del caso de uso Configurar Sistema, el actor Personal Directivo utiliza la IU Iniciar Sistema para ingresar los datos iniciales de configuración: datos del sistema (RIF, NIT, Código DEA, Teléfonos, Dirección,...), la IU Configurar Asignaturas para ingresar las asignaturas dictadas en la institución y

la IU Configurar Documentos para ingresar los documentos emitidos por el plantel; la clase de control Gestor Configurar Sistema se encarga de procesar dichos datos. Toda la información es cargada a la Base de Datos del Sistema por medio del actor Manejador de Base de Datos. (Figura 4.6).

#### **4.3.4. Diagrama de Colaboración**

Se organiza la estructura de los objetos que envían y reciben mensajes por medio de enlaces entre ellos. Su comportamiento define de forma más clara la manera como los usuarios interactúan con el sistema.

##### **4.3.4.1. Diagrama de Colaboración del Caso de Uso Procesar Empleados**

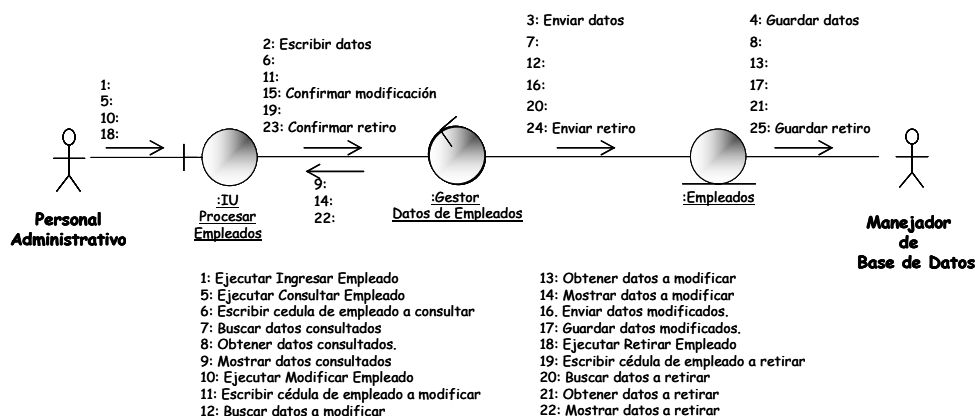
La colaboración del Caso de Uso Procesar Empleados presenta la elaboración de las actividades relacionadas al ingreso, actualización, consulta y retiro de los datos de los empleados del plantel.

La figura 4.7 muestra que el actor Personal Administrativo ejecuta el caso de uso Procesar Empleados para ingresar los datos del empleado en el sistema (enlace 1) que activa el objeto IU Procesar Alumnos, quien a su vez solicita al usuario los datos del empleado (enlace 2).

Una vez confirmado el ingreso, el objeto Gestor Datos de Empleados es activado para enviar los datos (enlace 3) y almacenarlos por medio del objeto Empleados (enlace 4) en la base de datos.

Para consultar los datos del empleado, el actor Personal Administrativo solicita la ejecución del proceso Consultar Empleado (enlace 5), en la interfaz escribe los datos para la consulta (enlace 6), cuando el actor solicita los datos del empleado el

objeto Gestor Datos de Empleados busca los datos del empleado (enlace 7), los obtiene por medio del objeto Empleados a través de una consulta a la base de datos (enlace 8) y luego los muestra en la interfaz (enlace 9).



**Figura 4.7. Diagrama de Colaboración del Caso de Uso Procesar Empleados**

Para modificar los datos del empleado, el actor Personal Administrativo solicita la ejecución del proceso Modificar Empleado (enlace 10), en la interfaz escribe los datos para la modificación (enlace 11), cuando el actor solicita los datos del empleado el objeto Gestor Datos de Empleados busca los datos del empleado (enlace 12), los obtiene por medio del objeto Empleados a través de una consulta a la base de datos (enlace 13) y luego los muestra en la interfaz (enlace 14); el actor realiza las modificaciones y cambios necesarios a los datos del empleado en la interfaz, confirma la modificación (enlace 15), activando al objeto Gestor Datos de Empleados para enviar la información (enlace 16) que se almacenará en la base de datos por medio del objeto Empleados (enlace 17).

Para retirar al empleado del plantel, el actor Personal Administrativo ejecuta el proceso Retirar Empleado (enlace 18), en la interfaz escribe los datos de retiro solicitando su búsqueda (enlace 19), cuando el actor solicita los datos del empleado el objeto Gestor Datos de Empleados los busca (enlace 20), obteniéndolos por medio del

objeto Empleados (enlace 21) a través de una consulta a la base de datos y luego los muestra en la interfaz (enlace 22), finalmente el usuario confirma el retiro (enlace 23). La información de retiro es enviada (enlace 24) a la base de datos y guardada por medio del objeto Empleados (enlace 25).

#### **4.3.4.2. Diagrama de Colaboración del Caso de Uso Ingresos y Egresos**

La Figura 4.8 muestra que la colaboración del Caso de Uso Ingresos y Egresos presenta la elaboración de las actividades relacionadas a los ingresos y pagos realizados por la institución.

El actor Personal Administrativo ejecuta el caso de uso Ingresos y Egresos por medio de los objetos: IU Ingresos (enlace 1) o IU Egresos (enlace 9).

Para el cobro de mensualidades el usuario facilita la cédula del alumno para la búsqueda de datos en el sistema (enlace 2), el objeto Gestor Datos de Alumnos solicita los datos a la base de datos quien los busca (enlace 3) y los obtiene por medio del objeto Alumnos (enlace 4) para ser mostrados luego en la interfaz (enlace 5), el usuario confirma la matrícula (enlace 6) y el Gestor Ingresos envía el monto pagado por el alumno (enlace 7) a la base de datos. Toda la información se almacena por medio del objeto Ingresos (enlace 8).

A través de la IU Egresos el usuario facilita la cédula del empleado para la búsqueda de los datos en el sistema (enlace 10), el objeto Gestor Datos de Empleados solicita los datos a la base de datos buscándolos (enlace 11) con la ayuda del objeto Empleados y obteniéndolos (enlace 12) para ser mostrados luego en la interfaz (enlace 13), el usuario confirma el pago al empleado (enlace 14) y el Gestor Egresos envía los datos del pago (enlace 15) a la base de datos para que se almacene la información por medio del objeto Egresos (16).

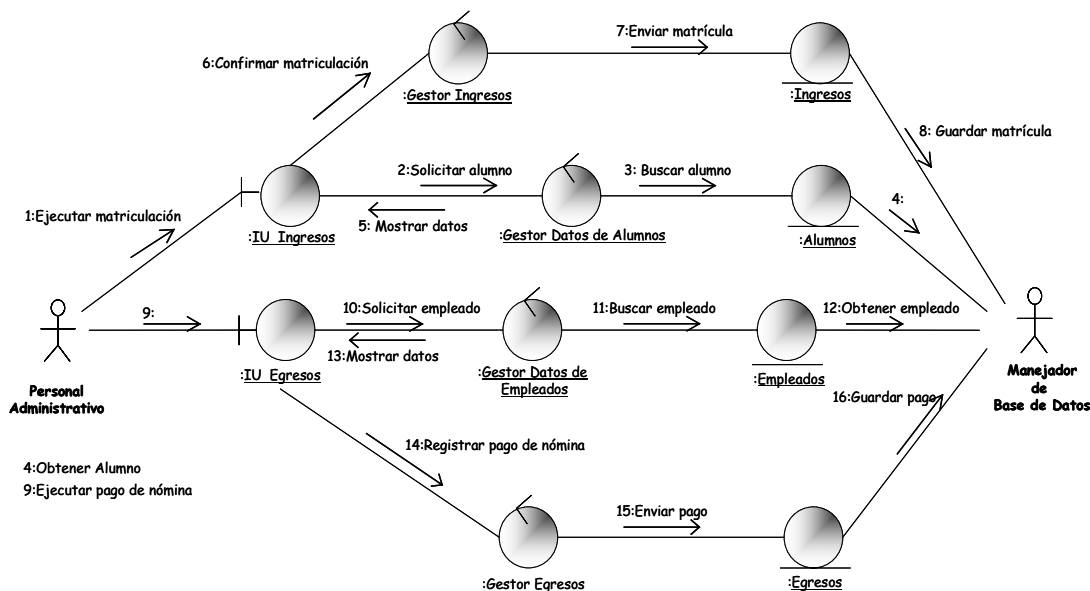


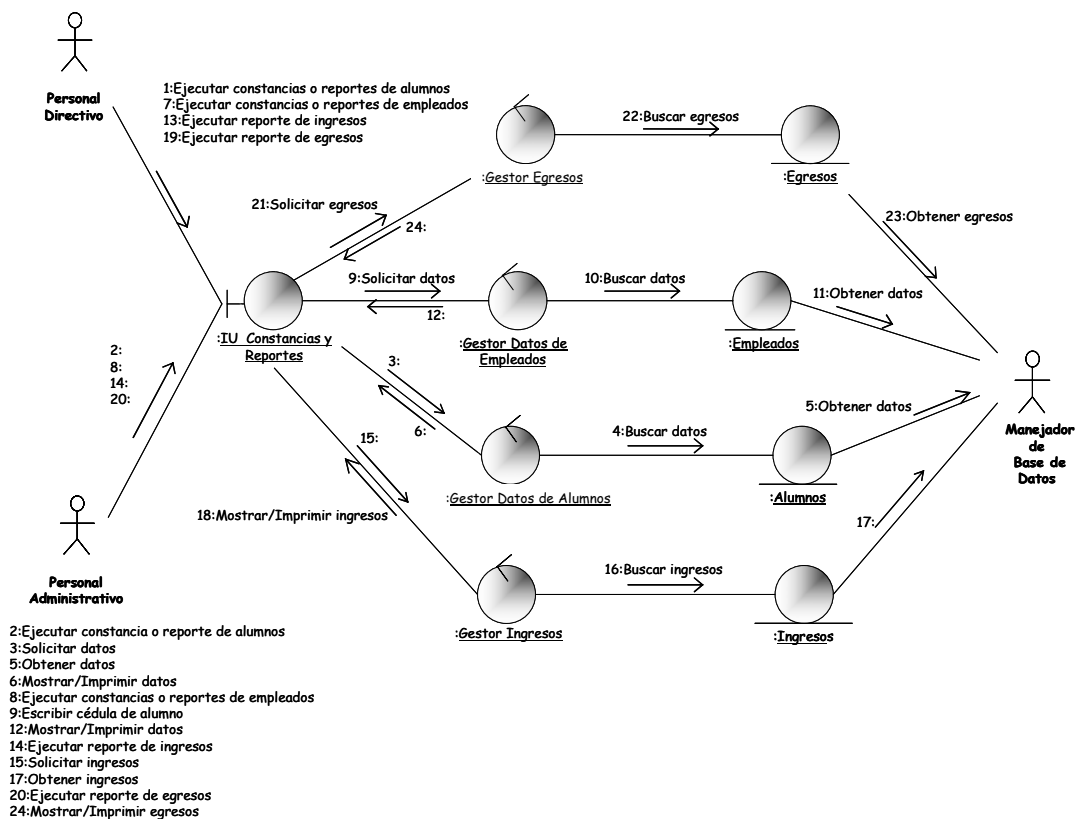
Figura 4.8. Diagrama de Colaboración del Caso de Uso Ingresos y Egresos

#### 4.3.4.3. Diagrama de Colaboración del Caso de Uso Constancias y Reportes

La colaboración del Caso de Uso Constancias y Reportes presenta la elaboración de las actividades relacionadas a la muestra/impresión de documentos personales de alumnos y empleados, así como documentos del plantel.

La Figura 4.9 muestra que el actor Personal Administrativo y el actor Personal Directivo ejecutan el caso de uso Constancias y Reportes por medio del objeto IU Constancias y Reportes. Para obtener constancias (Constancia de Estudio, Constancia de Buena Conducta) y reportes (Alumnos por Sexo, Alumnos por Grado, Alumnos por Año Escolar, Boletín Informativo) el usuario solicita el documento o reporte (enlace 1 y enlace 2) facilitando la información necesaria para la búsqueda de datos (enlace 3), luego el Gestor Datos de Alumnos solicita a la base de datos la información requeridos (enlace 4) y los obtiene por medio del objeto Alumnos (enlace 5) para mostrarlos en pantalla (enlace 6) o en su defecto por medio de una impresión si el usuario así lo requiere.





**Figura 4.9. Diagrama de Colaboración del Caso de Uso Constancias y Reportes.**

Si el usuario requiere reportes de empleados (Empleados del Plantel), ejecuta el proceso indicado (enlace 7 y enlace 8), el objeto Gestor Datos de Empleado solicita a la BDD la información (enlace 9) y por medio de una consulta (enlace 10) busca los datos demandados con la ayuda del objeto Egresos, obteniéndolos (enlace 11) para luego ser mostrados en pantalla o en su defecto por medio de una impresión si el usuario así lo solicita (enlace 12).

Si el usuario requiere reporte de ingresos (Ingresos Diarios, Ingresos Mensuales, Ingresos Anuales) ejecuta el proceso Reporte de Ingresos (enlace 13 y enlace 14), el objeto Gestor Ingresos solicita la información a la base de datos (enlace 15) y busca los datos demandados (enlace 16) al consultar la base de datos (enlace 17) por medio del objeto Ingresos para luego ser mostrados en pantalla a través de la

interfaz o en su defecto por medio de una impresión si el usuario así lo solicita (enlace 18).

Si el usuario requiere reportes de egresos (Egresos Diarios, Egresos Mensuales, Egresos Anuales) ejecuta el proceso deseado por medio de la interfaz (enlace 19) y enlace 20), el Gestor Egresos solicita a la BDD la información (enlace 21) para la búsqueda (enlace 22) de datos con la ayuda de Egresos y por medio de una consulta la obtiene (enlace 23) para luego ser mostrada en la interfaz o por medio de impresión si el usuario lo desea (enlace 24).

#### 4.3.4.4. Diagrama de Colaboración del Caso de Uso Calificaciones

La colaboración del Caso de Uso Calificaciones presenta la elaboración de las actividades relacionadas al ingreso de las calificaciones de los alumnos del plantel al sistema.

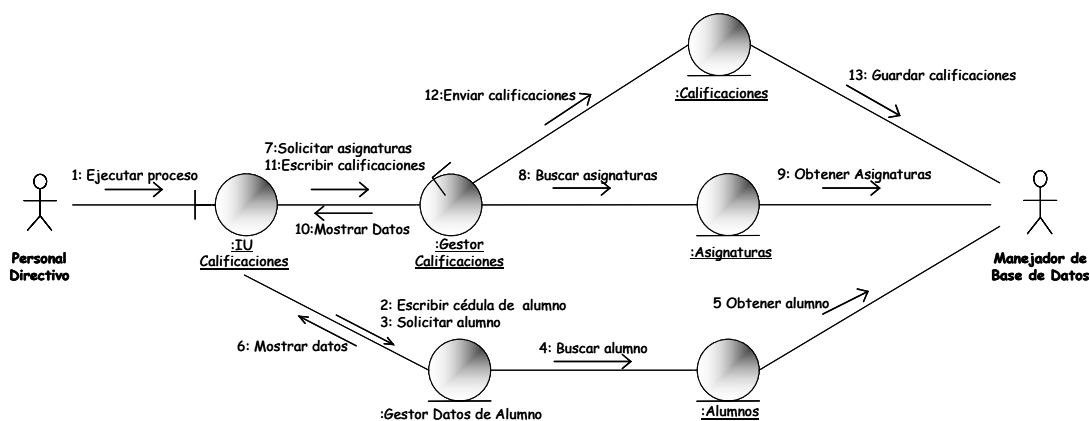


Figura 4.10. Diagrama de Colaboración del Caso de Uso Calificaciones.

En la figura 4.10, se muestra que el actor Personal Directivo ejecuta el caso de uso Calificaciones por medio del objeto IU Calificaciones (enlace 1). El usuario facilita la cédula del alumno para el registro de sus calificaciones (enlace 2), el objeto Gestor Datos de Alumno solicita los datos del alumno (enlace 3) quien por medio de

una consulta (enlace 4) busca y obtiene los datos (enlace 5) a través del objeto Alumnos para luego ser mostrados en la interfaz de usuario (enlace 6), inmediatamente el Gestor Calificaciones solicita a la base de datos las asignaturas dictadas en el plantel (enlace 7) quien por medio de una consulta a la BDD (enlace 8) obtiene los datos (enlace 9) para mostrar la información en la interfaz (enlace 10), paso seguido el usuario escribe las calificaciones (enlace 11) obtenidas por el alumno las cuales envía a la base de datos (enlace 12).

Finalmente por medio del objeto Calificaciones se almacenan las calificaciones (enlace 13) en la BDD del sistema.

#### 4.3.4.5. Diagrama de Colaboración del Caso de Uso Mostrar Ayuda

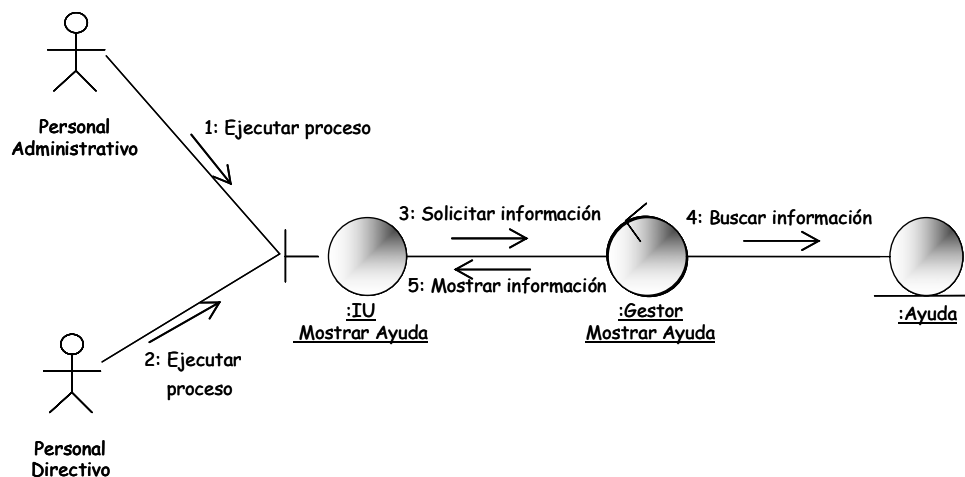


Figura 4.11. Diagrama de Colaboración del Caso de Uso Mostrar Ayuda.

La colaboración del Caso de Uso Mostrar Ayuda presenta la elaboración de las actividades relacionadas a la consulta de la ayuda sobre el manejo del sistema SAPSEE por parte de los usuarios. El actor Personal Administrativo y el actor Personal Directivo ejecutan el caso de uso Mostrar Ayuda por medio del objeto IU Mostrar Ayuda (enlace 1 y 2). Cualquiera de los usuarios solicita la ayuda del sistema

(enlace 3), el objeto Gestor Mostrar Ayuda busca la información (enlace 4) y por medio del objeto Ayuda genera los datos (enlace 5) (Figura 4.11).

#### 4.3.4.6. Diagrama de Colaboración del Caso de Uso Configurar Sistema

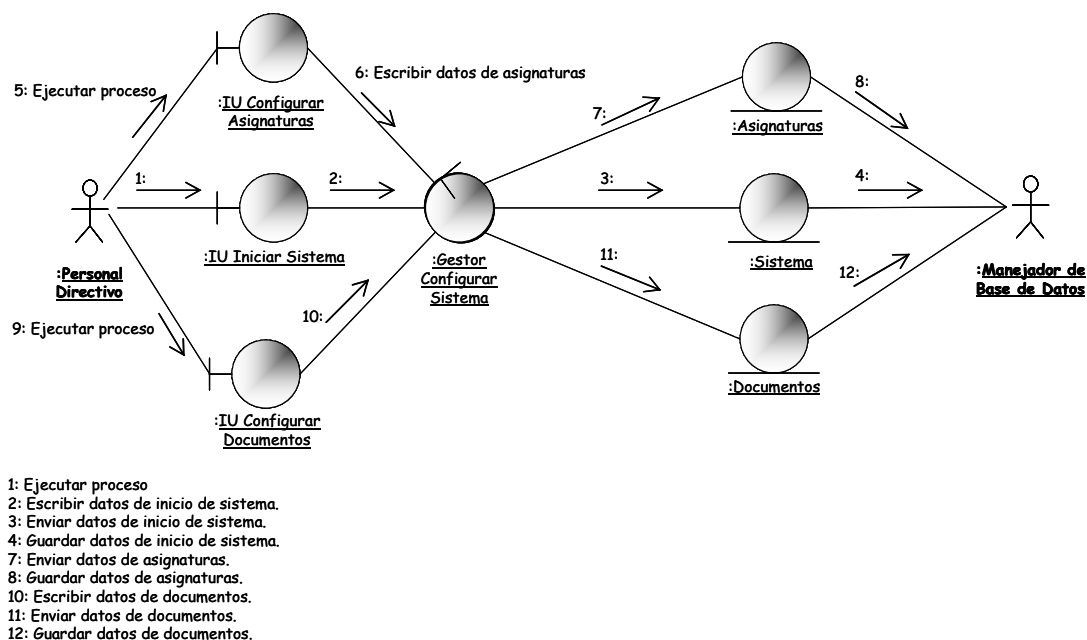


Figura 4.12. Diagrama de Colaboración del Caso de Uso Configurar Sistema.

La colaboración del Caso de Uso Configurar Sistema presenta la elaboración de las actividades relacionadas a la configuración del sistema por medio del ingreso de datos de inicio del sistema. La Figura 4.12 muestra que el actor Personal Directivo ejecuta el caso de uso Configurar Sistema por medio de la IU Configurar Sistema (enlace 1), IU Configurar Asignaturas (enlace 5) y la IU Configurar Documentos (enlace 9).

El usuario facilita los datos de inicio del sistema (enlace 2), las asignaturas dictadas en la institución (enlace 6) y los documentos emitidos por el plantel (enlace 10). El objeto Gestor Configurar Sistema se encarga de enviar los datos de inicio:

Nombre del Plantel, Código DEA, (enlace 3), las asignaturas dictadas en la institución (enlace 6) y los documentos emitidos por el plantel (enlace 11) y almacenarlos dentro del sistema (enlace 4, 8 y 12) con la ayuda de los objetos Sistema, Asignaturas y Documentos.

#### **4.4. Diseño**

Durante el diseño, el resultado del análisis es expandido a una solución técnica.

Se agregan nuevas clases que proveen la infraestructura del sistema: interfaces de usuario, manejo de bases de datos para almacenar objetos, comunicaciones con otros sistemas, etc.

El diseño resulta en especificaciones detalladas para la siguiente fase (construcción). Se modelará el sistema y se dará forma a fin de que soporte todos los requisitos, tomando como punto de partida el análisis que previamente fue realizado.

##### **4.4.1. Subsistemas de Diseño**

Durante esta fase se ha identificado un nuevo subsistema de diseño: Gestores, el cual prestará servicios a los subsistemas de diseño previamente identificados en la fase de inicio, según la Figura 3.11.

El subsistema de servicio Gestores contiene todos los gestores que son usados por la aplicación (Figura 4.13).

Cada gestor es un subsistema de diseño y en total se identificaron 7 gestores para esta fase del desarrollo.

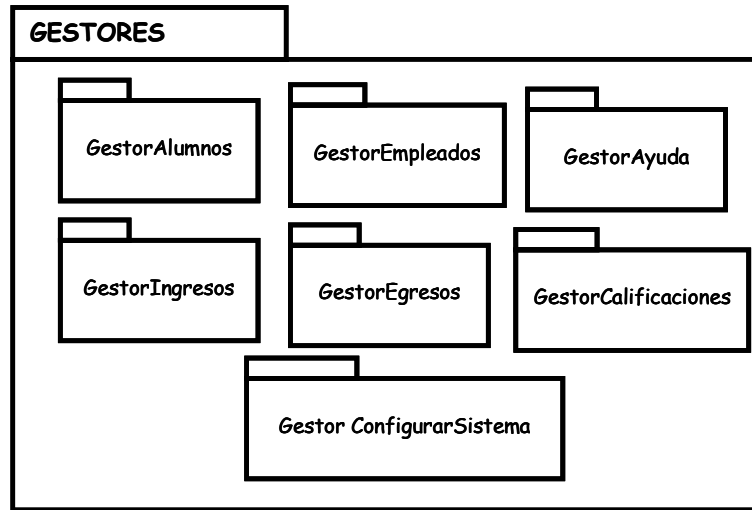


Figura 4.13. Subsistema de Servicio Gestores

#### 4.4.2. Dependencias y Capas del Diseño

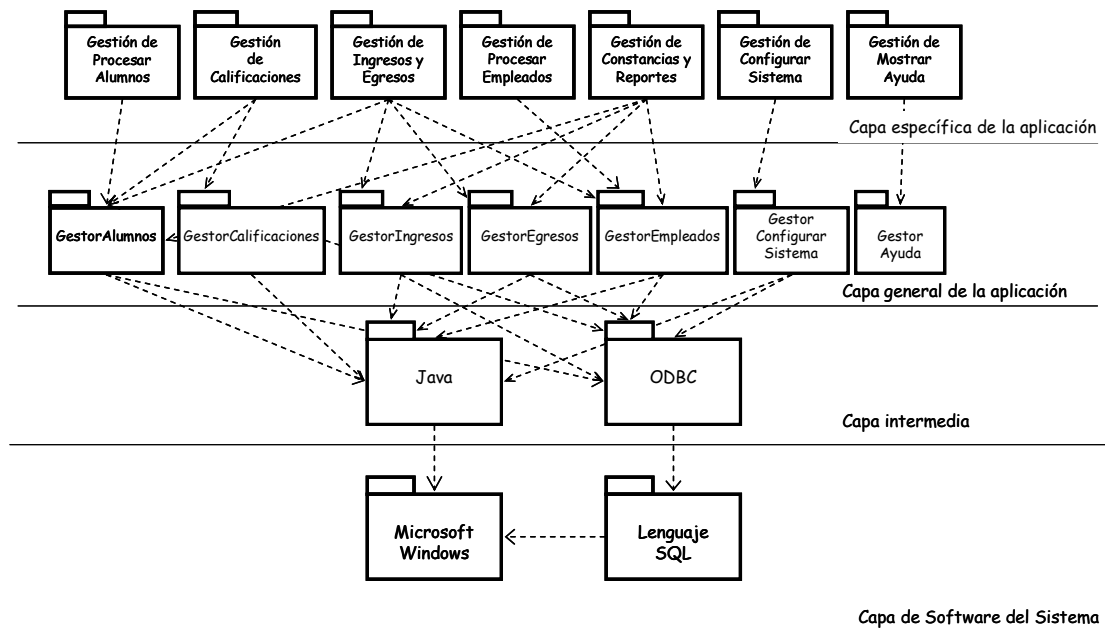


Figura 4.14. Dependencias y Capas de Diseño.

La Figura 4.14, muestra el diseño de la arquitectura en capas desde el punto de vista de los subsistemas que lo conforman.

Está dividida en cuatro capas en las que se distribuye la funcionalidad del sistema junto con otros requisitos como sistemas y marcos de trabajos necesarios para su desarrollo.

En la capa específica de la aplicación se encuentran los subsistemas de gestión, los cuales se obtuvieron durante el análisis en los paquetes de análisis, que contienen la realización de los casos de uso del sistema.

En la capa general de la aplicación, se encuentran todas las clases que son usadas por los procesos relacionados con los subsistemas de gestión; las cuales ejecutan todas las operaciones establecidas en la capa específica de la aplicación. Finalmente se encuentra la capa del software del sistema en el cual está el sistema operativo sobre el cual corre la aplicación (Microsoft Windows).

En la capa intermedia encontramos dos subsistemas, Java con la biblioteca de controles para la manipulación de bases de datos (ODBC) y para ofrecer un ambiente visual y agradable al manejar, conformando los bloques de construcción de la aplicación, específicamente del gestor de la capa anterior.

#### **4.4.3. Diagrama de Clases del Diseño**

Permiten mostrar lo que el sistema puede hacer, además, de cómo puede ser construido. En esta fase se desarrollarán los restantes casos de uso.

Para las realizaciones de casos de uso se emplearán diagramas de clases y de interacción para especificar la comunicación entre los objetos del diseño. Se observará de manera más clara el funcionamiento del sistema ya que los diagramas de secuencia hacen énfasis en la ordenación temporal de los mensajes entre los objetos del diseño.

#### 4.4.3.1. Realización del Caso de Uso Procesar Empleados



**Figura 4.15. Diagrama de Clases del Diseño del Caso de Uso Procesar Empleados**

La Figura 4.15 muestra las clases del diseño para el caso de uso Procesar Empleados. La clase `ProcesarEmpleados` es la interfaz de usuario, la cual permite el ingreso, consulta, eliminación y actualización de los datos de los empleados. La clase `GestorEmpleados` es el encargado de realizar las operaciones solicitadas por la interfaz. `Empleados`, representa información perdurable en el tiempo que se almacena en el sistema relacionada con los datos de los empleados.

La Figura 4.16, muestra el diagrama de secuencia del caso de uso Procesar Empleados, donde el Personal Administrativo lo ejecuta para ingresar los datos de los empleados por medio de la interfaz `ProcesarEmpleados`. Al conformar todos los datos, la interfaz llama al método `GuardarEmpleado()` de la clase `GestorEmpleados`, quien va a la base de datos para almacenar la información indicada en la interfaz.

Para consultar los datos del empleado, el Personal Administrativo ingresa las cédulas de los empleados por medio de la interfaz para activar el método `BuscarEmpleado()` de la clase `GestorEmpleados` y así obtenerlos por medio de una consulta a la BDD del sistema para que la interfaz finalmente los muestre.

Para retirar a los empleados, el Personal Administrativo ingresa por medio de la interfaz las cédulas, se activa el método `BuscarEmpleado()` de la clase `GestorEmpleados` para obtener los datos por medio de una consulta a la BDD del sistema, el usuario confirma el retiro y la interfaz activa el método `RetirarEmpleado()` de la clase `GestorEmpleados`, almacenando el proceso en el sistema.



Para modificar los datos de los empleados el Personal Administrativo ingresa las cédulas a través de la interfaz, se activa el método `BuscarEmpleado()` de la clase `GestorEmpleados` para obtener los datos por medio de una consulta a la BDD del sistema, el usuario modifica los datos en pantalla confirma el retiro y la interfaz activa el método `ModificarEmpleado()` de la clase `GestorEmpleados()`, almacenando la información en el sistema.

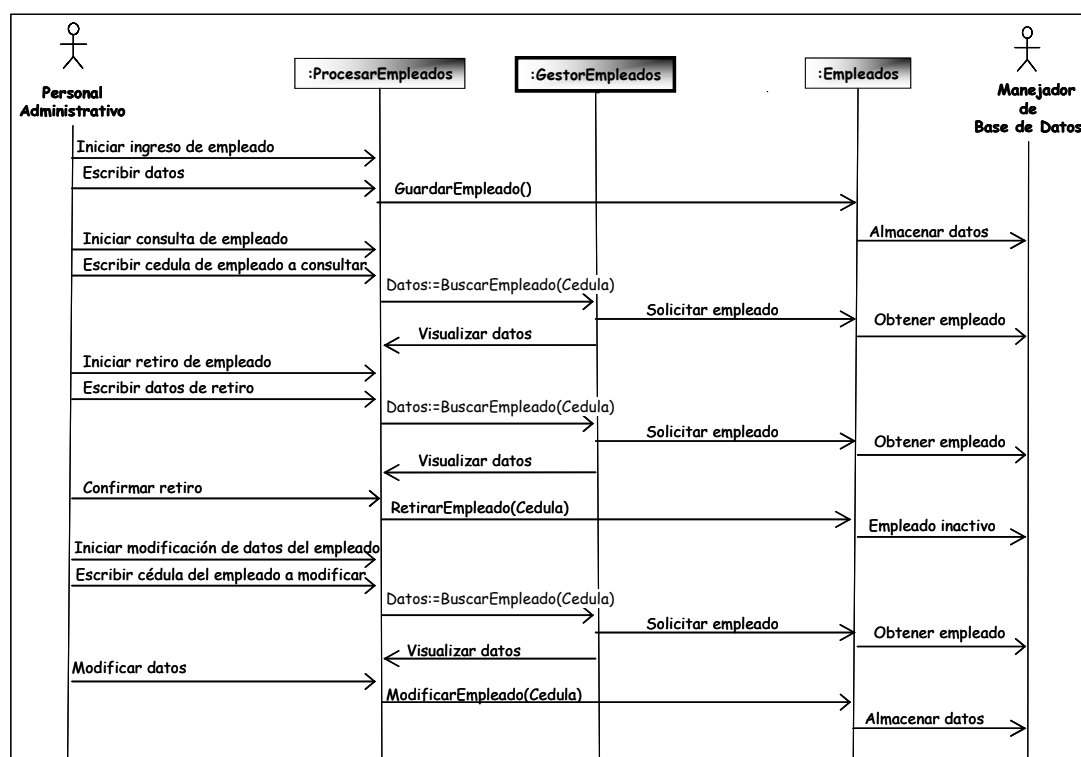
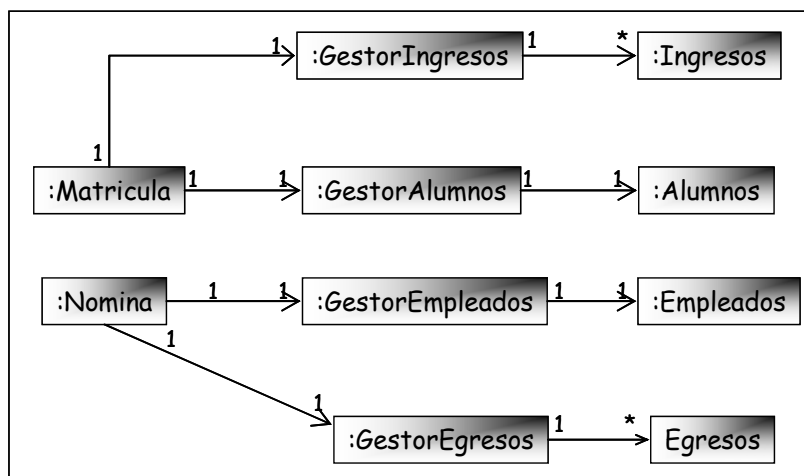


Figura 4.16. Diagrama de Secuencia del Caso de Uso Procesar Empleados

#### 4.4.3.2. Realización del Caso de Uso Ingresos y Egresos.

La Figura 4.17 muestra las clases del diseño para el caso de uso Ingresos y Egresos. Las clases: `Matricula` y `Nomina` son las interfaces de usuario. La clase `Matricula` permite ingresar al sistema los pagos de los alumnos. La clase `Nomina` permite ingresar al sistema los pagos que la institución realiza a los empleados. La

clase GestorAlumnos es la encargada de realizar las solicitudes realizadas por la clase Matricula; en este caso buscar los datos de los alumnos a matricularse. La clase GestorIngresos es la encargada de realizar las solicitudes hechas por la clase Matricula; en este caso procesar los pagos de los alumnos. La clase GestorEmpleados es la encargada de realizar las solicitudes hechas por la clase Nomina, en este caso buscar los datos de los empleados que la institución le cancelará el sueldo. La clase GestorEgresos es la encargada de realizar las solicitudes hechas por la clase Nomina. Ingresos, Alumnos, Empleados, y Egresos representan información perdurable que se almacena en la BDD del sistema para los ingresos y pagos al personal.



**Figura 4.17. Diagrama de Clases del Diseño del Caso de Uso Ingresos y Egresos.**

La Figura 4.18, muestra el diagrama de secuencia del caso de uso Ingresos y Egresos, donde el Personal Administrativo lo ejecuta para realizar algunas operaciones administrativas tanto de ingreso como de egreso de dinero durante el transcurso del año escolar en el plantel. Para ingresar los pagos de matrícula, el usuario usa la interfaz Matricula para ingresar la cédula del alumno y conseguir los datos necesarios en la ejecución del proceso, la clase Matricula activa el método BuscarAlumno() de la clase GestorAlumnos quien va a la BDD y por medio de una consulta obtiene los datos del alumno para que la interfaz los muestre,

inmediatamente el usuario confirma el ingreso del pago al sistema para que la clase Matricula active el método GuardarMatricula() de la clase GestorIngresos. Luego esta clase envía la información a la BDD para que sea almacenada.

Para hacer los pagos a los empleados el usuario facilita la cédula del empleado por medio de la interfaz Nomina y así obtener los datos necesarios para efectuar el pago, al confirmar la búsqueda, la clase Nomina activa al método BuscarEmpleado() de la clase GestorEmpleados quien va a la BDD del sistema y adquiere la información solicitada e inmediatamente la interfaz la muestra en pantalla, cuando el usuario confirma el pago la interfaz activa el método GuardarPago() de la clase GestorEgresos la cual envía los datos a la BDD para que sean almacenados.

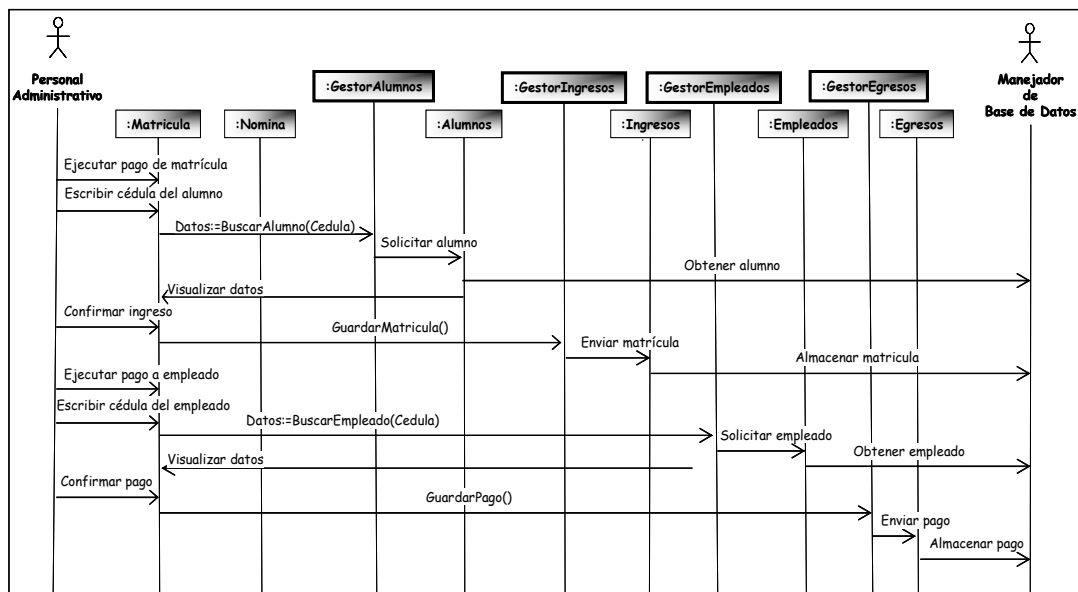


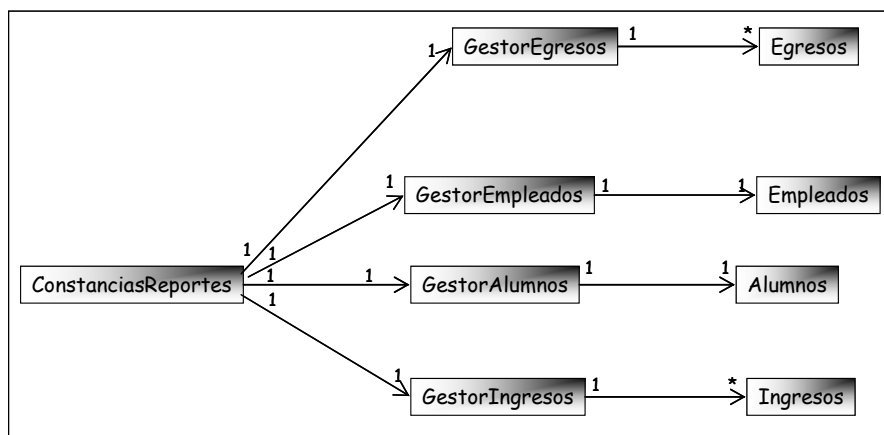
Figura 4.18. Diagrama de Secuencia del Caso de Uso Ingresos y Egresos.

#### 4.4.3.3. Realización del caso Constancias y Reportes

La Figura 4.19 muestra las clases del diseño del caso de uso Constancias y Reportes. La clase ConstanciasReportes es la interfaz de usuario. La clase

GestorAlumnos permite buscar los datos para la muestra en pantalla o la impresión de las constancias y reportes de los alumnos del plantel. La clase GestorEmpleados permite buscar los datos para la muestra en pantalla o la impresión de las constancias y reportes de los empleados del plantel. La clase GestorIngresos permite buscar los datos para la muestra en pantalla o la impresión de los reportes de ingresos del plantel.

La clase GestorEgresos permite buscar los datos de los egresos para la muestra o la impresión en pantalla de los reportes de egresos del plantel. Alumnos, Empleados, Ingresos, y Egresos representan información perdurable en el tiempo para su posterior uso.

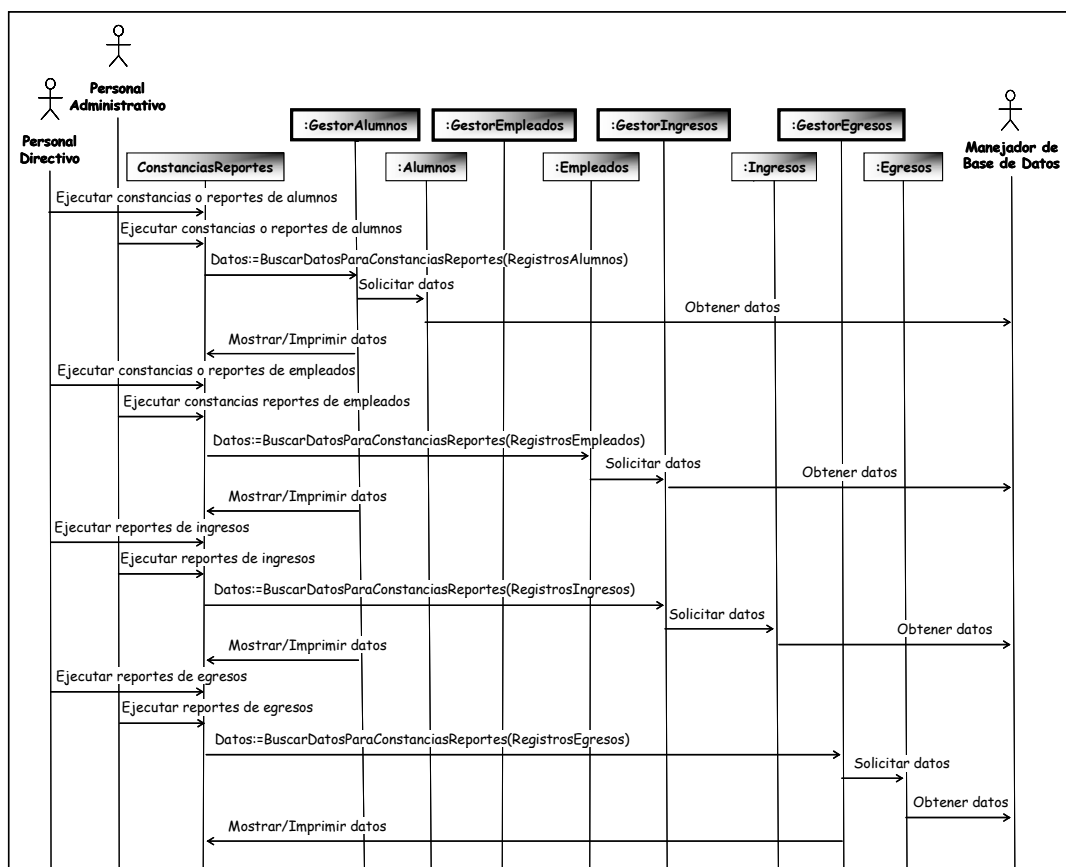


**Figura 4.19. Diagrama de Clases del Diseño del Caso de Uso Constancias y Reportes.**

La Figura 4.20, muestra el diagrama de secuencia del caso de uso Constancias y Reportes donde el Personal Directivo y el Personal Administrativo lo ejecutan para obtener las constancias o reportes de: alumnos, empleados, ingresos o egresos del plantel, a través de la pantalla o por impresiones en papel.

Para generar las constancias o reportes de alumnos el usuario invoca el proceso indicado facilitando los datos necesarios a través de la interfaz, la clase

ConstanciasReportes activa el método `BuscarDatosParaConstanciasReportes()` de la clase `GestorAlumnos` para obtener los datos y mostrarlos en pantalla o por medio de impresión si se requiere.



**Figura 4.20. Diagrama de Secuencia del Caso de Uso Constancias y Reportes.**

Para generar las constancias o reportes de empleados el usuario invoca el proceso indicado facilitando los datos necesarios a través de la interfaz, la clase `ConstanciasReportes` activa el método `BuscarDatosParaConstanciasReportes()` de la clase `GestorEmpleados` para obtener los datos y mostrarlos en pantalla o por medio de impresión si el usuario así lo requiere. Para generar las constancias o reportes de los ingresos de dinero en el plantel, el usuario invoca el proceso indicado facilitando los datos necesarios a través de la interfaz, la clase `ConstanciasReportes` activa el método

BuscarDatosParaConstanciasReportes() de la clase GestorIngresos para obtener los datos y mostrarlos en pantalla o por medio de impresión si el usuario así lo requiere. Para generar las constancias o reportes de los egresos de dinero en el plantel, el usuario invoca el proceso indicado facilitando los datos necesarios a través de la interfaz, la clase ConstanciasReportes activa el método BuscarDatosParaConstanciasReportes() de la clase GestorEgresos para obtener los datos y mostrarlos en pantalla o por medio de impresión si el usuario así lo requiere.

#### 4.4.3.4. Realización del Caso de Uso Calificaciones

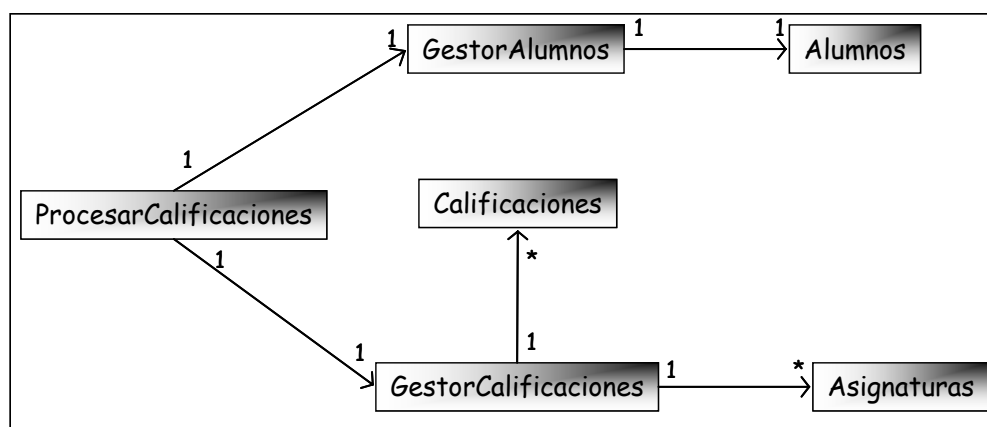
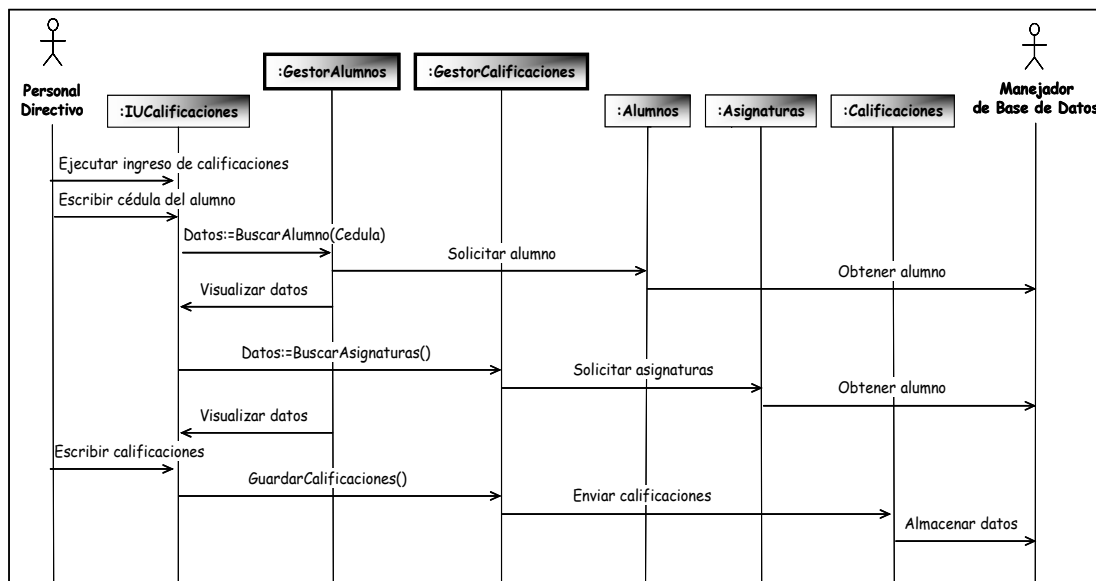


Figura 4.21. Diagrama de Clases del Diseño del Caso de Uso Calificaciones

La Figura 4.21 muestra las clases del diseño del caso de uso Calificaciones.

La clase **ProcesarCalificaciones** es la interfaz de usuario. La clase **GestorAlumnos** es la encargada de buscar los datos necesarios para el procesamiento de las calificaciones de los alumnos del plantel. La clase **GestorCalificaciones** se encarga de buscar las asignaturas dictadas en la institución para los alumnos y también de enviar todas las calificaciones a la BDD para que sean almacenadas; **Alumnos**, **Calificaciones** y **Asignaturas** representan información perdurable en el tiempo.

La Figura 4.22 muestra el diagrama de secuencia del caso de uso Calificaciones donde el Personal Directivo por medio de la interfaz facilita la cédula del alumno para procesar las calificaciones obtenidas durante el año escolar. La clase ProcesarCalificaciones activa el método BuscarAlumno() de la clase GestorAlumnos, quien obtiene los datos y los muestra en pantalla. La clase ProcesarCalificaciones activa el método BuscarAsignaturas() de la clase GestorAsignaturas solicitando los datos a la BDD del sistema y obteniéndolos por medio de una consulta para mostrarlos en pantalla, el usuario ingresa las calificaciones por medio de la interfaz confirmando su almacenamiento en el sistema, para lo cual la clase ProcesarCalificaciones activa el método GuardarCalificaciones() enviando los datos a la BDD para que sean almacenados para su posterior uso.

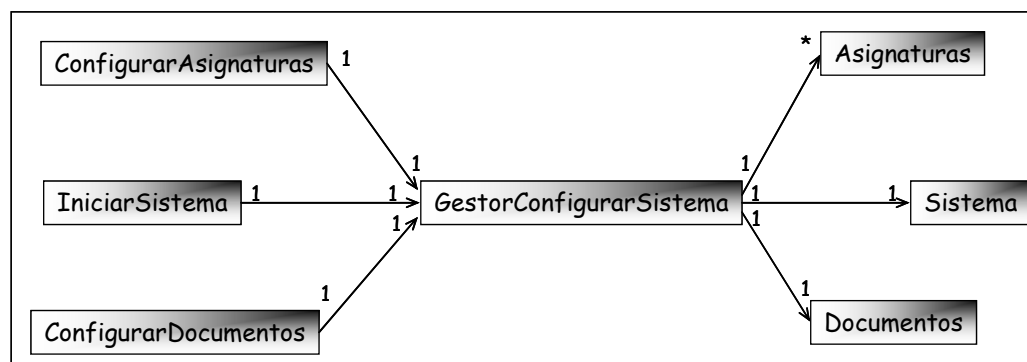


**Figura 4.22. Diagrama de Secuencia del Caso de Uso Constancias y Reportes.**

#### 4.4.3.5. Realización del Caso de Uso Configurar Sistema

La Figura 4.23, muestra las clases del diseño del caso de uso Configurar Sistema. Las clases: IniciarSistema, ConfigurarAsignaturas y ConfigurarDocumentos

son las interfaces de usuarios. La clase `GestorConfigurarSistema` es la encargada de realizar las solicitudes de las interfaces; `Asignaturas`, `Sistema` y `Documentos` representan información perdurable en el tiempo para su uso posterior.



**Figura 4.23. Diagrama de Clases del Diseño del Caso de Uso Configurar Sistema.**

La Figura 4.24 muestra el diagrama de secuencia del caso de uso Configurar Sistema donde el Personal Directivo lo ejecuta para configurar los datos de inicio del sistema (nombre del plantel, RIF,...), las asignaturas dictadas en la institución y los documentos que se emiten en la misma. Para configurar los datos de inicio el usuario ejecuta la interfaz relacionada a este proceso por la clase `IniciarSistema`, cuando todos los datos estén conformados la clase llama al método `GuardarDatosDeInicioDelSistema()` de la clase `GestorConfigurarSistema` quien envía la información a la BDD del sistema para su posterior almacenamiento. Para configurar las asignaturas dictadas en la institución el usuario emplea la interfaz relacionada al proceso, activando la clase `ConfigurarAsignaturas` para ingresar los datos solicitados por la interfaz, una vez confirmados los datos, la clase `ConfigurarAsignaturas` llama al método `GuardarAsignaturas()` de la clase `GestorConfigurarSistema` quien envía la información a la BDD del sistema para su posterior almacenamiento. Para configurar los documentos emitidos por el plantel el usuario emplea la interfaz relacionada al proceso por medio de la clase `ConfigurarDocumentos` para ingresar los datos solicitados en pantalla, una vez



confirmados los datos, la clase ConfigurarDocumentos llama al método GuardarDocumentos() de la clase GestorConfigurarSistema quien envía la información a la base de datos para su posterior almacenamiento.

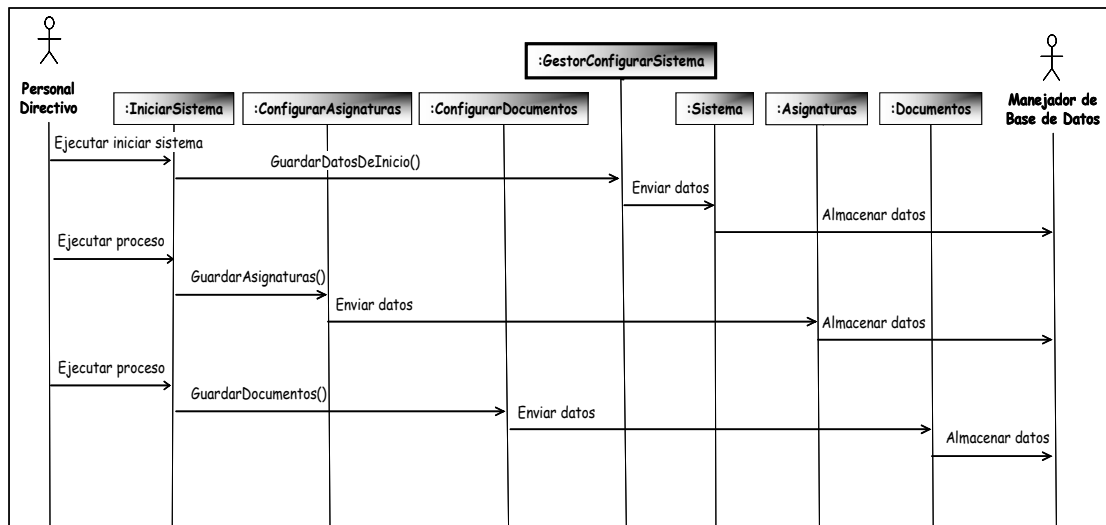


Figura 4.24. Diagrama de Secuencia del Caso de Uso Configurar Sistema.

#### 4.4.3.6. Realización del Caso de Uso Mostrar Ayuda

La Figura 4.25 muestra las clases del diseño del caso de uso Mostrar Ayuda.

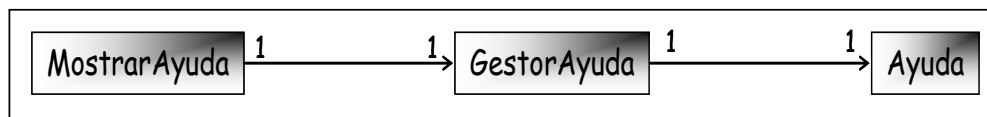


Figura 4.25. Diagrama de Clases del Diseño del Caso de Uso Mostrar Ayuda

La clase MostrarAyuda es la interfaz de usuario.

La clase GestorAyuda es la encargada de realizar las solicitudes hechas por la clase MostrarAyuda y Ayuda representa la información solicitada por la clase GestorAyuda.

La Figura 4.26 muestra el diagrama de secuencia del caso de uso Mostrar Ayuda donde El Personal Directivo y el Personal Administrativo lo ejecutan para obtener la ayuda del Sistema SAPSEE, la clase GestorAyuda se encarga de buscar la información llamando el método BuscarAyuda() de la clase GestorAyuda para luego presentarla en pantalla por medio de la clase Ayuda para el usuario que la requiera.

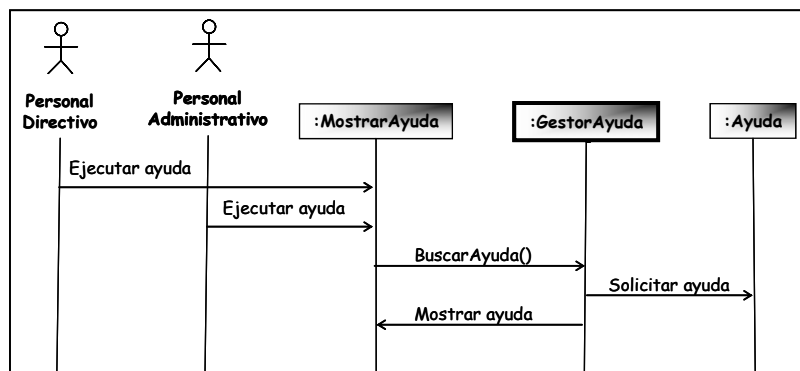


Figura 4.26. Diagrama de Secuencia del Caso de Uso Mostrar Ayuda.

#### 4.4.4. Diseño de la Base de Datos

El Proceso Unificado de Desarrollo de Software no contempla el diseño e implementación de la base de datos de manera explícita.

Por esta razón ha sido agregada esta sección en la que se describe el diseño de la base de datos que almacenará la información persistente del sistema.

Se ha tomado el conjunto de clases de entidad estudiadas en el análisis como punto de partida para la identificación de los componentes de la base de datos.

El diseño de la misma se realizará basado en el modelo relacional ya que suele dar un rendimiento para datos tabulares cuya representación lógica es más sencilla, además de ofrecer una tecnología robusta.

A continuación se describe su comportamiento:

#### 4.4.4.1. Descripción de Tablas

Tabla Alumnos: Almacena los datos personales y escolares de los alumnos durante el proceso de inscripción de alumnos nuevos y regulares; los cuales serán referenciados constantemente por otras tablas de la base de datos del sistema SAPSEE.

El esquema de esta tabla es el siguiente:

1. Cedula: varchar(11) → Clave Principal.
2. Nacionalidad: char (1).
3. Sexo: char (1).
4. Apellidos: varchar (MAX).
5. Nombres: varchar (MAX).
6. Direccion: varchar (MAX).
7. Telefono: varchar (50).
8. FechaNacimiento: datetime.
9. CiudadNacimiento: varchar(MAX).
10. EstadoNacimiento: varchar(MAX).
11. PaisNacimiento: varchar(max).
12. Etapa: varchar(7).
13. Escolaridad: varchar (10).
14. Grado: char (1).
15. Seccion: char (1).
16. AñoEscolar: varchar (9).
17. FechaInscripcion: datetime.
18. Retirado: bit.

19. MotivoRetiro: varchar (11).

20. FechaRetiro: datetime.

Tabla Empleados: Almacena toda la información académica y administrativa de los empleados del plantel.

El esquema es el siguiente:

1. Cedula: varchar (11) → Clave Principal.
2. Nacionalidad: char (1).
3. Sexo: char (1).
4. Apellidos: varchar (max).
5. Nombres: varchar (max).
6. FechaNacimiento: datetime.
7. Profesion: varchar (max).
8. EstadoCivil: varchar (13).
9. Cargo: varchar (26).
10. Direccion: varchar (max).
11. Telefono: varchar (50).
12. FechaIngreso: datetime.
13. AñoEscolar: varchar (9).
14. Retirado: bit
15. MotivoRetiro: varchar (23).
16. FechaRetiro: datetime.

Tabla Docentes: Almacena: cédula, grado, sección y año escolar del empleado que en el campo Cargo, de la tabla Empleados esté registrado como Docente de Aula.

El esquema es el siguiente:

1. idDocente: int → Clave Principal.
2. Cedula: varchar (11).
3. Grado: char (1).
4. Seccion: char (1).
5. AñoEscolar: varchar (9).

El campo Cedula hace referencia al campo Cedula de la tabla Empleados.

Tabla HistorialDeAlumnos: Almacena los datos relacionados al transcurso del alumno por la institución, para ello se relaciona con la tabla Docentes, la cual contiene el Grado, Sección y Año Escolar. El esquema de esta tabla es el siguiente:

1. Cedula: varchar (11) → Clave Principal.
2. idDocente: int → Clave Principal.
3. Escolaridad: varchar (10).

El campo Cedula hace referencia al campo Cedula de la tabla Alumnos. El campo idDocente hace referencia al campo idDocente de la tabla Docentes.

Tabla PadresRepresentantes: Almacena la información de los padres y representantes de los alumnos inscritos en la institución.

El esquema de esta tabla es el siguiente:

1. CedulaAlumno: varchar (11) → Clave Principal.
2. CedulaRepresentante: varchar (11) → Clave Principal.
3. Nacionalidad: char (1).
4. Apellidos: varchar (MAX).
5. Nombres: varchar (MAX).

6. Direccion: varchar (MAX).
7. Telefono: varchar (50).
8. Parentesco: varchar (5).
9. RepresentanteEconómico: bit.

El campo CedulaAlumno referencia al campo Cedula de la tabla Alumnos.

Tabla Asignaturas: Almacena el código y el nombre de las asignaturas dictadas en la institución, el código es generado por el mismo sistema cada vez que se genera una nueva asignatura.

El esquema es el siguiente:

1. idAsignatura: int → Clave Principal.
2. Nombre: varchar (MAX)

Tabla Calificaciones: Almacena toda la información relacionada a las calificaciones obtenidas por los alumnos durante el año escolar.

El esquema de esta tabla es el siguiente:

1. Cedula: varchar (11) → Clave Principal.
2. idAsignatura: int → Clave Principal: int.
3. AñoEscolar: varchar (9) → Clave Principal.
4. Lapso: char (3) → Clave Principal.
5. Calificacion: char (1).

El campo Cedula hace referencia al campo Cedula en la tabla Alumnos y el campo idAsignatura hace referencia al campo idAsignatura de la tabla Asignaturas.

Tabla Ingresos: Almacena los ingresos (matricula, constancias) recibos en la institución. El esquema es el siguiente:

1. NroRecibo: varchar (50) → Clave Principal.
2. Cedula: varchar(11).
3. FechaPago: datetime.
4. Concepto: varchar (50).
5. Monto: money.
6. TipoPago: varchar (8).
7. NroCheque: varchar (12).
8. NroCuenta: varchar (20).
9. Banco: varchar (100).

El campo Cedula hace referencia al campo Cedula de la tabla Alumnos.

Tabla IngresosConstancias: Almacena los ingresos por concepto de los pagos realizados a razón de la emisión de constancias por parte del plantel a los alumnos.

El esquema es el siguiente:

1. NroRecibo: varchar (50) → Clave Principal.
2. idDocumento: int.

El campo NroRecibo referencia al campo NroRecibo de la tabla Ingresos.

El campo idDocumento hace referencia al campo idDocumento de la tabla Documentos.

Tabla Egresos: Almacena los egresos (Nómina) emitidos por la institución.

El esquema es el siguiente:

1. NroRecibo: varchar (50) → Clave principal.
2. Concepto: varchar (100).
3. FechaPago. Datetime.
4. Monto: money.
5. NroCheque: varchar (12).
6. NroCuenta: varchar (20).
7. Banco: varchar (100).

Tabla Sistema: Almacena los datos de identificación del plantel como Nombres, RIF, NIT, Dirección.

El esquema es el siguiente

1. CodDEA: varchar(11) → Clave principal.
2. Nombre: varchar(MAX).
3. RIF: varchar(12).
4. NIT: varchar(11).
5. Direccion: varchar(MAX).
6. Teléfono: varchar(50).

Tabla Caja: Almacena la cantidad de dinero disponible por concepto de ingresos y egresos en el plantel.

El esquema es el siguiente:

1. idCaja: int → Clave principal.
2. DineroDisponible: money.



## **4.5. Implementación**

En este flujo de trabajo se toma el resultado del diseño y se implementa en términos de componentes, es decir, ficheros de código fuente, ejecutables.

A continuación se implementarán un 10% de los casos de usos definidos en la fase de inicio.

### **4.5.1. Implementación del Caso de Uso Procesar Alumnos.**

Este caso de uso se implementó a través de la clase de diseño ProcesarAlumnos, que es una clase derivada de la clase Dialog de Java y es ejecutada por medio de la clase ProcesarAlumnos.java que permite el ingreso, consulta, modificación y retiro de los alumnos dentro del sistema.

Estas operaciones permiten manejar los datos que serán usados para generar constancias, reportes, ingreso de pago de mensualidades, ingreso de padres y representantes. Para poder llevar a cabo todas las operaciones de forma correcta es necesario que el usuario tenga conocimiento acerca de los datos a ingresar en el Sistema, ya que estos representan información necesaria para evitar conflictos posteriores.

Después que los datos del alumno estén ingresados en el sistema, es necesario que el usuario ejecute algunas actividades para completar el proceso de inscripción como: ingresar los datos de los padres y representantes e ingresar los pagos relacionados con dicha inscripción. Esto hará que el proceso de inscripción sea completo y no se carezca de información necesaria para realizar otros procesos.

La pantalla relacionada con el ingreso de alumnos es la siguiente:

**Inscripción - Nuevos Ingresos Año Escolar 2008-2009** ✕

**Datos Personales** Fecha Inscripción: **2/11/2008**

Cédula Escolar:   Sexo:  Apellidos:

Dirección:  Nombres:

Teléfono:






**Datos de Nacimiento**

Fecha de Nacimiento:    Estado:

País:  Ciudad:

**Datos Escolares**

Etapas:  Escolaridad:  Grado:  Sección:

**Figura 4.27. Pantalla de Ingresar Alumno**

La Figura 4.28 muestra la pantalla relacionada con el ingreso de los datos de los Padres y Representantes, se activa cuando el usuario presiona el segundo icono ubicado de izquierda a derecha.

La Figura 4.29 muestra la pantalla relacionada al ingreso de los pagos por concepto de inscripción, se activa cuando el usuario presiona el tercer icono ubicado de izquierda a derecha. La Figura 4.30 muestra la pantalla relacionada con la consulta de los datos de los alumnos; se activa cuando el usuario solicita esta actividad en la interfaz principal. La Figura 4.31 muestra la pantalla relacionada con el retiro de los alumnos del plantel, se activa cuando el usuario solicita esta actividad en la interfaz principal. La Figura 4.32 muestra la pantalla relacionada con la modificación de los datos de los alumnos, se activa cuando el usuario solicita esta actividad en la interfaz principal.

**Ingresar Padres y Representantes** [X]

**Datos de la Madre**  Representante

Cedula: [icon] [ ] Teléfono: [ ]

Apellidos: [ ] Nombres: [ ]

Dirección: [ ]

**Datos del Padre**  Representante

Cedula: [icon] [ ] Teléfono: [ ]

Apellidos: [ ] Nombres: [ ]

Dirección: [ ]

[icon] [icon] [icon]

Figura 4.28. Pantalla de Ingresar datos de Padres y Representantes.

**Pago de Matrícula / Año Escolar 2008-2009** [X]

**Datos Personales** **Recibo Nro: I-6**

Cédula: [ ] Grado: [ ] Sección: [ ]

Apellidos: [ ] Nombres: [ ]

**Datos de Pago** Año Escolar de Pago: 2008-2009 [+ -] Fecha de Pago: 2/11/2008

Monto: [ ] Bs. Tipo de Pago: Efectivo [v]

N° Cheque: [ ] Concepto de Pago: Inscripción [v]

Banco: [ ] N° Cuenta: [ ]

[icon] [icon] [icon] [icon]

Figura 4.29. Pantalla de Ingresar Pagos por concepto de inscripción

La pantalla relacionada con la consulta de los datos de los alumnos es la siguiente:

**Consultar Alumno** [X]

**Datos Personales** Fecha Inscripción:

Cédula Escolar:  Sexo:  Apellidos:

Dirección:  Nombres:

Teléfono:

**Datos de Nacimiento** Fecha de Nacimiento:  Estado:

País:  Ciudad:

**Datos de Retiro** Motivo:

Fecha:

**Datos Escolares** Etapa:  Escolaridad:  Grado:  Sección:




  

Figura 4.30. Pantalla de Consultar Alumno

La pantalla relacionada con el retiro de alumnos del plantel es la siguiente:

**Retirar Alumno** [X]

**Datos Personales** Fecha Inscripción:

Cédula Escolar:  Sexo:  Apellidos:

Dirección:  Nombres:

Teléfono:

**Datos de Nacimiento** Fecha de Nacimiento:  Estado:

País:  Ciudad:

**Datos de Retiro** Motivo:

Fecha:

**Datos Escolares** Etapa:  Escolaridad:  Grado:  Sección:





   

Figura 4.31. Pantalla de Retirar Alumno

La pantalla relacionada con la modificación de los datos del alumno es la siguiente:

Figura 4.32. Pantalla de Modificar Alumno

El código fuente de esta clase es el siguiente:

```

package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.lang.*;
import javax.swing.border.*;
import java.io.*;
import Gestores.*;
public class ProcesarAlumnos extends JDialog {
    JPanel panel1 = new JPanel();
    private GestorAlumnos Alumno = new Gestores.GestorAlumnos();
    private GestorIngresos Ingresos = new Gestores.GestorIngresos();
    private String NACIONALIDAD[] = {"V","E"};

```

```
private String SEXO[] = {"F", "M"};
private String ETAPA[] = {"Primera", "Segunda"};
private String ESCOLARIDAD[] = {"Regular", "Repitiente"};
private String MOTIVODERETIRO[] = {"Culminación", "Voluntario",
"Expulsado"};
private String MESES[] = {"", "enero", "febrero", "marzo", "abril", "mayo", "junio",
    "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"};
private String AñoEscolarInscripcion = "";
private String SentenciaSQL = "";
private String Interfaz = "";
private ResultSet Datos;
private ImageIcon Guardar;
private ImageIcon Eliminar;
private ImageIcon Modificar;
private ImageIcon Salir;
private ImageIcon Actualizar;
private ImageIcon Padres;
private ImageIcon Buscar;
private ImageIcon Correcto;
private ImageIcon Pregunta;
private ImageIcon Matricula;
private JLabel DatosPersonales = new JLabel();
private JLabel cedula = new JLabel();
private JComboBox Nacionalidad = new JComboBox(NACIONALIDAD);
private JLabel sexo = new JLabel();
private JComboBox Sexo = new JComboBox(SEXO);
private JLabel apellidos = new JLabel();
private JLabel nombres = new JLabel();
private JLabel direccion = new JLabel();
```

```
private JLabel escolar = new JLabel();
private JLabel telefono = new JLabel();
private JTextField Telefono = new JTextField();
private JLabel DatosEscolares = new JLabel();
private JLabel grado = new JLabel();
private JLabel seccion = new JLabel();
private JLabel etapa = new JLabel();
private JComboBox Etapa = new JComboBox(ETAPA);
private JComboBox Grado = new JComboBox();
private JLabel escolaridad = new JLabel();
private JComboBox Escolaridad = new JComboBox(ESCOLARIDAD);
private JButton Fondo3 = new JButton();
private JLabel DatosDeNacimiento = new JLabel();
private JLabel estadonacimiento = new JLabel();
private JLabel ciudadnacimiento = new JLabel();
private JLabel paisnacimiento = new JLabel();
private JComboBox Dia = new JComboBox();
private JComboBox Mes = new JComboBox();
private JComboBox Año = new JComboBox();
private JButton Fondo2 = new JButton();
private JButton GuardarAlumno = new JButton();
private JButton ActualizarPantalla = new JButton();
private JButton CerrarIngresarAlumno = new JButton();
private JTextField Cedula = new JTextField();
private JTextField Apellidos = new JTextField();
private JTextField Nombres = new JTextField();
private JButton Fondo1 = new JButton();
private JButton DatosPadresRepresentantes = new JButton();
private JEditorPane FechaNacimiento = new JEditorPane();
```

```
private JTextField PaisNacimiento = new JTextField();
private JEditorPane Direccion = new JEditorPane();
private TitledBorder titledBorder2;
private JTextField EstadoNacimiento = new JTextField();
private JTextField CiudadNacimiento = new JTextField();
private JButton BuscarAlumno = new JButton();
private JLabel fechainscripcion = new JLabel();
private JTextField FechaInscripcion = new JTextField();
private JButton DatosMatricula = new JButton();
private JLabel DatosDeRetiro = new JLabel();
private JComboBox Motivo = new JComboBox(MOTIVODERETIRO);
private JLabel motivo = new JLabel();
private JLabel fechaderetiro = new JLabel();
private JTextField FechaRetiro = new JTextField();
private JButton Fondo4 = new JButton();
private JButton ModificarAlumno = new JButton();
private JButton RetirarAlumno = new JButton();
private JComboBox Seccion = new JComboBox();
public ProcesarAlumnos(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    Alumno.ObtenerFechaActualAñoEscolarActual();
    try {
        jbInit();
        pack();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}
```



```
public ProcesarAlumnos() {
    this(null, "", false);
}
private void jbInit() throws Exception {
    titledBorder2 = new TitledBorder("");
    panel1.setLayout(null);
    this.setResizable(false);
    Eliminar = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "eliminar.png"));
    Modificar = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "modificar.png"));
    Guardar = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "guardar.png"));
    Salir = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "salir.png"));
    Actualizar = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "actualizar.png"));
    Padres = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "padres.png"));
    Buscar = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "buscar.png"));
    Correcto = new ImageIcon(sistemasapsee.MenuInicio.class.getResource(
        "informa.png"));
    Pregunta = new ImageIcon(sistemasapsee.SAPSEE.class.getResource(
        "pregunta.png"));
    Matricula = new ImageIcon(sistemasapsee.SAPSEE.class.getResource(
        "dinero.png"));
    DatosPersonales.setFont(new java.awt.Font("Century Gothic", 1, 11));
    DatosPersonales.setForeground(SystemColor.activeCaption);
}
```

```
DatosPersonales.setText("Datos Personales");
DatosPersonales.setBounds(new Rectangle(17, 14, 98, 15));
cedula.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
cedula.setRequestFocusEnabled(true);
cedula.setText("Cédula");
cedula.setBounds(new Rectangle(19, 34, 35, 15));
Nacionalidad.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Nacionalidad.setBounds(new Rectangle(67, 38, 34, 19));
sexo.setFont(new java.awt.Font("Dialog", 0, 10));
sexo.setText("Sexo:");
sexo.setBounds(new Rectangle(234, 39, 31, 15));
Sexo.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Sexo.setSelectedIndex(0);
Sexo.setBounds(new Rectangle(263, 37, 43, 19));
apellidos.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
apellidos.setText("Apellidos:");
apellidos.setBounds(new Rectangle(311, 38, 47, 15));
nombres.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
nombres.setText("Nombres:");
nombres.setBounds(new Rectangle(312, 64, 46, 15));
direccion.setBounds(new Rectangle(17, 75, 51, 15));
direccion.setText("Dirección:");
direccion.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
escolar.setBounds(new Rectangle(20, 44, 38, 15));
escolar.setText("Escolar:");
escolar.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
telefono.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
telefono.setText("Teléfono:");
telefono.setBounds(new Rectangle(309, 90, 45, 15));
```

```
Telefono.setBounds(new Rectangle(360, 88, 165, 19));
Telefono.setText("");
Telefono.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
DatosEscolares.setBounds(new Rectangle(16, 214, 88, 15));
DatosEscolares.setText("Datos Escolares");
DatosEscolares.setForeground(SystemColor.activeCaption);
DatosEscolares.setFont(new java.awt.Font("Century Gothic", 1, 11));
grado.setBounds(new Rectangle(292, 245, 33, 15));
grado.setText("Grado:");
grado.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
seccion.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
seccion.setText("Sección:");
seccion.setBounds(new Rectangle(369, 245, 41, 15));
etapa.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
etapa.setText("Etapa:");
etapa.setBounds(new Rectangle(20, 243, 30, 15));
Etapa.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Etapa.setToolTipText("");
Etapa.setSelectedIndex(0);
Etapa.setBounds(new Rectangle(50, 242, 86, 19));
Etapa.addActionListener(new ProcesarAlumnos_Etapa_actionAdapter(this));
Grado.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Grado.setToolTipText("");
Grado.setBounds(new Rectangle(325, 243, 38, 19));
escolaridad.setBounds(new Rectangle(142, 244, 59, 15));
escolaridad.setText("Escolaridad:");
escolaridad.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
Escolaridad.setBounds(new Rectangle(202, 243, 85, 19));
Escolaridad.setToolTipText("");
```

```
Escolaridad.setSelectedIndex(0);
Escolaridad.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Escolaridad.setMaximumSize(new Dimension(32767, 32767));
Fondo3.setText("");
Fondo3.setSelectedIcon(null);
Fondo3.setSelected(false);
Fondo3.setBorderPainted(true);
Fondo3.setEnabled(false);
Fondo3.setBounds(new Rectangle(11, 235, 444, 34));
Fondo3.setBackground(SystemColor.activeCaption);
DatosDeNacimiento.setFont(new java.awt.Font("Century Gothic", 1, 11));
DatosDeNacimiento.setForeground(SystemColor.activeCaption);
DatosDeNacimiento.setText("Datos de Nacimiento");
DatosDeNacimiento.setBounds(new Rectangle(15, 127, 119, 15));
estadonacimiento.setBounds(new Rectangle(231, 153, 33, 15));
estadonacimiento.setText("Estado:");
estadonacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
ciudadnacimiento.setBounds(new Rectangle(231, 180, 33, 15));
ciudadnacimiento.setText("Ciudad:");
ciudadnacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
paisnacimiento.setBounds(new Rectangle(52, 182, 25, 15));
paisnacimiento.setText("País:");
paisnacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
Dia.setBounds(new Rectangle(79, 152, 40, 19));
Dia.setToolTipText("");
Dia.setSelectedIndex(-1);
Dia.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Mes.setBounds(new Rectangle(122, 152, 41, 19));
Mes.setToolTipText("");
```

```
Mes.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Año.setBounds(new Rectangle(166, 152, 59, 19));
Año.setToolTipText("");
Año.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Fondo2.setBackground(SystemColor.activeCaption);
Fondo2.setBounds(new Rectangle(12, 145, 415, 61));
Fondo2.setEnabled(false);
Fondo2.setBorderPainted(true);
Fondo2.setSelected(false);
Fondo2.setSelectedIcon(null);
Fondo2.setText("");
GuardarAlumno.setBounds(new Rectangle(186, 275, 40, 41));
GuardarAlumno.setBorderPainted(true);
GuardarAlumno.setIcon(Guardar);
GuardarAlumno.setText("");
GuardarAlumno.addActionListener(new
    ProcesarAlumnos_GuardarAlumno_actionAdapter(this));
GuardarAlumno.setToolTipText("Inscribir al alumno");
ActualizarPantalla.setBounds(new Rectangle(310, 275, 40, 41));
ActualizarPantalla.setToolTipText("Actualizar Pantalla");
ActualizarPantalla.setIcon(Actualizar);
ActualizarPantalla.setText("");
ActualizarPantalla.addActionListener(new
    ProcesarAlumnos_ActualizarPantalla_actionAdapter(this));
CerrarIngresarAlumno.setBounds(new Rectangle(351, 275, 40, 41));
CerrarIngresarAlumno.setDebugGraphicsOptions(0);
CerrarIngresarAlumno.setRequestFocusEnabled(true);
CerrarIngresarAlumno.setBorderPainted(true);
CerrarIngresarAlumno.setContentAreaFilled(true);
```

```
CerrarIngresarAlumno.setFocusPainted(true);
CerrarIngresarAlumno.setIcon(Salir);
CerrarIngresarAlumno.setRolloverEnabled(true);
CerrarIngresarAlumno.setText("");
CerrarIngresarAlumno.addActionListener(new
    ProcesarAlumnos_CerrarIngresarAlumno_actionAdapter(this));
CerrarIngresarAlumno.setToolTipText("Salir");
panel1.setMinimumSize(new Dimension(1, 1));
panel1.setOpaque(true);
panel1.setPreferredSize(new Dimension(555, 325));
Cedula.setBackground(Color.white);
Cedula.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Cedula.setText("");
Cedula.setBounds(new Rectangle(104, 38, 128, 19));
Apellidos.setBounds(new Rectangle(359, 36, 166, 19));
Apellidos.setText("");
Apellidos.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Nombres.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Nombres.setText("");
Nombres.setBounds(new Rectangle(360, 63, 166, 19));
Fondo1.setText("");
Fondo1.setSelectedIcon(null);
Fondo1.setSelected(false);
Fondo1.setBorderPainted(true);
Fondo1.setEnabled(false);
Fondo1.setBounds(new Rectangle(11, 31, 524, 87));
Fondo1.setBackground(SystemColor.activeCaption);
DatosPadresRepresentantes.setBounds(new Rectangle(227, 275, 40, 41));
DatosPadresRepresentantes.setToolTipText(
```

```
"Datos de Padres y Representantes del alumno");
DatosPadresRepresentantes.setIcon(Padres);
DatosPadresRepresentantes.setText("");
DatosPadresRepresentantes.addActionListener(new
    ProcesarAlumnos_DatosPadresRepresentantes_actionAdapter(this));
FechaNacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
FechaNacimiento.setOpaque(false);
FechaNacimiento.setRequestFocusEnabled(true);
FechaNacimiento.setEditable(false);
FechaNacimiento.setText("Fecha de Nacimiento:");
FechaNacimiento.setBounds(new Rectangle(16, 148, 63, 28));
PaisNacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
PaisNacimiento.setText("");
PaisNacimiento.setBounds(new Rectangle(79, 179, 146, 20));
Direccion.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
Direccion.setBorder(titledBorder2);
Direccion.setText("");
Direccion.setBounds(new Rectangle(67, 61, 240, 49));
EstadoNacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
EstadoNacimiento.setSelectionStart(1);
EstadoNacimiento.setText("");
EstadoNacimiento.setBounds(new Rectangle(268, 151, 153, 20));
CiudadNacimiento.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
CiudadNacimiento.setText("");
CiudadNacimiento.setBounds(new Rectangle(268, 179, 153, 20));
BuscarAlumno.setToolTipText("Buscar datos del alumno");
BuscarAlumno.setText("");
BuscarAlumno.addActionListener(new
    ProcesarAlumnos_BuscarAlumno_actionAdapter(this));
```

```
BuscarAlumno.setIcon(Buscar);
BuscarAlumno.setBorderPainted(true);
BuscarAlumno.setBounds(new Rectangle(399, 275, 40, 41));
fechainscripcion.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
fechainscripcion.setText("Fecha Inscripción:");
fechainscripcion.setBounds(new Rectangle(364, 9, 90, 15));
FechaInscripcion.setBounds(new Rectangle(452, 6, 83, 20));
FechaInscripcion.setBackground(Color.white);
FechaInscripcion.setFont(new java.awt.Font("Comic Sans MS", 1, 11));
FechaInscripcion.setEditable(false);
FechaInscripcion.setHorizontalAlignment(SwingConstants.CENTER);
DatosMatricula.setBounds(new Rectangle(268, 275, 41, 41));
DatosMatricula.setToolTipText("Ingresar pago de inscripción del alumno");
DatosMatricula.setIcon(Matricula);
DatosMatricula.setText("");
DatosMatricula.addActionListener(new
    ProcesarAlumnos_DatosMatricula_actionAdapter(this));
DatosDeRetiro.setBounds(new Rectangle(445, 126, 82, 15));
DatosDeRetiro.setText("Datos de Retiro");
DatosDeRetiro.setForeground(SystemColor.activeCaption);
DatosDeRetiro.setFont(new java.awt.Font("Century Gothic", 1, 11));
Motivo.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
Motivo.setBounds(new Rectangle(440, 162, 84, 20));
motivo.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
motivo.setText("Motivo:");
motivo.setBounds(new Rectangle(445, 147, 38, 15));
fechaderetiro.setBounds(new Rectangle(445, 183, 32, 15));
fechaderetiro.setText("Fecha:");
fechaderetiro.setFont(new java.awt.Font("Comic Sans MS", 0, 10));
```



```
FechaRetiro.setHorizontalAlignment(SwingConstants.CENTER);
FechaRetiro.setEditable(false);
FechaRetiro.setText("");
FechaRetiro.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
FechaRetiro.setBackground(Color.white);
FechaRetiro.setBounds(new Rectangle(441, 199, 83, 20));
Fondo4.setBackground(SystemColor.activeCaption);
Fondo4.setBounds(new Rectangle(434, 145, 96, 80));
Fondo4.setEnabled(false);
Fondo4.setBorderPainted(true);
Fondo4.setSelected(false);
Fondo4.setSelectedIcon(null);
Fondo4.setText("");
ModificarAlumno.setBounds(new Rectangle(440, 275, 40, 41));
ModificarAlumno.setToolTipText("Modificar datos del alumno");
ModificarAlumno.setIcon(Modificar);
ModificarAlumno.setText("");
ModificarAlumno.addActionListener(new
ProcesarAlumnos_ModificarAlumno_actionAdapter(this));
RetirarAlumno.setText("");
RetirarAlumno.addActionListener(new
    ProcesarAlumnos_RetirarAlumno_actionAdapter(this));
RetirarAlumno.setToolTipText("Retirar al alumno del plantel");
RetirarAlumno.setIcon(Eliminar);
RetirarAlumno.setBounds(new Rectangle(481, 275, 40, 41));
Seccion.setBounds(new Rectangle(411, 242, 37, 19));
this.getContentPane().add(pane1, BorderLayout.WEST);
pane1.add(DatosPersonales, null);
pane1.add(cedula, null);
```

```
panel1.add(escolar, null);
panel1.add(Nacionalidad, null);
panel1.add(Cedula, null);
panel1.add(sexo, null);
panel1.add(Sexo, null);
panel1.add(apellidos, null);
panel1.add(Apellidos, null);
panel1.add(direccion, null);
panel1.add(Direccion, null);
panel1.add(nombres, null);
panel1.add(telefono, null);
panel1.add(Telefono, null);
panel1.add(Nombres, null);
panel1.add(Fondo1, null);
panel1.add(DatosDeNacimiento, null);
panel1.add(FechaNacimiento, null);
panel1.add(Dia, null);
panel1.add(Mes, null);
panel1.add(Año, null);
panel1.add(paisnacimiento, null);
panel1.add(PaisNacimiento, null);
panel1.add(ciudadnacimiento, null);
panel1.add(estadonacimiento, null);
panel1.add(EstadoNacimiento, null);
panel1.add(CiudadNacimiento, null);
panel1.add(Fondo2, null);
panel1.add(DatosEscolares, null);
panel1.add(FechaInscripcion, null);
panel1.add(fechainscripcion, null);
```

```
panel1.add(DatosPadresRepresentantes, null);
panel1.add(DatosMatricula, null);
panel1.add(ActualizarPantalla, null);
panel1.add(CerrarIngresarAlumno, null);
panel1.add(GuardarAlumno, null);
panel1.add(BuscarAlumno, null);
panel1.add(fechaderetiro, null);
panel1.add(FechaRetiro, null);
panel1.add(motivo, null);
panel1.add(Motivo, null);
panel1.add(DatosDeRetiro, null);
panel1.add(etapa, null);
panel1.add(Etapa, null);
panel1.add(escolaridad, null);
panel1.add(Escolaridad, null);
panel1.add(ModificarAlumno, null);
panel1.add(RetirarAlumno, null);
panel1.add(Fondo4, null);
panel1.add(Grado, null);
panel1.add(seccion, null);
panel1.add(grado, null);
panel1.add(Seccion, null);
panel1.add(Fondo3, null);
}
//Verifica que la Plataforma es Windows
private boolean SobrePlataformaWindows(){
    String SistemaOperativo = System.getProperty("os.name");
    if(SistemaOperativo != null && SistemaOperativo.startsWith("Windows"))
        return true;
}
```

```
    return false;
}
//Muestra en Pantalla el Reporte Solicitado
private void MostrarReporte(){
    boolean Windows = SobrePlataformaWindows();
    String VentanaParaAbrir = null;
    try{
        if(Windows){
            VentanaParaAbrir = "rundll32 url.dll,FileProtocolHandler C:Reporte.html";
            Process Ejecutar = Runtime.getRuntime().exec(VentanaParaAbrir);
        }
        else{
            VentanaParaAbrir = "netscape-remote openURL(C:Reporte.html";
            Process Ejecutar = Runtime.getRuntime().exec(VentanaParaAbrir);
        }
        try{
            intCodigoDeSalida = Ejecutar.waitFor();
            if(CodigoDeSalida != 0){
                VentanaParaAbrir = "netscape C:Reporte.html";
                Ejecutar = Runtime.getRuntime().exec(VentanaParaAbrir);
            }
        }
        catch(InterruptedException e){
            JOptionPane.showMessageDialog(null, e.getMessage(),"Error", 0);
        }
    }
    catch(IOException e){
        JOptionPane.showMessageDialog(null, e.getMessage(),"Error", 0);
    }
}
```

```

}
//Crea el Archivo para Generar el Reporte
private void CopiarDatosEnReporte(StringCodigo){
    try {
        BufferedWriter Archivo = new BufferedWriter(new FileWriter(
            "C:Reporte.html"));
        Archivo.write(Codigo);
        Archivo.close();
    }
    catch (IOException ioe) {
        JOptionPane.showMessageDialog
            (null, ioe.getMessage(),
            "Error", 0);
    }
}
//Crea una constancia de estudio de un Alumno
private String CrearConstanciaDeRetiro(){
    String Texto = "";
    String NombrePlantel = "";
    AsignarSentenciaSQL(7);
    Datos = Alumno.BuscarDatosParaConstanciasReportes(SentenciaSQL);
    try{
        if (Datos.next()) {
            Texto = "" +
                "<html>\n" +
                "<head>\n" +
                "<title>Constancia de Retiro</title>\n" +
                "<style type = text/css>\n" +
                "<!--\n" +

```





```

+ "<tr><td><div align=center>_____</div></td></tr>\n"
+ "<tr><th><div align=center>Director (a)</div></th></tr>\n"
+ "<tr><th><div align=center>" + NombreDirector + "</tr></th></div>\n"
+   "<tr><th><div   align=center>C.I.:"   +   CedulaDirector   +
"</tr></th></div>\n"
+ "</table>\n"
+ "</body>\n"
+ "</html>\n";
}
else
    JOptionPane.showMessageDialog(null,
        "Faltan Datos del Director para realizar la Constancia", "Error",
        0);
}
else
    JOptionPane.showMessageDialog(null,
        "Faltan Datos del Plantel para realizar la Constancia", "Error", 0);
}
catch(Exception e){}
return Texto;
}

//Crea una constancia de estudio de un Alumno
private String CrearConstanciaDeInscripción(){
    String Texto = "";
    String NombrePlantel = "";
    AsignarSentenciaSQL(7);
    Datos = Alumno.BuscarDatosParaConstanciasReportes(SentenciaSQL);
    try{
        if (Datos.next()) {

```







```

    + "Constancia que se expide de parte interesada a los " + Alumno.Dia + "
días del mes de " + Meses.getItemAt(Alumno.Mes) + " de " + Alumno.Año +
".</p>\n"
    + "<p>&nbsp;</p></p><p>&nbsp;</p>\n"
    + "<table width=100% border=0>\n"
    + "<tr><td><div align=center>_____</div></td></tr>\n"
    + "<tr><th><div align=center>Director (a)</div></th></tr>\n"
    + "<tr><th><div align=center>" + NombreDirector + "</tr></th></div>\n"
    + "    <tr><th><div align=center>C.I.:" + CedulaDirector +
"</tr></th></div>\n"
    + "</table>\n"
    + "</body>\n"
    + "</html>\n";
}
else
    JOptionPane.showMessageDialog(null,
        "Faltan Datos del Director para realizar la Constancia", "Error",
        0);
}
else
    JOptionPane.showMessageDialog(null,
        "Faltan Datos del Plantel para realizar la Constancia", "Error", 0);
}
catch(Exception e){ }
return Texto;
}
//Asigna la sentencia SQL a ejecutarse
private void AsignarSentenciaSQL(int Accion){
    switch(Accion){

```

//Guarda los datos del Alumno en el Sistema

case 1:

```
SentenciaSQL = ""
+ "INSERT INTO Alumnos VALUES ("
+ Cedula.getText() + ","
+ Nacionalidad.getSelectedItem() + ","
+ Sexo.getSelectedItem() + ","
+ Apellidos.getText() + ","
+ Nombres.getText() + ","
+ Direccion.getText() + ","
+ Telefono.getText() + ","
+ Dia.getSelectedItem() + "-"
+ Mes.getSelectedItem() + "-"
+ Año.getSelectedItem() + ","
+ CiudadNacimiento.getText() + ","
+ EstadoNacimiento.getText() + ","
+ PaisNacimiento.getText() + ","
+ Etapa.getSelectedItem() + ","
+ Escolaridad.getSelectedItem() + ","
+ Grado.getSelectedItem() + ","
+ Seccion.getSelectedItem() + ","
+ Alumno.AñoEscolar + ","
+ FechaInscripcion.getText()
+ "','False','Ninguno','");
```

break;

//Almacena el Historial del Alumno

case 2:

```
SentenciaSQL = ""
+ "INSERT INTO HistorialDeAlumnos VALUES ("
```

```

+ Cedula.getText() + ","
+ Alumno.BuscarIdentificadorDelDocente(Alumno.AñoEscolar,"" +
Grado.getSelectedItem(),"" + Seccion.getSelectedItem()) + ","
+ Escolaridad.getSelectedItem() + ","
+ Grado.getSelectedItem() + ","
+ Seccion.getSelectedItem() + ","
+ Etapa.getSelectedItem() + ","
+ Escolaridad.getSelectedItem() + ","
+ Alumno.AñoEscolar + ")";

break;

//Modificar los datos del Alumno en el Sistema
case 3:
SentenciaSQL = ""
+ "UPDATE Alumnos SET Nacionalidad=" +
Nacionalidad.getSelectedItem()
+ ",Sexo=" + Sexo.getSelectedItem()
+ ",Apellidos=" + Apellidos.getText()
+ ",Nombres=" + Nombres.getText()
+ ",Direccion=" + Direccion.getText()
+ ",Telefono=" + Telefono.getText()
+ ",FechaNacimiento="
+ Dia.getSelectedItem() + "-"
+ Mes.getSelectedItem() + "-"
+ Año.getSelectedItem()
+ ",CiudadNacimiento = " + CiudadNacimiento.getText()
+ ",EstadoNacimiento = " + EstadoNacimiento.getText()
+ ",PaisNacimiento = " + PaisNacimiento.getText()
+ ",Etapa = " + Etapa.getSelectedItem()
+ ",Escolaridad = " + Escolaridad.getSelectedItem()

```

```

+ ",Grado = " + Grado.getSelectedItemAt()
+ ",Seccion = " + Seccion.getSelectedItemAt()
+ ",Retirado = 'False',MotivoRetiro = 'Ninguno',FechaRetiro = " WHERE
Cedula=" + Cedula.getText() + """;
break;
//Modifica el Historial del Alumno
case 4:
SentenciaSQL = ""
+ "UPDATE HistorialDeAlumnos SET idDocente = "
+ Alumno.BuscarIdentificadorDelDocente(AñoEscolarInscripcion,"" +
Grado.getSelectedItemAt(),"" + Seccion.getSelectedItemAt())
+ ",Grado = " + Grado.getSelectedItemAt()
+ ",Seccion = " + Seccion.getSelectedItemAt()
+ ",Etapa = " + Etapa.getSelectedItemAt()
+ ",Escolaridad = " + Escolaridad.getSelectedItemAt()
+ ",AñoEscolar = " + Alumno.AñoEscolar + """;
//Retira al Alumno del Plantel
case 5:
SentenciaSQL = ""
+ "UPDATE Alumnos SET Retirado = 'True', MotivoRetiro = "
+ Motivo.getSelectedItemAt()
+ ", FechaRetiro="
+ FechaRetiro.getText() +
" WHERE Cedula = "
+ Cedula.getText() + """;
break;
case 6:// Buscar Directivo para Reportes diversos
SentenciaSQL = "" + "SELECT Nacionalidad,Cedula,Apellidos,Nombres
FROM Empleados WHERE Cargo = 'Director(a)'"

```

```

        + " AND Retirado = 'False'";
    break;
case 7://Buscar Datos del Plantel para Constancias
    SentenciaSQL = ""
        + "SELECT * FROM Sistema";
    break;
default:
    break;
}
}
// Muestra los datos del alumno en la interfaz
private void MostrarAlumno() {
    String FechaDeNacimiento = "";
    try {
        if(Datos.next()){
            Cedula.setEditable(false);
            if (Interfaz.equals("Consultar") || Interfaz.equals("Retirar")) {
                Nacionalidad.addItem(Datos.getString("Nacionalidad"));
                Sexo.addItem(Datos.getString("Sexo"));
            }
            else {
                Nacionalidad.setSelectedItem(Datos.getString("Nacionalidad"));
                Sexo.setSelectedItem(Datos.getString("Sexo"));
            }
            Apellidos.setText(Datos.getString("Apellidos"));
            Nombres.setText(Datos.getString("Nombres"));
            Direccion.setText(Datos.getString("Direccion"));
            Telefono.setText(Datos.getString("Telefono"));
            FechaDeNacimiento = Alumno.TransformarFecha("FechaNacimiento"

```

```

    , "Alumnos", " WHERE Cedula = '" + Cedula.getText() + "'");
if (Interfaz.equals("Consultar") || Interfaz.equals("Retirar")) {
    Dia.addItem(FechaDeNacimiento.substring(0, 2));
    Mes.addItem(FechaDeNacimiento.substring(3, 5));
    Año.addItem(FechaDeNacimiento.substring(6, 10));
}
else {
    Dia.setSelectedItem(FechaDeNacimiento.substring(0, 2));
    Mes.setSelectedItem(FechaDeNacimiento.substring(3, 5));
    Año.setSelectedItem(FechaDeNacimiento.substring(6, 10));
}
CiudadNacimiento.setText(Datos.getString("CiudadNacimiento"));
EstadoNacimiento.setText(Datos.getString("EstadoNacimiento"));
PaisNacimiento.setText(Datos.getString("PaisNacimiento"));
if (Interfaz.equals("Consultar") || Interfaz.equals("Retirar")) {
    Etapa.addItem(Datos.getString("Etapa"));
    Escolaridad.addItem(Datos.getString("Escolaridad"));
    Grado.removeAllItems();
    Grado.addItem(Datos.getString("Grado"));
    Seccion.addItem(Datos.getString("Seccion"));
}
else {
    Etapa.setSelectedItem(Datos.getString("Etapa"));
    Escolaridad.setSelectedItem(Datos.getString("Escolaridad"));
    Grado.setSelectedItem(Datos.getString("Grado"));
    Seccion.setSelectedItem(Datos.getString("Seccion"));
}
AñoEscolarInscripcion = Datos.getString("AñoEscolar");
FechaInscripcion.setText(Alumno.TransformarFecha(

```



```

        "FechaInscripcion"
        , "Alumnos", " WHERE Cedula = " + Cedula.getText() + ""));
boolean Retirado = Datos.getBoolean("Retirado");
if (Interfaz.equals("Consultar")) {
    Motivo.addItem(Datos.getString("MotivoRetiro"));
    if(Retirado)
        FechaRetiro.setText(Alumno.TransformarFecha("FechaRetiro"
            , "Alumnos", " WHERE Cedula = " + Cedula.getText() + ""));
    else
        FechaRetiro.setText("**/**/**");
}
else
if(Interfaz.equals("Retirar")){
    if(Retirado){
        ActualizarVariables();
        JOptionPane.showMessageDialog
            (null,
            "El Alumno fue retirado el " + Datos.getString("FechaRetiro")
            + " por el motivo de " + Datos.getString("MotivoRetiro"),
            "Error",
            0);
    }
    else
if(Ingresos.AlumnoEsSolvente(Cedula.getText(),AñoEscolarInscripcion,Alumno.Año
Escolar,Alumno.Mes,Alumno.Dia) == false){
        ActualizarVariables();
        JOptionPane.showMessageDialog
            (null,
            "El Alumno no está solvente para ser Retirado del Plantel",

```

```

        "Error",
        0);
    }
}
else
if(Interfaz.equals("Modificar") && Retirado){
    int Respuesta = JOptionPane.showConfirmDialog(null,
        "Desea reingresar el Alumno al Plantel",
        "Modificar Alumno", 0, 0, Pregunta);
    if (Respuesta == 1)
        ActualizarVariables();
    }
}
else{
    JOptionPane.showMessageDialog
        (null,
        "El Alumno no está registrado en el sistema",
        "Error",
        0);
    Cedula.setText("");
}
}
catch (Exception e) {
}
}
//Muestra la interfaz para consultar alumnos
public void InterfazModificarAlumno(){
    Interfaz = "Modificar";
    FechaInscripcion.setFont(new java.awt.Font("Comic Sans MS", 0, 11));

```

```
FechaRetiro.setFont(new java.awt.Font("Comic Sans MS", 0, 11));
GuardarAlumno.setVisible(false);
DatosPadresRepresentantes.setVisible(false);
DatosMatricula.setVisible(false);
RetirarAlumno.setVisible(false);
DatosDeRetiro.setVisible(false);
motivo.setVisible(false);
Motivo.setVisible(false);
fechaderetiro.setVisible(false);
FechaRetiro.setVisible(false);
Fondo4.setVisible(false);
setTitle("Actualizar Alumno");
BuscarAlumno.setLocation(GuardarAlumno.getLocation());
ModificarAlumno.setLocation(DatosPadresRepresentantes.getLocation());
CerrarIngresarAlumno.setLocation(ActualizarPantalla.getLocation());
ActualizarPantalla.setLocation(DatosMatricula.getLocation());
MostrarDatosDeInicioDeInterfaz();
Sexo.setSelectedIndex(-1);
Nacionalidad.setSelectedIndex(-1);
Etapa.setSelectedIndex(-1);
Escolaridad.setSelectedIndex(-1);
Dia.setSelectedIndex(-1);
Mes.setSelectedIndex(-1);
Año.setSelectedIndex(-1);
Grado.setSelectedIndex(-1);
Seccion.setSelectedIndex(-1);
}
//Muestra la interfaz para retirar alumnos
public void InterfazRetiraraAlumno(){
```

```

Interfaz = "Retirar";
FechaInscripcion.setFont(new java.awt.Font("Comic Sans MS", 0,11));
FechaRetiro.setFont(new java.awt.Font("Comic Sans MS", 1, 11));
GuardarAlumno.setVisible(false);
DatosPadresRepresentantes.setVisible(false);
DatosMatricula.setVisible(false);
ModificarAlumno.setVisible(false);
setTitle("Retirar Alumno");
FechaRetiro.setText
    (""
    + Alumno.Dia + "/"
    + Alumno.Mes + "/"
    + Alumno.Año);
BuscarAlumno.setLocation(GuardarAlumno.getLocation());
RetirarAlumno.setLocation(DatosPadresRepresentantes.getLocation());
CerrarIngresarAlumno.setLocation(ActualizarPantalla.getLocation());
ActualizarPantalla.setLocation(DatosMatricula.getLocation());
Nacionalidad.removeAllItems();
Sexo.removeAllItems();
Etapa.removeAllItems();
Escolaridad.removeAllItems();
}
// Muestra la interfaz para consultar los alumnos
public void InterfazConsultarAlumno(){
    Interfaz = "Consultar";
    FechaInscripcion.setFont(new java.awt.Font("Comic Sans MS",
        0, 11));
    GuardarAlumno.setVisible(false);
    ModificarAlumno.setVisible(false);

```

```

RetirarAlumno.setVisible(false);
DatosPadresRepresentantes.setVisible(false);
DatosMatricula.setVisible(false);
BuscarAlumno.setLocation(DatosPadresRepresentantes.getLocation());
CerrarIngresarAlumno.setLocation(ActualizarPantalla.getLocation());
ActualizarPantalla.setLocation(DatosMatricula.getLocation());
setTitle("Consultar Alumno");
Nacionalidad.removeAllItems();
Sexo.removeAllItems();
Etapa.removeAllItems();
Escolaridad.removeAllItems();
Motivo.removeAllItems();
}
//Muestra la interfaz para inscribir nuevos alumnos
public void InterfazInscribirAlumno(){
    Interfaz = "Inscribir";
    setTitle("Inscripción - Nuevos Ingresos Año Escolar " + Alumno.AñoEscolar);
    FechaInscripcion.setText("
        + Alumno.Dia + "/"
        + Alumno.Mes + "/"
        + Alumno.Año);
    DatosDeRetiro.setVisible(false);
    motivo.setVisible(false);
    Motivo.setVisible(false);
    fechaderetiro.setVisible(false);
    FechaRetiro.setVisible(false);
    BuscarAlumno.setVisible(false);
    ModificarAlumno.setVisible(false);
    RetirarAlumno.setVisible(false);
}

```

```
Fondo4.setVisible(false);
MostrarDatosDeInicioDeInterfaz();
Etapa.setSelectedIndex(0);
}
//Muestra las secciones para los el grado de estudio del alumno
private void MostrarSecciones(){
    char C;
    for(C = 'A'; C <= 'Z' && C >= 'A'; C++)
        Seccion.addItem("" + C);
}
// Muestra los datos para fecha de nacimiento del alumno
private void MostrarFechaDeNacimiento(){
    int i;
    //Dias del mes
    for (i = 1; i <= 31; i++)
        if (i < 10)
            Dia.addItem("0" + i);
        else
            Dia.addItem("" + i);
    //Meses del Año
    for (i = 1; i <= 12; i++)
        if (i < 10)
            Mes.addItem("0" + i);
        else
            Mes.addItem("" + i);
    //Años
    for (i = 1900; i <= Alumno.Año; i++)
        Año.addItem("" + i);
}
```

```
// Muestra los datos para iniciar la interfaz
private void MostrarDatosDeInicioDeInterfaz() {
    MostrarFechaDeNacimiento();
    MostrarSecciones();
}

// Inicializa campos y variables
private void ActualizarVariables() {
    Cedula.setText("");
    Cedula.setEditable(true);
    Apellidos.setText("");
    Nombres.setText("");
    Direccion.setText("");
    Telefono.setText("");
    EstadoNacimiento.setText("");
    CiudadNacimiento.setText("");
    PaisNacimiento.setText("");
    if(Interfaz.equals("Inscribir")){
        Nacionalidad.setSelectedIndex(0);
        Sexo.setSelectedIndex(0);
        Etapa.setSelectedIndex(0);
        Grado.setSelectedIndex(0);
        Seccion.setSelectedIndex(0);
        Escolaridad.setSelectedIndex(0);
        Dia.setSelectedIndex(0);
        Mes.setSelectedIndex(0);
        Año.setSelectedIndex(0);
        Motivo.setSelectedIndex(0);
    }
    else
```

```
if (Interfaz.equals("Consultar")) {
    FechaInscripcion.setText("");
    FechaRetiro.setText("");
    Nacionalidad.removeAllItems();
    Sexo.removeAllItems();
    Etapa.removeAllItems();
    Grado.removeAllItems();
    Seccion.removeAllItems();
    Escolaridad.removeAllItems();
    Dia.removeAllItems();
    Mes.removeAllItems();
    Año.removeAllItems();
    Motivo.removeAllItems();
}
else
if (Interfaz.equals("Retirar")){
    FechaInscripcion.setText("");
    Nacionalidad.removeAllItems();
    Sexo.removeAllItems();
    Etapa.removeAllItems();
    Grado.removeAllItems();
    Seccion.removeAllItems();
    Escolaridad.removeAllItems();
    Dia.removeAllItems();
    Mes.removeAllItems();
    Año.removeAllItems();
}
else
if(Interfaz.equals("Modificar")){
```



```
FechaInscripcion.setText("");
Sexo.setSelectedIndex( -1);
Nacionalidad.setSelectedIndex( -1);
Etapas.setSelectedIndex( -1);
Escolaridad.setSelectedIndex( -1);
Dia.setSelectedIndex( -1);
Mes.setSelectedIndex( -1);
Año.setSelectedIndex( -1);
Grado.setSelectedIndex( -1);
Seccion.setSelectedIndex( -1);
}
}
//Verifica que los datos dispuestos en la interfaz estén correctos
private boolean DatosSonValidos() {
    if (Cedula.getText().length() == 0) {
        JOptionPane.showMessageDialog(null,
            "Indique la cédula del Alumno a inscribir",
            "Error",
            0);
        return false;
    }
    if (Alumno.ValidarCodigo(Cedula) == false) {
        JOptionPane.showMessageDialog(null,
            "Cédula del Alumno inválida, verifíquela",
            "Error",
            0);
        return false;
    }
    if (Cedula.getText().length() > 11) {
```

```
JOptionPane.showMessageDialog(null, "Tamaño de cédula excedido", "Error", 0);
return false;
}
if (Etapa.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog(null, "¿Etapa de estudio del Alumno?",
"Advertencia", 1);
    return false;
}
if (Grado.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog(null, "¿Grado del Alumno?", "Advertencia", 1);
    return false;
}
if (Seccion.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog(null, "¿Sección del Alumno?", "Error", 0);
    return false;
}
if (Escolaridad.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog(null, "¿Escolaridad del Alumno?",
"Advertencia", 1);
    return false;
}
if(Telefono.getText().length() > 50){
    JOptionPane.showMessageDialog(null, "Tamaño de teléfono excedido", "Error", 0);
    return false;
}
if(Nacionalidad.getSelectedIndex() == -1){
    JOptionPane.showMessageDialog(null, "¿Nacionalidad del Alumno?", "Error", 0);
    return false;
}
```

```

if(Sexo.getSelectedIndex() == -1){
    JOptionPane.showMessageDialog(null, "¿Sexo del Alumno?", "Error", 0);
    return false;
}
if (Alumno.ValidarFecha("" + Dia.getSelectedItem() + "-" +
Mes.getSelectedItem() + "-" + Año.getSelectedItem()) == false) {
    JOptionPane.showMessageDialog(null, "Verifique la fecha de nacimiento del
Alumno", "Error", 0);
    return false;
}
return true;
}
//Acción del botón CerrarIngresarAlumno
void CerrarIngresarAlumno_actionPerformed(ActionEvent e) {
    dispose();
}
//Acción del JComboBox Etapa
void Etapa_actionPerformed(ActionEvent e) {
    if(Etapa.getSelectedIndex() > -1){
        int i;
        Grado.removeAllItems();
        if (Etapa.getSelectedIndex() == 0)
            for (i = 1; i <= 3; i++)
                Grado.addItem("" + i);
        else
            for (i = 4; i <= 6; i++)
                Grado.addItem("" + i);
    }
}
}

```

```

//Acción del botón ActualizarPantalla
void ActualizarPantalla_actionPerformed(ActionEvent e) {
    ActualizarVariables();
}

//Muestra la interfaz para ingresar los datos de los Padres y Representantes del
Alumno
private void InterfazPadresRepresentantes(){
    DatosPadresRepresentantes PadresRepresentantes =
        new DatosPadresRepresentantes();
    PadresRepresentantes.CedulaAlumno = Cedula.getText();
    Dimension dlgSize = PadresRepresentantes.getPreferredSize();
    Dimension frmSize = getSize();
    Point loc = getLocation();
    PadresRepresentantes.setLocation( (frmSize.width - dlgSize.width) / 2 +
        loc.x,
        (frmSize.height - dlgSize.height) / 2 +
        loc.y);
    PadresRepresentantes.setTitle("Ingresar Padres y Representantes");
    PadresRepresentantes.setModal(true);
    PadresRepresentantes.show();
}

//Acción del botón MostrarPantallaPadresRepresentantes
void MostrarPantallaPadresRepresentantes() {
    Datos = Alumno.BuscarAlumno(Cedula.getText());
    try {
        if (Datos.next()) {
            Datos = Alumno.BuscarPadresRepresentantes(Cedula.getText());
            if (Datos.next()) {
                JOptionPane.showMessageDialog

```

```

        (null,
            "Los datos de Padres y Representantes del Alumno ya están registrados en
el Sistema",
            "Error",
            0);
        Cedula.setText("");
    }
else
    InterfazPadresRepresentantes();
}
else
    JOptionPane.showMessageDialog(null,
        "El Alumno no está registrado en el Sistema aún",
        "Error",
        0);
}
catch (Exception e) {}
}
//Acción del boton DatosPadresRepresentantes
void DatosPadresRepresentantes_actionPerformed(ActionEvent e) {
    if(Alumno.ValidarCodigo(Cedula) == false)
        JOptionPane.showMessageDialog
            (null,"Verifique el dato Cédula, es inválido",
            "Error",0);
    else
        if(Cedula.getText().length() > 0)
            MostrarPantallaPadresRepresentantes();
        else
            JOptionPane.showMessageDialog

```

```

        (null,"Debe facilitar la cédula escolar del Alumno, en la casilla Cédula",
        "Error",0);
    }
//Imprime la Constancia de Retiro
private void ImprimirConstanciaDeRetiro(){
    CopiarDatosEnReporte(CrearConstanciaDeRetiro());
    MostrarReporte();
}
//Imprime la Constancia de Inscripción
private void ImprimirConstanciaDeInscripción(){
    CopiarDatosEnReporte(CrearConstanciaDeInscripción());
    MostrarReporte();
}
//Acción del botón GuardarAlumno
void GuardarAlumno_actionPerformed(ActionEvent e) {
    if (DatosSonValidos()) {
        AsignarSentenciaSQL(1);
        if (Alumno.DocenteEstaRegistrado("" + Grado.getSelectedItem(),
            "" + Seccion.getSelectedItem(),
            Alumno.AñoEscolar)) {
            if (Alumno.GuardarAlumno(SentenciaSQL)) {
                AsignarSentenciaSQL(2);
                Alumno.GuardarHistorialDelAlumno(SentenciaSQL);
                ImprimirConstanciaDeInscripción();
            }
        }
    }
    else
        JOptionPane.showMessageDialog
            (null,

```

```

        "El alumno no puede ser inscrito porque no existe Docente para el " +
        Grado.getSelectedItem() + " Grado Sección " +
        Seccion.getSelectedItem() +
        "", en el Sistema. Ingrese al Docente primero",
        "Error",
        0);
    ActualizarVariables();
}
}
//Acción del botón BuscarAlumno
void BuscarAlumno_actionPerformed(ActionEvent e) {
    if (Cedula.getText().length() == 0)
        JOptionPane.showMessageDialog
            (null,
            "Indique la cédula del Alumno",
            "Error",
            0);
    else{
        Datos = Alumno.BuscarAlumno(Cedula.getText());
        MostrarAlumno();
    }
}
//Muestra la interfaz para la matriculación del Alumno
private void Matricular(){
    Matricula PagoDeMatricula = new Matricula();
    PagoDeMatricula.Cedula.setText(Cedula.getText());
    PagoDeMatricula.InterfazIngresarMatricula();
    Dimension dlgSize = PagoDeMatricula.getPreferredSize();
    Dimension frmSize = getSize();

```

```

Point loc = getLocation();
PagoDeMatricula.setLocation( (frmSize.width - dlgSize.width) / 2 + loc.x,
                             (frmSize.height - dlgSize.height) / 2 + loc.y);
PagoDeMatricula.setModal(true);
PagoDeMatricula.show();
}
//Acción del botón DatosMatricula
void DatosMatricula_actionPerformed(ActionEvent e) {
    if (Cedula.getText().length() == 0)
        JOptionPane.showMessageDialog(null, "Debe facilitar la cédula del Alumno",
                                    "Error", 0);
    else
        if (Alumno.ValidarCodigo(Cedula) == false)
            JOptionPane.showMessageDialog(null, "El dato cédula es incorrecto,
verifíquelo",
                                    "Error", 0);
    else
        Matricular();
}
// Acción del botón RetirarAlumno
void RetirarAlumno_actionPerformed(ActionEvent e) {
    if (Cedula.isEditable()) {
        JOptionPane.showMessageDialog
            (null,
             "No existen datos de Alumno en la Interfaz para realizar el retiro",
             "Error",
             0);
    }
    else

```



```

if (Motivo.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog
        (null,
         "Indique el motivo por el cual se retira el Alumno del Plantel",
         "Error",
         0);
}
else{
    if (JOptionPane.showConfirmDialog(null,
                                     "¿Desea retirar al Alumno del Plantel?",
                                     "Retirar Alumno", 0, 0, Pregunta) == 0) {
        AsignarSentenciaSQL(5);
        Alumno.RetirarAlumno(SentenciaSQL);
        ImprimirConstanciaDeRetiro();
    }
    else
        JOptionPane.showMessageDialog
            (null,
             "El Alumno continúa activo en el Plantel",
             "Alumno Activo",
             0, Correcto);
}
ActualizarVariables();
}
// Acción del botón ModificarAlumno
void ModificarAlumno_actionPerformed(ActionEvent e) {
    if (Cedula.isEditable()) {
        JOptionPane.showMessageDialog
            (null,

```

```

        "No existen datos en la interfaz para realizar la actualización",
        "Error",
        0);
    }
    else
    if(DatosSonValidos()){
        if(Alumno.DocenteEstaRegistrado("'" + Grado.getSelectedItem(), "'" +
Seccion.getSelectedItem(),Alumno.AñoEscolar) == false){
            JOptionPane.showMessageDialog
                (null,
                "No se pueden modificar los datos del Alumno porque no hay Docente
registrado en el Sistema para el " + Grado.getSelectedItem() + " Grado Sección " +
Seccion.getSelectedItem() + "'",
                "Error",
                0);
        }
        else
        if (JOptionPane.showConfirmDialog(null,
                "Desea modificar los datos del Alumno",
                "Modificar Alumno", 0, 0, Pregunta) ==
            0){
            AsignarSentenciaSQL(4);
            Alumno.ModificarHistorialDelAlumno(SentenciaSQL);
            AsignarSentenciaSQL(3);
            Alumno.ModificarAlumno(SentenciaSQL);
            ActualizarVariables();
        }
        else{
            ActualizarVariables();

```

```

OptionPane.showMessageDialog
    (null,
     "Los datos del Alumno no sufrieron ningún cambio",
     "Advertencia",
     1);
    }
    }
    }
}
class ProcesarAlumnos_CerrarIngresarAlumno_actionAdapter implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_CerrarIngresarAlumno_actionAdapter(ProcesarAlumnos adaptee)
    {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.CerrarIngresarAlumno_actionPerformed(e);
    }
}
class ProcesarAlumnos_Etapa_actionAdapter implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_Etapa_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.Etapa_actionPerformed(e);
    }
}

```

```

}
class      ProcesarAlumnos_ActualizarPantalla_actionAdapter      implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_ActualizarPantalla_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.ActualizarPantalla_actionPerformed(e);
    }
}

```

```

class      ProcesarAlumnos_DatosPadresRepresentantes_actionAdapter      implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_DatosPadresRepresentantes_actionAdapter(ProcesarAlumnos
adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.DatosPadresRepresentantes_actionPerformed(e);
    }
}

```

```

class      ProcesarAlumnos_GuardarAlumno_actionAdapter      implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_GuardarAlumno_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
}

```

```

public void actionPerformed(ActionEvent e) {
    adaptee.GuardarAlumno_actionPerformed(e);
}
}

class        ProcesarAlumnos_BuscarAlumno_actionAdapter        implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_BuscarAlumno_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.BuscarAlumno_actionPerformed(e);
    }
}

class        ProcesarAlumnos_DatosMatricula_actionAdapter        implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_DatosMatricula_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.DatosMatricula_actionPerformed(e);
    }
}

class        ProcesarAlumnos_RetirarAlumno_actionAdapter        implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_RetirarAlumno_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
}

```

```

    }
    public void actionPerformed(ActionEvent e) {
        adaptee.RetirarAlumno_actionPerformed(e);
    }
}
class ProcesarAlumnos_ModificarAlumno_actionAdapter implements
java.awt.event.ActionListener {
    ProcesarAlumnos adaptee;
    ProcesarAlumnos_ModificarAlumno_actionAdapter(ProcesarAlumnos adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.ModificarAlumno_actionPerformed(e);
    }
}
//Fin de la clase ProcesarAlumnos

```

Seguidamente se muestra el código fuente de métodos usados por la clase `ProcesarAlumnos.class` relacionados con las clases: `GestorAlumnos.java` y `GestorIngresos.java`:

Código fuente de `GestorAlumnos.ObtenerFechaActualAñoEscolarActual()`:

```

// Obtiene del Sistema el día, mes y año actual, además crea el Año Escolar actual
public void ObtenerFechaActualAñoEscolarActual(){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
    }
}

```

```

Datos = Sentencia.executeQuery
    ("SELECT DATEPART(DAY,GETDATE())AS DiaActual, "
    + "DATEPART(MONTH,GETDATE()) AS MesActual, "
    + "DATEPART(YEAR,GETDATE()) as AñoActual");
if (Datos.next()) {
    Dia = Datos.getInt("DiaActual");
    Mes = Datos.getInt("MesActual");
    Año = Datos.getInt("AñoActual");
    if ( (Mes == 7 && Dia < 17) || Mes < 7)
        AñoEscolar = "" + (Año - 1) + "-" + Año;
    else
        if ( (Mes == 7 && Dia > 17) || Mes > 7)
            AñoEscolar = "" + Año + "-" + (Año + 1);
    }
}
catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "" + e.getMessage().substring(47),
    "Error",0);
}
}
//Fin de ObtenerFechaActualAñoEscolarActual.

```

Código fuente de GestorAlumnos.ValidarFecha(StringFecha):

```

// Verifica si los datos de fecha de nacimiento dados son correctos en los intervalos de
fecha existentes en el sistema
public boolean ValidarFecha(String FECHA){
    Statement Sentencia;
    try{

```

```

Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
Sentencia = Conexion.createStatement();
Datos = Sentencia.executeQuery("SELECT CAST("'" + FECHA +
                                "' AS DATETIME) AS Fecha");
}
catch(SQLException e){
    return false;
}
return true;
}
//Fin de ValidarFecha

```

Código fuente de GestorAlumnos.ValidarCodigo(JTextField Codigo):

```

// Verifica si el texto dado es sólo número
public boolean ValidarCodigo(JTextField Codigo){
    int i;
    // Codigo vacío
    if(Codigo.getText().length() == 0)
        return false;

    // Código con caracteres no válidos
    for(i = 0; i < Codigo.getText().length(); )
        if (Codigo.getText().charAt(i) >= '0' && Codigo.getText().charAt(i) <= '9')
            i++;
        else
            return false;
    return true;
} //Fin de ValidarCodigo

```



Código fuente de GestorAlumnos.GuardarAlumno(String SentenciaSQL):

```
//Almacena los datos de los Alumnos en el sistema
public boolean GuardarAlumno(String SentenciaSQL) {
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
        JOptionPane.showMessageDialog
            (null, "El Alumno ha sido inscrito para el periodo escolar presente",
            "Inscripción de Alumno", 1, Correcto);
    }
    catch (SQLException e) {
        if (e.getErrorCode() == 2627) {
            JOptionPane.showMessageDialog
                (null,
                "La cédula facilitada aparece registrada en el Sistema o el Alumno está
                retirado del Plantel",
                "Error", 0);
            return false;
        }
        else{
            JOptionPane.showMessageDialog
                (null, "" + e.getMessage().substring(47), "Error", 0);
            return false;
        }
    }
    return true;
}
```

```
//Fin de GuardarAlumno
```

Código fuente de GestorAlumnos.GuardarHistorialDelAlumno(String SentenciaSQL):

```
// Guarda el historial de estudio del Alumno
```

```
public void GuardarHistorialDelAlumno(String SentenciaSQL) {
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null,e.getMessage().substring(47), "Error",0);
    }
}
//Fin de GuardarHistorialDelAlumno
```

Código fuente de GestorAlumnos.DocenteEstaRegistrado(String Grado, String Seccion, String AñoEscolar):

```
//Verifica si el Docente está registrado en el Sistema para el Alumno de nuevo ingreso
```

```
public boolean DocenteEstaRegistrado(String Grado, String Seccion, String AñoEscolar){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
```

```

Sentencia = Conexion.createStatement();
Datos = Sentencia.executeQuery(
    "SELECT Cedula FROM Docentes WHERE Grado = "
    + Grado + " AND Seccion = "
    + Seccion + " AND AñoEscolar = "
    + AñoEscolar + "");
if(Datos.next())
    return true;
}
catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "" + e.getMessage().substring(47),
Error",0);
}
return false;
} //Fin de DocenteEstaRegistrado

```

Código fuente de GestorAlumnos.BuscarIdentificadorDelDocente(String AñoEscolar,String Grado,String Seccion):

```

// Busca el docente para el Grado, Sección y Año Escolar dados
public int BuscarIdentificadorDelDocente(String AñoEscolar,String Grado,String
Seccion){
    int idDocente = 0;
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos = Sentencia.executeQuery(
            "SELECT idDocente FROM Docentes WHERE AñoEscolar = "

```

```

    + AñoEscolar + "" AND Grado = "" + Grado + "" AND Seccion = ""
    + Seccion + """);
if (Datos.next())
    idDocente = Datos.getInt("idDocente");
}
catch (SQLException e) {
    JOptionPane.showMessageDialog
        (null,
         e.getMessage().substring(47),
         "Error", 0);
}
return idDocente;
} //Fin de BuscarIdentificadorDelDocente

```

Código fuente de GestorAlumnos.BuscarAlumno(String SentenciaSQL):

```

// Busca un alumno en el Sistema
public ResultSet BuscarAlumno(String Cedula){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos = Sentencia.executeQuery("SELECT * FROM Alumnos where Cedula = ""
+ Cedula + """);
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage().substring(47), "Error",0);
    }
    return Datos;
}

```

```
}//Fin de BuscarAlumno
```

Código fuente de GestorAlumnos.BuscarPadresRepresentantes(String Cedula):

```
// Busca los datos de los Padres y Representantes de los alumnos registrados en el sistema
```

```
public ResultSet BuscarPadresRepresentantes(String Cedula){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos = Sentencia.executeQuery
            ("SELECT * FROM PadresRepresentantes WHERE CedulaAlumno = "
            + Cedula + "");
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog
            (null, e.getMessage().substring(47), "Error",0);
    }
    return Datos;
}
//Fin de BuscarDatosPadresRepresentantes.
```

Código fuente de GestorAlumnos.RetirarAlumno(SentenciaSQL):

```
// Coloca inactivo a los alumnos que son retirados del Plantel
```

```
public void RetirarAlumno(String SentenciaSQL) {
    Statement Sentencia;
    try {
```

```

Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
Sentencia = Conexion.createStatement();
Sentencia.executeUpdate(SentenciaSQL);
JOptionPane.showMessageDialog
    (null,
     "El Alumno ha sido retirado del Plantel",
     "Retirar Alumno", 1, Correcto);
}
catch (SQLException e) {
    JOptionPane.showMessageDialog(null,
                                e.getMessage().substring(47), "Error", 0);
}
} //Fin de RetirarAlumno

```

Código fuente de ModificarHistorialDeAlumno(String SentenciaSQL):

```

// Modifica los datos de los alumnos inscritos en el Sistema
public void ModificarHistorialDeAlumno(String SentenciaSQL){
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null,
                                    e.getMessage().substring(47),"Error",0);
    }
} //Fin de ModificarHistorialDeAlumno

```

Código fuente de GestorAlumnos.ModificarAlumno(String SentenciaSQL):

```
// Modifica los datos de los alumnos inscritos en el Sistema
public void ModificarAlumno(String SentenciaSQL){
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
        JOptionPane.showMessageDialog (null, "Los datos del alumno han sido
modificados", "Modificar Empleado", 2, Correcto);
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null,e.getMessage().substring(47),"Error",0);
    }
} //Fin de ModificarAlumno.
```

Código fuente de GestorIngresos.AlumnoEsSolvente(String Cedula, String AñoEscolarInscripcion, String AñoEscolarActual, int Mes,int Dia)

```
//Verifica si el Alumno tiene deudas pendientes con el Plantel
public boolean AlumnoEsSolvente(String Cedula, String AñoEscolarInscripcion,
String AñoEscolarActual, int Mes,int Dia){
    int Inicio = Integer.parseInt(AñoEscolarInscripcion.substring(0,4));
    JComboBox Pagos = new JComboBox();
    for( ; Inicio < Integer.parseInt(AñoEscolarActual.substring(5,9)); Inicio++){
        Datos = BuscarPagos(Cedula,"" + Inicio + "-" + (Inicio + 1));
        try {
            Pagos.removeAllItems();
```





### 4.5.2. Diagrama del Modelo Relacional.

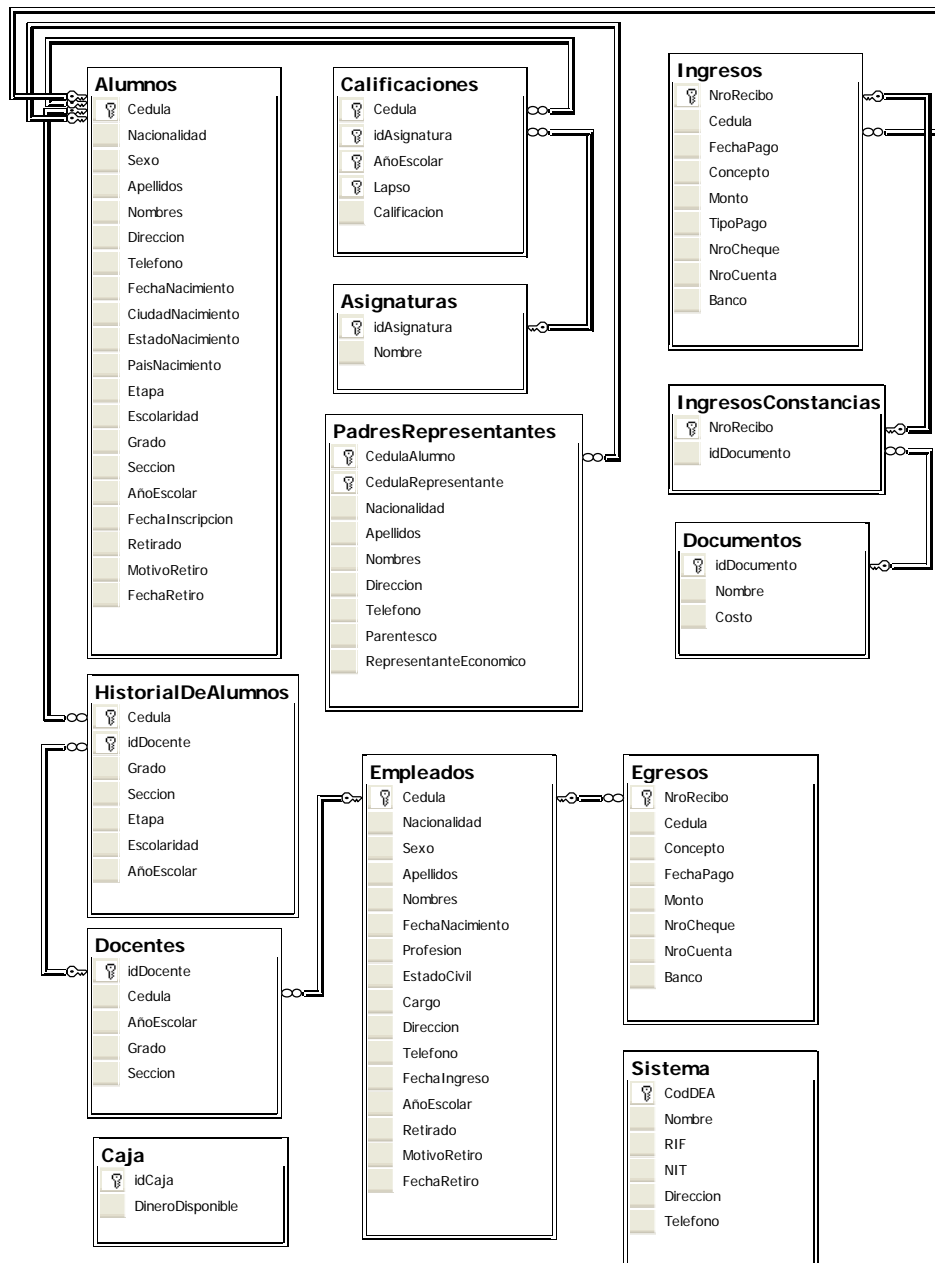


Figura 4.33. Diagrama del Modelo Relacional de la Base de Datos de SAPSEE.

Algunas de las tablas identificadas y descritas anteriormente, están vinculadas a otra por alguno de sus atributos.

Las dependencias identificadas en el diagrama 4.33 corresponden a claves foráneas o externas.

En el diagrama las relaciones apuntan directamente a los atributos que guardan una dependencia funcional entre sí.

La llave al final de la línea indica que ese atributo determina funcionalmente al atributo ubicado en el extremo contrario de la línea, es decir, el atributo del extremo contrario depende funcionalmente del atributo señalado por la llave.

El código para generar la base de datos es el siguiente:

```
create table Sistema
(CodDEA varchar(11) not null,
Nombre varchar(MAX),
RIF varchar(12),
NIT varchar(11),
Direccion varchar(MAX),
Telefono varchar(50),
primary key(CodDEA))
```

```
create table Caja
(idCaja int not null,
DineroDisponible money,
primary key(idCaja))
insert into Caja values('1','0.00')
```

```
create table Alumnos
(Cedula varchar(11) not null,
```

Nacionalidad char(1),  
Sexo char(1),  
Apellidos varchar(max),  
Nombres varchar(max),  
Direccion varchar(max),  
Telefono varchar(50),  
FechaNacimiento datetime,  
CiudadNacimiento varchar(max),  
EstadoNacimiento varchar(max),  
PaisNacimiento varchar(max),  
Etapa varchar(7),  
Escolaridad varchar(10),  
Grado char(1),  
Seccion char(1),  
AñoEscolar varchar(9),  
FechaInscripcion datetime,  
Retirado bit,  
MotivoRetiro varchar(11),  
FechaRetiro datetime,  
primary key(Cedula))  
create table PadresRepresentantes  
(CedulaAlumno varchar(11) not null,  
CedulaRepresentante varchar (11) not null,  
Nacionalidad char(1),  
Apellidos varchar(max),  
Nombres varchar (max),  
Direccion varchar (max),  
Telefono varchar (50),  
Parentesco varchar (5),

RepresentanteEconomico bit,  
primary key (CedulaAlumno,CedulaRepresentante),  
foreign key (CedulaAlumno) references Alumnos (Cedula))

```
create table Asignaturas
(idAsignatura int not null,
Nombre varchar(max),
primary key(idAsignatura))
```

```
create table Calificaciones
(Cedula varchar(11) not null,
idAsignatura int not null,
AñoEscolar varchar(9) not null,
Lapso varchar(3) not null,
Calificacion char(1),
primary key(Cedula,idAsignatura,AñoEscolar,Lapso),
foreign key (idAsignatura) references Asignaturas (idAsignatura)
foreign key (Cedula) references Alumnos (Cedula))
```

```
create table Documentos
(idDocumento varchar(4) not null,
Nombre varchar(25),
Costo money,
primary key (idDocumento))
insert into Documentos values
('DOC1','Constancia de Conducta','0.00')
insert into Documentos values
('DOC2','Constancia de Estudios','0.00')
insert into Documentos values
```

```
('DOC3','Constancia de Retiro','0.00')
insert into Documentos values
('DOC4','Solvencia','0.00')
insert into Documentos values
('DOC5','Boletín Informativo','0.00')
create table Ingresos
(NroRecibo varchar (50) not null,
Cedula varchar (11),
FechaPago datetime,
Concepto varchar (50),
Monto money,
TipoPago varchar (8),
NroCheque varchar (12),
NroCuenta varchar (20),
Banco varchar (100),
primary key (NroRecibo),
foreign key (Cedula) references Alumnos (Cedula))
create table IngresosConstancias
(NroRecibo varchar (50) not null,
idDocumento varchar (4),
primary key (NroRecibo),
foreign key (idDocumento) references Documentos (idDocumento),
foreign key (NroRecibo) references Ingresos (NroRecibo))

create table Empleados
(Cedula varchar (11) not null,
Nacionalidad char(1),
Sexo char (1),
Apellidos varchar (max),
```

Nombres varchar (max),  
FechaNacimiento datetime,  
Profesion varchar (max),  
EstadoCivil varchar (13),  
Cargo varchar (26),  
Direccion varchar (max),  
Telefono varchar (50),  
FechaIngreso datetime,  
AñoEscolar varchar (9),  
Retirado bit,  
MotivoRetiro varchar (23),  
FechaRetiro datetime,  
primary key (Cedula))

create table Docentes  
(idDocente int not null,  
Cedula varchar (11),  
AñoEscolar varchar (9),  
Grado char (1),  
Seccion char (1),  
primary key (idDocente),  
foreign key (Cedula) references Empleados (Cedula))

create table HistorialDeAlumnos  
(Cedula varchar (11) not null,  
idDocente int not null,  
Grado char (1),  
Seccion char (1),  
Etapa varchar (7),

Escolaridad varchar (10),  
AñoEscolar varchar (9),  
primary key (Cedula,idDocente),  
foreign key (Cedula) references Alumnos (Cedula),  
foreign key (idDocente) references Docentes (idDocente))

create table Egresos  
(NroRecibo varchar (50) not null,  
Cedula varchar (11),  
Concepto varchar (11),  
FechaPago datetime,  
Monto money,  
NroCheque varchar (12),  
NroCuenta varchar (20),  
Banco varchar (100),  
primary key (NroRecibo),  
foreign key (Cedula) references Empleados (Cedula))

#### **4.6. Planificación de las Sigüientes Fases.**

1. Fase de Construcción: Se usarán los requerimientos para el desarrollo, es decir, convirtiéndose en una versión beta, se termina el software con su arquitectura, la línea base de la arquitectura se convierte en un sistema completo, estable y en su contenido están todos los casos de uso acabados para su desarrollo. Se obtienen los modelos completos del sistema.
2. Fase de Transición: Durante esta fase se implementará el software de instalación del sistema, así como la construcción del manual de usuario y la arquitectura final.

#### **4.7. Resumen de la Fase de Elaboración**

1. La línea base de la arquitectura representada en el Modelo de Análisis (Diagramas de Clases del Análisis y Diagramas de Colaboración de cada caso de uso) y el Modelo del Diseño (Diagramas de Clases del Diseño de cada caso de uso).
2. Implementación del 10% de los casos de uso del sistema SAPSEE.
3. Diseño de la Base de Datos del Sistema SAPSEE.
4. Planificación de las siguientes fases.



## **CAPÍTULO 5**

### **FASE DE CONSTRUCCIÓN**

#### **5.1. Introducción**

En esta fase se desarrolla un producto software listo para su operación inicial en el entorno del usuario. El objetivo primordial es obtener una versión beta capaz de ser distribuida, ser sometida a pruebas e integración. La Fase de Construcción construye el sistema completo dejando cerrados los modelos de análisis, diseño e implementación para mantener la integridad de la arquitectura.

#### **5.2. Implementación**

En este flujo de trabajo se toma el resultado del diseño de la fase de elaboración y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, ejecutables. Se implementan las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente. Se prueban los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en un ejecutable. Una vez identificados los componentes ejecutables, se han detallados los componentes de implementación que son necesarios para la construcción del sistema. Las clases de diseño definidas han de ser implementadas en código fuente dentro de componentes ficheros, incluye el código fuente generado a partir de la clase de diseño y la implementación de las operaciones en forma de métodos. El código fuente que implementará las clases de diseño residirá en componentes fichero “.java” que es la extensión empleada para identificar los archivos fuente del lenguaje Java. Cada componente contendrá la implementación en código fuente de una sola clase de diseño.

A continuación una breve descripción de las clases de los casos de uso de la aplicación del Sistema SAPSEE.

Se llevará a cabo siguiendo el siguiente formato:

1. Breve descripción de la clase (en qué consiste).
2. Identificación de los datos a manejar por la clase (datos que serán introducidos por el usuario a través de la interfaz en proceso).
3. Imagen referencial de la pantalla (interfaz) relacionada con la clase.

### 5.2.1. Implementación del Caso de Uso Procesar Empleados

La pantalla relacionada con el ingreso de empleados es la siguiente:

**Ingresar Empleado para Año Escolar 2008-2009**

**Datos Personales**

Cédula:   Apellidos:

Fecha de Nacimiento:    Nombres:

Estado Civil:  Profesión:

Sexo:  Dirección:

Teléfono:

**Datos Laborales**




Cargo:  Grado:  Sección:    

Figura 5.1. Pantalla de Ingresar Empleado.

Este caso de uso se implementó a través de la clase de diseño `ProcesarEmpleados`, derivada de la clase `Dialog` de Java y ejecutada por medio de la

clase `ProcesarEmpleados.java` que permite el ingreso, consulta, modificación y retiro de los empleados en el sistema. Estas operaciones permiten manejar los datos que serán usados para generar constancias, reportes, pago de quincenas, iniciar el sistema.

Para poder llevar a cabo todas las operaciones de forma correcta es necesario que el usuario tenga conocimiento acerca de los datos a ingresar en el Sistema, ya que estos representan información necesaria para evitar conflictos posteriores.

A continuación se muestran otras pantallas generadas por esta clase para la consulta, modificación y retiro de los empleados del plantel.

La Figura 5.2, muestra la pantalla para consultar los datos de los empleados que se activa cuando el usuario solicita esta actividad desde la interfaz principal.

**Figura 5.2. Pantalla para Consultar Empleado.**

La Figura 5.3, muestra la pantalla para modificar los datos de los empleados que se activa cuando el usuario solicita esta actividad desde la interfaz principal.

**Modificar Empleado**

**Datos Personales**

Cédula:  Apellidos:

Fecha de Nacimiento:    Nombres:

Estado Civil:  Profesión:

Sexo:  Dirección:

Teléfono:

**Datos Laborales**

Cargo:

**Datos de Retiro**

Motivo:  Fecha Retiro:





   

Figura 5.3. Pantalla para Modificar Empleado

**Retirar Empleado**

**Datos Personales**

Cédula:  Apellidos:

Fecha de Nacimiento:    Nombres:

Estado Civil:  Profesión:

Sexo:  Dirección:

Teléfono:

**Datos Laborales**

Cargo:

**Datos de Retiro**

Motivo:  Fecha Retiro:





   

Figura 5.4. Pantalla para Retirar Empleado.

La Figura 5.4 muestra la pantalla para retirar los datos de los empleados en el sistema que se activa cuando el usuario solicita esta actividad desde la interfaz principal.

El código fuente de la clase ProcesarEmpleados.java es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import java.lang.String;
import javax.swing.border.*;
import Gestores.*;
public class ProcesarEmpleados extends JDialog {
    JPanel panel1 = new JPanel();
    ...}
public ProcesarEmpleados(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch (Exception ex) {ex.printStackTrace(); }
    Empleados.ObtenerFechaActualAñoEscolarActual();
}
public ProcesarEmpleados() {
    this(null, "Ingresar Empleado", false);
}
private void jbInit() throws Exception {
    getContentPane().add(panel1);
    ... }
// Asigna la sentencia SQL a ejecutarse
```

```

private void AsignarSentenciaSQL(int Accion){
    switch (Accion) {
        case 1: //Guarda los datos de un Empleado
            SentenciaSQL = ""
                + "INSERT INTO Empleados VALUES ("
                + Cedula.getText() + ","
                + Nacionalidad.getSelectedItemAt() + ","
                + Sexo.getSelectedItemAt() + ","
                + Apellidos.getText() + ","
                + Nombres.getText() + ","
                + DiaNacimiento.getSelectedItemAt() + "-"
                + MesNacimiento.getSelectedItemAt() + "-"
                + AñoNacimiento.getSelectedItemAt() + ","
                + Profesion.getText() + ","
                + EstadoCivil.getSelectedItemAt() + ","
                + CARGO + ","
                + Direccion.getText() + ","
                + Telefono.getText() + ","
                + Empleados.Dia + "-" + Empleados.Mes + "-" + Empleados.Año + ","
                + Empleados.AñoEscolar
                + "','False','Ninguno','");
            break;
        case 2: //Modifica los datos de un Empleado
            SentenciaSQL = ""
                + "UPDATE Empleados SET Nacionalidad=" +
                Nacionalidad.getSelectedItemAt()
                + "','Sexo=" + Sexo.getSelectedItemAt()
                + "','Apellidos=" + Apellidos.getText()
                + "','Nombres=" + Nombres.getText()

```

```

+ ",FechaNacimiento="
+ DiaNacimiento.getSelectedItemAt() + "-"
+ MesNacimiento.getSelectedItemAt() + "-"
+ AñoNacimiento.getSelectedItemAt()
+ ",Profesion=" + Profesion.getText()
+ ",EstadoCivil=" + EstadoCivil.getSelectedItemAt()
+ ",Cargo=" + Cargo.getSelectedItemAt()
+ ",Direccion=" + Direccion.getText()
+ ",Telefono=" + Telefono.getText()
+ ",AñoEscolar=" + Empleados.AñoEscolar
+ ",Retirado='False',"
+ "MotivoRetiro='Ninguno',"
+ "FechaRetiro="
+ " WHERE Cedula=" + Cedula.getText() + """;

```

break;

case 3: //Retira al Empleado del Plantel

```

SentenciaSQL = ""
+ "Update Empleados SET Retirado = 'True', MotivoRetiro="
+ Motivo.getSelectedItemAt()
+ ",FechaRetiro="
+ Empleados.Dia + "-" + Empleados.Mes + "-" + Empleados.Año +
+ " WHERE Cedula="
+ Cedula.getText() + """;

```

break;

case 4: //Guarda los datos de los docentes del plantel

```

SentenciaSQL = ""
+ "INSERT INTO Docentes VALUES ("
+ Empleados.GenerarIdentificadorDelDocente() + ","
+ Cedula.getText() + "," + Empleados.AñoEscolar + ","

```

```

        + Grado.getSelectedItemAt() + "," + Seccion.getSelectedItemAt() + "));
    break;
    case 5: //Verifica que el docente para Grado, Sección y Año Escolar dado no esté
registrado
        SentenciaSQL = "" +
            "SELECT idDocente,Cedula FROM Docentes WHERE Grado = "
            + Grado.getSelectedItemAt() + " AND Seccion = "
            + Seccion.getSelectedItemAt() + " AND AñoEscolar = "
            + Empleados.AñoEscolar + """;
        break;
    case 6: //Seleccionar los datos de un Docente
        SentenciaSQL = ""
            + "SELECT * FROM Docentes WHERE Cedula = "
            + Cedula.getText() + """;
        break;
    default:
        break;
}
}
//Muestra la interfaz de ingreso de empleado
public void InterfazIngresarEmpleado(){
    MostrarDatosDeInterfaz();
    Apellidos.setEditable(true);
    Nombres.setEditable(true);
    Telefono.setEditable(true);
    Direccion.setEditable(true);
    Profesion.setEditable(true);
    Interfaz = "Ingresar Empleado Normal";
    setSize(555, 250);
}
}

```



```

    BuscarEmpleado.setVisible(false);
    ModificarEmpleado.setVisible(false);
    RetirarEmpleado.setVisible(false);
    Motivo.setVisible(false);
    DatosRetiro.setVisible(false);
    Fondo3.setVisible(false);
    motivo.setVisible(false);
    fecharetiro.setVisible(false);
    FechaRetiro.setVisible(false);
    setTitle("Ingresar Empleado para Año Escolar " + Validar.AñoEscolar);
}
// Valida los datos ingresados por el usuario en la interfaz
private boolean DatosSonValidos(){
    if(Cedula.getText().length() == 0){
        JOptionPane.showMessageDialog(null,"Indique la cédula del
Empleado", "Error", 0);
        return false;
    }
    if(Empleados.ValidarCodigo(Cedula) == false){
        JOptionPane.showMessageDialog(null,"Verifique la cédula del
Empleado", "Error", 0);
        return false;
    }
    if(Empleados.ValidarFecha("" + DiaNacimiento.getSelectedItemAt() + "-" +
MesNacimiento.getSelectedItemAt() + "-" + AñoNacimiento.getSelectedItemAt()) ==
false){
        JOptionPane.showMessageDialog(null,"Verifique fecha de nacimiento del
Empleado", "Error", 0);
        return false;
    }
}

```

```

    }
    if(Telefono.getText().length() > 50 ){
        JOptionPane.showMessageDialog(null,"Tamaño de teléfono excedido",
"Error",0);
        return false;
    }
    if(Cargo.getSelectedIndex() == -1){
        JOptionPane.showMessageDialog(null,"¿Cargo del Empleado?","Error",0);
        return false;
    }
    if(Sexo.getSelectedIndex() == -1){
        JOptionPane.showMessageDialog(null,"¿Sexo del Empleado?","Error",0);
        return false;
    }
    if(Nacionalidad.getSelectedIndex() == -1){
        JOptionPane.showMessageDialog(null,"¿Nacionalidad del
Empleado?","Error",0);
        return false;
    }
    if(EstadoCivil.getSelectedIndex() == -1){
        JOptionPane.showMessageDialog(null,"¿Estado Civil del Empleado?","Error",0);
        return false;
    }
    return true;
}
// Cierra la pantalla de ingreso de empleados
void CerrarPantalla_actionPerformed(ActionEvent e) {
    dispose();
}

```

```

//Muestra los datos de los empleados en pantalla
private void MostrarDatos(){
    try{
        if(Datos.next() == false){
            ActualizarVariables();
            JOptionPane.showMessageDialog
                (null,
                 "El Empleado no está registrado en el Sistema",
                 "Error", 0);
        }
        else
            if(Interfaz.equals("Consultar")){
                String Fecha = "";
                Nacionalidad.removeAllItems();
                DiaNacimiento.removeAllItems();
                MesNacimiento.removeAllItems();
                AñoNacimiento.removeAllItems();
                EstadoCivil.removeAllItems();
                Sexo.removeAllItems();
                Cargo.removeAllItems();
                Grado.removeAllItems();
                Seccion.removeAllItems();
                Motivo.removeAllItems();
                Nacionalidad.addItem(Datos.getString("Nacionalidad"));
                Sexo.addItem(Datos.getString("Sexo"));
                Apellidos.setText(Datos.getString("Apellidos"));
                Nombres.setText(Datos.getString("Nombres"));
                Fecha = Validar.TransformarFecha("FechaNacimiento", "Empleados", "
WHERE Cedula = " + Cedula.getText() + """);
            }
    }
}

```

```
DiaNacimiento.addItem(Fecha.substring(0, 2));
MesNacimiento.addItem(Fecha.substring(3, 5));
AñoNacimiento.addItem(Fecha.substring(6, 10));
Profesion.setText(Datos.getString("Profesion"));
EstadoCivil.addItem(Datos.getString("EstadoCivil"));
Cargo.addItem(Datos.getString("Cargo"));
Direccion.setText(Datos.getString("Direccion"));
Telefono.setText(Datos.getString("Telefono"));
boolean RETIRADO = Datos.getBoolean("Retirado");
Motivo.addItem(Datos.getString("MotivoRetiro"));
if (RETIRADO)
    FechaRetiro.setText(Validar.TransformarFecha("FechaRetiro",
        "Empleados", " WHERE Cedula = " + Cedula.getText() + ""));
else
    FechaRetiro.setText("**/**/**");
if (Cargo.getSelectedItem().equals("Docente"))
    MostrarDocente();
Cedula.setEditable(false);
}
else
if(Interfaz.equals("Retirar")){
    String Fecha = "";
    Nacionalidad.removeAllItems();
    DiaNacimiento.removeAllItems();
    MesNacimiento.removeAllItems();
    AñoNacimiento.removeAllItems();
    EstadoCivil.removeAllItems();
    Sexo.removeAllItems();
    Cargo.removeAllItems();
```

```

Grado.removeAllItems();
Seccion.removeAllItems();
Nacionalidad.addItem(Datos.getString("Nacionalidad"));
Sexo.addItem(Datos.getString("Sexo"));
Apellidos.setText(Datos.getString("Apellidos"));
Nombres.setText(Datos.getString("Nombres"));
Fecha = Validar.TransformarFecha("FechaNacimiento", "Empleados",
    " WHERE Cedula = " +
    Cedula.getText() + "");
DiaNacimiento.addItem(Fecha.substring(0, 2));
MesNacimiento.addItem(Fecha.substring(3, 5));
AñoNacimiento.addItem(Fecha.substring(6, 10));
Profesion.setText(Datos.getString("Profesion"));
EstadoCivil.addItem(Datos.getString("EstadoCivil"));
Cargo.addItem(Datos.getString("Cargo"));
Direccion.setText(Datos.getString("Direccion"));
Telefono.setText(Datos.getString("Telefono"));
Retirado = Datos.getBoolean("Retirado");
if(Retirado){
    Motivo.setSelectedItem(Datos.getString("MotivoRetiro"));
    JOptionPane.showMessageDialog
    (null, "El Empleado fue retirado el "
    + Empleados.TransformarFecha("FechaRetiro", "Empleados", " WHERE
Cedula = " + Cedula.getText() + "")
    + " por el motivo de " + Motivo.getSelectedItem(), "Error", 0);
ActualizarVariables();
}
else{
    if (Cargo.getSelectedItem().equals("Docente"))

```

```

        MostrarDocente();
        Cedula.setEditable(false);
    }
}
else
if(Interfaz.equals("Modificar")){
    String Fecha = "";
    Cargo.removeAllItems();
    Grado.removeAllItems();
    Seccion.removeAllItems();
    Motivo.removeAllItems();
    Nacionalidad.setSelectedItem(Datos.getString("Nacionalidad"));
    Sexo.setSelectedItem(Datos.getString("Sexo"));
    Apellidos.setText(Datos.getString("Apellidos"));
    Nombres.setText(Datos.getString("Nombres"));
    Fecha = Validar.TransformarFecha("FechaNacimiento", "Empleados", " WHERE
Cedula = " + Cedula.getText() + "");
    DiaNacimiento.setSelectedItem(Fecha.substring(0, 2));
    MesNacimiento.setSelectedItem(Fecha.substring(3, 5));
    AñoNacimiento.setSelectedItem(Fecha.substring(6, 10));
    Profesion.setText(Datos.getString("Profesion"));
    EstadoCivil.setSelectedItem(Datos.getString("EstadoCivil"));
    Cargo.addItem(Datos.getString("Cargo"));
    Direccion.setText(Datos.getString("Direccion"));
    Telefono.setText(Datos.getString("Telefono"));
    String AñoEscolarEmpleado = Datos.getString("AñoEscolar");
    Retirado = Datos.getBoolean("Retirado");
    Motivo.addItem(Datos.getString("MotivoRetiro"));
    if(Retirado){

```

```

        FechaRetiro.setText(Empleados.TransformarFecha("FechaRetiro",
"Empleados", " WHERE Cedula = " + Cedula.getText() + ""));
        if(AñoEscolarEmpleado.equals(Validar.AñoEscolar))
            ErrorEmpleadoRetirado();
        else
            EmpleadoParaContratoNuevo();
    }
else{
    if (Cargo.getSelectedItem().equals("Docente"))
        MostrarDocente();
        FechaRetiro.setText("**/**/**");
        Cedula.setEditable(false);
    }}}
catch(Exception e){ }
}
//Acción del botón GuardarEmpleado
void GuardarEmpleado_actionPerformed(ActionEvent e) {
    if (DatosSonValidos()) {
        int Respuesta = JOptionPane.showConfirmDialog(null, "¿Está seguro de Ingresar
los datos del Empleado dentro del Sistema?", "Ingresar Empleado", 0, 0, Pregunta);
        switch (Respuesta) {
            case 0:
                if (CargoDirectivosAdministrativos())
                    GuardarEmpleadoDirectivoAdministrativo();
                else
                    if (Cargo.getSelectedItem().equals("Docente"))
                        GuardarEmpleadoDocente();
                    else {
                        AsignarSentenciaSQL(1);

```

```
        Empleados.GuardarEmpleado(SentenciaSQL);
    }
    break;
default:
    break;
}
}
}
public void GuardarEmpleado(String SentenciaSQL){
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
        JOptionPane.showMessageDialog (null, "El Empleado ha sido incluido en el
Sistema", "Ingresar Empleado", 1, Correcto);
    }
    catch (SQLException e) {
        if (e.getErrorCode() == 2627) {
            JOptionPane.showMessageDialog(null, "La cédula dada está registrada en el
Sistema", "Error", 0);
        }
        else
            JOptionPane.showMessageDialog
            (null,
            e.getMessage().substring(47),
            "Error", 0);
    }
}
```



### 5.2.2. Implementación del Caso de Uso Ingresos y Egresos

Este caso de uso se implementó a través de las clases de diseño: Matricula y Nomina. La clase Matricula se derivada de la clase Dialog de Java y ejecutada por medio de la clase Matricula.java que permite el ingreso y consulta de los pagos de matrícula por parte de los alumnos. Estas operaciones permiten manejar los datos que serán usados para generar constancias, reportes, consultar deudas. Para poder llevar a cabo todas las operaciones de forma correcta es necesario que el usuario tenga conocimiento acerca de los datos a ingresar en el Sistema, ya que estos representan información necesaria para evitar conflictos posteriores. La Figura 5.5 muestra una de las pantallas generadas por la clase Matricula, cuando se procede a ingresar los pagos de matrícula al sistema.

Figura 5.5. Pantalla de Ingreso de Pago de Matrícula.

El código es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.lang.*;
import java.io.*;
import Gestores.*;
public class Matricula extends JDialog { // Declaración de variables
    JPanel panel1 = new JPanel();
    ...
}
public Matricula(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
    Alumno.ObtenerFechaActualAñoEscolarActual();
}
public Matricula() {
    this(null, "", false);
}
private void jbInit() throws Exception {
    getContentPane().add(panel1);
    ...
}
```

```

}
//Muestra la interfaz para ingresar las matrículas o pagos de mensualidades de los
alumnos.
public void InterfazIngresarMatricula() {
    int NUMERO = 1 + Ingresos.GenerarReciboIngresos(), i = 0;
    Matricula = true;
    IdentificadorRecibo.setVisible(false);
    BuscarMatricula.setVisible(false);
    Recibo.setText("I-" + NUMERO);
    FechaPago.setText(""
        + Alumno.Dia + "/"
        + Alumno.Mes + "/"
        + Alumno.Año);
    setTitle("Pago de Matrícula / Año Escolar " + Alumno.AñoEscolar);
    AñoEscolar.setText(Alumno.AñoEscolar);
    if(Cedula.getText().length() > 0 )
        SolicitarAlumno();
}
// Muestra las deudas pendientes del Alumno con respecto a un año escolar dado
private boolean MostrarPagosPendientes(String PagoDefinido) {
    int j = 0;
    for (j = 0; j < ConceptosRegistrados.getItemCount(); j++)
        if (ConceptosRegistrados.getItemAt(j).equals(PagoDefinido))
            return false;
    return true;
}
// Busca los registros de pago de un año escolar específico
private void ObtenerRegistrosDePago(String AñoEscolarDePago) {
    ConceptosRegistrados.removeAllItems();
}

```

```

Concepto.removeAllItems();
Datos = Ingresos.BuscarPagos(Cedula.getText(), AñoEscolarDePago);
try {
    String Aux = "";
    while (Datos.next()){
        Aux = Datos.getString("Concepto");
        ConceptosRegistrados.addItem(Aux.substring(0,Aux.indexOf(' ')));
    }
}
catch (Exception e) {}
SeleccionarPagosNoRegistrados();
}

// Asigna la sentencia SQL a ejecutarse
private void AsignarSentenciaSQL(int Accion) {
    switch (Accion) {
        case 1: //Guarda los datos de la Matrícula
            SentenciaSQL = ""
                + "INSERT INTO Ingresos VALUES ("
                + Recibo.getText() + ","
                + Cedula.getText() + ","
                + FechaPago.getText() + ","
                + Concepto.getSelectedItem() + " - Año Escolar "
                + AñoEscolar.getText() + ","
                + Validar.QuitarComas(Monto.getText()) + ","
                + TipoDePago.getSelectedItem() + ","
                + NroCheque.getText() + ","
                + NroCuenta.getText() + ","
                + Banco.getText() + ")";
            break;
    }
}

```

```

case 2: //Aumenta el Dinero disponible de la Caja
    SentenciaSQL = ""
        + "UPDATE Caja SET DineroDisponible="
        + DineroDisponible
        + "" WHERE Código='1";
    break;
case 3://Buscar Datos del Plantel para Recibo de Pago
    SentenciaSQL = ""
        + "SELECT * FROM Sistema";
    break;
default:
    break;
}
}
// Muestra los datos del alumno necesarios para el ingreso de matrícula al sistema
private void MostrarDatos() {
    try {
        if (Datos.next()) {
            Apellidos.setText(Datos.getString("Apellidos"));
            Nombres.setText(Datos.getString("Nombres"));
            Grado.setText(Datos.getString("Grado"));
            Seccion.setText(Datos.getString("Seccion"));
            AñoEscolarInscripcion = Datos.getString("AñoEscolar");
            if(Datos.getBoolean("Retirado")){
                ActualizarVariables();
                JOptionPane.showMessageDialog(null, "El Alumno fue retirado el " +
                    Datos.getString("FechaRetiro") + " por el motivo " +
                    Datos.getString("MotivoRetiro"), "Error", 0);
            }
        }
    }
}

```

```
        else
            Cedula.setEditable(false);
        }
    else {
        JOptionPane.showMessageDialog (null, "El Alumno no existe en el Sistema",
        "Error", 0);
        Cedula.setText("");
    }
}
catch (Exception e) {
}
}
// Acción del JComboBox TipoDePago
void TipoDePago_actionPerformed(ActionEvent e) {
    if (Matricula)
        if (TipoDePago.getSelectedIndex() == 0) {
            NroCuenta.setText("");
            NroCheque.setText("");
            Banco.setText("");
            NroCuenta.setEditable(false);
            NroCheque.setEditable(false);
            Banco.setEditable(false);
        }
    else {
        NroCuenta.setEditable(true);
        NroCheque.setEditable(true);
        Banco.setEditable(true);
    }
}
```

```

//Solicita los datos del alumno
private void SolicitarAlumno(){
    Datos = Alumno.BuscarAlumno(Cedula.getText());
    MostrarDatos();
    ObtenerRegistrosDePago(AñoEscolar.getText());
}
// Acción del botón BuscarAlumno
void BuscarAlumno_actionPerformed(ActionEvent e) {
    if (Cedula.isEditable())
        if (Cedula.getText().length() == 0) {
            JOptionPane.showMessageDialog
                (null,
                 "Indique la Cédula del Alumno",
                 "Error", 0);
        }
    else
        if (Alumno.ValidarCodigo(Cedula) == false) {
            JOptionPane.showMessageDialog
                (null,
                 "Verifique la Cédula del Alumno",
                 "Error", 0);
        }
    else
        SolicitarAlumno();
}
// Valida los datos facilitados en la interfaz
private boolean ValidarDatos() {
    if (Monto.getText().length() == 0) {
        JOptionPane.showMessageDialog

```

```
(null,
    "Indique el monto de cancelación de matrícula del Alumno",
    "Error", 0);
return false;
}
if (Alumno.VerificarDecimales(Monto.getText()) == false) {
    JOptionPane.showMessageDialog
        (null,
            "Verifique el monto de cancelación de matrícula del Alumno",
            "Error", 0);
    return false;
}
if (TipoDePago.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog
        (null,
            "Seleccione tipo de pago",
            "Error", 0);
    return false;
}
if (Concepto.getSelectedIndex() == -1) {
    JOptionPane.showMessageDialog
        (null,
            "Seleccione concepto de pago",
            "Error", 0);
    return false;
}
switch (TipoDePago.getSelectedIndex()) {
    case 1:
        if (Alumno.ValidarCodigo(NroCheque) == false) {
```



```
JOptionPane.showMessageDialog
    (null,
     "Verifique el Número de Cheque. No es válido",
     "Error", 0);
return false;
}
if(NroCheque.getText().length() > 12 ){
    JOptionPane.showMessageDialog(null, "Tamaño del Número del Cheque
excedido", "Error", 0);
    return false;
}
if (Alumno.ValidarCodigo(NroCuenta) == false) {
    JOptionPane.showMessageDialog
        (null, "Verifique el Número de Cuenta. No es válida",
         "Error", 0);
    return false;
}
if(NroCuenta.getText().length() > 20 ){
    JOptionPane.showMessageDialog(null, "Tamaño del Número de Cuenta
excedido", "Error", 0);
    return false;
}
if (Banco.getText().length() == 0) {
    JOptionPane.showMessageDialog (null, "Indique el Nombre del Banco",
"Error", 0);
    return false;
}
if(Banco.getText().length() > 100 ){
    JOptionPane.showMessageDialog(null,
```

```

"Tamaño del nombre del Banco excedido", "Error", 0);
    return false;
}
break;
default:
    break;
}
return true;
}
//Crea el Archivo para Generar el Recibo
private void CopiarDatosEnReporte(StringCodigo){
    try{
        BufferedWriter Archivo = new BufferedWriter(new
FileWriter("C:Reporte.html"));
        Archivo.write(Codigo);
        Archivo.close();
    }
    catch(IOException ioe){
        JOptionPane.showMessageDialog(null, ioe.getMessage(), "Error", 0);
    }
}
//Imprime el Recibo de Pago del Empleado
private void ImprimirRecibo(){
    CopiarDatosEnReporte(CrearReciboDePago());
    MostrarReporte();
}
//Acción del botón Guardar Ingreso
void GuardarIngreso_actionPerformed(ActionEvent e) {
    if (Cedula.isEditable()) {

```

```

JOptionPane.showMessageDialog (null,"Indique los datos del Alumno", "Error", 0);
}
else {
    if (ValidarDatos()) {
        AsignarSentenciaSQL(1);
        if (Ingresos.GuardarMatricula(SentenciaSQL)) {
            DineroDisponible = Ingresos.DineroDisponibleEnCaja() +
                Double.parseDouble(Validar.QuitarComas(
                    Monto.getText()));
            Ingresos.AumentarDineroDisponible("" + DineroDisponible);
            ImprimirRecibo();
            MostrarNumeroReciboDePago();
        }
        Concepto.removeItem(Concepto.getSelectedItem());
        Monto.setText("");
        NroCheque.setText("");
        NroCuenta.setText("");
        Banco.setText("");
        NroCheque.setEditable(false);
        NroCuenta.setEditable(false);
        Banco.setEditable(false);
    }
}
}

// Acción del Botón BuscarMatricula
void BuscarMatricula_actionPerformed(ActionEvent e) {
    if (IdentificadorRecibo.getText().length() == 0)
        JOptionPane.showMessageDialog (null, "Indique el Recibo de Matrícula para la
        búsqueda", "Error", 0);
}

```

```

else {
    Datos = Ingresos.BuscarReciboDeMatricula(IdentificadorRecibo.getText());
    try {
        if (Datos.next()){
            MostrarReciboMatricula();
            if (JOptionPane.showConfirmDialog(null, "¿Desea imprimir el recibo de
            pago?", "Pago de Nómina", 0, 0, Pregunta) == 0)
                ImprimirRecibo();
        }
    }
    else {
        JOptionPane.showMessageDialog(null, "Recibo de Matrícula no existente en el
        Sistema", "Error", 0);
        ActualizarVariables();
    }
}
catch (Exception ex) {}
}
}

// Guarda los datos de la matriculación de los alumnos del Plantel
public boolean GuardarMatricula(String SentenciaSQL){
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
        JOptionPane.showMessageDialog(null, "Matrícula registrada en el Sistema",
        "Pago de Matrícula", 1, Correcto);
    }
    catch (SQLException e) {

```

```

JOptionPane.showMessageDialog(null,e.getMessage().substring(47), "Error", 0);
return false;
}
return true;
}
}...

```

La Figura 5.6 muestra otra de las pantallas generadas por esta clase, cuando se consultan los recibos de pago de matrículas registrados en el sistema.

**Figura 5.6. Pantalla de Consultar Matrícula.**

La clase de diseño Nomina es una clase derivada de la clase Dialog de Java y es implementada por medio de la clase Nomina.java que permite el ingreso y consulta de los pagos de sueldos emitidos a los empleados del plantel. Estas operaciones permiten manejar los datos que serán usados para generar constancias y reportes. Para poder llevar a cabo todas las operaciones de forma correcta es necesario que el

usuario tenga conocimiento acerca de los datos a ingresar en el Sistema, ya que estos representan información necesaria para evitar conflictos posteriores.

La Figura 5.7 muestra una de las pantallas que es generada por la clase Nomina, cuando se procede a ingresar los pagos de nómina al sistema.

**Figura 5.7. Pantalla de Ingresar Pago de Nómina.**

El código fuente de la clase es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.lang.*;
import java.io.*;
```

```

import Gestores.*;
public class Nomina extends JDialog {
private JPanel panel1 = new JPanel();
...
public Nomina(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace(); }
    }
public Nomina() {
    this(null, "", false);
}
private void jbInit() throws Exception {
    panel1.setLayout(null);
    ...
}
//Muestra la interfaz para el pago de nómina
public void InterfazPagoDeNomina(){
    Empleado.ObtenerFechaActualAñoEscolarActual();
    int RECIBO = 1 + Egresos.GenerarReciboNomina();
    GenerarConceptosDePago();
    RealizarPago = true;
    IdentificadorRecibo.setVisible(false);
    BuscarPagoEmpleado.setVisible(false);
    añoscolardepago.setVisible(false);
}

```

```

AñoEscolarDePago.setVisible(false);
Recibo.setText("E-" + RECIBO);
FechaPago.setText("'" + Empleado.Dia + "/" + Empleado.Mes + "/" +
Empleado.Año);
setTitle("Pago de Nómina / Año Escolar " + Empleado.AñoEscolar);
}
// Asigna la sentencia SQL a Ejecutarse
private void AsignarSentenciaSQL(int Accion){
switch(Accion){
case 1: //Guardar pago de quincena hecho a los empleados del Plantel
SentenciaSQL = ""
+ "INSERT INTO Egresos VALUES ("
+ Recibo.getText() + ","
+ Cedula.getText() + ","
+ Pago.getSelectedItem() + " - Año Escolar "
+ Empleado.AñoEscolar + ","
+ FechaPago.getText() + ","
+ Empleado.QuitarComas(Monto.getText()) + ","
+ NroCheque.getText() + ","
+ NroCuenta.getText() + ","
+ Banco.getText() + ")";
break;
case 2: //Modificar el dinero disponible en caja
SentenciaSQL = ""
+ "UPDATE Caja SET DineroDisponible=" + Empleado.QuitarPuntos
("" + + (Egresos.DineroDisponibleEnCaja() -
Double.parseDouble(Empleado.QuitarComas(Monto.getText())))) + "" WHERE
idCaja='1";
break;

```



```

case 3://Busca los datos del Plantel para la impresión de los recibos
    SentenciaSQL = "" + SELECT * FROM Sistema";
    break;
default:
    break;
}
}
// Muestra los datos del empleado consultado
private void MostrarDatos(){
    try {
        Apellidos.setText(Datos.getString("Apellidos"));
        Nombres.setText(Datos.getString("Nombres"));
        Cargo.setText(Datos.getString("Cargo"));
        if(Datos.getBoolean("Retirado")){
            JOptionPane.showMessageDialog
                (null,
                 "Empleado retirado del Sistema el dia "
                 + Empleado.TransformarFecha("FechaRetiro", "Empleados", " WHERE
Cedula = " + Cedula.getText() + """)
                 + " por el motivo " + Datos.getString("MotivoRetiro"), "Error", 0);
            ActualizarVariables();
            Cedula.setEditable(true);
        }
        else
            Cedula.setEditable(false);
    }
    catch (SQLException e) {}
}
//Busca los datos para presentarlos en la interfaz

```

```

private void BuscarDatos(){
    Datos = Empleado.BuscarEmpleado(Cedula.getText());
    try {
        if (Datos.next()) {
            MostrarDatos();
            if (Cedula.isEditable() == false) {
                ObtenerPagosRegistrados();
                SeleccionarPagosNoRegistrados();
                if (Pago.getItemCount() == 0) {
                    JOptionPane.showMessageDialog
                        (null, "Todos los pagos al Empleado han sido efectuados, para el
Año Escolar" + Empleado.AñoEscolar, "Error", 0);
                    ActualizarVariables();
                }
            }
        }
        else
            ActualizarVariables();
    }
    else {
        JOptionPane.showMessageDialog(null, "El Empleado no existe en el Sistema",
"Error", 0);
        ActualizarVariables();
    }
}
catch (Exception e) {}
}

// Verifica que los datos de búsqueda del empleados sean correctos
void BuscarEmpleado_actionPerformed(ActionEvent e) {
    if (Cedula.getText().length() == 0) {

```

```

        JOptionPane.showMessageDialog(null, "Indique la Cédula del Empleado",
"Error", 0);
    }
    else
        if (Empleado.ValidarCodigo(Cedula) == false) {
JOptionPane.showMessageDialog(null,"Verifique la Cédula del Empleado","Error",
0);
        }
        else
            BuscarDatos();
    }
// Verifica que los datos de pago estén correctos para su almacenaje
private boolean ValidarDatosDePago(){
    if(Monto.getText().length() == 0){
        JOptionPane.showMessageDialog(null, "Indique el Monto a Pagar", "Error", 0);
        return false;
    }
    if(Empleado.VerificarDecimales(Monto.getText()) == false){
        JOptionPane.showMessageDialog(null, "Verifique el Monto a Pagar", "Error", 0);
        return false;
    }
    if(Empleado.ValidarCodigo(NroCheque) == false){
        JOptionPane.showMessageDialog(null, "Verifique el Número de Cheque. Es
incorrecto",
            "Error", 0);
        return false;
    }
    if(Empleado.ValidarCodigo(NroCuenta) == false){
        JOptionPane.showMessageDialog(null, "Verifique el Número de Cuenta. Es
incorrecto",
            "Error", 0);
    }
}

```

```

        return false;
    }
    if(Pago.getSelectedIndex() == -1){
        JOptionPane.showMessageDialog(null, "Indique el Concepto de Pago al
Empleado",
"Error", 0);
        return false;
    }
    if(NroCheque.getText().length() > 12 ){
        JOptionPane.showMessageDialog(null, "Tamaño del Número del Cheque
excedido",
"Error", 0);
        return false;
    }
    if(NroCuenta.getText().length() > 20 ){
        JOptionPane.showMessageDialog(null, "Tamaño del Número de Cuenta
excedido",
"Error", 0);
        return false;
    }
    if(Banco.getText().length() > 100 ){
        JOptionPane.showMessageDialog(null, "Tamaño del nombre del Banco
excedido", "Error", 0);
        return false;
    }
    return true;
}
// Acción del Botón GuardarPago
void GuardarPago_actionPerformed(ActionEvent e) {
    if (Cedula.isEditable()) {
        JOptionPane.showMessageDialog(null,

```

```

        "Indique los datos del Empleado para el pago de nómina
correspondiente", "Error", 0);
    }
    else
    if (ValidarDatosDePago()) {
        if((Egresos.DineroDisponibleEnCaja() -
            Double.parseDouble(Empleado.QuitarComas(Monto.getText())) < 0){
            JOptionPane.showMessageDialog
                (null,
                "El pago sobrepasa al dinero disponible en Caja. Dinero Disponible = "
                + (Empleado.QuitarPuntos("
                + Egresos.DineroDisponibleEnCaja())) + ", el pago debe hacerse por un
monto menor","Error", 0);
                Monto.setText("");
            }
        else {
            EnviarPago();
            ImprimirRecibo();
            if(JOptionPane.showConfirmDialog(null,"¿Desea realizar otro pago al
Empleado?", "Pago de Nómina",0,0,Pregunta) == 0){
                Pago.removeItemAt(Pago.getSelectedIndex());
                Monto.setText("");
                NroCheque.setText("");
                NroCuenta.setText("");
                Banco.setText("");
            }
            else
                ActualizarVariables();
        } }

```

```
}
}...
```

La Figura 5.8 muestra la pantalla generada por esta clase cuando se requiere consultar los pagos emitidos al personal.

Figura 5.8. Pantalla para Consultar Pagos de Nómina.

### 5.2.3. Implementación del Caso de Uso Calificaciones

Este caso de uso se implementó a través de la clase de diseño `ProcesarCalificaciones`, derivada de la clase `Dialog` de Java y ejecutada por medio de la clase `ProcesarCalificaciones.java` que permite el ingreso y consulta de las calificaciones obtenidas durante un año escolar por el alumno. Estas operaciones permiten manejar los datos que serán usados para generar el Boletín Informativo (Boleta de Calificaciones). Para poder llevar a cabo todas las operaciones de forma correcta es necesario que el usuario tenga conocimiento acerca de los datos a ingresar

en el Sistema, ya que estos representan información necesaria para evitar conflictos posteriores. La Figura 5.9 muestra la pantalla para ingresar calificaciones al sistema.

Figura 5.9. Pantalla para Ingresar Calificaciones.

El código fuente de esta clase es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import Gestores.*;
public class ProcesarCalificaciones extends JDialog {
private JPanel panel1 = new JPanel();
...}
public ProcesarCalificaciones(Frame frame, String title, boolean modal) {
super(frame, title, modal);
```

```

try {
    jbInit();
    pack();
}
catch(Exception ex) {
    ex.printStackTrace();
}
ObtenerAsignaturas();
Alumnos.ObtenerFechaActualAñoEscolarActual();
}
public ProcesarCalificaciones() {
    this(null, "", false);
}
private void jbInit() throws Exception {
    getContentPane().add(panel1, BorderLayout.CENTER);
    ...
}
//Muestra la interfaz para ingresar calificaciones
public void InterfazIngresarCalificaciones(){
    Interfaz = "Ingresar";
    AñoEscolar.setText("" + Alumnos.AñoEscolar);
    setTitle("Ingresar Calificaciones Año Escolar " + Alumnos.AñoEscolar);
    AsignaturasDelSistema.setSelectedIndex(-1);
    Aumentar.setVisible(false);
    Disminuir.setVisible(false);
    Calificacion.setVisible(false);
}
//Asigna la sentencia SQL a ejecutarse
private void AsignarSentenciaSQL(int Accion){

```



```

switch(Accion){
    case 1: //Guarda la calificación de una Asignatura
        SentenciaSQL = ""
            + "INSERT INTO Calificaciones VALUES ("
            + Cedula.getText() + ","
            + CalificacionesAlumno.ObtenerCodigoAsignatura("
Asignaturas.getSelectedItem()) +
            ","
            + AñoEscolar.getText() + ","
            + Lapso.getSelectedItem() + ","
            + Calificaciones.getSelectedItem() +
            ")";
        break;
    case 2:
        SentenciaSQL = ""
            + "SELECT  Calificaciones.Calificacion,  Asignaturas.Nombre  FROM
Calificaciones INNER JOIN Asignaturas ON Calificaciones.IdAsignatura =
Asignaturas.idAsignatura WHERE Calificaciones.Cedula = "
            + Cedula.getText()
            + " AND Calificaciones.AñoEscolar = "
            + AñoEscolar.getText()
            + " AND Calificaciones.Lapso = "
            + Lapso.getSelectedItem() + """;
        break;
    default:
        break;
}
}
// Muestra los datos de grado en curso del Alumno

```

```

private void MostrarDatos(){
    try{
        Apellidos.setText(Datos.getString("Apellidos"));
        Nombres.setText(Datos.getString("Nombres"));
        Grado.setText(Datos.getString("Grado"));
        Seccion.setText(Datos.getString("Seccion"));
        if(Datos.getBoolean("Retirado")){
            ActualizarVariables();
            JOptionPane.showMessageDialog(null, "El Alumno fue retirado el " +
Datos.getString("FechaRetiro") + " por el motivo " +
Datos.getString("MotivoRetiro"), "Error", 0);
        }
        else{
            Cedula.setEditable(false);
            Lapso.setEnabled(true);
        }
    }
    catch(Exception e){
    }
}

// Busca las Asignaturas registradas en el Sistema
private void ObtenerAsignaturas(){
    try {
        Datos = CalificacionesAlumno.BuscarAsignaturas();
        while (Datos.next())
            AsignaturasDelSistema.addItem(Datos.getString("Nombre"));
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(null, ""+e.getMessage().substring(47),"Error",0);
    }
}

```

```

}
if (AsignaturasDelSistema.getItemCount() == 0){
    JOptionPane.showMessageDialog
        (null, "En el Sistema no hay Asignaturas registradas",
        "Advertencia",0);
    dispose();
}
AsignaturasDelSistema.setSelectedIndex( -1);
}
//Acción del botón CerrarPantalla
void CerrarPantalla_actionPerformed(ActionEvent e) {
    dispose();
}
//Acción del botón BuscarDatosDeAlumno
void BuscarDatosDeAlumno_actionPerformed(ActionEvent e) {
    Datos = Alumnos.BuscarAlumno(Cedula.getText());
    try{
        if (Datos.next()){
            if (Interfaz.equals("Ingresar") || Interfaz.equals("Históricas"))
                MostrarDatos();
            if (Interfaz.equals("Consultar")){
                MostrarDatos();
                BuscarAsignaturasRegistradasDelAlumno();
            }
        }
    }
    else {
        ActualizarVariables();
        JOptionPane.showMessageDialog
            (null, "El Alumno no existe en el Sistema", "Error",

```

```

        0);
    }
}
catch(Exception ex){}
}
//Muestra las asignaturas del alumno que han sido registradas en el Sistema para un
lapso y año escolar dado
private void BuscarAsignaturasRegistradasDelAlumno(){
    if (Lapso.getSelectedIndex() > -1) {
        AsignarSentenciaSQL(2);
        Datos = CalificacionesAlumno.BuscarAsignaturasRegistradasDelAlumno(
            SentenciaSQL);
        if(Interfaz.equals("Ingresar") || Interfaz.equals("Históricas")){
            AsignaturasRegistradasDelAlumno.removeAllItems();
            Asignaturas.removeAllItems();
            try {
                while (Datos.next())
                    AsignaturasRegistradasDelAlumno.addItem(Datos.getString("Nombre"));
            }
            catch (Exception e) {}
            MostrarAsignaturasNoRegistradasDelAlumno();
        }
        //Consulta para Asignaturas y Calificaciones del Alumno
        if(Interfaz.equals("Consultar")){
            Asignaturas.removeAllItems();
            Calificaciones.removeAllItems();
            try {
                while (Datos.next()) {
                    Calificaciones.addItem(Datos.getString("Calificacion"));
                }
            }
        }
    }
}

```

```

        Asignaturas.addItem(Datos.getString("Nombre"));
    }
}
catch (Exception e) {}
MensajePorConsulta();
}
}
}
//Acción del JComboBox Lapso
void Lapso_actionPerformed(ActionEvent e) {
    BuscarAsignaturasRegistradasDelAlumno();
}
//Acción del JComboBox Asignaturas
void Asignaturas_actionPerformed(ActionEvent e) {
    if(Cedula.isEditable() == false && Interfaz.equals("Consultar") &&
Asignaturas.getSelectedIndex() > -1){
        Calificacion.setText("");
        JComboBox Copia = (JComboBox)e.getSource();
        Calificacion.setText("" + Calificaciones.getItemAt(Copia.getSelectedIndex()));
    }
}
//Busca las asignaturas dictadas por el Plantel
public ResultSet BuscarAsignaturas(){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos = Sentencia.executeQuery("SELECT * FROM Asignaturas");
    }
}

```

```

catch (SQLException e) {
    JOptionPane.showMessageDialog(null,""+e.getMessage().substring(47),"Error",0);
}
return Datos;
}
public int GuardarCalificacion(String SentenciaSQL){
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
    }
    catch (SQLException e) {
        if (e.getErrorCode() == 2627)
            return 1;
        else{
            JOptionPane.showMessageDialog(null,
            e.getMessage().substring(47),
            "Error",
            0);
            return 2;
        }
    }
    return 0;
}...
}

```

La Figura 5.10, muestra la pantalla para ingresar calificaciones históricas (obtenidas en años anteriores por los alumnos).

**Ingresar Calificaciones Históricas**

Cédula:  Grado:

Nombres:  Sección:

Apellidos:  Lapso:

Año Escolar: 2007-2008

**Asignatura**  **Nota**




  

Figura 5.10. Pantalla para Ingresar Calificaciones Históricas.

**Consultar Calificaciones**

Cédula:  Grado:

Nombres:  Sección:

Apellidos:  Lapso:

Año Escolar: 2008-2009

**Asignatura**  **Nota**




  

Figura 5.11. Pantalla para Consultar Calificaciones.

La Figura 5.11, muestra la pantalla para consultar las calificaciones de los alumnos.

Ambas pantallas son generadas por la clase `ProcesarCalificaciones.java`

#### 5.2.4. Implementación del Caso de Uso Mostrar Ayuda

Este caso de uso se implementó a través de la clase de diseño MostrarAyuda, derivada de la clase Dialog de Java y ejecutada por medio de la clase MostrarAyuda.java que permite consultar la ayuda para el uso del sistema. Esta ayuda es activada por cualquier usuario del sistema.

Esta clase genera una pantalla para informar al usuario acerca de: el nombre, la versión, el año de creación y el creador del sistema SAPSEE.

La Figura 5.12 muestra la pantalla que muestra dicha información.

Cuando el usuario solicita la información de cómo usar el sistema, la clase genera un documento llamado “Ayuda SAPSEE.pdf” para que el usuario visualice.

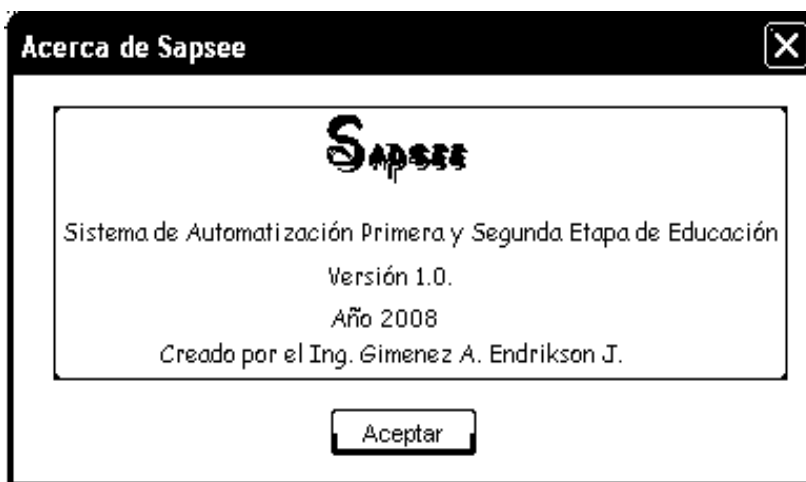


Figura 5.12. Pantalla Acerca de SAPSEE.

El código de la clase MostrarAyuda.java es el siguiente:

```
package sistemasapsee;
```



```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import Gestores.*;
public class MostrarAyuda extends JDialog {
private JPanel panel1 = new JPanel();
...
}
public MostrarAyuda(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
public MostrarAyuda() {
    this(null, "Acerca de Sapsee", false);
}
private void jbInit() throws Exception {
    getContentPane().add(panel1);
}
//Muestra en Pantalla la ayuda de SAPSEE para el usuario
public void MostrarAyuda(){
    Ayuda.BuscarAyuda();
}
void Aceptar_actionPerformed(ActionEvent e) {
```

```

    dispose();
}
}
//Busca la ayuda de SAPSEE y la muestra en pantalla
public void BuscarAyuda(){
    boolean windows = SobrePlataformaWindows();
    String VentanaPorAbrir = null;
    try{
        if(windows){
            VentanaPorAbrir = "rundll32 url.dll,FileProtocolHandler C:Ayuda
SAPSEE.pdf";
            Process Ejecutar = Runtime.getRuntime().exec(VentanaPorAbrir);
        }
        else{
            VentanaPorAbrir = "netscape-remote openURL(C:Ayuda SAPSEE.pdf)";
            Process Ejecutar = Runtime.getRuntime().exec(VentanaPorAbrir);
            try{
                int CodigoDeSalida = Ejecutar.waitFor();
                if(CodigoDeSalida != 0){
                    VentanaPorAbrir = "netscape C:Ayuda SAPSEE.pdf";
                    Ejecutar = Runtime.getRuntime().exec(VentanaPorAbrir);
                }
            }
            catch(InterruptedException e){
                JOptionPane.showMessageDialog(null, e.getMessage(),"Error", 0);
            }
        }
    }
    catch(IOException e){

```

```

        JOptionPane.showMessageDialog(null, e.getMessage(), "Error", 0);
    }
}
//Verifica que la Plataforma es Windows
private boolean SobrePlataformaWindows(){
    String SistemaOperativo = System.getProperty("os.name");
    if(SistemaOperativo != null && SistemaOperativo.startsWith("Windows"))
        return true;
    return false;
}
}
}

```

### 5.2.5. Implementación del Caso de Uso Configurar Sistema

Este caso se uso se implementó a través de las clases de diseño: IniciarSistema, ConfigurarAsignaturas y ConfigurarDocumentos.

La clase IniciarSistema se deriva de la clase Dialog de Java y ejecutada por medio de la clase IniciarSistema.java que permite el ingreso, modificación, eliminación de los datos del plantel: nombre, Registro en el Ministerio del Poder Popular para la Educación, teléfono; así como también del ingreso del personal directivo y administrativo. Estas operaciones permiten manejar los datos que serán usados para generar constancias y reportes. Es necesario que el usuario posea conocimiento acerca de los datos a ingresar al sistema ya que luego serán usados por el sistema para realizar operaciones tanto académicas como administrativas y para que no produzca ningún tipo de conflictos posteriores.

La Figura 5.13 muestra la pantalla generada por la clase IniciarSistema.

Figura 5.13. Pantalla para Iniciar Sistema

La siguiente pantalla es generada cuando el usuario presiona el quinto botón, de izquierda a derecha, de la pantalla identificada como la Figura 5.13.

Figura 5.14. Pantalla para Ingresar Directivos del Plantel.

El código de la clase `IniciarSistema.java` es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.lang.*;
import javax.swing.border.*;
import Gestores.*;
public class IniciarSistema extends JDialog {
    private JPanel panel1 = new JPanel();
    ...
}
public IniciarSistema(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
public IniciarSistema() {
    this(null, "Configurar Sistema - Iniciar Plantel", false);
}
private void jbInit() throws Exception {
    getContentPane().add(panel1, BorderLayout.CENTER);
    ...
}
```

```
public boolean ExistenDatosDeInicioDeSistema(){
    try{
        Datos = Sistema.BuscarDatosDeInicioDeSistema();
        if(Datos.next())
            return true;
    }
    catch(Exception e){
    }
    return false;
}

// Muestra los valores de Datos de Inicio del Sistema
void MostrarDatosDeInicioDeSistema(){
    try{
       CodigoDEA.setText(Datos.getString("CodDEA"));
        Plantel.setText(Datos.getString("Nombre"));
        RIF.setText(Datos.getString("RIF"));
        NIT.setText(Datos.getString("NIT"));
        Direccion.setText(Datos.getString("Direccion"));
        Telefono.setText(Datos.getString("Telefono"));
        CodigoDEA.setEditable(false);
    }
    catch(Exception e){}
}

boolean ValidarDatos(){
    if(CodigoDEA.getText().length() == 0){
        JOptionPane.showMessageDialog(null,
            "Indique el Código DEA",
            "Error", 0);
    }
    return false;
}
```

```
}  
if(CodigoDEA.getText().length() > 11){  
    JOptionPane.showMessageDialog(null,  
        "Verifique el Código DEA. Tamaño excedido",  
        "Error", 0);  
    return false;  
}  
if(Plantel.getText().length() == 0){  
    JOptionPane.showMessageDialog(null,  
        "¿Nombre del Plantel?",  
        "Advertencia", 1);  
    return false;  
}  
if(RIF.getText().length() > 11){  
    JOptionPane.showMessageDialog(null,  
        "Verifique el RIF del Plantel, tamaño excedido",  
        "Error", 0);  
    return false;  
}  
if(NIT.getText().length() > 0 && Sistema.ValidarCodigo(NIT) == false){  
    JOptionPane.showMessageDialog(null,  
        "Verifique el NIT del Plantel",  
        "Error", 0);  
    return false;  
}  
if(NIT.getText().length() > 11){  
    JOptionPane.showMessageDialog(null,  
        "Verifique el NIT del Plantel, tamaño excedido",  
        "Error", 0);
```

```

    return false;
}
return true;
}

void Guardar_actionPerformed(ActionEvent e) {
    if(ExistenDatosDeInicioDeSistema()){
        JOptionPane.showMessageDialog(null,
            "Ya existen Datos de Inicio del Sistema",
            "Advertencia", 1);

        MostrarDatosDeInicioDeSistema();
    }
    else
    if(ValidarDatos()){
        SentenciaSQL = "" + "INSERT INTO Sistema VALUES ("
            + CodigoDEA.getText() + ","
            + Plantel.getText() + ","
            + RIF.getText() + ","
            + NIT.getText() + ","
            + Direccion.getText() + ","
            + Telefono.getText() + ")";

        Sistema.GuardarDatosDeInicioDeSistema(SentenciaSQL);
        ActualizarVariables();
    }
}

void Buscar_actionPerformed(ActionEvent e) {
    if(ExistenDatosDeInicioDeSistema()){
        JOptionPane.showMessageDialog(null, "Datos de Inicio del Sistema
encontrados", "Advertencia", 1);

        MostrarDatosDeInicioDeSistema();
    }
}

```



```

    }
    else
        JOptionPane.showMessageDialog(null, "No existen Datos de Inicio de Sistema
para modificar", "Error", 0);
    }
}
...
public ResultSet BuscarDatosDeInicioDeSistema(){
    try {
        Statement Sentencia;
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos = Sentencia.executeQuery("SELECT * FROM Sistema");
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null,
e.getMessage().substring(47),
"Error", 0);
    }
    return Datos;
}
}
}

```

La clase ConfigurarAsignaturas se deriva de la clase Dialog de Java y ejecutada por medio de la clase ConfigurarAasignaturas.java que permite el ingreso modificación, eliminación de los datos de las asignaturas dictadas por el plantel. Estas operaciones permiten manejar los datos que serán usados para generar constancias y reportes. A continuación, la Figura 5.15 muestra la pantalla que es generada por esta clase y seguidamente el código de compilación.



Figura 5.15. Pantalla de Configurar Asignaturas.

```

package sistemasapsee;
import java.sql.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import Gestores.*;
public class ConfigurarAsignaturas extends JDialog {
private JPanel panel1 = new JPanel();
...
}

public ConfigurarAsignaturas(Frame frame, String title, boolean modal) {
super(frame, title, modal);
try {
jInit();
pack();
}
catch(Exception ex) {
ex.printStackTrace();
}
MostrarIdentificadorDeAsignaturas();

```

```

    }
    public ConfigurarAsignaturas() {
        this(null, "Crear Asignatura", false);
    }
    private void jbInit() throws Exception {
        getContentPane().add(panel1, BorderLayout.NORTH);
        ...
    }
    public void AsignarSentenciaSQL(int Accion){
        switch(Accion){
            case 1:
                SentenciaSQL = ""
                + "INSERT INTO Asignaturas VALUES ("
                + idAsignatura.getText() + ","
                + Nombre.getText() + ")";
                break;
            case 2:
                SentenciaSQL = ""
                + "UPDATE Asignaturas SET Nombre = "
                + Nombre.getText()
                + " WHERE idAsignatura = "
                + idAsignatura.getText()
                + """;
                break;
            case 3:
                SentenciaSQL = ""
                + "DELETE FROM Asignaturas WHERE idAsignatura = "
                + idAsignatura.getText()
                + """;

```

```

        break;
    default:
        break;
    }
}

// Valida los datos entrante de la Asignatura
boolean ValidarDatosAsignatura(){
    if(Nombre.getText().length() == 0){
        JOptionPane.showMessageDialog
            (null,"Indique el Nombre de la Asignatura","Error",0);
        return false;
    }
    return true;
}

void Guardar_actionPerformed(ActionEvent e) {
    if(ValidarDatosAsignatura()){
        AsignarSentenciaSQL(1);
        Sistema.GuardarAsignaturas(SentenciaSQL);
        ActualizarVariables();
    }
}

...
}

/*Busca los datos de una Asignatura en el Sistema*/
public ResultSet BuscarAsignatura(StringCodigo){
    Statement Sentencia;
    try{
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
    }
}

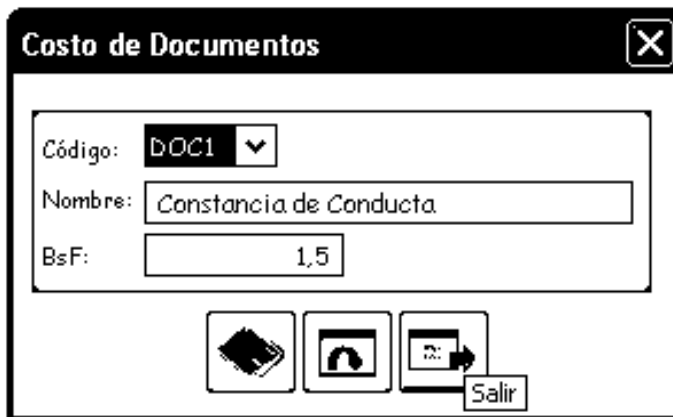
```

```

    Datos = Sentencia.executeQuery ("SELECT * FROM Asignaturas WHERE
idAsignatura = " + Codigo + "");
}
catch(SQLException e){
    JOptionPane.showMessageDialog (null,e.getMessage().substring(47),
"Error",0);
}
return Datos;
}

```

La clase ConfigurarDocumentos se deriva de la clase Dialog de Java y ejecutada por medio de la clase ConfigurarDocumentos.java que permite el ingreso modificación, eliminación de los documentos emitidos por el plantel. Estas operaciones permiten manejar los datos que serán usados para generar constancias, reportes. La Figura 5.16 muestra la pantalla generada por esta clase.



**Figura 5.16. Pantalla de Configurar Documentos**

El código de la clase ConfigurarDocumentos.java es el siguiente:

```

package sistemasapsee;

```

```

import java.awt.*;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import Gestores.*;
public class ConfigurarDocumentos extends JDialog {
private JPanel panel1 = new JPanel();
...
}
public ConfigurarDocumentos(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    ObtenerDocumentos();
    try {
        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
    MostrarDatos();
}
public ConfigurarDocumentos() {
    this(null, "Costo de Documentos", false);
}
private void jbInit() throws Exception {
getContentPane().add(panel1);
}
private void AsignarSentenciaSQL(){
    SentenciaSQL = ""

```

```

        + "UPDATE Documentos SET Costo = "
        + Sistema.QuitarComas(Precio.getText())
        + " WHERE idDocumento = "
        + idDocumento.getSelectedItemId()
        + """;
    }

    public void MostrarDatos(){
        Nombres.setSelectedIndex(idDocumento.getSelectedIndex());
        Costos.setSelectedIndex(idDocumento.getSelectedIndex());
        Nombre.setText("" + Nombres.getSelectedItemId());
        Precio.setText("" + Costos.getSelectedItemId());
    }

    public void ObtenerCostos(){
        Costos.removeAllItems();
        Datos = Sistema.BuscarDocumentos();
        try{
            while(Datos.next())
                Costos.addItem(""
                    + Sistema.QuitarPuntos("" + Datos.getDouble("Costo")));
        }
        catch(Exception e){ }
    }

    public void ObtenerDocumentos(){
        Datos = Sistema.BuscarDocumentos();
        try{
            while(Datos.next()){
                idDocumento.addItem("" + Datos.getString("idDocumento"));
                Nombres.addItem("" + Datos.getString("Nombre"));
                Costos.addItem("" + Sistema.QuitarPuntos("" + Datos.getDouble("Costo")));
            }
        }
    }

```

```

    }
}
catch(Exception e){}
}

void Guardar_actionPerformed(ActionEvent e) {
    if(Sistema.VerificarDecimales(Precio.getText()) == false)
        JOptionPane.showMessageDialog
            (null,
             "Verifiquer el costo del Documento. No es válido",
             "Error", 0);
    else{
        AsignarSentenciaSQL();
        Sistema.GuardarDocumento(SentenciaSQL);
        ObtenerCostos();
        idDocumento.setSelectedIndex(0);
    }
}

/*Busca los datos de los Documento en el Sistema*/
public ResultSet BuscarDocumentos() {
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Datos =
            Sentencia.executeQuery("SELECT * FROM Documentos");
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog
            (null, e.getMessage().substring(47), "Error", 0);
    }
}

```



```

    }
    return Datos;
}
...
}
/*Guarda los datos de un Documento expedido por el Plantel en el Sistema*/
public void GuardarDocumento(String SentenciaSQL) {
    Statement Sentencia;
    try {
        Conexion = DriverManager.getConnection("jdbc:odbc:sistema");
        Sentencia = Conexion.createStatement();
        Sentencia.executeUpdate(SentenciaSQL);
        JOptionPane.showMessageDialog(null,
"Los datos del Documento se han registrado satisfactoriamente",
        "Agregar Documento",
        1,
        Correcto);
    }
    catch (SQLException e) {
        if (e.getErrorCode() == 2627) {
            JOptionPane.showMessageDialog(null,
"El código del documento ya existe en el Sistema",
"Error",
0);
        }
        else
            JOptionPane.showMessageDialog(null, e.getMessage().substring(47), "Error", 0);
    }
}
}

```

### 5.2.6. Implementación del Caso de Uso Constancias y Reportes.

Este caso se uso se implementó a través de la clase de diseño: ConstanciasReportes. La clase ConstanciasReportes se deriva de la clase Dialog de Java y ejecutada por medio de la clase ConstanciasReportes.java que permite la impresión de constancias y reportes que son emitidos por el plantel. Esta clase genera diversas pantallas. A continuación se muestran las pantallas (activadas por el usuario desde la interfaz principal) que son generadas por el código fuente.

La Figura 5.17 muestra la pantalla para crear constancias de inscripción.



**Constancia de Inscripción - Año Escolar 2008-2009** [X]

**Datos para Búsqueda**

Cédula:

**Datos Registrados en el Sistema**

Grado:  Sección:  Etapa:

Alumno:

L. de Nac.

F. de Nac.

Escolaridad:  Fecha de Inscripción

[Magnifying Glass] [Map] [Refresh] [Print]

Figura 5.17. Pantalla para generar Constancias de Inscripción.

La Figura 5.18 muestra la pantalla para generar constancias de estudios.

La Figura 5.19 muestra la pantalla para generar boletines informativos.

**Constancia de Estudios - Año Escolar 2008-2009**

**Datos para Búsqueda**

Cédula:

**Datos Registrados en el Sistema**

Grado:  Sección:  Etapa:

Alumno:

L. de Nac.:

F. de Nac.:

Escolaridad:  Fecha de Inscripción:

Figura 5.18. Pantalla para generar Constancias de Estudios.

**Boletín Informativo**

**Datos para Búsqueda**

Cédula:  Año Escolar: 2008-2009

**Datos Registrados en el Sistema**

Grado:  Sección:  Lapso:

Etapa:

Apellidos:

Nombres:

Escolaridad:

Figura 5.19. Pantalla para generar Boletín Informativo

La Figura 5.20 muestra la pantalla para generar el reporte de alumnos por grado.

**Reporte de Alumnos por Grado**

**Datos para Búsqueda**

Grado: 1 Sección: A Año Escolar: 2008-2009

**Datos Registrados en el Sistema**

Nº:  Sexo:

Cédula:

Apellidos:

Nombres:

Escolaridad:  Fecha de Nacimiento:





   

Figura 5.20. Pantalla para generar el Reporte de Alumnos por Grado

La Figura 5.21 muestra la pantalla para generar el reporte de alumnos por sexo.

**Reporte de Alumnos por Sexo**

**Datos para Búsqueda**

Sexo: F Año Escolar: 2008-2009

**Datos Registrados en el Sistema**

Nº:  Grado:  Sección:

Cédula:

Apellidos:

Nombres:

Escolaridad:  Fecha de Nacimiento:





   

Figura 5.21. Pantalla para generar el Reporte de Alumnos por Sexo

La Figura 5.22 muestra la pantalla para generar el reporte de alumnos por año escolar.

Figura 5.22. Pantalla para generar el Reporte de Alumnos por Año Escolar.

Figura 5.23. Pantalla para generar el Reporte de Ingresos Diarios.

La Figura 5.23 muestra la pantalla para generar el reporte de los ingresos diarios. La Figura 5.24 muestra la pantalla para generar el reporte de los ingresos mensuales.

**Reporte de Ingresos Mensuales**

**Datos para Búsqueda**

Mes: diciembre ▼ Año: 2008 [+/-]

**Datos Registrados en el Sistema**

Recibo Nº: ▼ Monto Bs. [ ]

Cédula: [ ]

Alumno: [ ]

Concepto: [ ]

Hecho en: [ ] Fecha de Pago: [ ]

[Magnifying Glass] [Printer] [Refresh] [Right Arrow]

Figura 5.24. Pantalla para generar el Reporte de Ingresos Mensuales.

La Figura 5.25 es la pantalla para generar el reporte de los ingresos anuales.

**Reporte de Ingresos Anuales**

**Datos para Búsqueda**

Año: 2008 [+/-]

**Datos Registrados en el Sistema**

Recibo Nº: ▼ Monto Bs. [ ]

Cédula: [ ]

Alumno: [ ]

Concepto: [ ]

Hecho en: [ ] Fecha de Pago: [ ]

[Magnifying Glass] [Printer] [Refresh] [Right Arrow]

Figura 5.25. Pantalla para generar el Reporte de Ingresos Anuales.

La Figura 5.26 muestra la pantalla para generar el reporte de los egresos diarios.

Figura 5.26. Pantalla para generar el Reporte de Egresos Diarios.

Figura 5.27. Pantalla para generar el Reporte de Egresos Mensuales.

La Figura 5.27 muestra la pantalla para generar el reporte de los egresos mensuales.

La Figura 5.28 muestra la pantalla para generar el reporte de egresos anuales.

**Reporte de Egresos Anuales**

**Datos para Búsqueda**

Año: 2008 + -

**Datos Registrados en el Sistema**

Recibo N°:  Monto Bs.

Cédula:

Empleado:

Concepto:

Cheque N°:  Fecha de Pago:

Cuenta N°:

Banco:

Icons:    

Figura 5.28. Pantalla para generar Reportes de Egresos Anuales.

**Reporte: Empleados del Plantel 2008-2009**

**Datos para Búsqueda**

Año Escolar: 2008-2009

**Datos Registrados en el Sistema**

N°:  Sexo:  F. de Nac.:

Cédula:

Apellidos:

Nombres:

Cargo:

Icons:  

Figura 5.29. Pantalla para generar Reporte de los Empleados del Plantel.

La Figura 5.29 muestra la pantalla para generar el Reporte de Empleados del Plantel.



El código de la clase ConstanciasReportes.java es el siguiente:

```
package sistemasapsee;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.io.*;
import Gestores.*;
public class ConstanciasReportes extends JDialog {
...
}
public ConstanciasReportes(Frame frame, String title, boolean modal) {
    super(frame, title, modal);
    try {
        jbInit();
        pack();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
    ObtenerFechaActualAñoEscolarActual();
}
public ConstanciasReportes() {
    this(null, "", false);
}
private void jbInit() throws Exception {
...
}
```

```

//Crea el Archivo para Generar el Reporte
private void CopiarDatosEnReporte(String Codigo){
    try{
        BufferedWriter Archivo = new BufferedWriter(new
FileWriter("C:Reporte.html"));
        Archivo.write(Codigo);
        Archivo.close();
        MostrarReporte();
    }
    catch(IOException ioe){
        JOptionPane.showMessageDialog
        (null, ioe.getMessage(),
        "Error", 0);
    }
}

//Activa la interfaz para Mostrar/Imprimir el Boletín Informativo (Boleta de
Calificaciones)
public void InterfazBoletinInformativo(){
    datobusqueda1.setVisible(false);
    DatoBusqueda1.setVisible(false);
    datobusqueda2.setVisible(false);
    DatoBusqueda2.setVisible(false);
    CantidadRegistros.setEnabled(false);
    datoregistrado2.setLocation(numero.getLocation());
    DatoRegistrado2.setLocation(CantidadRegistros.getLocation());
    datoregistrado3.setLocation
    (DatoRegistrado2.getLocation().x + DatoRegistrado2.getSize().width + 3,
    datoregistrado2.getLocation().y);
    DatoRegistrado3.setLocation

```

```

        (datoregistrado3.getLocation().x + datoregistrado3.getSize().width + 3,
        DatoRegistrado2.getLocation().y);
datoregistrado1.setSize(datoregistrado5.getSize());
datoregistrado1.setText("Etapa:");
datoregistrado1.setLocation(datoregistrado4.getLocation());
DatoRegistrado1.setLocation(DatoRegistrado4.getLocation());
DatoRegistrado1.setSize(DatoRegistrado5.getSize());
datoregistrado4.setLocation(datobusqueda1.getLocation());
datoregistrado4.setSize(35,datoregistrado4.getSize().height);
DatoRegistrado4.setEditable(true);
DatoRegistrado4.setSize(100,DatoRegistrado4.getSize().height);
DatoRegistrado4.setLocation(datoregistrado4.getLocation().x
datoregistrado4.getSize().width + 3,DatoBusqueda1.getLocation().y);
DatoRegistrado7.setSize(DatoRegistrado6.getSize());
datoregistrado8.setVisible(false);
DatoRegistrado8.setVisible(false);
numero.setText("Lapso:");
numero.setSize(30,numero.getSize().height);
numero.setLocation(DatoRegistrado3.getLocation().x
DatoRegistrado3.getSize().width + 3,datoregistrado3.getLocation().y);
CantidadRegistros.addItem("I");
CantidadRegistros.addItem("II");
CantidadRegistros.addItem("III");
CantidadRegistros.setSelectedIndex(-1);
CantidadRegistros.setLocation(numero.getLocation().x + numero.getSize().width +
3,DatoRegistrado3.getLocation().y);
setTitle("Boletín Informativo");
Interfaz = "Boletín Informativo";
DatoBusqueda3.setText(AñoEscolar);

```

```

}
//Activa la interfaz para Mostrar/Imprimir el Reporte de Ingresos Anuales
public void InterfazIngresosAnuales(){
    datobusqueda1.setVisible(false);
    DatoBusqueda1.setVisible(false);
    datobusqueda2.setVisible(false);
    DatoBusqueda2.setVisible(false);
    datoregistrado2.setVisible(false);
    DatoRegistrado2.setVisible(false);
    datoregistrado3.setVisible(false);
    DatoRegistrado3.setVisible(false);
    DatoBusqueda3.setText("" + Año);
    datobusqueda3.setText("Año:");
    datobusqueda3.setSize(20,15);
    datobusqueda3.setLocation(datobusqueda1.getLocation());
    DatoBusqueda3.setLocation(DatoBusqueda1.getLocation());
    DatoBusqueda3.setSize(40,22);
    DatoBusqueda3.setLocation(datobusqueda3.getLocation().x
                                                                    +
datobusqueda3.getSize().width + 3,DatoBusqueda1.getLocation().y);
    Arriba.setSize(20,22);
    Arriba.setLocation(DatoBusqueda3.getLocation().x
+ DatoBusqueda3.getSize().width + 2, DatoBusqueda3.getLocation().y);
    Abajo.setSize(20,22);
    Abajo.setLocation(Arriba.getLocation().x +
Arriba.getSize().width, Arriba.getLocation().y);
    Fondo.setSize(120,Fondo.getSize().height);
    numero.setText("Recibo N°:");
    datoregistrado1.setText("Monto Bs.");
    datoregistrado1.setSize(45,15);

```

```

DatoRegistrado1.setSize(149,20);
DatoRegistrado1.setLocation(datoregistrado1.getLocation().x      +
datoregistrado1.getSize().width + 3,DatoRegistrado1.getLocation().y);
datoregistrado5.setText("Alumno:");
datoregistrado6.setText("Concepto");
datoregistrado7.setText("Hecho en:");
datoregistrado8.setText("Fecha de Pago:");
Interfaz = "Ingresos Anuales";
setTitle("Reporte de Ingresos Anuales");
}
//Activa la interfaz para Mostrar/Imprimir Egresos Mensuales
public void InterfazEgresosAnuales(){
datobusqueda1.setVisible(false);
DatoBusqueda1.setVisible(false);
datobusqueda2.setVisible(false);
DatoBusqueda2.setVisible(false);
Fondo1.setSize(Fondo1.getSize().width, Fondo1.getSize().height + 50);
DatoBusqueda3.setText("" + Año);
datobusqueda3.setText("Año:");
datobusqueda3.setSize(20, 15);
datobusqueda3.setLocation(datobusqueda1.getLocation());
DatoBusqueda3.setSize(40, 22);
DatoBusqueda3.setLocation(datobusqueda3.getLocation().x +
                        datobusqueda3.getSize().width + 3,
                        DatoBusqueda1.getLocation().y);
Arriba.setSize(20, 22);
Arriba.setLocation(DatoBusqueda3.getLocation().x +
                    DatoBusqueda3.getSize().width + 2,
                    DatoBusqueda3.getLocation().y);

```

```
Abajo.setSize(20, 22);
Abajo.setLocation(Arriba.getLocation().x + Arriba.getSize().width,
    Arriba.getLocation().y);
Fondo.setSize(120, Fondo.getSize().height);
numero.setText("Recibo N°:");
datoregistrado1.setText("Monto Bs.");
datoregistrado1.setSize(45, 15);
DatoRegistrado1.setSize(149, 20);
DatoRegistrado1.setLocation(datoregistrado1.getLocation().x +
    datoregistrado1.getSize().width + 3,
    DatoRegistrado1.getLocation().y);
datoregistrado5.setText("Empleado:");
datoregistrado6.setText("Concepto");
datoregistrado7.setText("Cheque N°:");
DatoRegistrado7.setSize(100,DatoRegistrado7.getSize().height);
datoregistrado3.setSize(65,datoregistrado3.getSize().height);
datoregistrado3.setText("Fecha de Pago:");
DatoRegistrado3.setSize(79,DatoRegistrado3.getSize().height);
datoregistrado3.setLocation(DatoRegistrado7.getLocation().x +
    DatoRegistrado7.getSize().width + 3,
    datoregistrado7.getLocation().y);
DatoRegistrado3.setLocation
    (datoregistrado3.getLocation().x +
    datoregistrado3.getSize().width + 3,
    DatoRegistrado7.getLocation().y);
datoregistrado8.setText("Cuenta N°:");
datoregistrado8.setLocation(datoregistrado7.getLocation().x,
    datoregistrado7.getLocation().y + 26);
DatoRegistrado8.setSize(DatoRegistrado6.getSize());
```

```

    DatoRegistrado8.setLocation(DatoRegistrado7.getLocation().x,
DatoRegistrado7.getLocation().y + DatoRegistrado7.getSize().height + 6);
    datoregistrado2.setText("Banco:");
    datoregistrado2.setLocation(datoregistrado8.getLocation().x,
        datoregistrado8.getLocation().y + 26);
    DatoRegistrado2.setSize(DatoRegistrado6.getSize());
    DatoRegistrado2.setLocation(DatoRegistrado8.getLocation().x,
        DatoRegistrado8.getLocation().y +
        DatoRegistrado8.getSize().height + 6);
    BuscarReporte.setLocation(BuscarReporte.getLocation().x,
        BuscarReporte.getLocation().y + 50);
    ImprimirReporte.setLocation(ImprimirReporte.getLocation().x,
        ImprimirReporte.getLocation().y + 50);
    ActualizarPantalla.setLocation(ActualizarPantalla.getLocation().x,
        ActualizarPantalla.getLocation().y + 50);
    CerrarPantalla.setLocation(CerrarPantalla.getLocation().x,CerrarPantalla.getLocation
().y + 50);
    Interfaz = "Egresos Anuales";
    setTitle("Reporte de Egresos Anuales");
}
//Activa la interfaz para Reportes de los Alumnos por Año Escolar
public void InterfazAlumnosPorAñoEscolar(){
    DatoBusqueda1.addItem("Ninguno");
    DatoBusqueda2.addItem("Ninguno");
    datobusqueda1.setVisible(false);
    DatoBusqueda1.setVisible(false);
    datobusqueda2.setVisible(false);
    DatoBusqueda2.setVisible(false);
    datobusqueda3.setBounds(new Rectangle(22, 52, 57, 15));

```

```

DatoBusqueda3.setBounds(new Rectangle(80, 47, 72, 21));
Arriba.setBounds(new Rectangle(153, 46, 15, 12));
Abajo.setBounds(new Rectangle(153, 57, 15, 12));
Fondo.setBounds(new Rectangle(16, 42, 155, 32));
Interfaz = "Alumnos por Año Escolar";
setTitle("Reporte de Alumnos por Año Escolar");
DatoBusqueda3.setText("" + AñoEscolar);
}

//Asigna la instrucción SQL a ejecutarse sobre el Sistema
private void AsignarSentenciaSQL(int Accion) {
    switch (Accion) {
        case 1: //Registros de Alumnos por Grado
            if (Interfaz.equals("Alumnos por Grado"))
                SentenciaSQL = ""
                    + "SELECT Nacionalidad,Alumnos.Cedula,Sexo,Apellidos,Nombres,"
                    +
                    "(SELECT      CONVERT(VARCHAR(10),FechaNacimiento,3))      AS
FechaNacimiento,"
                    +
                    "HistorialDeAlumnos.Grado,HistorialDeAlumnos.Seccion,HistorialDeAlumnos.Esco
laridad FROM Alumnos "
                    + " INNER JOIN  HistorialDeAlumnos "
                    + "ON Alumnos.Cedula = HistorialDeAlumnos.Cedula "
                    + "WHERE HistorialDeAlumnos.Grado = ""
                    + DatoBusqueda1.getSelectedItem()
                    + "" AND HistorialDeAlumnos.Seccion = ""
                    + DatoBusqueda2.getSelectedItem()
                    + "" AND HistorialDeAlumnos.AñoEscolar = ""

```



```

+ DatoBusqueda3.getText()
+ """;
else //Registros de Alumnos por Sexo
if (Interfaz.equals("Alumnos por Sexo"))
SentenciaSQL = ""
+ "SELECT Nacionalidad,Alumnos.Cedula,Sexo,Apellidos,Nombres,"
+
"(SELECT      CONVERT(VARCHAR(10),FechaNacimiento,3))      AS
FechaNacimiento,"
+
"HistorialDeAlumnos.Grado,HistorialDeAlumnos.Seccion,HistorialDeAlumnos.Esco
laridad FROM Alumnos "
+ " INNER JOIN HistorialDeAlumnos "
+ "ON Alumnos.Cedula = HistorialDeAlumnos.Cedula "
+ "WHERE HistorialDeAlumnos.AñoEscolar = "
+ DatoBusqueda3.getText()
+ " AND Sexo = "
+ DatoBusqueda1.getSelectedItem() + """;
else //Registros de Alumnos por Año Escolar
if (Interfaz.equals("Alumnos por Año Escolar"))
SentenciaSQL = ""
+ "SELECT Nacionalidad,Alumnos.Cedula,Sexo,Apellidos,Nombres,"
+
"(SELECT      CONVERT(VARCHAR(10),FechaNacimiento,3))      AS
FechaNacimiento,"
+
"HistorialDeAlumnos.Grado,HistorialDeAlumnos.Seccion,HistorialDeAlumnos.Esco
laridad "
+ "FROM Alumnos INNER JOIN HistorialDeAlumnos "

```

```

+ "ON Alumnos.Cedula = HistorialDeAlumnos.Cedula "
+ "WHERE HistorialDeAlumnos.AñoEscolar = "
+ DatoBusqueda3.getText() + """;
else //Constancias de Alumnos
if (Interfaz.equals("Constancia de Inscripción") ||
    Interfaz.equals("Constancia de Estudios"))
    SentenciaSQL = ""
+
+ "SELECT
Nacionalidad,Etapa,Grado,Seccion,Apellidos,Nombres,CiudadNacimiento,EstadoNac
imimiento,PaisNacimiento,CONVERT(VARCHAR(10),FechaNacimiento,3) AS
FechaNacimiento,Escolaridad,CONVERT(VARCHAR(10),FechaInscripcion,3) AS
FechaInscripcion, AñoEscolar "
+ " FROM Alumnos WHERE Cedula = "
+ DatoBusqueda3.getText() + " AND Retirado = 'False'";
else //Boletín Informativo
if (Interfaz.equals("Boletín Informativo") &&
    DatoRegistrado4.isEditable())
    SentenciaSQL = ""
+
+ "SELECT
HistorialDeAlumnos.Etapa,HistorialDeAlumnos.Grado,HistorialDeAlumnos.Seccion,
"
+ "Apellidos,Nombres,HistorialDeAlumnos.Escolaridad FROM "
+ "Alumnos INNER JOIN HistorialDeAlumnos "
+ "ON Alumnos.Cedula = HistorialDeAlumnos.Cedula "
+ "WHERE HistorialDeAlumnos.AñoEscolar ="
+ DatoBusqueda3.getText()
+ " AND HistorialDeAlumnos.Cedula ="
+ DatoRegistrado4.getText()
+ """;

```

```

else
if (Interfaz.equals("Boletín Informativo") &&
    DatoRegistrado4.isEditable() == false)
SentenciaSQL = ""
+
"SELECT Calificaciones.idAsignatura,Asignaturas.Nombre,Calificacion "
+ "FROM Calificaciones INNER JOIN Asignaturas "
+ "ON Calificaciones.idAsignatura = Asignaturas.idAsignatura "
+ "WHERE Calificaciones.AñoEscolar ="
+ DatoBusqueda3.getText()
+ "" AND Calificaciones.Cedula ="
+ DatoRegistrado4.getText()
+ "" AND Calificaciones.Lapso ="
+ CantidadRegistros.getSelectedItem()
+ """;
break;

case 2: //Registros de Empleados del Plantel
SentenciaSQL = ""
+
"SELECT
Nacionalidad,Cedula,Sexo,Apellidos,Nombres,CONVERT(VARCHAR(10),FechaNa
cimiento,103) AS FechaNacimiento,Cargo FROM Empleados WHERE Retirado =
'False'";
break;

case 3: //Registros de Ingresos y Egresos del Plantel
if (Interfaz.equals("Ingresos Diarios"))
SentenciaSQL = ""
+
"SELECT NroRecibo,Nacionalidad,Alumnos.Cedula,Apellidos,Nombres,"

```

```

+          "CONVERT(VARCHAR(10),FechaPago,3)          AS
FechaPago,Concepto,Monto,"
+ "TipoPago FROM Alumnos INNER JOIN Ingresos "
+ "ON Alumnos.Cedula = Ingresos.Cedula"
+ " WHERE FechaPago = '"
+ DatoBusqueda1.getSelectedIndex() + "/"
+ (DatoBusqueda2.getSelectedIndex() + 1) + "/"
+ DatoBusqueda3.getText() + "'";
else
if (Interfaz.equals("Ingresos Mensuales")) {
SentenciaSQL = "" +
    "SELECT NroRecibo,Nacionalidad,Alumnos.Cedula,Apellidos,Nombres,"
+          "CONVERT(VARCHAR(10),FechaPago,3)          AS
FechaPago,Concepto,Monto,"
+ "TipoPago FROM Alumnos INNER JOIN Ingresos "
+ "ON Alumnos.Cedula = Ingresos.Cedula ";
if ( (DatoBusqueda1.getSelectedIndex() + 1) < 12)
    SentenciaSQL = SentenciaSQL
+ "WHERE FechaPago >= '1/"
+ (DatoBusqueda1.getSelectedIndex() + 1) + "/"
+ DatoBusqueda3.getText()
+ "' AND FechaPago < '1/"
+ (DatoBusqueda1.getSelectedIndex() + 2) + "/"
+ DatoBusqueda3.getText()
+ "'";
else
    SentenciaSQL = SentenciaSQL
+ "WHERE FechaPago >= '1/"
+ (DatoBusqueda1.getSelectedIndex() + 1) + "/"

```

```

+ DatoBusqueda3.getText() + " AND FechaPago < '31/"
+ (DatoBusqueda1.getSelectedIndex()+1) + "/" + DatoBusqueda3.getText()
+ """;

}
else
if (Interfaz.equals("Ingresos Anuales"))
    SentenciaSQL = ""
    +
    "SELECT NroRecibo,Nacionalidad,Alumnos.Cedula,Apellidos,Nombres,"
    +
    "CONVERT(VARCHAR(10),FechaPago,3) AS
FechaPago,Concepto,Monto,"
    + "TipoPago FROM Alumnos INNER JOIN Ingresos "
    + "ON Alumnos.Cedula = Ingresos.Cedula "
    + "WHERE FechaPago >= '1/1/"
    + DatoBusqueda3.getText() + " AND FechaPago <='31/12/"
    + DatoBusqueda3.getText() + """;
else
if (Interfaz.equals("Egresos Diarios"))
    SentenciaSQL = ""
    +
    "SELECT NroRecibo,Nacionalidad,Empleados.Cedula,Apellidos,Nombres,"
    + "Concepto,Monto,NroCheque,NroCuenta,Banco,"
    + "CONVERT(VARCHAR(10),FechaPago,3) AS FechaPago "
    + "FROM Empleados INNER JOIN Egresos "
    + "ON Empleados.Cedula = Egresos.Cedula "
    + "WHERE Egresos.FechaPago = "
    + DatoBusqueda1.getSelectedIndex() + "/"
    + (DatoBusqueda2.getSelectedIndex() + 1) + "/"

```

```

        + DatoBusqueda3.getText()
        + """;
else
if (Interfaz.equals("Egresos Mensuales")) {
    SentenciaSQL = ""
        +
        "SELECT NroRecibo,Nacionalidad,Empleados.Cedula,Apellidos,Nombres,"
        + "Concepto,Monto,NroCheque,NroCuenta,Banco,"
        + "CONVERT(VARCHAR(10),FechaPago,3) AS FechaPago "
        + "FROM Empleados INNER JOIN Egresos "
        + "ON Empleados.Cedula = Egresos.Cedula ";
if ( (DatoBusqueda1.getSelectedIndex() + 1) < 12)
    SentenciaSQL = SentenciaSQL
        + "WHERE FechaPago >= '1/"
        + (DatoBusqueda1.getSelectedIndex() + 1) + "/"
        + DatoBusqueda3.getText()
        + "' AND FechaPago < '1/"
        + (DatoBusqueda1.getSelectedIndex() + 2) + "/"
        + DatoBusqueda3.getText()
        + """;
else
    SentenciaSQL = SentenciaSQL
        + "WHERE FechaPago >= '1/"
        + (DatoBusqueda1.getSelectedIndex() + 1) + "/"
        + DatoBusqueda3.getText()
        + "' AND FechaPago < '31/" + (DatoBusqueda1.getSelectedIndex() + 1) + "/" +
        DatoBusqueda3.getText() + """;
    }
else

```

```

if (Interfaz.equals("Egresos Anuales"))
    SentenciaSQL = ""
    +
    "SELECT NroRecibo,Nacionalidad,Empleados.Cedula,Apellidos,Nombres,"
    + "Concepto,Monto,NroCheque,NroCuenta,Banco,"
    + "CONVERT(VARCHAR(10),FechaPago,3) AS FechaPago "
    + "FROM Empleados INNER JOIN Egresos "
    + "ON Empleados.Cedula = Egresos.Cedula "
    + "WHERE FechaPago >= '1/1/"
    + DatoBusqueda3.getText() + "' AND FechaPago <='31/12/"
    + DatoBusqueda3.getText() + """;
break;

```

case 4: //Datos del Docente para Constancias

```

if (Interfaz.equals("Alumnos por Grado"))
    SentenciaSQL = ""
    + "SELECT Apellidos,Nombres,Nacionalidad,Empleados.Cedula "
    + " FROM Empleados INNER JOIN Docentes"
    + " ON Empleados.Cedula = Docentes.Cedula"
    + " WHERE Docentes.Grado = "
    + DatoBusqueda1.getSelectedText()
    + "' AND Docentes.Seccion = "
    + DatoBusqueda2.getSelectedText()
    + "' AND Docentes.AñoEscolar = "
    + DatoBusqueda3.getText() + """;

```

else //Datos del Docente para el Boletín Informativo

```

SentenciaSQL = ""
    + "SELECT Apellidos,Nombres,Nacionalidad,Empleados.Cedula "
    + " FROM Empleados INNER JOIN Docentes"
    + " ON Empleados.Cedula = Docentes.Cedula"

```

```

        + " WHERE Docentes.Grado = "
        + DatoRegistrado2.getText()
        + "' AND Docentes.Seccion = "
        + DatoRegistrado3.getText()
        + "' AND Docentes.AñoEscolar = "
        + DatoBusqueda3.getText() + "'";
    break;
case 5: // Buscar Directivo para Reportes diversos
    SentenciaSQL = ""
        +
        "SELECT  Nacionalidad,Cedula,Apellidos,Nombres  FROM  Empleados
WHERE Cargo = "
        + Directivo + "' AND Retirado = 'False'";
    break;
case 6: //Buscar datos del Plantel
    SentenciaSQL = "" + "SELECT * FROM Sistema";
    break;
case 7: //Buscar datos del Representante Económico del Alumno
    SentenciaSQL = "" + "SELECT
Apellidos,Nombres,Nacionalidad,CedulaRepresentante FROM PadresRepresentantes
WHERE CedulaAlumno = " + DatoRegistrado4.getText()
        + "' AND RepresentanteEconomico = 'True'";
    break;
default:
    break;
}
}
//Acción del botón ImprimirReporte
void ImprimirReporte_actionPerformed(ActionEvent e) {

```



```

if(DatosEncontrados){
    int Respuesta = JOptionPane.showConfirmDialog(null, "¿Desea imprimir el
documento?", "Constancias/Reportes", 0, 0, Pregunta);
    if(Respuesta == 0)
        if(Interfaz.equals("Alumnos por Sexo") || Interfaz.equals("Alumnos por Grado") ||
Interfaz.equals("Alumnos por Año Escolar")){
            CopiarDatosEnReporte(CrearReporteAlumnos());
        }
    else
        if (Interfaz.equals("Empleados del Plantel"))
            CopiarDatosEnReporte(CrearReporteEmpleadosDelPlantel());
        else
            if (Interfaz.equals("Ingresos Diarios"))
                CopiarDatosEnReporte(CrearReporteIngresos("Ingresos del " +
DatoBusqueda1.getSelectedItem() + "/" + (DatoBusqueda2.getSelectedIndex() + 1) +
"/" + DatoBusqueda3.getText()));
            else
                if (Interfaz.equals("Ingresos Mensuales"))
                    CopiarDatosEnReporte(CrearReporteIngresos("Ingresos del mes de " +
DatoBusqueda1.getSelectedItem() + " de " + DatoBusqueda3.getText()));
                else
                    if (Interfaz.equals("Ingresos Anuales"))
                        CopiarDatosEnReporte(CrearReporteIngresos("Ingresos del Año " +
DatoBusqueda3.getText()));
                    else
                        if (Interfaz.equals("Egresos Diarios"))
                            CopiarDatosEnReporte(CrearReporteEgresos("Egresos del " +
DatoBusqueda1.getSelectedItem() + "/" + (DatoBusqueda2.getSelectedIndex() + 1) +
"/" + DatoBusqueda3.getText()));

```

```

else
if (Interfaz.equals("Egresos Mensuales"))
    CopiarDatosEnReporte(CrearReporteEgresos("Egresos del mes de " +
DatoBusqueda1.getSelectedItem() + " de " + DatoBusqueda3.getText()));
else
if (Interfaz.equals("Egresos Anuales"))
    CopiarDatosEnReporte(CrearReporteEgresos("Egresos del Año " +
DatoBusqueda3.getText()));
else
if (Interfaz.equals("Constancia de Inscripción"))
    CopiarDatosEnReporte(CrearConstanciaDeInscripción());
else
if (Interfaz.equals("Constancia de Estudios")){
    ProcesarIngresosPorConstancias();
    CopiarDatosEnReporte(CrearConstanciaDeEstudios());
}
else
if (Interfaz.equals("Boletín Informativo")){
    ProcesarIngresosPorConstancias();
    CopiarDatosEnReporte(CrearBoletínInformativo());
}
}
else
JOptionPane.showMessageDialog
    (null,
    "No se encontraron datos para realizar el Reporte o Constancia requerida",
    "Error",
    0);
}...}

```

### **5.3. Prueba**

En este flujo de trabajo se verifica el resultado de la implementación probando la construcción del sistema.

Se planifican las pruebas necesarias por medio de un diseño, creando los casos de prueba que especifican lo que se va a examinar por medio de procedimientos.

Se realizan las distintas pruebas y se manejan los resultados sistemáticamente.

En el sistema SAPSEE se procedió a realizar la codificación de cada componente en el lenguaje seleccionado, debiendo realizar las pruebas necesarias con el objetivo de solventar los posibles errores antes de poner en funcionamiento el sistema.

Se seleccionó la prueba de caja negra la cual está ajustada a los requerimientos del sistema, es decir, ayuda a discernir conjunto de condiciones de entrada que satisfagan correctamente la ejecución de los casos de uso implementados.

#### **5.3.1. Prueba del Caso de Uso Procesar Alumnos**

Para este caso de uso se procedió a realizar las pruebas de los componentes que sirven de entrada de datos al sistema con relación a los datos de los alumnos durante el proceso de inscripción.

Lo ideal es obtener resultados precisos y de medidas exactas, conforme a los requerimientos iniciales del sistema. Las clases de equivalencia obtenidas se especifican en la Figura 5.30.

	Clases de Equivalencia	Válido	No Válido
1	Si Cedula > 0 caracteres y Cedula <= 11 caracteres	X	
2	Si Cedula = 0 caracteres ó Cedula > 11 caracteres		X
3	Si Cedula posee caracteres numéricos	X	
4	Si Cedula posee números y caracteres especiales		X
5	Si DiaNacimiento >= "1" y DiaNacimiento <= "31"	X	
6	Si DiaNacimiento = ""		X
7	Si MesNacimiento >= "1" y MesNacimiento <= "12"	X	
8	Si MesNacimiento = ""		X
9	Si AñoNacimiento = un valor de la lista	X	
10	Si AñoNacimiento = ""		X
11	Si Etapa = "Primera" ó Etapa = "Segunda"	X	
12	Si Etapa = ""		X
13	Si Grado >= "1" y Grado <= "6"	X	
14	Si Grado = ""		X
15	Si Motivo = un valor de la lista	X	
16	Si Motivo = ""		X
17	Si Seccion >= "A" y Seccion <= "Z"	X	
18	Si Seccion = ""		X
19	Si Nacionalidad = "V" ó Nacionalidad = "E"	X	
20	Si Nacionalidad = ""		X
21	Si Sexo = "F" ó Sexo = "M"	X	
22	Si Sexo = ""		X
23	Si Telefono > 0 caracteres y Telefono <= 50 caracteres	X	
24	Si Telefono > 50		X

Figura 5.30. Clases de Equivalencia del Caso de Uso Procesar Alumnos.

La Figura 5.31 muestra la prueba de caja negra del caso de uso Procesar Alumnos.

<b>Caso de Prueba</b>	<b>Entrada</b>	<b>Clases cubiertas</b>
Cedula = "19813689716"	Válida	1,3
Cedula = "12457856217%,:"	No válida	2,4
DiaNacimiento = "20"	Válida	5
DiaNacimiento = ""	No válida	6
MesNacimiento = ""	No válida	8
AñoNacimiento = "1998"	Válida	9
AñoNacimiento = ""	No válida	10
Etapa = ""	No válida	12
Grado = ""	No válida	14
Seccion = ""	No válida	18
Etapa = "Primera"	Válida	11
Grado = "3"	Válida	13
Seccion = "U"	Válida	17
MesNacimiento = "12"	Válida	7
Motivo = "Culminación"	Válida	15
Nacionalidad = ""	No válida	20
Sexo = ""	No válida	22
Teléfono = "02812764632"	Válida	23

Figura 5.31. Prueba de Caja Negra del Caso de Uso Procesar Alumnos.

### 5.2.2. Prueba del Caso de Uso Ingresos y Egresos

La Figura 5.32 muestra las clases de equivalencia del caso de uso Ingresos y Egresos.

La figura 5.33 muestra la prueba de caja negra del caso de uso Ingresos y Egresos.

	<b>Clases de Equivalencia</b>	<b>Válido</b>	<b>No Válido</b>
1	Si Monto = expresión entera ó Monto = expresión decimal	X	
2	Si Monto = 0 caracteres ó Monto posee caracteres no numéricos ó Monto # expresión decimal		X
3	Si TipoDePago = "Efectivo" ó TipoDePago = "Cheque"	X	
4	Si TipoDePago = ""		X
5	Si Concepto = un valor de la lista	X	
6	Si Concepto = ""		X
7	Si NroCheque >= 1 caracter y NroCheque <= 12 caracteres	X	
8	Si NroCheque = 0 caracter ó NroCheque > 12 caracteres		X
9	Si NroCheque posee caracteres numéricos	X	
10	Si NroCheque posee caracteres numéricos y no numéricos		X
11	Si NroCheque posee caracteres no numéricos		X
12	Si NroCuenta >= 1 caracter y NroCuenta <= 20 caracteres	X	
13	Si NroCuenta = 0 caracteres ó NroCuenta > 20 caracteres		X
14	Si NroCuenta posee caracteres numéricos	X	
15	Si NroCuenta posee caracteres numéricos y no numéricos		X
16	Si NroCuenta posee caracteres no numéricos		X
17	Si Banco >= 1 caracter y Banco <= 100 caracteres	X	
18	Si Banco > 100 caracteres ó Banco = ""		X

**Figura 5.32. Clases de Equivalencia del Caso de Uso Ingresos y Egresos.**

Para el resto de los casos de uso del sistema no se realizaron clases de equivalencias debido a que se consideró que las clases de equivalencias realizadas anteriormente servirían para hacer las pruebas faltantes al sistema. Se realizaron casos de pruebas de los restantes casos de uso y los resultados obtenidos fueron satisfactorios.

Caso de Prueba	Entrada	Clases cubiertas
Monto = "123,20"	Válida	1
TipoDePago = "Efectivo"	Válida	3
Concepto = ""	No válida	6
NroCheque = "12345hy5\$"	No válida	10
NroCuenta = ""	No válida	13
NroCuenta = "1254712ghs32"	No válida	15
Banco = "Venezuela"	Válida	17
Monto = "12458.fret3"	No válida	2
TipoDePago = ""	No válida	4
Concepto = "1era Quincena Diciembre Año Escolar 2008-2009"	Válida	14
NroCheque = "84012478541"	Válida	7,9
NroCuenta = "00418745971230457841"	Válida	12,14
Banco = ""	No válida	18

Figura 5.33. Prueba de Caja Negra del Caso de Uso Ingresos y Egresos.

#### 5.4. Planificación de las Sigüientes Fases.

1. Fase de Transición: Durante esta fase se implementará el software de instalación del sistema, así como la construcción del manual de usuario y la arquitectura final.

#### 5.5. Resumen de la Fase de Construcción

Al concluir la tercera fase del proceso unificado se obtuvo:

1. Implementación de los casos de uso del sistema SAPSEE.
2. Prueba de SAPSEE por medio de clases de equivalencia y caja negra.

3. Planificación de las siguientes fases.
4. Resumen de la Fase de Construcción.



# **CAPÍTULO 6**

## **FASE DE TRANSICIÓN**

### **6.1. Introducción**

En esta fase de desarrollo el sistema ha alcanzado la capacidad operativa inicial. Se ha culminado el estudio completo del sistema por lo que no se hace énfasis en los flujos de trabajo principales. Además el producto obtenido (software ) se le entregará a los usuarios para que sean ellos mismos quienes se encarguen de hacer la implementación del sistema y realizar las pruebas pertinentes.

En este momento se ha cumplido con los requisitos establecidos en las fases anteriores, hasta la satisfacción de todos los usuarios. Se ha determinado que el sistema hace lo que se le demanda ya que no se encontraron fallas, ni se descubrieron riesgos inesperados cuando se realizaron las pruebas por medio de las clases de equivalencias y caja negra. En fin, no fueron encontrados problemas que ameritaran cambios de ningún tipo en el diseño del sistema.

### **6.2. Resumen de la Fase de Transición**

Al finalizar la cuarta y última fase del proceso unificado se obtuvo:

1. El producto o software ejecutable.
2. La versión completa y corregida de la línea base de la versión del software, incluyendo todos los modelos del sistema.
3. Manuales y material de información para el usuario.

# CAPÍTULO 7

## MANUAL DE USUARIO

### 7.1. Introducción

Este manual le guiará de una forma breve para el uso del sistema SAPSEE.

La descripción se hará por medio de los menús y submenús que se encuentran en la interfaz principal.

### 7.2. Interfaz Principal

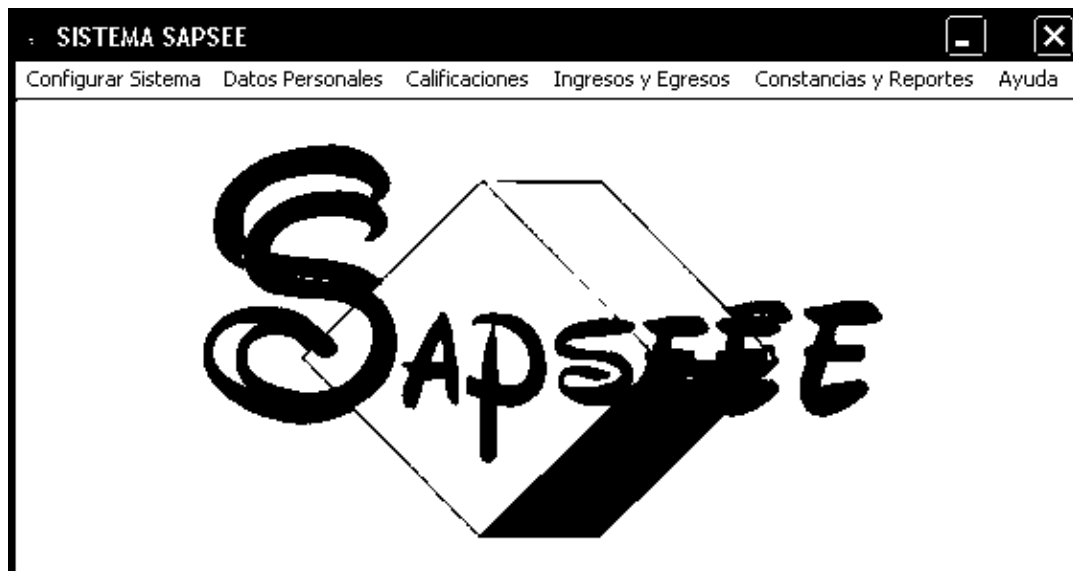
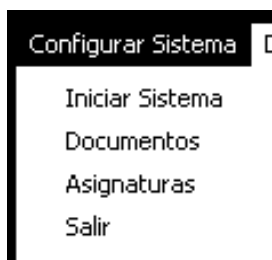


Figura 7.1. Interfaz Principal del Sistema SAPSEE.

#### 7.2.1. Menú Configurar Sistema

Este menú se usa para configurar el sistema: ingresando los datos del plantel, las asignaturas dictadas en el plantel y los datos de los documentos que se emiten.



**Figura 7.2. Menú para Configurar el Sistema.**

Para configurar los datos del plantel, active el renglón Iniciar Sistema del menú Configurar Sistema (Figura 7.2) dando clic sobre el y obteniendo la siguiente pantalla:

 A screenshot of a software window titled "Configurar Sistema - Iniciar Plantel". The window contains several input fields for configuration: "Cód. D.E.A.", "RIF", "NIT", "Nombre", "Dirección", and "Teléfono". Below the input fields is a row of seven icons: a book, a person with a magnifying glass, a person with a checkmark, a person with a checkmark, two people, a person with a checkmark, and a person with a checkmark.

**Figura 7.3. Interfaz para Iniciar el Sistema.**

1. Presione el primer botón con el cual se guardarán los datos del plantel, siempre que los datos estén copiados en la interfaz.
2. Busque los datos en el sistema por medio del segundo botón, pero antes, dichos datos deben estar registrados en el sistema.
3. Modifique los datos del plantel, pero primero realice el paso anterior; luego efectúe las modificaciones respectivas y presione el tercer botón. El

sistema deberá guardar toda la información modificada en la base de datos respectiva.

4. Elimine los datos del plantel, pero primero realice el paso 2 y luego presionar el cuarto botón, el sistema deberá eliminar toda la información de la base de datos respectiva.
5. Ingrese el Personal Directivo del plantel presionando el quinto botón, se abrirá una nueva pantalla para el ingreso de personal directivo según Figura 7.17, cuyo proceso se explica en la sección 7.4.
6. Limpie todos los campos de la interfaz presionando el sexto botón.
7. Salga de la Interfaz de Inicio de Sistema presionando el séptimo botón.

Para configurar los datos de los documentos emitidos por el plantel, active el renglón Documentos del menú Configurar Sistema (Figura 7.2) dando clic sobre el y obteniendo una pantalla como la Figura 7.4. El único dato a configurar es el costo del documento en Bs., el cual debe ser escrito en decimales o enteros, ejemplos: 12,50 ó 12.



**Figura 7.4. Interfaz para Configurar Documentos.**

1. Guarde los datos presionando el primer botón. Inmediatamente el sistema guarda la información en la base de datos.
2. Inicie el proceso presionando el botón número dos.
3. Salga de la interfaz presionando el botón número tres.

Para configurar los datos de los asignaturas dictadas por el plantel, active el renglón Asignaturas del menú Configurar Sistema (Figura 7.2) dando clic sobre el y obteniendo la interfaz según Figura 7.5. El único dato a configurar es el nombre de la asignatura.



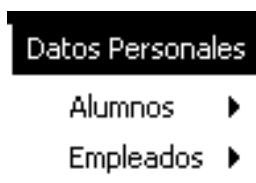
**Figura 7.5. Interfaz para Configurar Asignaturas.**

1. Guarde los datos de las asignaturas presionando el primer botón.
2. Cree un nuevo código de asignatura presionando el botón con el signo “+” y bajar hasta el código más pequeño presionando el botón “-“. Los códigos comienzan desde el número 1.
3. Modifique los datos de las asignaturas presionando el segundo botón pero antes dichos datos deben estar registrados en el sistema.
4. Elimine los datos de las asignaturas presionando el segundo botón, pero antes, dichos datos deben estar registrados en el sistema.
5. Inicie el proceso presionando el cuarto botón. Esto limpiará todos los campos de la interfaz para que usted vuelva a escribirlos.
6. Cierre la interfaz, presionando el quinto botón.

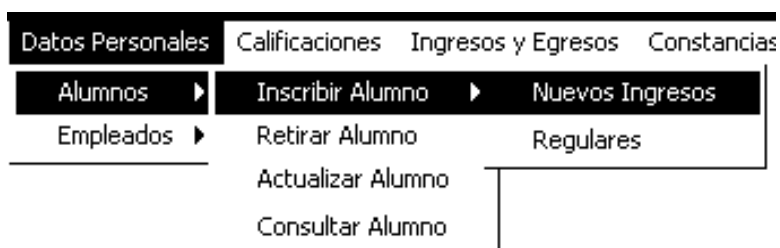
### **7.2.2. Menú Datos Personales**

Este menú se usa para ingresar, eliminar, actualizar y consultar los datos de los alumnos y empleados del plantel (Figura 7.6).

Para ingresar los datos de los alumnos, active el renglón Alumnos del menú Datos Personales (Figura 7.6) y se ubica en el submenú Ingresar Alumno y luego en el submenú Nuevos Ingresos según Figura 7.7; finalmente obtiene una interfaz como la Figura 7.8.



**Figura 7.6. Menú Datos Personales**



**Figura 7.7. Submenú Inscribir Alumno.**

1. Presione el primer botón para guardar los datos que ya ha escrito en la interfaz.
2. El sistema automáticamente guarda toda la información del alumno en la base de datos.
3. Invoque la interfaz para guardar los datos de los padres y representantes al presionar el segundo botón (Figura 7.9).
4. Invoque la interfaz para guardar los datos de pago de matrícula al presionar el tercer botón (Figura 7.30). La explicación se hace en la sección 7.6 (Menú Ingresos y Egresos).
5. Limpie los valores de la interfaz presionando el cuarto botón. Esto le permitirá escribir los datos nuevamente en la interfaz.
6. Salga de la interfaz presionando el quinto botón.

Figura 7.8. Interfaz para Ingresar Alumnos Nuevos.

Figura 7.9. Interfaz para Ingresar datos de los Padres y Representantes.

Para consultar los datos de los alumnos, active el renglón Alumnos del menú Datos Personales (Figura 7.6) y se ubica en el submenú Consultar Alumno según Figura 7.10; finalmente obtendrá una interfaz como la Figura 7.11.



Figura 7.10. Submenú Consultar Alumno.

Figura 7.11. Interfaz para Consultar los datos de los alumnos.

1. Ingresar el número de cédula escolar que identifica al alumno a consultar.
2. Presionar el primer botón y seguidamente el sistema muestra los datos del alumno consultado.
3. Limpiar los valores de la interfaz presionando el segundo botón.
4. Salir de la interfaz presionando el tercer botón.

Para retirar a los alumnos del plantel, activa el renglón Alumnos del menú Datos Personales (Figura 7.6) y se ubica en el submenú Retirar Alumno según Figura 7.12; finalmente obtendrá una interfaz como la figura 7.13.



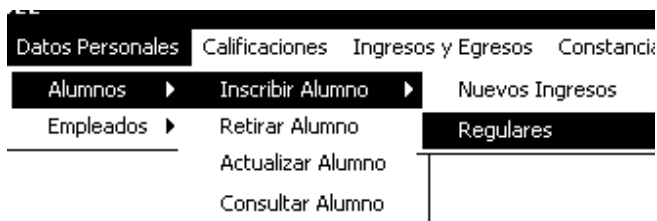


Figura 7.12. Submenú Retirar Alumno.

Figura 7.13. Interfaz para Retirar Alumnos del plantel.

1. Ingresa la cédula del alumno en la interfaz, luego presiona el primer botón al hacerlo el sistema busca los datos y los presenta en pantalla.
2. Confirme los datos del alumno en pantalla y presiona el segundo botón. El alumno será eliminado si está registrado en el sistema.
3. Limpie los valores de la interfaz presionando el tercer botón.
4. Salga de la interfaz presionando el cuarto botón.

Para realizar la prosecución de los alumnos (continuidad de estudio), active el renglón Alumnos del menú Datos personales (Figura 7.6), se ubica en el submenú Ingresar Alumno y seguidamente en el submenú Regulares (Figura 7.14).



**Figura 7.14. Submenú Regulares.**

Seguidamente se obtiene una interfaz como la siguiente:

 A screenshot of a software window titled 'Inscripción de Alumnos Regulares 2008-2009'. The window contains a form for student registration. At the top right, it says 'Fecha Inscripción: 4/12/2008'. The form fields include: 'Cédula:', 'Apellidos:', 'Escolaridad:', 'Nombres:', 'Etapa:', 'Grado:', and 'Sección: A'. Below the form are five icons: a book, a person with a gear, a globe, a refresh symbol, and a right-pointing arrow.

**Figura 7.15. Interfaz para Inscribir Alumnos Regulares.**

1. Ingrese la cédula escolar del alumno en la interfaz, presiona el segundo botón para que el sistema busque los datos y los presente en pantalla. Si el alumno tiene deudas presentes se emite el mensaje: “El alumno tiene deudas pendientes” sino se verifica que el alumno no esté inscrito en el año escolar presente, si lo está entonces se emite el mensaje: “El alumno ya está inscrito en el año escolar presente” sino se procede a inscribir al alumno en un nuevo año escolar.
2. Ingrese los pagos correspondientes presionando el segundo botón para que se active una interfaz como la Figura 7.30. La explicación está en la sección 7.6 (Menú Ingresos y Egresos).
3. Limpie los datos de la interfaz presionando el cuarto botón.
4. Salga de la interfaz presionando el quinto botón.

Para ingresar los datos de los empleados, active el renglón Empleados del menú Datos Personales (Figura 7.6.) y se ubica en el submenú Ingresar Empleados según Figura 7.16 para obtener una interfaz como la Figura 7.17.

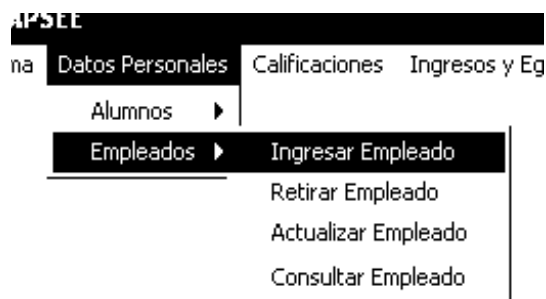


Figura 7.16. Submenú Ingresar Empleado.

 A screenshot of a web-based form titled 'Ingresar Empleado para Año Escolar 2008-2009'. The form is divided into two main sections: 'Datos Personales' and 'Datos Laborales'. 
   
 In the 'Datos Personales' section, there are several input fields:
 

- 'Cédula:': A dropdown menu with 'V' selected and an adjacent text box.
- 'Apellidos:': A text box.
- 'Fecha de Nacimiento:': Three dropdown menus for day (01), month (01), and year (1900).
- 'Nombres:': A text box.
- 'Estado Civil:': A dropdown menu with 'Soltero(a)' selected.
- 'Profesión:': A text box.
- 'Sexo:': A dropdown menu with 'F' selected.
- 'Dirección:': A large text box.
- 'Teléfono:': A text box.

 In the 'Datos Laborales' section, there are three dropdown menus:
 

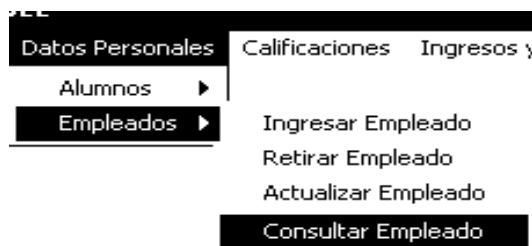
- 'Cargo:': Set to 'Docente'.
- 'Grado:': Set to '1'.
- 'Sección:': Set to 'A'.

 At the bottom right of the form, there are three buttons: a save button (floppy disk icon), a clear button (eraser icon), and a back button (arrow icon).

Figura 7.17. Interfaz para Ingresar Empleados.

1. Ingrese todos los datos del empleado en la interfaz, al finalizar presiona el primer botón para guardar los datos del empleado en el sistema. El sistema debe almacenar toda la información en la base de datos respectiva.
2. Limpie todos los datos de la interfaz para iniciar el proceso nuevamente presionando el segundo botón.
3. Salga de la interfaz presionando el tercer botón.

Para consultar los datos de los empleados, active el renglón Empleados del menú Datos Personales (Figura 7.6.) y se ubica en el submenú Consultar Empleado según Figura 7.18; finalmente obtendrá una interfaz como la Figura 7.19.



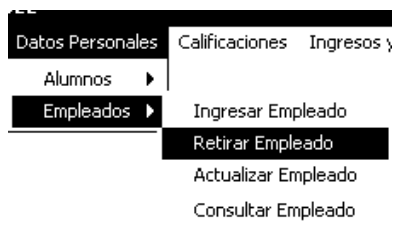
**Figura 7.18. Submenú Consultar Empleado.**

 A screenshot of a software window titled 'Consultar Empleado'. The window is divided into three sections: 'Datos Personales', 'Datos Laborales', and 'Datos de Retiro'. 
 - 'Datos Personales' includes fields for 'Cédula:' (with a dropdown arrow), 'Fecha de Nacimiento:' (with three dropdown arrows), 'Estado Civil:' (with a dropdown arrow), 'Sexo:' (with a dropdown arrow), 'Dirección:', 'Teléfono:', 'Apellidos:', 'Nombres:', and 'Profesión:'.
 - 'Datos Laborales' includes a 'Cargo:' dropdown menu.
 - 'Datos de Retiro' includes a 'Motivo:' dropdown menu and a 'Fecha Retiro:' text field.
 At the bottom right of the window, there are three icons: a magnifying glass (search), a refresh symbol, and a right-pointing arrow (next).

**Figura 7.19. Interfaz para Consultar los datos de los Alumnos.**

1. Ingresar la cédula del alumno y presionar el primer botón para que el sistema busque los datos y los muestre en pantalla.
2. Limpie todos los datos de la interfaz presionando el segundo botón.
3. Salga de la interfaz presionando el tercer botón.

Para retirar a los empleados del plantel, active el renglón Empleados del menú Datos Personales (Figura 7.6.) y se ubica en el submenú Retirar Empleado según Figura 7.20; finalmente obtendrá una interfaz como la Figura 7.21.



**Figura 7.20. Submenú Retirar Empleado.**

 A screenshot of a web application window titled 'Retirar Empleado'. It contains three sections:
 

- Datos Personales:** Fields for Cédula (with a dropdown), Apellidos, Fecha de Nacimiento (with three dropdowns), Nombres, Estado Civil (with a dropdown), Profesión, Sexo (with a dropdown), Dirección, and Teléfono.
- Datos Laborales:** A dropdown field for Cargo.
- Datos de Retiro:** A dropdown for Motivo (set to 'Culminación de Contrato') and a text field for Fecha Retiro (set to '3/11/2008').

 At the bottom right, there are four icons: a magnifying glass, a pencil, a refresh symbol, and a close button.

**Figura 7.21. Interfaz para Retirar Empleados del Plantel.**

1. Ingrese la cédula del empleado, presione el primer botón y el sistema busca los datos y los presenta en pantalla.
2. Presione el segundo botón para que el sistema retire al empleado del plantel. Los datos del empleado deben estar registrados.
3. Presione el tercer botón para limpiar todos los datos de la interfaz.
4. Presione el cuarto botón y se cerrará la interfaz.

Para actualizar los datos de los empleados del plantel, active el renglón Empleados del menú Datos Personales (Figura 7.6.) y se ubica en el submenú Actualizar Empleado según Figura 7.22; finalmente obtendrá una interfaz como la Figura 7.23.

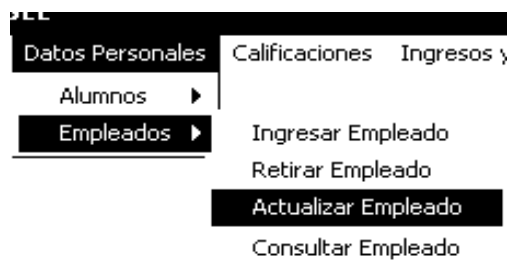


Figura 7.22. Submenú Actualizar Empleado

 A screenshot of a web application window titled 'Modificar Empleado'. The window contains three main sections:
 

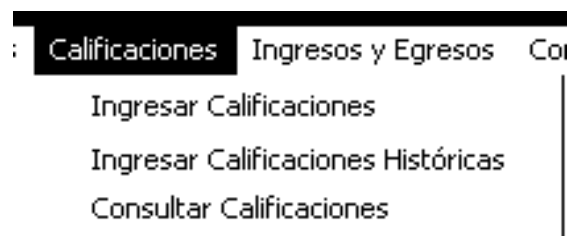
- Datos Personales:** Includes fields for 'Cédula:' (with a dropdown arrow), 'Apellidos:', 'Fecha de Nacimiento:' (with three date dropdowns), 'Nombres:', 'Estado Civil:' (with a dropdown arrow), 'Profesión:', 'Sexo:' (with a dropdown arrow), 'Dirección:', and 'Teléfono:'.
- Datos Laborales:** Includes a 'Cargo:' dropdown menu.
- Datos de Retiro:** Includes a 'Motivo:' dropdown menu, a 'Fecha Retiro:' text field, and four icons: a magnifying glass, a trash can, a refresh symbol, and a close button.

Figura 7.23. Interfaz para Actualizar los datos de los Empleados.

1. Ingrese la cédula del empleado, presione el primer botón y el sistema buscará los datos y los presentará en pantalla.
2. Realice las modificaciones requeridas, presione el segundo botón y el sistema guardará las actualizaciones de los datos en la base de datos correspondiente.
3. Presione el tercer botón para limpiar todos los datos de la interfaz e iniciar el proceso nuevamente para un nuevo empleado.
4. Presione el cuarto botón y se cerrará la interfaz.

### 7.2.3. Menú Calificaciones

Este menú se usa para ingresar y consultar las calificaciones de los alumnos (Figura 7.24).



**Figura 7.24. Menú Calificaciones**

Para ingresar las calificaciones de los alumnos, active el renglón Ingresar Calificaciones (Figura 7.24) y obtendrá una interfaz como la Figura 7.25. Las actividades para este proceso son las siguientes:

1. Ingrese la cédula escolar del alumno y presione el primer botón para que el sistema busque los datos y los muestre en pantalla. Paso seguido debe elegir el Lapso de estudio en el campo Lapso para que el sistema presente las asignaturas dictadas por la institución en el campo Asignatura, finalmente ingrese las calificaciones en el campo Nota seleccionando la calificación obtenida por el alumno, desde “A” hasta “E”. Cuando lo haga se guardará automáticamente por el sistema en la base de datos.
2. Presione el segundo botón y se limpiarán todos los datos de la interfaz.
3. Presione el tercer botón y se cerrará la interfaz.

Para ingresar las calificaciones de años anteriores, active el renglón Ingresar Calificaciones Históricas (Figura 7.24) y finalmente obtendrá una interfaz como la Figura 7.26.

Figura 7.25. Interfaz para Ingresar las Calificaciones de los alumnos.

Figura 7.26. Interfaz para Ingresar Calificaciones Históricas.

Actividades para el ingreso de calificaciones históricas:

1. Ingrese la cédula escolar del alumno y presione el primer botón para que el sistema busque los datos y los presente en pantalla.



2. Elija el lapso de estudio usando el campo Lapso y el año escolar usando los botones con los signos “+” ó “-”. Inicialmente aparecerá el año escolar que se está cursando actualmente. Al seleccionar el Lapso aparecerán las asignaturas dictadas por el plantel; seleccione la calificación del alumno en el campo Nota para que automáticamente se registre en el sistema.
3. Presione el segundo botón y se limpiarán todos los datos de la interfaz.
4. Presione el tercer botón y se cerrará la interfaz.

Para consultar las calificaciones, active el renglón Consultar Calificaciones (Figura 7.24) y obtendrá una interfaz como la Figura 7.27.

**Figura 7.27. Interfaz para Consultar Calificaciones.**

1. Ingrese la cédula escolar del alumno y presione el primer botón para que el sistema busque los datos y los presente en pantalla. Elija el lapso de estudio usando el campo Lapso y el año escolar usando los botones con los signos “+” ó “-”. Inicialmente aparecerá el año escolar que se está cursando actualmente. Al seleccionar el Lapso aparecerán las asignaturas registradas para el alumno con su respectiva calificación en el campo Nota.

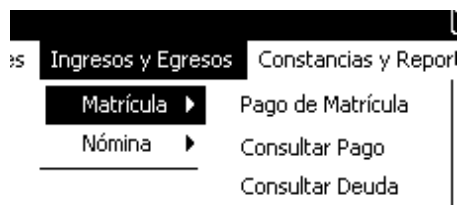
2. Presione el segundo botón y se limpiarán todos los datos de la interfaz.
3. Presione el tercer botón y se cerrará la interfaz.

#### 7.2.4. Menú Ingresos y Egresos.



**Figura 7.28. Menú Ingresos y Egresos.**

Este menú se usa para ingresar y consultar los ingresos y egresos del plantel (Figura 7.28). Para registrar los ingresos del plantel, elija el submenú Matrícula y el renglón Pago de Matrícula (Figura 7.29); finalmente obtendrá una interfaz como la Figura 7.30.



**Figura 7.29. Submenú Matrícula.**

1. Ingrese la cédula escolar y presione el segundo botón para que el sistema busque los datos y los presente en pantalla. Seguidamente coloque los datos que restan en la interfaz según la figura anteriormente indicada. Presione el primer botón para que el sistema guarde la información del pago.
2. Use el tercer botón para limpiar los campos de la interfaz y reiniciar el proceso.

- Use el cuarto botón para cerrar la interfaz y realizar otro proceso.

Inicialmente el sistema presentará la fecha de pago, el año escolar de pago (usted puede cambiarlo usando los botones con el signo “+” y “-”) y el tipo de pago: “Efectivo”.

**Figura 7.30. Interfaz para Ingresar Pagos de Matrícula.**

Para consultar los ingresos del plantel, elija el renglón Consultar Pago del submenú Matrícula (Figura 7.29) y obtendrá una interfaz como la Figura 7.31.

Actividades para consultar los ingresos del plantel:

- Ingrese el número de recibo y presione el primer botón para que el sistema busque los datos del pago y los presente en pantalla.
- Limpie los valores de los campos usando el segundo botón.
- Cierre la interfaz usando el tercer botón.

Figura 7.31. Interfaz para Consultar pagos de Matrícula.

Para consultar la deuda pendiente de los alumnos, elija el renglón Consultar Deuda del submenú Matrícula (Figura 7.29) y obtendrá una interfaz como la Figura 7.32.

Figura 7.32. Interfaz para Consultar Deudas Pendientes.

1. Ingrese la cédula escolar y presione el primer botón; el sistema buscará los datos escolares del alumno para presentarlos en pantalla.
2. Limpie los valores de los campos usando el segundo botón.
3. Cierre la interfaz usando el tercer botón.

Para registrar los egresos del plantel, elija el submenú Nómina y el renglón Pago de Nómina (Figura 7.33) y obtendrá una interfaz como la Figura 7.34.

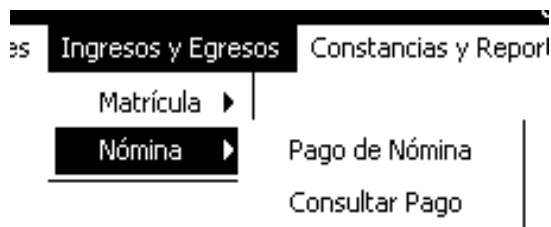


Figura 7.33. Submenú Ingresos y Egresos.

 A screenshot of a web application window titled 'Pago de Nómina / Año Escolar 2008-2009'. The window contains two main sections: 'Datos Personales' and 'Datos de Pago'. 
   
 In the 'Datos Personales' section, there are input fields for 'Cédula:', 'Apellidos:', 'Nombres:', and 'Cargo:'. To the right, 'Recibo Nro:' is displayed as 'E-1'.
   
 In the 'Datos de Pago' section, there are input fields for 'Monto:', 'Nro Cheque:', and 'Nro Cuenta:'. There is also a 'Fecha de Pago:' field with the value '4/11/2008'. A 'Concepto de Pago:' dropdown menu is set to a value with a dot. A 'Banco:' field is also present.
   
 At the bottom of the form, there are four icons: a folder, a magnifying glass, a refresh symbol, and a right-pointing arrow.

Figura 7.34. Interfaz para Registrar Pagos de Nómina.

1. Ingrese la cédula del empleado, presione el primer botón y el sistema buscará los datos laborales del empleado para presentarlos en pantalla, coloque los datos restantes en la interfaz según la figura indicada y presione el primer botón para que el sistema guarde la información.
2. Limpie los valores de los campos usando el segundo botón y así podrá comenzar el proceso con otro empleado del plantel o con el mismo en cuestión.

3. Cierre la interfaz usando el tercer botón.

Para consultar los egresos del plantel, el usuario elige el submenú Nómina y el renglón Consultar Pago (Figura 7.33) y obtiene una interfaz como la Figura 7.35.

1. Ingrese el número de recibo, presione el primer botón y el sistema buscará los datos del pago para presentarlos en pantalla.
2. Limpie los valores de los campos usando el segundo botón.
3. Cierre la interfaz usando el tercer botón.

Figura 7.35. Interfaz para la Consulta de Pagos de Nómina.

### 7.2.5. Menú Constancias y Reportes

Este menú se usa para generar constancias y reportes acerca de todas las actividades realizadas por el plantel, tanto académicas como administrativas (Figura 7.36.)



Figura 7.36. Menú Constancias y Reporte.

Para emitir una constancia de inscripción, elija el submenú Constancias de Inscripción del renglón Constancias (Figura 7.37) y finalmente obtendrá una interfaz como la Figura 7.38.



Figura 7.37. Submenú Constancias.

 A screenshot of a web application window titled 'Constancia de Inscripción - Año Escolar 2008-2009'. The window contains two main sections:
 

- Datos para Búsqueda:** A single text input field labeled 'Cédula:'.
- Datos Registrados en el Sistema:** A form with several fields:
  - Grado: [ ] Sección: [ ] Etapa: [ ]
  - Alumno: [ ]
  - L. de Nac. [ ]
  - F. de Nac. [ ]
  - Escolaridad: [ ] Fecha de Inscripción [ ]

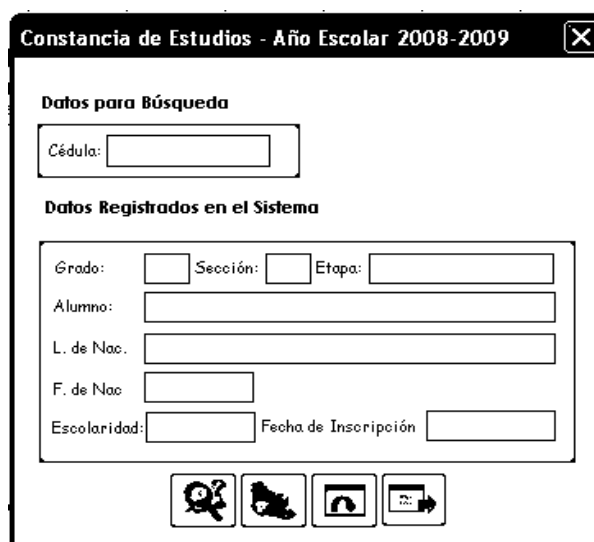
 At the bottom of the form are four icons: a person, a map, a magnifying glass, and a document with an arrow.

Figura 7.38. Interfaz para generar Constancias de Inscripción.

Actividades para generar una constancia de inscripción:

1. Ingrese la cédula escolar, presione el primer botón y el sistema buscará los datos escolares del alumno y los presentará en pantalla.
2. Presione el segundo botón y el sistema generará una constancia de inscripción para el alumno.
3. Inicie el proceso presionando el tercer botón
4. Salga de la interfaz presionando el cuarto botón.

Para emitir una constancia de estudios, elija el submenú Constancias de Estudios del renglón Constancias (Figura 7.37) y finalmente obtendrá una interfaz como la Figura 7.39.



**Figura 7.39. Interfaz para generar Constancias de Estudios.**

1. Ingrese la cédula escolar, presione el primer botón y el sistema buscará los datos escolares del alumno y los presentará en pantalla.
2. Presione el segundo botón y el sistema generará una constancia de estudios para el alumno.
3. Inicie el proceso presionando el tercer botón
4. Salga de la interfaz presionando el cuarto botón.



Para emitir un boletín informativo, elija el submenú Boletín Informativo del renglón Constancias (Figura 7.37) y finalmente obtendrá una interfaz como la Figura 7.40.

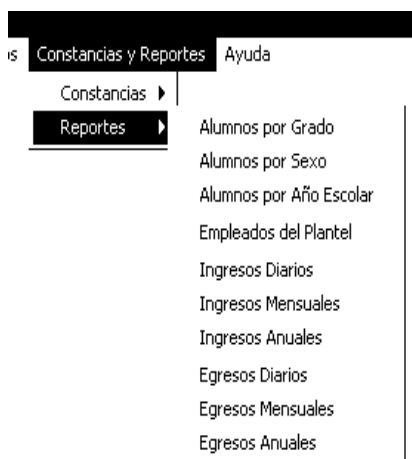
**Figura 7.40. Interfaz para generar Boletines Informativos.**

1. Ingrese la cédula escolar, presione el primer botón y el sistema buscará los datos escolares del alumno y los presentará en pantalla.
2. Seleccione el lapso para la emisión del boletín.
3. Presione el segundo botón y el sistema generará el boletín informativo para el alumno.
4. Inicie el proceso presionando el tercer botón.
5. Salga de la interfaz presionando el cuarto botón.

Para generar el reporte de alumnos por grado, elija el submenú Alumnos por Grado del renglón Reportes (Figura 7.41) y obtendrá una interfaz como la Figura 7.42.

Actividades para este proceso:

1. Seleccione el grado, sección y año escolar de los alumnos, presione el primer botón y el sistema buscará los datos para presentarlos en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte de los alumnos por grado.
3. Presione el tercer botón y se iniciará el proceso nuevamente limpiando los valores de la interfaz.
4. Salga de la interfaz presionando el cuarto y último botón.



**Figura 7.41. Submenú Reportes.**

Para generar el reporte de alumnos por sexo, elija el submenú Alumnos por Sexo del renglón Reportes (Figura 7.41) y obtendrá una interfaz como la Figura 7.43.

Actividades para generar reportes de alumnos por sexo:

1. Seleccione el sexo de los alumnos y el año escolar, presiona el primer botón y el sistema buscará los datos para presentarlos en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte solicitado.
3. Presione el tercer botón y limpiará todos los datos de la interfaz para iniciar el proceso.

4. Presione el cuarto botón y se cerrará la interfaz.



The screenshot shows a window titled "Reporte de Alumnos por Grado". It contains two main sections: "Datos para Búsqueda" and "Datos Registrados en el Sistema".

**Datos para Búsqueda:** This section includes three input fields: "Grado:" with a dropdown menu showing "1", "Sección:" with a dropdown menu showing "A", and "Año Escolar:" with a text box containing "2008-2009" and small "+" and "-" buttons.

**Datos Registrados en el Sistema:** This section includes several input fields: "N°:" with a dropdown menu, "Sexo:" with a text box, "Cédula:" with a text box, "Apellidos:" with a text box, "Nombres:" with a text box, "Escolaridad:" with a text box, and "Fecha de Nacimiento:" with a text box.

At the bottom of the window, there are four icons: a magnifying glass, a printer, a refresh symbol, and a close button (X).

Figura 7.42. Interfaz para generar Reportes de Alumnos por Grado.



The screenshot shows a window titled "Reporte de Alumnos por Sexo". It contains two main sections: "Datos para Búsqueda" and "Datos Registrados en el Sistema".

**Datos para Búsqueda:** This section includes two input fields: "Sexo:" with a dropdown menu showing "F" and "Año Escolar:" with a text box containing "2008-2009" and small "+" and "-" buttons.

**Datos Registrados en el Sistema:** This section includes several input fields: "N°:" with a dropdown menu, "Grado:" with a text box, "Sección:" with a text box, "Cédula:" with a text box, "Apellidos:" with a text box, "Nombres:" with a text box, "Escolaridad:" with a text box, and "Fecha de Nacimiento:" with a text box.

At the bottom of the window, there are four icons: a magnifying glass, a printer, a refresh symbol, and a close button (X).

Figura 7.43. Interfaz para generar Reportes de Alumnos por Sexo.

Para generar el reporte de alumnos por año escolar, elija el submenú Alumnos por Año Escolar del renglón Reportes (Figura 7.41) y obtendrá la interfaz solicitada (Figura 7.44).

**Figura 7.44. Interfaz para generar Reportes de Alumnos por Año Escolar.**

1. Escoja el año escolar para el reporte, presione el primer botón y el sistema buscará los datos y los presentará en la interfaz.
2. Para emitir el reporte físico, presione el segundo botón.
3. Si presiona el tercer botón borrará todos los datos de la interfaz y se iniciará de nuevo el proceso.
4. Presione el cuarto botón y cierra la interfaz.

Para generar reportes de los empleados del plantel, elija la opción Empleados del Plantel en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.45).

1. Si existen empleados registrados se presentarán los datos de los empleados en la interfaz sino el sistema emitirá el mensaje: “No existen empleados en el sistema”.
2. Presione el primer botón y se generará el reporte correspondiente.
3. Para salir de la interfaz presione el segundo botón.

**Reporte: Empleados del Plantel 2008-2009**

**Datos para Búsqueda**

Año Escolar: 2008-2009

**Datos Registrados en el Sistema**

Nº:  Sexo:  F. de Nac.:

Cédula:

Apellidos:

Nombres:

Cargo:

**Figura 7.45. Interfaz para generar Reportes de Empleados del Plantel.**

Para generar reportes de los ingresos diarios del plantel, elija la opción Ingresos Diarios en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.46).

**Reporte de Ingresos Diarios**

**Datos para Búsqueda**

Día: 01 Mes: diciembre Año: 2008

**Datos Registrados en el Sistema**



Recibo Nº:  Monto Bs.:

Cédula:

Alumno:

Concepto:

Hecho en:

**Figura 7.46. Interfaz para generar Reportes de Ingresos Diarios.**

1. Seleccione el día, mes y año para el reporte; presione el primer botón y el sistema buscará los datos y los presentará en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte requerido.
3. Si presiona el tercer botón borrará todos los datos de la interfaz.

4. Cierre la interfaz presionando el cuarto botón.

Para generar reportes de los ingresos mensuales del plantel, elija la opción Ingresos Mensuales en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.47).

**Reporte de Ingresos Mensuales**

**Datos para Búsqueda**

Mes: diciembre Año: 2008

**Datos Registrados en el Sistema**

Recibo Nº: Monto Bs.

Cédula:

Alumno:

Concepto:

Hecho en: Fecha de Pago:

**Figura 7.47. Interfaz para generar Reportes de Ingresos Mensuales.**

1. Seleccione el mes y año para el reporte; presione el primer botón y el sistema buscará los datos y los presentará en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte requerido.
3. Inicie el proceso presionando el tercer botón.
4. Cierre la interfaz presionando el cuarto botón.

Para generar reportes de los ingresos anuales del plantel, elija la opción Ingresos Anuales en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.48).

Actividades para generar el reporte de ingresos anuales:

1. Seleccione el año para el reporte; presione el primer botón y el sistema buscará los datos y los presentará en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte requerido.
3. Inicie el proceso presionando el tercer botón.
4. Cierre la interfaz presionando el cuarto botón.

**Figura 7.48. Interfaz para generar Reportes de Ingresos Anuales.**

Para generar reportes de los egresos diarios del plantel, elija la opción Egresos Diarios en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.49).

1. Seleccione día, mes y año para el reporte; presione el primer botón y el sistema buscará los datos para presentarlos en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte requerido.
3. Si presiona el tercer botón borrará todos los datos de la interfaz y se iniciará de nuevo el proceso.
4. Cierre la interfaz presionando el cuarto botón.

**Reporte de Egresos Diarios**

**Datos para Búsqueda**

Día: 01 ▼ Mes: diciembre ▼ Año: 2008 [+/-]

**Datos Registrados en el Sistema**

Recibo Nº: ▼ Monto Bs.

Cédula:

Empleado:

Concepto

Cheque Nº:

Cuenta Nº:

Banco:

**Figura 7.49. Interfaz para generar Reportes de Egresos Diarios.**

Para generar reportes de los egresos mensuales del plantel, elija la opción Egresos Mensuales en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.50).

1. Seleccione el mes para el reporte; presione el primer botón y el sistema buscará los datos para presentarlos en la interfaz.
2. Presione el segundo botón y el sistema generará el reporte requerido.
3. Inicie el proceso presionando el tercer botón.
4. Cierre la interfaz presionando el cuarto botón.

Para generar reportes de los egresos anuales del plantel, elija la opción Egresos Anuales en el submenú Reportes (Figura 7.41) y obtendrá la interfaz correspondiente (Figura 7.51).



**Reporte de Egresos Mensuales**

**Datos para Búsqueda**

Mes: febrero Año: 2009

**Datos Registrados en el Sistema**

Recibo Nº:  Monto Bs.:

Cédula:

Empleado:

Concepto:

Cheque Nº:  Fecha de Pago:

Cuenta Nº:

Banco:

Icons: [Magnifying Glass] [Printer] [Refresh] [Right Arrow]

Figura 7.50. Interfaz para generar Reportes de Egresos Mensuales.

**Reporte de Egresos Anuales**

**Datos para Búsqueda**

Año: 2008 + -

**Datos Registrados en el Sistema**

Recibo Nº:  Monto Bs.:

Cédula:

Empleado:

Concepto:

Cheque Nº:  Fecha de Pago:

Cuenta Nº:

Banco:

Icons: [Magnifying Glass] [Printer] [Refresh] [Right Arrow]

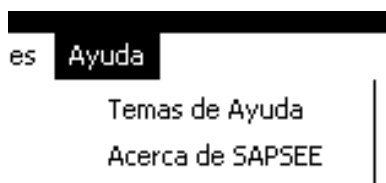
Figura 7.51. Interfaz para generar Reportes de Egresos Anuales.

### 7.2.6. Menú Ayuda

Con este menú obtendrá la ayuda necesaria para el uso del Sistema SAPSEE. Para generar la información acerca del nombre del producto, fecha de creación,

versión y creador del mismo, elija el menú Ayuda y se ubica en el submenú Acerca de SAPSEE (Figura 7.52) y obtendrá la información requerida.

Para generar la ayuda de SAPSEE, elija el menú Ayuda y se ubica en el submenú Temas de Ayuda (Figura 7.52) y obtendrá la información requerida.



**Figura 7.52. Menú Ayuda.**

## CONCLUSIONES

1. Los requisitos del plantel fueron transformados en el sistema software SAPSEE por medio del Proceso Unificado de Desarrollo de Software y el Lenguaje de Modelado Unificado (UML).
2. El Modelo de Casos de Uso permitió representar al sistema funcionalmente, para definir la interacción del usuario con el software.
3. La línea base de la arquitectura se estableció por medio del Modelo de Análisis, para ayudar a describir, a través de clases de análisis, una visión general del sistema; el Modelo de Diseño precisó la representación y el modelado del sistema, dándole forma, y soportando todos los requisitos tomados del análisis previamente realizado.
4. El diseño de la base de datos del sistema es consistente para el almacenamiento, organización y control de la información; usando como herramienta de consulta, el lenguaje SQL.
5. La implementación de la arquitectura final por medio de las clases del diseño permitió la construcción del sistema en forma de códigos fuentes para la obtención del software ejecutable.
6. Con el Modelo de Pruebas se obtuvo diversas clases de equivalencias, las cuales arrojaron resultados satisfactorios durante la integración, logrando la garantía de un sistema fiable, estable y sólido para su ejecución.

## RECOMENDACIONES

1. Desarrollar nuevos módulos para respaldar la información del plantel: datos de los alumnos, datos de los empleados, datos de ingresos y egresos, entre otros, para ayudar a mejorar el funcionamiento del sistema en caso de pérdida imprevista de algunos datos.
2. Realizar un mantenimiento constante del sistema SAPSEE con el propósito de mantener un excelente funcionamiento del software, permitiendo de esa forma la actualización de módulos existentes o de la inclusión de otros.
3. Entrenar a los futuros usuarios del Sistema para que lo manipulen correctamente y se produzca un excelente rendimiento de la aplicación.

## BIBLIOGRAFÍA

1. Rodríguez G., English J. “Desarrollo de un software que permita el control de las actividades asociadas a una Unidad Educativa Industrial en las modalidades: Educación Básica (III Etapa), Diversificada y Profesional”. Venezuela. 2005.
2. AutomatyControl SA de CV. Beneficios de un Sistema Automatizado. <http://www.automatycontrol.com/Beneficios-de-un-sistema-automatizado.1.htm>, México, 2005.
3. García M., María A. “Concepto de Base de Datos”. <http://barceo.eubd.uem.es/html/personalesenred/mantonia/decauto/tema1/concepto.htm>. Venezuela, 2005.
4. Moya G., Adrián E. “Desarrollo de un sistema para la automatización de los procesos de admisión y control de estudios en un Instituto Universitario”. Venezuela 2004.
5. Martra P., “Programación en castellano – Introducción a UML” <http://www.programacion.com/tutorial/UML>. 2004.
6. Simoni G., Miguel E. “Desarrollo de un sistema de información para la automatización de los procesos realizados en el Departamento de Ciencias de la Unidad de Estudios Básicos de la Universidad de Oriente, Núcleo de Anzoátegui”. Venezuela, 2004.

7. S. G. T. I. “Curso de Iniciación en SQL”.  
[http://www.itp.edu.mx/publica/tutoriales/basedat1/tema1\\_4.htm](http://www.itp.edu.mx/publica/tutoriales/basedat1/tema1_4.htm) . España, 2003.
8. Zavala R., "Ingeniería de Software". Soluciones Avanzadas, 2002,  
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>.
9. Uranga C., Raúl. “Base de Datos”.  
<http://www.monografias.com/trabajos12/basdat/basdat.shtml>, Venezuela. 2002.
10. Roger S. Pressman. “Ingeniería del Software. Un enfoque práctico”, Editorial McGraw-Hill, 5ª Edición, 2001.
11. Jacobson, I. Booch, G. y Rumbaugh, J. “El Proceso Unificado de Desarrollo de Software”, Editorial Pearson Educación, Primera Edición, Madrid, 2000.
12. Elmasri R. y Navathe S. Sistemas de Base de Datos. Segunda Edición. Editorial Addison-Wesley. México, 2000.
13. Hernández R., Yadira. ITP. “Tutorial de Base de Datos I”.  
[http://www.itp.edu.mx/publica/tutoriales/basedat1/tema1\\_4.htm](http://www.itp.edu.mx/publica/tutoriales/basedat1/tema1_4.htm).  
Venezuela, 1999.
14. Joyanes L. y Zahonero I. Estructura de Datos. Algoritmos, Abstracción y Objetos. McGraw-Hill. Madrid, 1998.
15. Circulo de Lectores, “Enciclopedia Multimedia”, Venezuela, 1998.

16. Gros, B. "Diseños y Programas Educativos", Barcelona Ariel, 1997.
17. Dalton P. Microsoft SQL Server Black Book. The Coriolis Group, 1997.
18. Pablo F, "Elementos Notacionales de UML"  
<http://www.cs.ualberta.ca/~pfiguero/soo/uml/>, 1997.
19. Lewis G., N "¿Qué es Ingeniería de Software?", DataPro, 1994.
20. Cota A. "Ingeniería de Software". Soluciones Avanzadas. México, 1994.
21. Greiff W. R, "Paradigma VS Metodología; El Caso de la POO (Parte II)"  
Soluciones Avanzadas, 1994.
22. HARWRYSZKIEWYCZ I. "Análisis y Diseño de Base de Datos".  
Editorial Megabyte. México, 1994.

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

TÍTULO	Desarrollo de un sistema para generar la automatización de los procesos académicos y administrativos vinculados a la I y II Etapa de Educación básica dentro de una Unidad Educativa
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Gimenez A., Endrikson J.	CVLAC: 13.689.716 E MAIL: endrikson_jose@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Desarrollo, Software, Automatización, Procesos, Administrativos, Educación, Actividades.



## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

ÁREA	SUBÁREA
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

## RESUMEN (ABSTRACT):

En la Primera y Segunda Etapa de Educación en Unidades Educativas, se realiza de forma manual el manejo de los datos y el procesamiento de la información, generando un completo desorden en el momento de ser solicitada. Por tal motivo surgió la necesidad de automatizar dichas actividades por medio del desarrollo de un sistema que permita la optimización de los datos de los alumnos, empleados, ingresos y egresos, procesamiento de constancias y reportes. Se utilizó una base de datos relacional porque ofrece un excelente rendimiento en la obtención y almacenamiento de datos, a través de consultas por medio del lenguaje SQL. Para desarrollar la aplicación se utilizó el Proceso Unificado de desarrollo de software, aplicado por medio de las fases de Inicio, Elaboración y Construcción, con su respectivo modelo de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación, modelo de prueba e integración del sistema. Como resultado se obtuvo una arquitectura robusta que lleva por nombre SAPSEE (Sistema de Automatización Primera y Segunda Etapa de Educación). Dicha herramienta permite optimizar las actividades primarias del plantel como: inscripción de alumnos, ingreso de empleados, procesamiento de calificaciones, ingresos y egresos de dinero y emisión de constancias y reportes.

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

## CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Mujica Yulitza	ROL	CA	AS X	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				
Dorta Pedro	ROL	CA	AS	TU	JU X
	CVLAC:				
	E_MAIL				
	E_MAIL				
Gerardino María	ROL	CA	AS	TU	JU X
	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				

## FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	03	25
AÑO	MES	DÍA

LENGUAJE. SPA

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

## ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.DesarrolloDeSoftware.doc	application/word

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

## ALCANCE

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

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

## TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

## NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

## ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

## INSTITUCIÓN:

Universidad de Oriente, Núcleo de Anzoátegui

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

### DERECHOS

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

Gimenez A., Endrikson José

AUTOR

Mujica Yulitza

TUTOR

Gerardino María

JURADO

Dorta Pedro

JURADO

Bastardo José Luis

POR LA SUBCOMISIÓN DE TESIS