



ESCUELA POLITECNICA DEL EJÉRCITO
EXTENSION LATACUNGA.

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA.

“SISTEMA DE INFORMACIÓN ACADÉMICO PARA LA ESCUELA
DE COCINA CUISSINE QUE AYUDE A TENER UN BUEN
CONTROL Y SEGUIMIENTO DEL ÁREA ACADÉMICA”

LUIS ANIBAL MOSCOSO CALVOPIÑA.

Tesis presentada como requisito previo a la obtención del grado de:

INGENIERO EN SISTEMAS E INFORMATICA.

AÑO 2013.

Declaración de Responsabilidad

ESCUELA POLITÉCNICA DEL EJÉRCITO
INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

LUIS ANIBAL MOSCOSO CALVOPIÑA

DECLARO QUE:

El proyecto de grado denominado “Sistema de Información Académico para la Escuela de Cocina Cuissine que ayude a tener un buen control y seguimiento del área académica”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 22 de Marzo de 2013.

Luis Aníbal Moscoso Calvopiña

Certificado de tutoría

ESCUELA POLITÉCNICA DEL EJÉRCITO
INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

ING. NANCY DEL PILAR JACHO GUANOLUISA. (DIRECTORA)

ING. FRANKLIN JAVIER MONTALUISA YUGLA. (CODIRECTOR)

CERTIFICAN

Que el trabajo titulado “Sistema de Información Académico para la Escuela de Cocina Cuissine que ayude a tener un buen control y seguimiento del área académica” realizado por Luis Aníbal Moscoso Calvopiña, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan al Sr. Luis Aníbal Moscoso Calvopiña que lo entregue a Ing. Luis Alberto Guerra Cruz, en su calidad de Director de la Carrera.

Latacunga, 22 de Marzo de 2013.

Ing. Nancy Jacho.
DIRECTORA

Ing. Javier Montaluisa.
CODIRECTOR

Autorización de publicación

ESCUELA POLITÉCNICA DEL EJÉRCITO
INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

YO, LUIS ANIBAL MOSCOSO CALVOPÍÑA.

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “Sistema de Información Académico para la Escuela de Cocina Cuissine que ayude a tener un buen control y seguimiento del área académica”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, 22 de Marzo de 2013.

Luis Aníbal Moscoso Calvopiña

DEDICATORIA

Este proyecto está dedicado a mis padres, quienes con su apoyo incondicional y buenos consejos, me enseñaron que todos los objetivos pueden cumplirse cuando se los realiza con esfuerzo y dedicación. A mis hermanos y amigos quienes de una u otra forma formaron parte de este sueño.

AGRADECIMIENTO

Gracias a Dios por darme la oportunidad de haber culminado esta etapa tan importante en mi vida, un sueño, una meta cumplida.

Un sueño que no hubiese sido posible sin el apoyo incondicional de mi familia, amigos, maestros y demás personas que a lo largo del tiempo han confiado en mí, gracias a todos ellos.

Mi agradecimiento a Carrera de Ingeniería en Sistemas e Informática, y a todos sus maestros, ya que sin ellos no hubiese podido finalizar tan anhelado sueño, sus conocimientos y experiencia me ayudaron a formarme como persona y a llegar a esta etapa de mi vida tan importante.

Gracias a todos de corazón.

Índice.

Contenido.

Declaración de Responsabilidad	i
Certificado de tutoría	ii
Autorización de publicación	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
Gracias a todos de corazón.....	v
Índice	vi
CAPITULO 1.....	3
INTRODUCCIÓN.....	3
1.1 INTRODUCCIÓN.....	3
1.2 ANTECEDENTES.....	4
1.2.1 DE LA INSTITUCIÓN.....	4
1.2.2 DEL PROYECTO.....	5
1.3 DEFINICIÓN DEL PROBLEMA.....	5
1.4 OBJETIVOS.....	8
1.4.1 OBJETIVO GENERAL.....	8
1.4.2 OBJETIVOS ESPECÍFICOS.....	8
1.5 JUSTIFICACIÓN.....	9
1.5.1 JUSTIFICACIÓN SOCIAL.....	9
1.5.2 JUSTIFICACIÓN TECNOLÓGICA.....	9
1.5.3 JUSTIFICACIÓN ECONÓMICA.....	9
1.6 ALCANCES Y LÍMITES.....	10
1.7 METODOLOGÍAS, TÉCNICAS Y HERRAMIENTAS.....	10
1.7.1 METODOLOGÍA DE PROCESO UNIFICADO RACIONAL – RUP.....	10
1.7.2 TÉCNICAS.....	12
1.7.3 HERRAMIENTAS.....	12
CAPITULO 2.....	14
MARCO TEORICO.....	14
2.1 INTRODUCCIÓN.....	14
2.2 INGENIERÍA DE REQUERIMIENTOS.....	14

2.2.1	FASES DE IMPLEMENTACIÓN.....	15
2.2.2	TÉCNICAS PRINCIPALES.....	16
2.2.3	ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE.....	16
2.2.4	IDENTIFICACIÓN DE LAS PERSONAS INVOLUCRADAS.....	16
2.2.5	CARACTERÍSTICAS DE LOS REQUERIMIENTOS.....	19
2.2.6	ACTIVIDADES DE LA INGENIERÍA DE REQUERIMIENTOS.....	20
2.3	INGENIERÍA DE SOFTWARE.....	22
2.4	METODOLOGÍA.....	23
2.4.1	METODOLOGÍA DE MODELADO RUP.....	24
2.4.2	DESCRIPCIÓN DE LAS ACTIVIDADES.....	26
2.5	ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.....	30
2.5.1	ANÁLISIS ORIENTADO A OBJETOS.....	30
2.5.2	DISEÑO ORIENTADO A OBJETOS.....	31
2.6	HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO DEL SOFTWARE. 31	
2.6.1	ARQUITECTURA CLIENTE/SERVIDOR.....	31
2.6.2	MODELO VISTA CONTROLADOR.....	32
2.6.3	PROGRAMACIÓN EN TRES CAPAS.....	33
2.6.4	LENGUAJE UNIFICADO DE MODELADO – UML.....	35
2.6.5	APACHE.....	37
2.6.6	PHP.....	38
2.6.7	JAVASCRIPT.....	41
2.6.8	AJAX.....	41
2.6.9	HOJAS DE ESTILO.....	42
2.6.10	MySQL.....	43
2.6.11	LIBRERÍAS ADODB.....	44
2.7	CALIDAD DEL SOFTWARE.....	45
2.7.1	ISO 9126.....	47
CAPITULO 3.....		51
MARCO APLICATIVO.....		51
3.1	INTRODUCCION.....	51
3.2	ANÁLISIS DE LA SITUACIÓN ACTUAL.....	51
3.2.1	DESCRIPCIÓN DE FUNCIONES.....	51
3.2.2	PROCESO ACADÉMICOS.....	52

3.3	FASE DE COMIENZO O INICIO CAPTURA DE REQUISITOS.....	53
3.3.1	ALCANCE DEL PROYECTO.....	53
3.3.2	MODELADO DEL SISTEMA.....	60
3.3.3	ACTORES.....	61
3.3.4	CASOS DE USO.....	61
3.4	FASE DE ELABORACIÓN.....	74
3.4.1	DIAGRAMA DE SECUENCIA DEL SISTEMA.....	74
3.4.2	CONTRATOS DEL SISTEMA.....	88
3.4.3	DIAGRAMAS DE ESTADO.....	107
3.4.4	CASOS DE USO REALES.....	114
3.4.5	DIAGRAMA DE CLASES.....	138
3.4.6	MODELO ENTIDAD/RELACIÓN.....	139
3.4.7	MODELO LÓGICO RELACIONAL.....	139
3.4.8	DIAGRAMA DE PAQUETES.....	146
3.5	FASE DE CONSTRUCCIÓN.....	147
3.5.1	DIAGRAMA NAVEGACIONAL.....	147
3.6	FASE DE TRANSICIÓN.....	147
3.6.1	ACTIVIDADES PARA LA IMPLEMENTACIÓN DEL SISTEMA.....	148
	CAPITULO 4.....	149
	CALIDAD DEL SOFTWARE.....	149
4.1	INTRODUCCIÓN.....	149
4.2	FUNCIONALIDAD.....	150
4.2.1	MÉTRICA DE ADECUADA.....	150
4.2.2	PUNTO FUNCIÓN.....	151
4.3	CONFIABILIDAD.....	156
4.3.1	MÉTRICA DE MADUREZ.....	157
4.4	USABILIDAD.....	158
4.4.1	MÉTRICA DE FACILIDAD DE COMPRENSIÓN.....	158
4.5	FACILIDAD DE MANTENIMIENTO.....	158
4.5.1	MÉTRICA DE FACILIDAD DE CAMBIO.....	159
	CAPITULO 5.....	160
	PRUEBAS Y SEGURIDAD.....	160
5.1	INTRODUCCIÓN.....	160

5.2	PRUEBAS DEL DISEÑO ARQUITECTÓNICO.....	160
5.3	SEGURIDAD.....	162
5.3.1	SEGURIDAD FÍSICA.	162
5.3.2	SEGURIDAD DE BASE DE DATOS.....	163
5.3.3	SEGURIDAD EN LA WEB	164
5.4	COSTO BENEFICIO.....	165
5.4.1	INTRODUCCIÓN.....	165
5.4.2	ANÁLISIS DE COSTO.	166
5.4.3	BENEFICIOS.....	168
	CAPITULO 6.....	170
	CONCLUSIONES Y RECOMENDACIONES.....	170
6.1	CONCLUSIONES.....	170
6.2	RECOMENDACIONES.	171
	BIBLIOGRAFÍA.....	172
	ANEXOS	174
	ANEXO 1: ARBOL DE PROBLEMAS	175
	ANEXO 2: ARBOL DE OBJETIVOS.....	177
	ANEXO 3: Código Fuente.....	178

ÍNDICE DE FIGURAS.

Figura 1. Modelado del Sistema.....	60
Figura 2. Diagrama de secuencia del caso de uso 1: Ingresar al Sistema.....	75
Figura 3. Diagrama de secuencia del caso de uso 2: Registrar Estudiante.....	76
Figura 4. Diagrama de secuencia del caso de uso 3: Registrar Familiar.....	77
Figura 5. Diagrama de secuencia del caso de uso: Registrar Parentesco.....	78
Figura 6. Diagrama de secuencia del caso de uso 5: Registrar Docente.....	79
Figura 7. Diagrama de secuencia del caso de uso 6: Registrar Materia.....	80
Figura 8. Diagrama de secuencia del caso de uso 7: Registrar Periodo.....	81
Figura 9. Diagrama de secuencia del caso de uso 8: Registrar Grado.....	82
Figura 10. Diagrama de secuencia del caso de uso 9: Registrar Horario.....	83
Figura 11. Diagrama de secuencia del caso de uso 10: Registrar Matricula.....	84
Figura 12. Diagrama de secuencia del caso de uso 11: Registrar Calificaciones.....	85
Figura 13. Diagrama de secuencia del caso de uso: Consultar Calificaciones.....	86
Figura 14. Diagrama de secuencia del caso de uso 13: Consultar Horarios.....	87
Figura 15. Diagrama de Estados: Ingresar al Sistema.....	107
Figura 16. Diagrama de Estados: Registrar Estudiante.....	108
Figura 17. Diagrama de Estados: Registrar Familiar.....	108
Figura 18. Diagrama de Estados: Registrar Parentesco.....	109
Figura 19. Diagrama de Estados: Registrar Profesor.....	109
Figura 20. Diagrama de Estados: Registrar Materia.....	110
Figura 21. Diagrama de Estados: Registrar Periodo.....	110
Figura 22. Diagrama de Estados: Registrar Grado.....	111
Figura 23 . Diagrama de Estados: Registrar Horario.....	111
Figura 24 Diagrama de Estados: Registrar Matricula.....	112
Figura 25. Diagrama de Estados: Registrar Calificaciones.....	112
Figura 26. Diagrama de Estados: Consultar Calificaciones.....	113
Figura 27. Diagrama de Estados: Consultar Horarios.....	113
Figura 28. Pantalla de Ingreso al Sistema.....	114
Figura 29. Pantalla Registrar Estudiante.....	116
Figura 30. Pantalla Lista de Estudiantes.....	116
Figura 31. Pantalla Registrar Familiar.....	118
Figura 32. Pantalla Lista de Familiares.....	118
Figura 33. Pantalla Registrar Parentesco.....	120
Figura 34. Pantalla Lista de Parentescos.....	120
Figura 35. Pantalla Registrar Profesor.....	122
Figura 36. Pantalla Lista de Profesores.....	122
Figura 37. Pantalla Registrar Materia.....	124
Figura 38. Pantalla Lista de Materias.....	124
Figura 39. Pantalla Registrar Periodo.....	126
Figura 40. Pantalla Lista de Periodos.....	126
Figura 41. Pantalla Registrar Grado.....	128
Figura 42. Pantalla Lista de Grados.....	128
Figura 43. Pantalla Registrar Horario.....	130
Figura 44. Pantalla Lista de Horarios.....	130

Figura 45. Pantalla Registrar Matricula.....	132
Figura 46. Pantalla Lista de Matriculas.....	132
Figura 47. Pantalla Lista de Grados del Periodo Actual.....	134
Figura 48. Pantalla Lista de Materias por Grado.....	134
Figura 49. Pantalla Lista de Estudiantes por Materia.....	134
Figura 50. Pantalla Consultar Calificaciones.....	136
Figura 51. Pantalla Consultar Horarios.....	137
Figura 52. Diagrama de Clases.....	138
Figura 53. Modelo Entidad Relación.....	139
Figura 54. Diagrama de Paquetes.....	146
Figura 55. Diagrama de Navegación.....	147
Figura 56. Grafo del sistema.....	161

ÍNDICE DE TABLAS

Tabla 1. Actividades de la Ingeniería de Requerimientos para diferentes modelos de procesos de Ingeniería de Software. [10].....	21
Tabla 2. Caso de Uso 1: Ingresar al Sistema.....	61
Tabla 3. Descripción del evento: Ingresar al Sistema.....	62
Tabla 4. Caso de Uso 2: Registrar Estudiante.....	63
Tabla 5. Descripción del evento: Registrar Estudiante.....	63
Tabla 6. Caso de uso 3: Registrar familiar.....	64
Tabla 7. Descripción del Evento: Registrar Familiar.....	64
Tabla 8. Caso de uso 4: Registrar parentesco.....	65
Tabla 9. Descripción del evento: Registrar parentesco.....	65
Tabla 10. Caso de uso 5: Registrar docente.....	66
Tabla 11. Descripción del evento: Registrar Docente.....	67
Tabla 12. Caso de uso 6: Registrar Materia.....	67
Tabla 13. Descripción del evento: Registrar Materia.....	67
Tabla 14. Caso de uso 7: Registrar Periodo.....	68
Tabla 15. Descripción del evento: Registrar Periodo.....	68
Tabla 16. Caso de uso 8: Registrar Grado.....	69
Tabla 17. Descripción del evento: Registrar Grado.....	69
Tabla 18. Caso de uso 9: Registrar Horario.....	70
Tabla 19. Descripción del evento: Registrar Horario.....	70
Tabla 20. Caso de uso 10: Registrar Matricula.....	71
Tabla 21. Descripción del evento: Registrar Matricula.....	71
Tabla 22. Caso de uso 11: Registrar Calificaciones.....	72
Tabla 23. Descripción del evento: Registrar Calificaciones.....	72
Tabla 24. Caso de uso 12: Consultar Calificaciones.....	73
Tabla 25. Descripción del evento: Consultar Calificaciones.....	73
Tabla 26. Caso de uso 13: Consultar Horarios.....	73
Tabla 27. Descripción del evento: Consultar Horarios.....	74
Tabla 28. Contrato: Ingresar al Sistema Función: Introducir Identificador.....	88
Tabla 29. Contrato: Registrar Estudiante Función: Ingresar Datos.....	89
Tabla 30. Contrato: Registrar Estudiante Función: Validar Datos.....	89
Tabla 31. Contrato: Registrar Estudiante - Función: Mostrar Información.....	90
Tabla 32. Contrato: Registrar Familiar. Función: Ingresar Datos.....	90
Tabla 33. Contrato: Registrar Familiar. Función: Validar Datos.....	91
Tabla 34. Contrato: Registrar Familiar. Función: Mostrar Información.....	91
Tabla 35. Contrato: Registrar Parentesco. Función: Ingresar Datos.....	92
Tabla 36. Contrato: Registrar Parentesco. Función: Validar Datos.....	93
Tabla 37. Contrato: Registrar Parentesco. Función: Mostrar Información.....	93
Tabla 38. Contrato: Registrar Profesor. Función: Ingresar Datos.....	94
Tabla 39. Contrato: Registrar Profesor. Función: Validar Datos.....	94
Tabla 40. Contrato: Registrar Profesor. Función: Mostrar Información.....	95
Tabla 41. Contrato: Registrar Materia. Función: Ingresar Datos.....	95
Tabla 42. Contrato: Registrar Materia. Función: Validar Datos.....	96
Tabla 43. Contrato: Registrar Materia. Función: Mostrar Información.....	96
Tabla 44. Contrato: Registrar Periodo. Función: Ingresar Datos.....	97

Tabla 45. Contrato: Registrar Periodo. Función: Validar Datos.....	97
Tabla 46. Contrato: Registrar Periodo. Función: Mostrar Información.....	98
Tabla 47. Contrato: Registrar Grado. Función: Ingresar Datos.	99
Tabla 48. Contrato: Registrar Grado. Función: Validar Datos.	99
Tabla 49. Contrato: Registrar Grado. Función: Mostrar Información.....	100
Tabla 50. Contrato: Registrar Horario. Función: Ingresar Datos.....	100
Tabla 51. Contrato: Registrar Horario. Función: Validar Datos.....	101
Tabla 52. Contrato: Registrar Horario. Función: Mostrar Información.	101
Tabla 53. Contrato: Registrar Matriculas. Función: Ingresar Datos.	102
Tabla 54. Contrato: Registrar Matriculas. Función: Validar Datos.	102
Tabla 55. Contrato: Registrar Matriculas. Función: Mostrar Información.....	103
Tabla 56. Contrato: Registrar Calificaciones. Función: Ingresar Datos.	104
Tabla 57. Contrato: Registrar Matriculas. Función: Validar Datos.	104
Tabla 58. Contrato: Registrar Matriculas. Función: Mostrar Información.....	105
Tabla 59. Contrato: Consultar Calificaciones. Función: Solicitar Información.	106
Tabla 60. Contrato: Consultar Calificaciones. Función: Mostrar Información.	106
Tabla 61. Caso de uso real: Ingresar al Sistema.	114
Tabla 62. Descripción del Evento: Ingresar al Sistema.	115
Tabla 63. Caso de uso real: Registrar Estudiante.....	116
Tabla 64. Descripción del Evento: Registrar Estudiante.....	117
Tabla 65. Caso de uso real: Registrar Familiar.....	117
Tabla 66. Descripción del Evento: Registrar Familiar.	119
Tabla 67. Caso de uso real: Registrar Parentesco.	119
Tabla 68. Descripción del Evento: Registrar Parentesco.....	121
Tabla 69. Caso de uso real: Registrar Docente.	121
Tabla 70. Descripción del Evento: Registrar Docente.	123
Tabla 71. Caso de uso real: Registrar Materia.....	123
Tabla 72. Descripción del Evento: Registrar Materia.....	125
Tabla 73. Caso de uso real: Registrar Periodo.	125
Tabla 74. Descripción del Evento: Registrar Periodo.	127
Tabla 75. Caso de uso real: Registrar Grado.....	127
Tabla 76. Descripción del Evento: Registrar Grado.....	129
Tabla 77. Caso de uso real: Registrar Horario.....	129
Tabla 78. Descripción del Evento: Registrar Horario.	131
Tabla 79. Caso de uso real: Registrar Matricula.	131
Tabla 80. Descripción del Evento: Registrar Matricula.	133
Tabla 81. Caso de uso real: Registrar Calificaciones.	133
Tabla 82. Descripción del Evento: Registrar Calificaciones.....	135
Tabla 83. Caso de uso real: Consultar Calificaciones.	135
Tabla 84. Descripción del Evento: Consultar Calificaciones.....	136
Tabla 85. Caso de uso real: Consultar Horarios.	136
Tabla 86. Descripción del Evento: Consultar Horarios.	137
Tabla 87. Tabla Alumnos.....	140
Tabla 88. Tabla Categorías.	140
Tabla 89. Tabla DetalleHorarios.....	141
Tabla 90. Tabla Familiares.....	141

Tabla 91. Tabla Grados.....	142
Tabla 92. Tabla Horarios.....	142
Tabla 93. Tabla Materias.....	142
Tabla 94. Tabla Matriculas.....	143
Tabla 95. Tabla Notas.....	144
Tabla 96. Tabla Parentesco.....	144
Tabla 97. Tabla Periodo.....	144
Tabla 98. Tabla Profesores.....	145
Tabla 99. Tabla RelacionFamiliar.....	145
Tabla 100. Tabla Usuarios.....	145
Tabla 101. Métrica de Adecuidad.....	151
Tabla 102. Archivos Lógicos Internos.....	151
Tabla 103. Archivos de Interfaz Externa.....	151
Tabla 104. Entradas Internas.....	152
Tabla 105. Salidas Externas.....	153
Tabla 106. Consultas Externas.....	153
Tabla 107. Calculo de punto función.....	154
Tabla 108. Valores de ajuste de complejidad.....	155
Tabla 109. Características Generales del Sistema.....	155
Tabla 110. Métrica de Madurez.....	157
Tabla 111. Métrica de Facilidad de Comprensión.....	158
Tabla 112 Escala.....	159
Tabla 113. Métrica de Facilidad de Cambio.....	159
Tabla 114. Tabla de los coeficientes para los diferentes proyectos.....	167
Tabla 115. Tablas de los costos del sistema.....	168

RESUMEN

El presente documento muestra todo el desarrollo de un sistema de información académica para la ESCUELA DE COCINA CUISSINE, mostrando los antecedentes, la investigación y los procesos que se realizaron para completar dicho proyecto.

Para el desarrollo del proyecto era fundamental conocer todos los aspectos relacionados con la institución y como se manejan los procesos, de esta manera determinamos el alcance del problema que se va a optimizar mediante el sistema.

Después de determinar el alcance del problema planteamos los objetivos a cumplir y la respectiva justificación para la elaboración del proyecto. Además establecimos la metodología a emplear en el desarrollo del sistema.

Como siguiente punto empezamos con la investigación y desarrollo de los contenidos teóricos acerca de la metodología y tecnologías que ayudaran y participaran en el desarrollo del proyecto. Luego mostraremos cada una de las fases de elaboración del sistema en base a la metodología y los resultados obtenidos.

ABSTRACT

This document shows all the development of an information system for Academic Cuisine Cooking School, showing the antecedents, research and processes that are performed to complete the project.

For the development of the project was essential to know all aspects of the institution and processes are handled in this way determine the scope of the problem to be a optimized by the system.

After determining the scope of the problem we propose to fulfill the objectives and the relevant justification for the development of the project. Also established the methodology used in developing the system.

As next point began to research and develop the theoretical about the methodology and technologies to assist and participate in the development of the project. Then show each of the stages of development of the system based on the methodology and results.

CAPITULO 1

INTRODUCCIÓN

1.1 INTRODUCCIÓN.

Desde la antigüedad el hombre se ha mostrado como un ser creativo, empleando instrumentos rudimentarios ha expresado situaciones cotidianas y creencias que hoy en día se consideran expresiones artísticas.

Actualmente en Ecuador existen instituciones que brindan servicios de formación artísticas en diferentes áreas, y para lograr su objetivo es que muchas de ellas optan por sujetarse al avance tecnológico para optimizar las funciones que desempeñan, debido a estas circunstancias la Escuela de Cocina Cuissine requiere el apoyo de un sistema informático que ayude a controlar y administrar la información y procesos que se manejan dentro de la institución.

El presente proyecto se enfoca en solucionar los problemas de la Escuela de Cocina Cuissine, que tiene como principal misión la formación de Chefs, orientando y reglamentando la oferta de estudios especializados que tiene como meta el ejercicio profesional en el campo de las artes. La Escuela de Cocina de Cuissine se constituye en una institución académica de formación artística, capaces de desempeñarse de manera eficiente en las necesidades gastronómicas del país.

La manipulación de información extensiva relacionada al seguimiento académico hace ver la necesidad de optar por un sistema informático que ayude a la institución a realizar sus funciones de manera eficiente, obteniendo información rápida, segura y confiable desde cualquier lugar mediante el acceso a internet.

Para ello se considera los siguientes puntos importantes: la inscripción de estudiantes nuevos, registro de familiares y profesores, la administración de materias, grados periodos y horarios, y un control de matriculas y calificaciones de los alumnos, que ayuden a la institución a manejar los procesos académicos.

1.2 ANTECEDENTES.

1.2.1 DE LA INSTITUCIÓN.

Le Cuissine abre sus puertas en la ciudad de Ambato, el 23 de junio del 2003 en una casa residencial, con Acuerdo Ministerial No. 4621, otorgado por el Ministerio de Educación del Ecuador. Inicio sus actividades con cuarenta alumnos, hoy cuenta con más de trescientos alumnos en aula y 14 promociones graduadas. Ocho años de historia labrando en esta hermosa ciudad de los Tres Juanes, no son producto de la casualidad, sino del trabajo serio, honesto y el esfuerzo de todos quienes laboran en la institución. *“Nuestra Experiencia hace la Diferencia”*.

Comprometidos con nuestra sociedad y gracias al espíritu de innovación y liderazgo, fundamos el COLEGIO TECNICO CUISSINE, el 08 de septiembre del 2008 después de una ardua labor. Hoy cuenta con 8vo, 9no, y 10mo Año de Educación Básica otorgando el Título de TECNICOS PRACTICOS EN GASTRONOMIA Y TURISMO, con opción de terminar el Bachillerato en Gastronomía y Turismo para que puedan continuar con sus estudios superiores en cualquier Universidad nacional o internacional.

Desde el inicio de su fundación la Escuela lleva el control de la información de estudiantes, familiares y docentes de forma manual lo cual conlleva a una excesiva acumulación de documentos e información. Otro factor que se debe

analizar es la dificultad en la búsqueda de información solicitada por docentes o alumnos.

1.2.2 DEL PROYECTO.

Los sistemas escolásticos posibilitan el logro de importantes mejoras en cuanto a la gestión de la información, automatizan para la toma de decisiones y el manejo y uso de información. Su implementación y generalización a diferentes sectores los procesos operativos, suministran una plataforma de información necesaria sobre todo los relacionados al ámbito educativo para el área de educación permanente donde se encuentran carrera artesanales como Estilistas, Chefs, Imagen personal, significan un avance y desarrollo tecnológico para las instituciones.

1.3 DEFINICIÓN DEL PROBLEMA.

Actualmente la institución carece de un sistema de información que contrólelos procesos en el área académica, produciendo volúmenes considerables de información, contrarrestando el problema con procesos burocráticos para reducir el trabajo de la parte administrativa. Pero a pesar de esto se nota en ciertas actividades como inscripciones y matriculación, el trabajo se acumula y las tareas se vuelven caóticas haciendo notar ineficiencia y desorden en las actividades diarias de la institución.

A continuación se presentan los siguientes procesos realizados manualmente referentes al seguimiento académico:

- Registro de nuevos estudiantes, el proceso se realiza mediante el llenado de formularios de inscripción por los alumnos, posteriormente la secretaria se encarga de guardar los mismos.

- Registro de calificaciones de los estudiantes a las diferentes materias de su horario.
- Seguimiento Académico de los estudiantes, incluye el mínimo de los datos personales, calificaciones de las diferentes materias de las materias correspondientes.
- Historial de docentes, que escasamente contiene información relacionada a los datos personales.
- Elaboración de listados de cada materia con el respectivo número de estudiantes, datos del estudiante, y docentes asignados a los mismos.
- Elaboración de certificados de notas.
- Matriculación de los estudiantes en los respectivos grados y periodos correspondientes, proceso que los estudiantes lo hacen de forma manual por medio de formularios que abarcan información de periodos y grados correspondientes.

Los problemas que conlleva cada proceso son los siguientes:

- **Inexistencia de un historial académico de estudiante y de un historial de docentes**, lo cual dificulta en la obtención de datos personales del estudiante para realizar sus respectivos papeleos e informes, en el caso de docentes la inexistencia del mismo dificulta ubicarlo para realizar determinada consulta, más aun si existe problemas con la pérdida de notas de estudiantes. .
- **La falta de control en inscripciones**, no existe una validación previa de la información proporcionada por los alumnos.
- **Volúmenes considerables de documentación**, los procesos de inscripción a estudiantes, inscripción de materias, registro de matrícula, registro de calificaciones y otros, acumulan bastante

información tangible lo que dificulta para su obtención de manera rápida.

- **Demora en la búsqueda de calificaciones de un estudiante**, se tiene una considerable pérdida de tiempo en la búsqueda de sus respectivas calificaciones para la obtención de sus certificados de notas como de su record académico.
- **Atención no apropiada a los estudiantes**, a principio y final de cada periodo los estudiantes realizan sus matriculas, solicitud de sus certificados de notas, los mismos realizan largas filas y deben esperar un tiempo considerable para que se les atienda, por tratarse de procesos realizados manualmente y varios de ellos son realizados al mismo tiempo, considerando también que los documentos están en diferentes archiveros.
- **Demora en la emisión de reportes**, la demora en la búsqueda de información ocasiona que la información no sea oportuna.
- **Listado incorrecto de estudiantes**, se inicia cuando a principio de cada periodo algo más de la mitad de los estudiantes se registra en las boletas de inscripción cancelando algún monto, y la otra cantidad de estudiantes se registra directamente con su respectivo docente sin haber llenado las boletas de inscripción y sin cancelar ningún monto.
- **Desorganización**, en los procesos referentes al seguimiento académico, existe actualmente una confusión en cuanto al manual de funciones de cada administrativo lo que implica desorden en el momento de desempeñar sus funciones.

Gran parte de los problemas existentes se deben a la extensa información y a su tratamiento, ocasionando en los administrativos de la institución un desorden en cuanto a las funciones que se desempeñan, no ayudando el que las actividades se desarrollan en forma manual, donde los datos

son recopilados y transcritos por empleados de la institución convirtiéndose en procesos tediosos, monótonos y rutinarios, que no ayudan a su buen desempeño.

Por lo visto anteriormente, para el manejo de información en la Institución se plantea en la siguiente interrogante:

¿De qué forma el “SISTEMA DE INFORMACIÓN ACADÉMICO PARA LA ESCUELA DE COCINA CUISSINE” ayudara a reducir el trabajo de los administrativos?

1.4 OBJETIVOS.

1.4.1 OBJETIVO GENERAL.

Desarrollar un sistema de información académico para la Escuela de Cocina Cuissine que ayude a tener un buen control y seguimiento del área académica, obteniendo de esta forma, información fidedigna, eficaz, confiable y oportuna para la institución.

1.4.2 OBJETIVOS ESPECÍFICOS.

El sistema tiene por objetivos específicos los siguientes:

Levantar los procesos de manera eficiente, con la automatización de procesos manuales para que la obtención de información sea de forma inmediata.

Diseñar una base de datos para estudiantes y docentes, que contenga información necesaria para la institución, para facilitar su obtención.

Desarrollar un sistema informático basado en la información y procesos

identificados dentro de la institución.

1.5 JUSTIFICACIÓN.

1.5.1 JUSTIFICACIÓN SOCIAL.

Actualmente la institución brinda servicios de formación artística a un conjunto de personas, el Sistema de Información Académica se justifica socialmente por ser una herramienta de obtención y procesamiento de información rápida y automática, que ayuda a la parte administrativa de la institución a optimizar sus recursos, y beneficiará a la parte estudiantil haciendo más fácil los procesos de inscripción e información académica.

1.5.2 JUSTIFICACIÓN TECNOLÓGICA.

En un mundo en el cual la tecnología tiende a un constante crecimiento, es ineludible estar al margen de este avance tecnológico, más aun si el incluir tecnologías que ayuden a mejorar el rendimiento del personal y el desempeño de la institución, favorezcan a la sociedad en común a obtener mejores servicios.

1.5.3 JUSTIFICACIÓN ECONÓMICA.

Económicamente el sistema minimizará el costo asociado al tiempo de obtención de información, reduciendo la cantidad de papeles que se generan en los procesos de seguimiento académico, alcanzando así resultados eficientes con el menor esfuerzo del personal administrativo, optimizando tiempo y dinero.

1.6 ALCANCES Y LÍMITES.

Con el presente proyecto de grado se pretende implementar un SISTEMA DE INFORMACIÓN ACADÉMICO PARA LA ESCUELA DE COCINA CUISSINE que ayude a administrar información académica relevante para la institución, automatizando los siguientes módulos:

Módulo de datos, que contiene información referente al seguimiento académico, es decir la información de estudiantes, familiares y profesores de la institución.

Módulo de Administración, que contiene toda la información sobre las materias, grados, horarios y periodos, los mismos que complementan el proceso de matriculación y calificaciones.

Módulo de Matriculación, que permite registrar un historial de matriculas de cada uno de los estudiantes de la institución.

Modulo de Calificaciones, que permite ingresar las diferentes calificaciones de los estudiantes de acuerdo al horario y periodo en vigencia.

1.7 METODOLOGÍAS, TÉCNICAS Y HERRAMIENTAS.

1.7.1 METODOLOGÍA DE PROCESO UNIFICADO RACIONAL – RUP.

El objetivo del análisis es desarrollar el modelo del funcionamiento del sistema que satisfaga las necesidades de la institución. El modelo se expresa en términos de objetos y relaciones, el control dinámico de flujo y las transformaciones funcionales, teniendo en cuenta que es una metodología muy organizativa y es adaptable a cualquier tipo de proyecto.

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado

UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas.

El ciclo de vida de la metodología RUP es la siguiente:

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

La metodología RUP define nueve actividades a realizarse en cada fase:

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Pruebas
6. Distribución
7. Gestión de configuración y cambios
8. Gestión de proyectos
9. Gestión de entorno

1.7.2 TÉCNICAS.

El análisis puede ser un proceso largo y arduo para el que se requiere de habilidades psicológicas. Los nuevos sistemas cambian el entorno y las relaciones entre las personas, así que es importante identificar a todas las personas implicadas y considerar sus necesidades, asegurando que entienden las implicaciones del nuevo sistema. Existen técnicas para obtener los requisitos del usuario, como por ejemplos los que se utilizan para el presente proyecto.

Entrevistas, la entrevista es el método más usual. Por lo general no se entrevista a toda las personas que se relacionará con el sistema, sino a un determinado grupo de personas que represente a todos los sectores críticos de la institución, con énfasis en los sectores más afectados o que harán uso frecuente del nuevo sistema.

Formulario, en lugar de una entrevista, se pueden llenar formularios indicando los requerimientos.

Prototipos, un prototipo es una pequeña muestra, de funcionalidad limitada, de cómo sería el producto final una vez terminado. Ayudan a conocer la opinión de los usuarios y rectificar algunos aspectos antes de llegar al producto terminado.

1.7.3 HERRAMIENTAS.

1.7.3.1 LENGUAJE DE PROGRAMACIÓN.

Php, especial para realizar sistemas en red ampliada vía Internet (Software libre)

1.7.3.2 BASE DE DATOS.

MySql, gestor de Base de datos que no requiere licencia.

1.7.3.3 SISTEMA OPERATIVO.

Servidor, Linux (software libre).

Clientes, Windows XP (Microsoft).

Herramientas de programación Ajax, para optimizar procesos.

Javascript, para realizar eventos con efectos y validaciones a nivel cliente.

CAPITULO 2

MARCO TEORICO

2.1 INTRODUCCIÓN.

La ingeniería de software y la ingeniería de requerimientos comprenden todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para la realización de software, tomando en cuenta los diversos requerimientos de los interesados.

El propósito de la ingeniería de requerimientos es hacer que los requerimientos alcancen un estado óptimo antes de seguir adelante con el proyecto.

2.2 INGENIERÍA DE REQUERIMIENTOS.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, permitiendo enfocarse en un área fundamental: la definición del producto que vamos a desarrollar. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema, de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

A continuación se darán algunas definiciones para ingeniería de requerimientos. "Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema" [1].

"Ingeniería de Requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes , ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones".

"Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos" [6].

Por tanto, la Ingeniería de requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto, que son las necesidades mínimas que debe cumplir el software.

2.2.1 FASES DE IMPLEMENTACIÓN.

Desde un punto de vista conceptual, las actividades que se realizan son de 5 clases:

- a) **Obtener requerimientos:** A través de cuestionarios, entrevistas, foros, se recopila los requerimientos de los diferentes usuarios que interactúan con el sistema.
- b) **Analizar requerimientos:** Detectar y corregir las falencias comunicativas, transformando los requerimientos obtenidos, en condiciones apropiadas para ser tratados por el diseño.
- c) **Documentar requerimientos:** Igual que todas las etapas, los requerimientos deben estar debidamente documentados.

- d) **Verificar los requerimientos:** Consiste en comprobar el correcto funcionamiento de un requerimiento en la aplicación.
- e) **Validar los requerimientos:** Comprobar que los requerimientos implementados se corresponden con lo que inicialmente se pretendía.

2.2.2 TÉCNICAS PRINCIPALES.

Los analistas pueden emplear varias técnicas para obtener los requisitos del usuario. Históricamente, esto ha incluido técnicas tales como las entrevistas, o talleres con grupos para crear listas de requisitos.

Técnicas más modernas incluyen los prototipos, y utilizan casos de uso. Cuando sea necesario, el analista empleará una combinación de estos métodos para establecer los requisitos exactos de las personas implicadas, para producir un sistema que resuelva las necesidades del negocio.

2.2.3 ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE.

Una especificación de requisitos del software es una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones entre los usuarios y el sistema.

También contiene requisitos no funcionales (o suplementarios). Los requisitos no funcionales son los requisitos que imponen restricciones al diseño o funcionamiento del sistema (tal como requisitos de funcionamiento o requisitos del diseño).

2.2.4 IDENTIFICACIÓN DE LAS PERSONAS INVOLUCRADAS.

Debido a que los cambios que introduce un sistema nuevo tienden a afectar a más de un tipo de usuario, los analistas de requerimientos van a tomar en consideración a todos los implicados para que se obtengan y depuren sus requerimientos de la forma más fidedigna posible.

Entre las personas implicadas hay que considerar:

- Organizaciones que integran la organización del analista que está diseñando el sistema.
- Organizaciones o sistemas de respaldo.
- Directivos.
- Usuarios.

2.2.4.1 PROBLEMAS RELACIONADOS CON LAS PERSONAS INVOLUCRADAS.

Existen varias razones que pueden dificultar la determinación de los requisitos, entre ellas tenemos:

- Los usuarios no tienen claro lo que desean.
- Los usuarios no se involucran en la elaboración de requisitos escritos.
- Los usuarios insisten en nuevos requisitos después de finalizada la fase de obtención de requisitos.
- La comunicación con los usuarios es lenta.
- Los usuarios no participan en revisiones o hay poca inserción de los usuarios en el desarrollo de software.
- Los usuarios no comprenden los problemas técnicos.
- Los usuarios desconocen el proceso a seguir cuando el software esta implementado.

Esto puede conducir a la situación donde las exigencias del usuario cambian, incluso cuando el desarrollo del producto ya finalizó.

2.2.4.2 RELACIONADOS CON LOS DESARROLLADORES.

Los problemas posibles causados por los desarrolladores durante el análisis de requisitos son:

- El personal técnico y los usuarios finales pueden tener diversos vocabularios y pueden llegar a creer incorrectamente que están de acuerdo, no dándose cuenta del desacuerdo hasta que se entrega el producto final.
- Los desarrolladores pueden intentar encajar el sistema en un modelo existente, en vez de desarrollar un sistema adaptado a las necesidades del usuario.
- El análisis de requisitos se puede realizar a menudo por los ingenieros o programadores, en vez de personal con habilidades de relación con las personas y el conocimiento para entender las necesidades de un usuario correctamente.

2.2.4.3 SOLUCIONES APLICADAS.

Una solución aplicada en los problemas de comunicaciones ha sido emplear a especialistas en análisis del sistema.

Algunas de las herramientas aplicadas para salvar las diferencias entre ambas partes y que permiten la comprobación de las aplicaciones son:

- Capacidad de capturar la lógica de la institución y la información necesaria.

- Capacidad de generar los prototipos que imitan fielmente el producto final.
- Capacidad para que usuarios remotos y distribuidos operen con el prototipo.
- Documentar eficientemente los requerimientos a desarrollar utilizando un estándar adecuado.

2.2.5 CARACTERÍSTICAS DE LOS REQUERIMIENTOS.

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, debe presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes:

Necesario, un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades de software.

Conciso, un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

Completo, un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

Consistente, un requerimiento es consistente si no es contradictorio con otro requerimiento.

No ambiguo, Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición no debe causar confusiones al lector.

Verificable, un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

2.2.6 ACTIVIDADES DE LA INGENIERÍA DE REQUERIMIENTOS.

En el presente documento serán presentadas secuencialmente, estas actividades son aplicadas de manera continua y en orden variado.

Dependiendo del tamaño del proyecto y del modelo de proceso de software utilizado para el ciclo de desarrollo, las actividades de la ingeniería de requerimientos varían tanto en número como en nombres. La tabla No.1 muestra las actividades identificadas para diferentes procesos. A pesar de las diferentes puntos de vista de autores y estándares que cada desarrollador tenga sobre el conjunto de actividades mostradas en la tabla, podemos identificar y extraer cinco actividades principales que son:

- Análisis del Problema
- Evaluación y Negociación
- Especificación
- Validación
- Evolución

MODELO	Oliver and Steiner 1996	EIA / IS-632	IEEE Std 1220-1994	CMM nivel Repetitivo (2)	RUP
Actividades	Evaluar la información disponible.	Análisis de requerimientos.	Análisis de Requerimientos.	Identificación de requerimientos.	Análisis del Problema.
	Definir métricas efectivas.	Análisis funcional.	Estudio de los requerimientos.	Identificación de restricciones del sistema a desarrollar.	Comprender las necesidades de los involucrados.
	Crear un modelo del comportamiento del sistema.	Síntesis.	Validación de requerimientos.	Análisis de los requerimientos.	Definir el sistema.
	Crear un modelo de los objetos.	Análisis y control del sistema.	Análisis funcional.	Representación de los requerimientos.	Analizar el alcance del proyecto.
	Ejecutar el análisis.		Evaluación y estudio de funciones.	Comunicación de los requerimientos.	Modificar la definición del sistema.
	Crear un plan secuencial de construcción y pruebas.		Verificación de funciones.	Validación de requerimientos.	Administrar los cambios de requerimientos.
			Síntesis.		
			Evaluación del diseño.		
			Verificación física.		
			Control.		

Tabla 1. Actividades de la Ingeniería de Requerimientos para diferentes modelos de procesos de Ingeniería de Software. [10]

2.3 INGENIERÍA DE SOFTWARE.

La expresión Ingeniería de software fue introducido a causa de la crisis del software caracterizada por los siguientes tres puntos: primero imprecisión en la planificación de los proyectos y estimación de los costos, segundo baja calidad existente en la creación del software, y tercero dificultad del mantenimiento de programas con un diseño poco estructurado.

El objetivo primordial de la Ingeniería de Software es la construcción y desarrollo de proyectos en los que se aplican métodos y técnicas para la realización del software, mejorando su calidad, facilitando el control del proceso y desarrollo del software, suministrando a los desarrolladores las bases para construir software de alta calidad en forma eficiente y dejando a un lado la artesanía que era en ese entonces el desarrollo del software.

Según diferentes autores se considera a la ingeniería de software como: un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, en otras palabras, se considera que, 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 económicas a los problemas de desarrollo de software.

El proceso de ingeniería de software se define como "Un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" [3].

El proceso de desarrollo de software "Es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es

probado, documentado y certificado para su uso operativo", [3].

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Software, probablemente la definición más formal del software es la atribuida a la IEEE en su estándar 729, “la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo” [2], bajo esta definición el concepto de software va mas allá

De los programas de cómputo en sus distintas formas. Por lo mencionado anteriormente, “Ingeniería de Software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo” [7].

2.4 METODOLOGÍA.

Una metodología es aquella guía que se sigue a fin realizar las acciones propias de una investigación, identifica qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación. Es posible definir una metodología como aquel enfoque que permite observar un problema de una forma total, sistemática y con cierta disciplina.

2.4.1 METODOLOGÍA DE MODELADO RUP.

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier tipo de proyecto [11]. El ciclo de vida de RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y haciendo hincapié en las distintas actividades. En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades

1. Inicio (puesta en marcha)
2. Elaboración (definición análisis y diseño)
3. Construcción (implementación)
4. Transición (fin del proyecto y puesta en producción)

En cada fase se ejecutaran una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas se seguirá el modelo cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas. Como tres características esenciales está dirigido por los Casos de Uso que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Como filosofía RUP maneja seis principios clave:

Adaptación del proceso.

El proceso deberá adaptarse a las características propias de la institución. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

Balancear prioridades.

Los requerimientos de los diversos usuarios pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

Colaboración entre equipos.

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados.

Demostrar valor iterativamente.

Los proyectos se entregan, aunque sea de un modo interno, en **etapas iteradas**. En cada iteración se analiza la opinión de los usuarios, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

Elevar el nivel de abstracción.

Este principio dominante motiva el uso de conceptos reutilizables tal como patrón del software, por ejemplo. Que se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

Enfocarse en la calidad.

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

RUP define nueve actividades a realizar en cada fase del proyecto.

1. Modelo del negocio
2. Análisis de requisitos
3. Análisis de diseño
4. Implementación
5. Prueba
6. Distribución
7. Gestión de configuración y cambios
8. Gestión de proyecto
9. Gestión de entorno

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la institución.

2.4.2 DESCRIPCIÓN DE LAS ACTIVIDADES.

2.4.2.1 FASE DE INICIO.

Durante la fase de inicio las iteraciones tienen mayor énfasis en actividades de modelado del negocio y de requisitos.

Modelado del negocio.

Contempla la familiarización del funcionamiento de la institución, sobre conocer sus procesos. Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado. Entender el

problema actual en la organización objetivo e identificar potenciales mejoras. Asegurar que los usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

Requisitos.

Los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos. Establecer y mantener un acuerdo entre usuarios y los desarrolladores sobre lo que el sistema debe hacer.

- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

2.4.2.2 FASE DE ELABORACIÓN.

En la fase de elaboración, las iteraciones se orientan al desarrollo de ciclos de desarrollo de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación.

Análisis y diseño.

En la actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema

- Transformar los requisitos al diseño del sistema
- Desarrollar una arquitectura para el sistema
- Adaptar el diseño para que sea consistente con el entorno de implementación

2.4.2.3 FASE DE CONSTRUCCIÓN.

Implementación.

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

- Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se integra el sistema siguiendo el plan.

Pruebas.

El flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.

Verificar que los requisitos tengan su apropiada implementación.

2.4.2.4 FASE DE TRANSICIÓN.

Despliegue.

La actividad tiene como objetivo, producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios.
- Migrar el software existente o convertir bases de datos.

2.4.2.5 DURANTE TODO EL PROYECTO.

Gestión del proyecto.

Se vigila el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto
- Proveer un marco de trabajo para gestionar riesgos

Configuración y control de cambios.

El control de cambios permite mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

Entorno.

La finalidad de esta actividad es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir. En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas.
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

2.5 ANALISIS Y DISEÑO ORIENTADO A OBJETOS.

2.5.1 ANÁLISIS ORIENTADO A OBJETOS.

El proceso de análisis comienza con la definición de los casos de uso. La técnica de modelado de clase, responsabilidades, colaboraciones se aplica para documentar las clases atributos y operaciones. También se proporciona una vista inicial de las colaboraciones que ocurren entre objetos. El siguiente paso es la clasificación de los objetos y la creación de la jerarquía de clases. “El objeto – relación proporciona información sobre las conexiones” [8].

2.5.2 DISEÑO ORIENTADO A OBJETOS.

El diseño orientado a objetos traduce el modelo del análisis orientado a objetos del mundo real. Para el diseño orientado a objetos existen métodos como ser UML [12] y otros métodos aproximan el proceso de diseño mediante dos niveles de abstracción: diseño de subsistemas (arquitectura), el diseño de objetos individuales [8].

2.6 HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO DEL SOFTWARE.

2.6.1 ARQUITECTURA CLIENTE/SERVIDOR.

La arquitectura cliente servidor es una relación entre procesos corriendo en máquinas separadas, que interactúan mediante un mecanismo de pasaje de mensajes pedido de servicio y respuesta WEB. Donde el servidor es un proveedor de servicios y el cliente es un consumidor de servicios.

Cliente, es el que inicia un requerimiento de servicio, donde el requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de las redes LAN [13] o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para los clientes[14].

Servidor Web, es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor Web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página Web o información de todo tipo de acuerdo a los comandos solicitados. En este punto es necesario aclarar lo siguiente: mientras que comúnmente se utiliza la palabra servidor para referirnos a

una computadora con un software servidor instalado, en estricto rigor un servidor es el software que permite la realización de las funciones descritas. El servidor vendría a ser la "casa" de los sitios que visitamos en la Internet. Los sitios se alojan en computadoras con servidores instalados, y cuando un usuario los visita son estas computadoras las que proporcionan al usuario la interacción con el sitio en cuestión. Gracias a los avances en conectividad y la gran disponibilidad de banda ancha, hoy en día es muy común establecer los servidores Web dentro de la propia empresa, sin tener que recurrir a caros alojamientos en proveedores externos.

2.6.2 MODELO VISTA CONTROLADOR.

Un problema muy frecuente para los programadores es que tienen problema en la solución o mejora de código existente. Para los casos en los que hay que resolver un problema parecido a algo que ya tienen hecho, mejorar el aspecto de un programa, mejorar su algoritmo. Esta tarea se facilita mucho si a la hora de programar tienen la precaución de separar el código en varias partes que sean susceptibles de ser reutilizadas sin modificaciones.

En casi cualquier programa que hagan podemos encontrar tres partes bien diferenciadas:

- a) Por un lado, está el problema que se trata de resolver. El problema suele ser independiente de cómo queramos que nuestro programa recoja los resultados o cómo queremos que los presente. El código constituiría el modelo.
- b) Otra parte clara es la presentación visual que queramos hacer de la aplicación. El código es la vista. La llamaré interfaz gráfica por ser lo más común, pero podría ser de texto, de comunicaciones con otro programa externo, con la impresora.

- c) La tercera parte de código es aquel código que toma decisiones, algoritmos. Es código que no tiene que ver con las ventanas visuales ni con las reglas de nuestro modelo.

Tras este tipo de ordenación, si se quiere reaprovechar cosas en futuros programas, está claro que el modelo debe ser independiente. Las clases (o funciones y estructuras) del modelo no deben ver a ninguna clase de los otros grupos. De esta forma podremos compilar el modelo en una librería independiente que podremos utilizar en cualquier programa que hagamos. Siguiendo con el orden de posibilidad de reutilización, el controlador podría (y suele) ver clases del modelo, pero no de la vista. De esta forma, el cambio de interfaces gráficas no implicará retocar el algoritmo y recompilarlo, con los consiguientes riesgos de estropearlo además del trabajo del retoque.

La vista es lo más cambiante, así que podemos hacer que vea clases del modelo y del controlador. Si cambiamos algo del controlador o del modelo, es bastante seguro que tengan como mínimo que recompilar la interface gráfica.

2.6.3 PROGRAMACIÓN EN TRES CAPAS.

La programación por capas es un estilo de programación, el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

El diseño de sistemas informáticos actual suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más utilizado actualmente es el diseño en

tres niveles (o en tres capas):

- a) **Capa de presentación:** Es la que visualiza el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable, entendible y fácil de usar para el usuario.

- b) **Capa de negocio:** Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio, pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

- c) **Capa de datos:** Es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en una única computadora. Si bien lo más usual es que haya una multitud de computadoras en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor).

Las capas de negocio y de datos pueden residir en la misma computadora, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más computadoras. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varias computadoras las cuales recibirán las peticiones de la computadora en que resida la capa de negocio.

2.6.4 LENGUAJE UNIFICADO DE MODELADO – UML.

El lenguaje unificado de modelado se ha vuelto el lenguaje de modelado estándar usado en análisis y diseño orientado a objetos.

Popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software. Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación. Para ello utiliza varios tipos diferentes de diagramas.

Algunos programas gratuitos para modelar en UML son: ArgoUML, Día, gModeler, MonoUML, StarUML, TCM, Umbrello Herramienta, UMLet.

UML no es un método de desarrollo. Es un estándar para representar los diagramas del análisis al diseño y de este al código. No son una serie de pasos que llevan a producir código a partir de unas especificaciones. UML al no ser un método de desarrollo es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

UML nació durante los ochenta y principios de los noventa, Grady Booch,

James Rumbaugh, e Ivar Jacobson trabajaban por separado en desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento. Booch había escrito "Object-Oriented Analysis and Design with Applications" un libro de referencia en el análisis y diseño orientado a objetos desarrollando su propia notación. Por su parte James Rumbaugh había desarrollado su propia notación de diseño orientado a objetos llamada OMT (Object Modeling Technique) en su libro "Object-Oriented Modeling and Design". Por otro lado Jacobson se había revelado como un visionario del análisis (padre de los casos de uso) y sobre todo del diseño orientado a objetos, sorprendiendo a todo el mundo en "Object-Oriented Software Engineering: A Use Case Driven Approach". A mediados de los noventa empezaron a intercambiar documentos y trabajar en conjunto produciendo grandes avances en el modelado de sistemas orientados a objetos. En 1994 Rational contrató a Rumbaugh en donde ya trabajaba Booch, un año después Jacobson se unía a ellos en Rational. En 1997 salió a la luz la versión 1.0 de UML.

Objetivos del UML.

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática se diseñó prácticamente para ser un estándar de entendimiento común entre distintas metodologías orientadas a objetos.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver

requisitos dirigidos por casos de uso.

- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

2.6.5 APACHE.

Servidor Web de distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996, con una implementación actual del 50% del total de servidores Web del mundo (hasta agosto de 2007). Apache fue la primera alternativa viable para el servidor Web de Netscape Communications, actualmente conocido como Sun Java System Web Server. Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation. La aplicación permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare y los sistemas basados en Unix. La versión 2 del servidor Apache fue una reescritura sustancial de la mayor parte del código de Apache 1.x, enfocándose en una mayor modularización y el desarrollo de una capa de portabilidad, el Apache Portable Runtime. Apache 2.x incluyó multitarea en UNIX [15], mejor soporte para plataformas no Unix (como Windows), una nueva API Apache y soporte para IPv6.

Características de Apache:

- Soporte para los lenguajes Perl, python, tcl y PHP
- Módulos de autenticación: mod_access, mod_auth y mod_digest
- Soporte para SSL y TLS
- Permite la configuración de mensajes de errores personalizados y negociación de contenido
- Permite autenticación de base de datos basada en SGBD

Uso de Apache es principalmente usado para servir páginas Web estáticas y dinámicas en la WWW. Apache es el servidor Web del popular sistema XAMP, junto con MySql y los lenguajes de programación PHP/ Perl/ Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP.

2.6.6 PHP.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas [16].

PHP es un acrónimo recursivo que significa PHP **Hypertext Pre-processor**. Fue creado originalmente por Rasmus Lerdof en 1994, sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar para PHP al no haber una especificación formal. Publicado bajo la **PHP License**, la **Free Software Foundation** considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser insertado dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor Web. La más reciente versión principal del PHP fue la versión 5.2.6 de 1 de mayo de 2008. Cuando el cliente hace una petición al servidor para que le envíe una página Web, el servidor ejecuta el intérprete de PHP. Éste procesa el código PHP solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio.

PHP 5.

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine 2. Actualmente existen muchas mejoras en la última versión que incluye todas las ventajas que provee el nuevo Zend Engine 2 como:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects[4].
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente rescrita.
- Mejor soporte a XML (XPath, DOM).
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones.

Ventajas.

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde php5).

Desventajas.

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, Unicode.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.

2.6.7 JAVASCRIPT.

JavaScript es un lenguaje de Alto Nivel con capacidades orientadas a objetos [17], incrustado en los navegadores como Netscape e Internet Explorer, es decir, no se necesita instalar ningún software adicional pues estos navegadores tienen a JavaScript integrado, lo único que se requiere es agregar los scripts en los documentos HTML bajo la sintaxis propia tanto de JavaScript como de HTML. El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.6.8 AJAX.

AJAX Acrónimo de **Asynchronous JavaScript And XML** (JavaScript

asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas [18]. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

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

2.6.9 HOJAS DE ESTILO.

Las hojas de estilo representan un avance importante para los diseñadores de páginas Web, al darles un mayor rango de posibilidades para mejorar la apariencia de sus páginas [19]. En los entornos científicos en que la Web fue concebida, la gente estaba más preocupada por el contenido de sus páginas que por su presentación. A medida que la Web era descubierta por un espectro mayor de personas de distintas procedencias, las limitaciones del HTML se convirtieron en fuente de continua frustración, y los autores se vieron forzados a superar las limitaciones estilísticas del HTML. Aunque las intenciones han sido buenas mejorar la presentación de las páginas Web, las técnicas para conseguirlo han tenido efectos secundarios negativos.

Entre estas técnicas, que dan buenos resultados para algunas personas, algunas veces, pero no siempre ni para todas las personas, se incluyen:

- La utilización de extensiones propietarias del HTML
- Conversión del texto en imágenes
- Utilización de imágenes para controlar el espacio en blanco
- La utilización de tablas para la organización de las páginas
- Escribir programas en lugar de usar HTML

Estas técnicas incrementan considerablemente la complejidad de las páginas Web, ofrecen una flexibilidad limitada, sufren de problemas de interoperabilidad, y crean dificultades para las personas con discapacidades.

Las hojas de estilo resuelven estos problemas al mismo tiempo que reemplazan al limitado rango de mecanismos de presentación del HTML. Con las hojas de estilo es más fácil especificar la cantidad de espacio entre líneas, el sangrado de las líneas, los colores a utilizar para el texto y el fondo, el tamaño y estilo de las fuentes, y otros muchos detalles.

2.6.10 MySQL.

MySQL es un sistema de gestión de base de datos, multi-hilo y multi usuario [20]. La siguiente es una lista de algunas de las características de MySQL:

- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.

Uso completo de multi-threaded mediante threads de kernel. Pueden usarse fácilmente múltiple CPUs si están disponibles.

Puede mezclar tablas de distintas bases de datos en la misma consulta (como un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite la verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

Soporte a grandes bases de datos. Se ha usado MySQL Server con bases de datos que contienen 50 millones de registros. También existen usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.

2.6.11 LIBRERÍAS ADODB.

Las funciones de acceso a base de datos en PHP no están estandarizadas. ADOdb es un conjunto de librerías de bases de datos para PHP y Python. Esta permite a los programadores desarrollar aplicaciones Web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada llamada a la base de datos realizada por la aplicación. Actualmente soporta MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, SQLite, Interbase (versiones de Firebird y Borland), Foxpro, Access, ADO, DB2, SAP DB and ODBC. Se puede conectar a Postgress y CacheLite vía ODBC. Este programa tiene licencia dual BSD-Style y LGPL. Esto significa que lo puedes usar tanto en productos comerciales como en propietarios.

2.7 CALIDAD DEL SOFTWARE.

Todas las metodologías y herramientas tienen un único fin producir software de gran calidad. Seguidamente se presentan algunas definiciones de Calidad de Software según algunos autores: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [8].

El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas.

Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad. Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad, existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad.

Para realizar software de calidad se debe realizar en todas las fases de desarrollo del software un aseguramiento de calidad y se define de esta manera: El aseguramiento de calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad. El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla y no después. Pretende dar confianza en que el producto tiene calidad, está presente en métodos y herramientas de análisis, diseño, programación y prueba inspecciones técnicas formales en

todos los pasos del proceso de desarrollo del software. Control de la documentación del software y de los cambios realizados, procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos), mecanismos de medida (métricas), registro de auditorías y realización de informes. Existen actividades para el aseguramiento de calidad del software, métricas de software para el control del proyecto, verificación y validación del software a lo largo del ciclo de vida, incluye las pruebas y los procesos de revisión e inspección. [9]

Existen factores que determinan la calidad del software y se clasifican en tres grupos:

a) **Operaciones del producto:** características operativa

- Corrección (¿Hace lo que se le pide?), el grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
- Fiabilidad (¿Lo hace de forma fiable todo el tiempo?), el grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.
- Eficiencia (¿Qué recursos hardware y software necesito?), La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.
- Integridad (¿Puedo controlar su uso?), el grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
- Facilidad de uso (¿Es fácil y cómodo de manejar?), el esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

b) **Revisión del producto:** capacidad para soportar cambios

- Facilidad de mantenimiento (¿Puedo localizar los fallos?), el esfuerzo requerido para localizar y reparar errores.
- Flexibilidad (¿Puedo añadir nuevas opciones?), el esfuerzo requerido para modificar una aplicación en funcionamiento
- Facilidad de prueba (¿Puedo probar todas las opciones?), el esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

c) **Transición del producto:** adaptabilidad a nuevos entornos

- Portabilidad (¿Podré usarlo en otra máquina?), el esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo
- Reusabilidad (¿Podré utilizar alguna parte del software en otra aplicación?), grado en que partes de una aplicación pueden utilizarse en otras aplicaciones
- Interoperabilidad (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?), el esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos

2.7.1 ISO 9126.

ISO 9126 es un estándar internacional para la evaluación del Software. El estándar ISO 9126 está conformado en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y sub-características de la siguiente manera:

a) **Funcionalidad**, un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.

- Idoneidad
- Exactitud
- Interoperabilidad
- Seguridad

b) **Fiabilidad**, un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período de tiempo establecido.

- Madurez
- Recuperabilidad
- Tolerancia a fallos

c) **Usabilidad**, conjuntos de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- Aprendizaje
- Comprensión
- Operatividad

d) **Eficiencia**, conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

- Comportamiento en el tiempo
- Comportamiento de recursos

e) **Mantenimiento**, conjunto de atributos relacionados con el esfuerzo necesitado para modificar las especificaciones.

- Estabilidad
- Facilidad de análisis
- Facilidad de cambio
- Facilidad de pruebas

f) **Movilidad**, conjunto de atributos relacionados con la habilidad del software para ser transferido desde un entorno a otro.

- Capacidad de instalación
- Capacidad de reemplazamiento
- Adaptabilidad

La sub-característica Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad. Cada sub-característica (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software. Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software. El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada

organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad. Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas). Métricas externas son aquellas aplicables al software en ejecución. La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales. Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso. Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Ellos describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Ellos describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Ellas son definidas y usadas para proveer una escala y método para la medida.

ISO 9126 distingue entre fallos y no conformidad, siendo un fallo el no cumplimiento de los requisitos previos, mientras que la no conformidad afecta a los requisitos especificados. Una distinción similar es hecha entre la validación y la verificación.

CAPITULO 3

MARCO APLICATIVO

3.1 INTRODUCCION.

Los modelos pertenecientes a la metodología RUP tendrán una representación de acuerdo a las necesidades del sistema a implementar.

3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL.

Antes de realizar las fases del modelo se debe dar la situación actual de la institución en cuanto a las actividades concernientes al seguimiento académico.

3.2.1 DESCRIPCIÓN DE FUNCIONES.

Director.

La máxima autoridad dentro de la institución las funciones y responsabilidades de su cargo: La administración del plantel administrativo y estudiantil de la academia, llegando a realizar las siguientes tareas: Diseño de las políticas académicas, resguardo de la documentación de los estudiantes e historial académico, inscripciones, entre otras.

Secretaria.

La persona que coadyuva en el desarrollo de las funciones administrativas y académicas de la academia. Su función conlleva las siguientes responsabilidades, recepción y registro de documentos del área académica, elaboración de los informes relacionados a los ingresos monetarios, cobro de

pensiones y matrícula, la administración de notas, la administración manual y sistémica de la documentación referente al historial académico.

3.2.2 PROCESO ACADÉMICOS.

Director.

- Requiere reporte de estudiantes inscritos por materia.
- Requiere reporte del rendimiento académico.

Docentes.

- Solicita lista de estudiantes en su asignatura.
- Llena su correspondiente acta de notas.

Secretaria.

- Realiza control de inscripción por periodo.
- Recepción y archivo de acta de notas.
- Recepción y archivo de documentación de estudiantes por especialidad.
- Inscribe estudiantes nuevos.

Estudiante.

- Solicita inscripción
- Solicita calificaciones
- Consulta de calificaciones
- Consulta de horarios
- Cambio de contraseña

3.3 FASE DE COMIENZO O INICIO CAPTURA DE REQUISITOS.

Esta fase esta principalmente dirigida al buen entendimiento de requisitos y determinación del alcance del proyecto.

3.3.1 ALCANCE DEL PROYECTO.

En las funciones del sistema están lo que el sistema deberá de hacer, y con su determinación llegamos a determinar los límites y alcances del proyecto en curso.

Funciones Básicas.

Ingreso al sistema.

- R1.** El usuario debe ingresar su usuario y contraseña, la aplicación comprobara la existencia y concordancia de los datos.
- R2.** La aplicación mostrara un menú de funciones de acuerdo al tipo de usuario que ingrese al sistema.

Gestión de Estudiantes.

- R3.** Para registrar un nuevo estudiante el administrador debe ingresar cedula, nombres, apellidos, fecha de nacimiento, ciudad de nacimiento, genero nacionalidad, personas con las que vive, escuela de procedencia y una observación del mismo.
- R4.** La aplicación asignara un código único a cada estudiante, que se maneja posteriormente en todos los procesos de la institución.
- R5.** Al registrar al nuevo estudiante la aplicación generara un usuario y contraseña para el mismo, por defecto será el código descrito en el

requisito 4, pero que puede modificarse de acuerdo al criterio del estudiante.

- R6.** El administrador puede modificar datos de un estudiante excepto el código generado por el sistema.
- R7.** Al eliminar la información de un estudiante, ésta quedara fuera de la aplicación. En caso de que el estudiante se encuentre matriculado, los datos de calificaciones quedaran intactos.
- R8.** La aplicación muestra una lista de todos los estudiantes registrados, donde se puede realizar una búsqueda por el código, nombres, apellidos, la escuela de procedencia, personas con las que vive y la observación.

Gestión de Familiares.

- R9.** Para ingresar la información de familiar el administrador debe llenar datos como: cedula, nombres, apellidos, dirección de domicilio, teléfono del domicilio, profesión, lugar de trabajo, teléfono móvil y observación.
- R10.** El administrador puede modificar la información de un familiar excepto su cedula.
- R11.** Al eliminar la información de un familiar, esta se borrara de la lista de familiares, pero los datos se mantendrán en cualquier otro proceso que esté interviniendo.
- R12.** La aplicación muestra una lista de todos los familiares registrados, donde se puede realizar una búsqueda por la cedula, nombres, apellidos, dirección de domicilio, profesión y el teléfono del domicilio.

Gestión de Parentescos.

- R13.** El administrador puede registrar un parentesco, para esto la aplicación

permite realizar una búsqueda de estudiantes y familiares por sus nombres y/o apellidos.

- R14.** El administrador solo puede modificar la relación de parentesco entre el estudiante y familiar seleccionado.
- R15.** La aplicación muestra una lista de todos los parentescos registrados, donde se puede realizar una búsqueda por el nombre del estudiante, el nombre del familiar o el parentesco.

Gestión de Profesores.

- R16.** Para registrar un nuevo profesor el administrador debe llenar datos como: cedula, nombres, apellidos, dirección del domicilio, teléfono del domicilio, teléfono móvil, profesión y observación.
- R17.** El administrador puede modificar la información de un profesor de la institución, excepto la cedula.
- R18.** Al eliminar la información de un profesor, esta se borra de la lista de profesores, pero los datos se mantendrán en cualquier otro proceso que esté interviniendo.
- R19.** La aplicación muestra una lista de todos los profesores registrados, donde se puede realizar una búsqueda por la cedula, nombres, apellidos dirección, profesión y teléfono de domicilio.

Gestión de Categorías.

- R20.** Para registrar una nueva categoría el administrador llenara los siguientes datos: código, nombre y estado de la misma.
- R21.** El administrador puede modificar la información de una categoría, excepto el código.
- R22.** La aplicación permite eliminar una categoría seleccionada por el administrador, en caso de existir materias asociadas a dicha categoría

están también serán eliminadas.

R23. La aplicación muestra una lista de todas las categorías, donde se puede realizar una búsqueda por el código, nombre o estado de la misma.

Gestión de Materias.

R24. El administrador puede registrar la información de materias para lo cual llenara los siguientes campos: código de la materia, nombre, categoría, profesor, observación y estado.

R25. El administrador puede modificar la información de una materia, excepto el código.

R26. La aplicación permite eliminar una materia seleccionada por el administrador, la información relacionada con la materia también se ver afectada.

R27. La aplicación muestra una lista de todas las materias, donde se puede realizar una búsqueda por código, nombre, categoría, profesor, observación y estado.

Gestión de Periodos.

R28. El administrador puede registrar la información de periodos para lo cual llenara los siguientes campos: código del periodo, fecha de inicio, fecha de finalización y observación.

R29. El sistema asignara por defecto el periodo en curso o vigencia en base a las fechas de inicio y fechas de finalización del periodo.

R30. El administrador puede modificar la información de un periodo, excepto el código del mismo.

R31. La aplicación permite eliminar un periodo seleccionado por el administrador, toda la información relacionada con el periodo se verá

afectada.

R32. La aplicación muestra una lista de todos los periodos existentes, donde se puede realizar una búsqueda por código, fecha de inicio, fecha de finalización y estado.

Gestión de Grados.

R33. El administrador puede registrar la información de los grados existentes en la institución para lo cual llenara los siguientes campos: grado, paralelo, dirigente, ubicación y observación.

R34. El administrador puede modificar la toda la información del grado seleccionado.

R35. La aplicación permite eliminar un grado seleccionado por el administrador, toda la información relacionada con éste se verá afectada.

R36. La aplicación muestra una lista de todos los grados existentes, donde se puede realizar una búsqueda por grado, paralelo, dirigente, ubicación y observación.

Gestión de Horarios.

R37. El administrador puede registrar la información de los horarios para lo cual seleccionara el periodo y grado, adicional a esto se hará una búsqueda del profesor encargado del horario.

R38. En cada horario se podrá registrar el detalle del horario, es decir la materia, el día y la hora correspondientes.

R39. El sistema evitara sobreescribir las horas y días ocupados en el detalle del horario.

R40. El detalle del horario estará definido en 5 días laborables cada uno con 8 horas de clases.

- R41.** El administrador puede modificar la toda la información del horario seleccionado, así como su respectivo detalle.
- R42.** La aplicación muestra una lista de todos los horarios existentes, donde se puede realizar una búsqueda por periodo, grado y/o profesor.

Matriculación.

- R43.** El administrador puede registrar la matricula de un estudiante ingresando los datos solicitados en el requisito R3, en grado determinado y un periodo específico.
- R44.** El sistema controlara que el alumno tenga registrado su respectivo representante.
- R45.** El sistema permitirá realizar la búsqueda de estudiantes por el nombre, apellido o su respectivo código.
- R46.** El administrador únicamente puede modificar el campo observaciones de las matriculadas registradas en el sistema.
- R47.** La aplicación permitirá eliminar la matricula seleccionada, esto afectara la información relacionada con la misma.
- R48.** La aplicación muestra una lista de todas las matriculas existentes, donde se puede realizar una búsqueda por numero de matrícula, folio, alumno, representante, periodo, grado y las observaciones.

Gestión de Calificaciones.

- R49.** La aplicación mostrara la lista de grados registrados en los horarios y de acuerdo al periodo en vigencia detallado en el requisito 30.
- R50.** Dentro de cada grado se mostrara un detalle de las materias asignadas a éste, en base al detalle del horario registrado.
- R51.** En cada materia habrá un detalle de los alumnos matriculados en un periodo determinado y grado seleccionado con las respectivas notas y

aportes a registrarse.

- R52.** El registro de calificaciones empieza con el ingreso a la opción de calificaciones, al hacer esto la aplicación mostrara la información detallada en el requisito 51, el administrador seleccionara el grado deseado y el sistema mostrara las materias en base al requisito 52; a continuación el usuario escogerá la materia donde desee registrar las calificaciones haciendo que el navegador muestre una lista de estudiante según como lo describe el requisito 53. El sistema controlara que las calificaciones estén en un rango de 0 a10 puntos.
- R53.** Las calificaciones estarán restringidas en base a los parciales, por ejemplo al principio solo puede registrar las notas del primer periodo del primer quimestre, una vez ingresada dicha calificación el sistema habilitara el registro del segundo periodo. Posteriormente cuando se ingrese la calificación del segundo periodo la aplicación bloqueara la calificación del primer periodo y así sucesivamente con las demás calificaciones.

3.3.2 MODELADO DEL SISTEMA.

En la figura 1 se muestra el modelo del sistema:

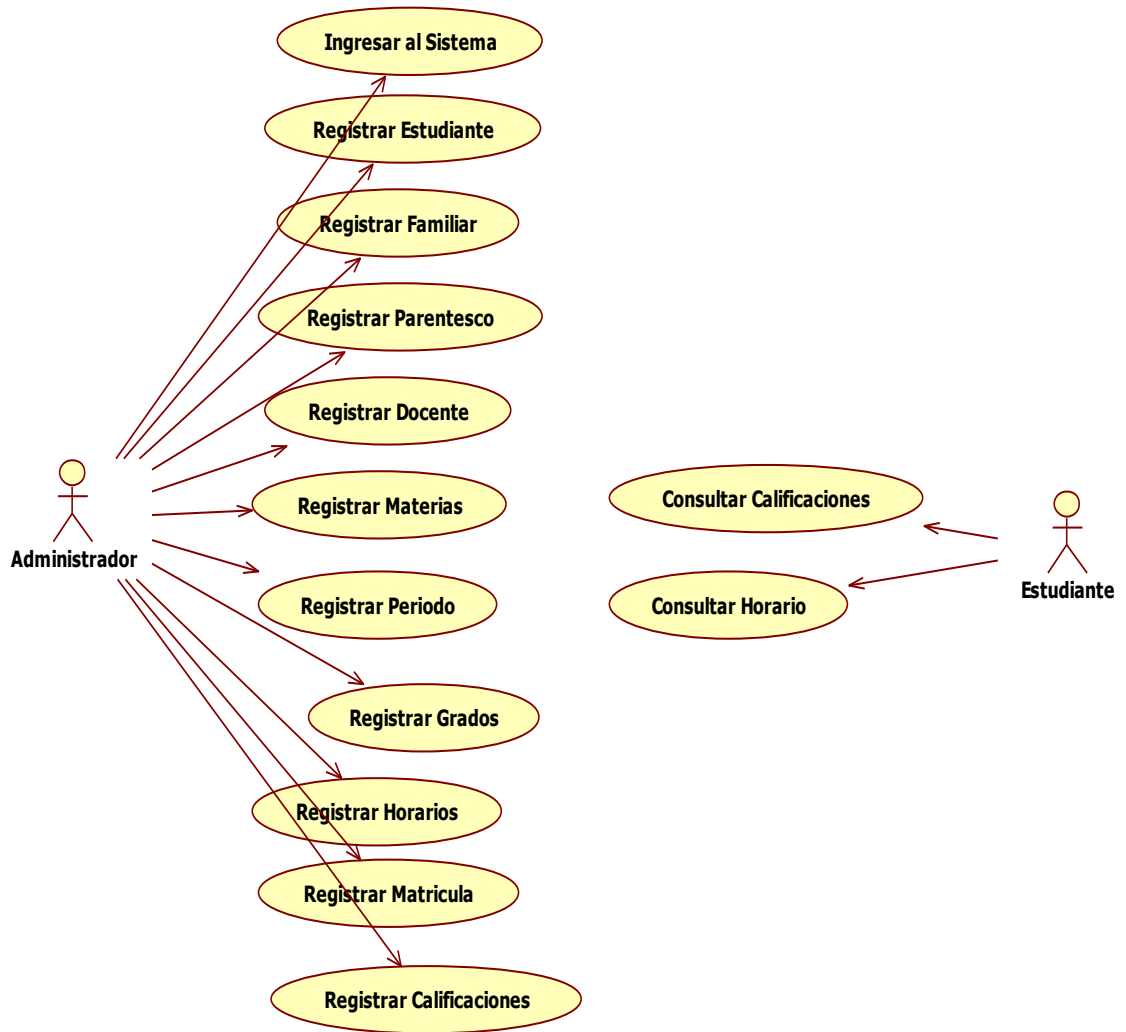


Figura 1. Modelado del Sistema.

3.3.3 ACTORES.

Administrador del sistema.

Es el encargado de dar soporte al sistema, incluyendo crear, modificar y eliminar usuarios.

Administrador.

Podrá registrar estudiantes nuevos, registrar familiares, registrar docentes, registrar materias, registrar grados, registrar horarios, registrar matriculas, registrar calificaciones y generar reportes

Estudiante.

Solicita su inscripción y realiza su seguimiento académico.

3.3.4 CASOS DE USO.

Caso de uso 1: Ingresar al sistema.

Caso de uso:	Ingresar al sistema.
Actores:	Administrador del sistema, administrador, estudiante.
Propósito:	Verificar la validez del usuario.
Resumen:	Un usuario ingresa a la página del sistema, ingresa su identificador único y su contraseña para poder ingresar al sistema.
Tipo:	Primario, esencial.
Referencias Cruzadas:	R1 y R2.

Tabla 2. Caso de Uso 1: Ingresar al Sistema.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-Este caso de uso comienza cuando un usuario ingresa a la página de autenticación del sistema.	2.-El sistema muestra una pantalla donde debe de introducir su identificador y su contraseña.
3.-El usuario ingresa su identificador y contraseña.	
4.-Envía la información.	5.-El sistema verifica la existencia del usuario en la base de datos. 6.-El sistema encuentra al usuario. 7.-Muestra en pantalla la interfaz dependiendo al tipo de usuario.

Tabla 3. Descripción del evento: Ingresar al Sistema.

Cursos alternos.

Línea 6.- El usuario no está registrado en el sistema.

Línea 7.- El sistema muestra la pantalla de identificación del usuario.

Caso de uso 2: Registrar estudiante.

Caso de uso:	Registrar Estudiante.
Actores:	Estudiante (iniciador), administrador.
Propósito:	Registrar la inscripción de estudiantes nuevos a la academia.
Resumen:	El estudiante se aproxima a instalaciones de la academia solicitando su incorporación, el administrador solicita sus datos personales y los documentos a presentar, posteriormente los registra.
Tipo:	Principal, esencial.

Referencias Cruzadas:	R2, R3, R4, R5, R6, R7, R8.
-----------------------	-----------------------------

Tabla 4. Caso de Uso 2: Registrar Estudiante.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
<p>1.-El estudiante se aproxima a la secretaria solicitando su incorporación.</p> <p>2.-El administrador solicita sus datos personales y los documentos necesarios.</p> <p>3.-El administrador registra la información.</p> <p>4.-El administrador envía la información.</p> <p>8.-El administrador acepta las entradas.</p>	<p>5.-El sistema verifica la validez de los datos.</p> <p>6.-Registra la información.</p> <p>7.-Muestra en pantalla la información entrante.</p>

Tabla 5. Descripción del evento: Registrar Estudiante.

Caso de uso 3: Registrar familiar.

Caso de uso:	Registrar Familiar.
Actores:	Administrador, estudiante.
Propósito:	Registrar la inscripción de familiares al sistema.
Resumen:	El administrador solicita datos personales de los familiares, estos se verifican y luego se registran.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R9, R10, R11, R12.

Tabla 6. Caso de uso 3: Registrar familiar.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador solicita datos personales y los documentos necesarios. 2.-El administrador registra la información. 3.-El administrador envía la información. 7.-El administrador acepta las entradas.	4.-El sistema verifica la validez de los datos. 5.-Registra la información. 6.-Muestra en pantalla la información entrante.

Tabla 7. Descripción del Evento: Registrar Familiar.

Caso de uso 4: Registrar parentesco.

Caso de uso:	Registrar parentesco.
Actores:	Administrador, estudiante.
Propósito:	Registrar la inscripción de familiares al sistema.
Resumen:	El administrador registra la relación entre los estudiantes y familiares registrados en el sistema.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R13, R14, R15, R16.

Tabla 8. Caso de uso 4: Registrar parentesco.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador relaciona a cada estudiante con su familiar. 2.-El administrador registra la información. 3.-El administrador envía la información. 7.-El administrador acepta las entradas.	4.-El sistema verifica la validez de los datos. 5.-Registra la información. 6.-Muestra en pantalla la información entrante.

Tabla 9. Descripción del evento: Registrar parentesco.

Caso de uso 5: Registrar docente.

Caso de uso:	Registrar Docente.
Actores:	Docente (iniciador), administrador.
Propósito:	Registrar la inscripción de profesores nuevos en la institución.
Resumen:	El docente se aproxima a instalaciones de la institución solicitando su incorporación, el administrador solicita sus datos personales y los documentos a presentar, posteriormente los registra.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R17, R18, R19, R20.

Tabla 10. Caso de uso 5: Registrar docente.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El docente se aproxima al administrador solicitando su incorporación. 2.-El administrador solicita sus datos personales y los documentos necesarios. 3.-El administrador registra la información. 4.-El administrador envía la información. 8.-El administrador acepta las	5.-El sistema verifica la validez de los datos. 6.-Registra la información. 7.-Muestra en pantalla la información entrante.

entradas.	
-----------	--

Tabla 11. Descripción del evento: Registrar Docente.

Caso de uso 6: Registrar Materia.

Caso de uso:	Registrar Materia.
Actores:	Administrador.
Propósito:	Registrar la información de las materias en el sistema.
Resumen:	El administrador registra la información de las materias que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R25, R26, R27, R28.

Tabla 12. Caso de uso 6: Registrar Materia.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 13. Descripción del evento: Registrar Materia.

Cursos alternos.

Línea 3.- El código de la materia no se puede repetir, y la información no se registrara.

Caso de uso 7: Registrar Periodo.

Caso de uso:	Registrar Periodo.
Actores:	Administrador
Propósito:	Registrar la información de las materias en el sistema.
Resumen:	El administrador registra la información de las materias que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R29, R30, R31, R32, R33.

Tabla 14. Caso de uso 7: Registrar Periodo.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 15. Descripción del evento: Registrar Periodo

Cursos alternos.

Línea 3.- El código del periodo no se puede repetir, y la información no se registrara.

Caso de uso 8: Registrar Grado.

Caso de uso:	Registrar Grado.
Actores:	Administrador
Propósito:	Registrar la información de los grados en el sistema.
Resumen:	El administrador registra la información de los grados que existen en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R34, R35, R36, R37.

Tabla 16. Caso de uso 8: Registrar Grado.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 17. Descripción del evento: Registrar Grado

Caso de uso 9: Registrar Horario.

Caso de uso:	Registrar Horario.
Actores:	Administrador.
Propósito:	Registrar la información de los horarios en el sistema.
Resumen:	El administrador registra la información de los horarios que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R38, R39, R40, R41, R42, R43.

Tabla 18. Caso de uso 9: Registrar Horario.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 19. Descripción del evento: Registrar Horario.

Cursos alternos.

Línea 3.- Solo se puede registrar un horario para cada grado en cada periodo.

Caso de uso 10: Registrar Matricula.

Caso de uso:	Registrar Matricula.
Actores:	Administrador.
Propósito:	Registrar la información de matriculas en el sistema.
Resumen:	El administrador registra la información de matriculas de acuerdo al periodo en vigencia.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R44, R45, R46, R47, R48, R49, R50.

Tabla 20. Caso de uso 10: Registrar Matricula.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 21. Descripción del evento: Registrar Matricula

Cursos alternos.

Línea 3.- Solo se puede registrar la matricula si existe un periodo en vigencia.

Caso de uso 11: Registrar Calificaciones.

Caso de uso:	Registrar Calificaciones.
Actores:	Administrador.
Propósito:	Registrar la información de calificaciones en el sistema.
Resumen:	El administrador registra las calificaciones por cada grado y materia seleccionados y de acuerdo al
Tipo:	Principal, esencial.
Referencias Cruzadas:	R51, R52, R53, R54.

Tabla 22. Caso de uso 11: Registrar Calificaciones.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 23: Descripción del evento: Registrar Calificaciones

Caso de uso 12: Consultar Calificaciones.

Caso de uso:	Consultar Calificaciones.
Actores:	Estudiante.
Propósito:	Consultar las calificaciones.

Resumen:	El estudiante consulta las calificaciones obtenidas durante todo el periodo lectivo.
Tipo:	Secundario.
Referencias Cruzadas	R54.

Tabla 24. Caso de uso 12: Consultar Calificaciones.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El estudiante ingresa al sistema.	2.-El sistema almacena su usuario. 3.-Consulta la información. 4.-Muestra en pantalla la información.

Tabla 25. Descripción del evento: Consultar Calificaciones

Caso de uso 13: Consultar Horarios.

Caso de uso:	Consultar Horarios.
Actores:	Estudiante.
Propósito:	Consultar el horario del periodo lectivo en vigencia.
Resumen:	El estudiante consulta el horario que le corresponde en base al periodo en vigencia.
Tipo:	Secundario.
Referencias Cruzadas:	R43.

Tabla 26. Caso de uso 13: Consultar Horarios.

Curso normal de eventos

Acción de actores.	Respuesta del sistema.
1.-El estudiante ingresa al sistema.	2.-El sistema almacena su usuario. 3.-Consulta la información 4.-Muestra en pantalla la información.

Tabla 27. Descripción del evento: Consultar Horarios

3.4 FASE DE ELABORACIÓN.

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura y desarrollar el plan del proyecto.

3.4.1 DIAGRAMA DE SECUENCIA DEL SISTEMA.

El diagrama de secuencia del sistema muestra de manera grafica los eventos que propician los actores directos con el sistema. Para su elaboración deberá haberse construido previamente los casos de uso esenciales de los cuales se obtiene el curso normal de eventos teniendo en cuenta los cursos opcionales más interesantes. [5]

Caso de uso 1: Ingresar al sistema.

Curso normal de eventos.

- 1.-Este caso de uso comienza cuando un usuario ingresa a la página de autenticación del sistema.
- 2.-El sistema muestra una pantalla donde debe de introducir su identificador y su contraseña.
- 3.-El usuario ingresa su identificador y contraseña.
- 4.-Envía la información.
- 5.-El sistema verifica la existencia del usuario en la base de datos.
- 6.-El sistema encuentra al usuario.
- 7.-Muestra en pantalla la interfaz dependiendo al tipo de usuario.
- 8.-El usuario ingresa al sistema.

En la figura 2 se muestra el diagrama de secuencia del caso de uso ingresar al sistema.

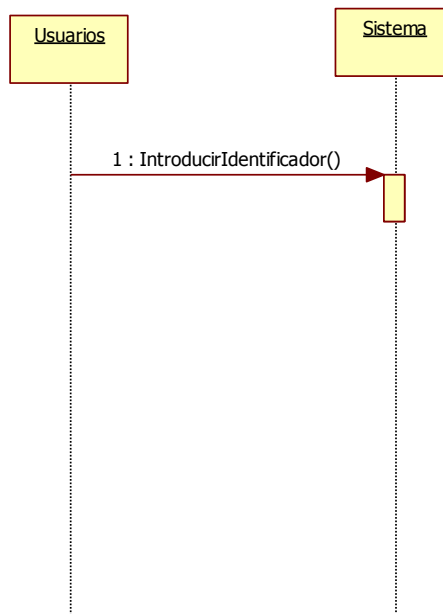


Figura 2. Diagrama de secuencia del caso de uso 1: Ingresar al Sistema.

Caso de uso 2: Registrar estudiante.

Curso normal de eventos.

- 1.-El estudiante se aproxima al administrador solicitando su incorporación.
- 2.-El administrador solicita sus datos personales y los documentos necesarios.
- 3.-El administrador registra la información.
- 4.-El administrador envía la información.
- 5.-El sistema verifica la validez de los datos.
- 6.-Registra la información.
- 7.-Muestra en pantalla la información entrante.
- 8.-El administrador acepta las entradas.

En la figura 3 se muestra el diagrama de secuencia del caso de uso registrar estudiante.

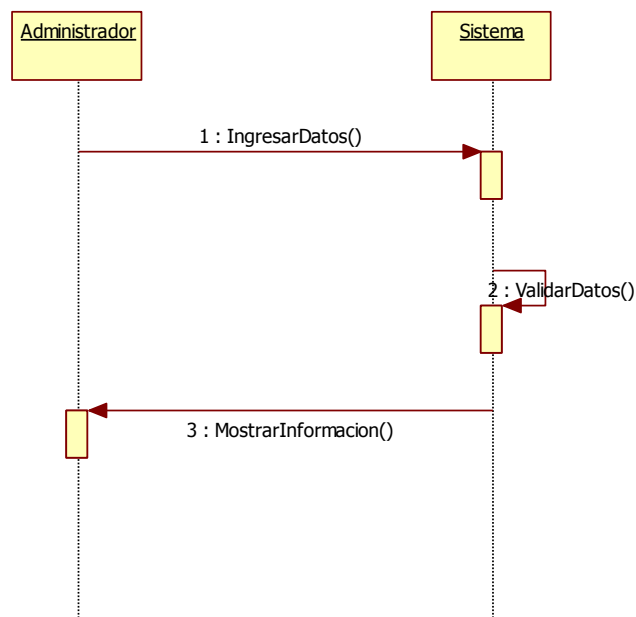


Figura 3. Diagrama de secuencia del caso de uso 2: Registrar Estudiante.

Caso de uso 3: Registrar familiar.

Curso normal de eventos.

- 1.-El administrador solicita datos personales y los documentos necesarios.
- 2.-El administrador registra la información.
- 3.-El administrador envía la información.
- 4.-El sistema verifica la validez de los datos.
- 5.-Registra la información
- 6.-Muestra en pantalla la información entrante.
- 7.-El administrador acepta las entradas.

En la figura 4 se muestra el diagrama de secuencia del caso de uso Registrar Familiar.

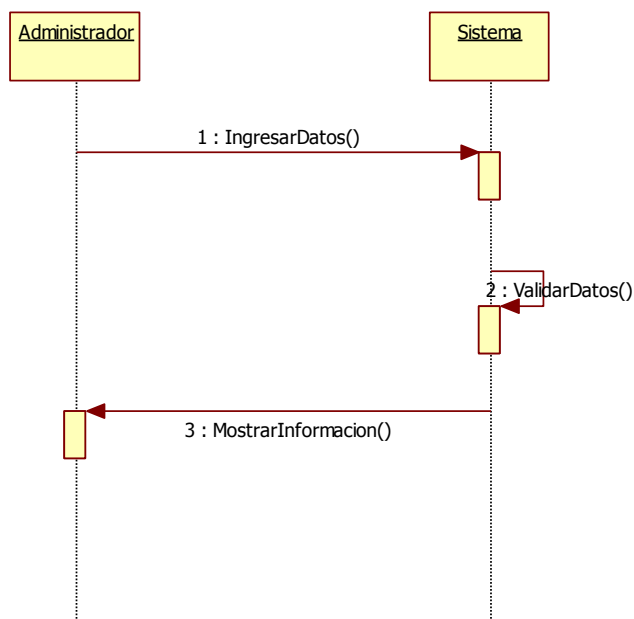


Figura 4. Diagrama de secuencia del caso de uso 3: Registrar Familiar.

Caso de uso 4: Registrar Parentesco.

Curso normal de eventos.

- 1.-El administrador relaciona a cada estudiante con su familiar.
- 2.-El administrador registra la información.
- 3.-El administrador envía la información.
- 4.-El sistema verifica la validez de los datos.
- 5.-Registra la información
- 6.-Muestra en pantalla la información entrante.
- 7.-El administrador acepta las entradas.

En la figura 5 se muestra el diagrama de secuencia del caso de uso Registrar parentesco.

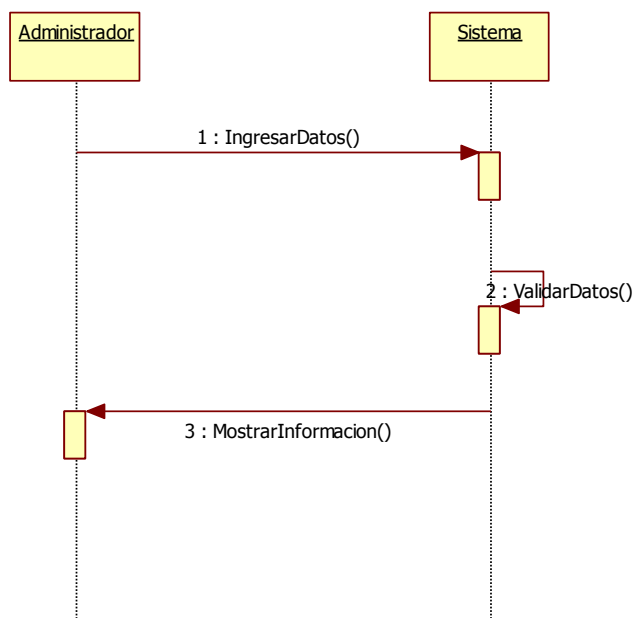


Figura 5. Diagrama de secuencia del caso de uso: Registrar Parentesco.

Caso de uso 5: Registrar Docente.

Curso normal de eventos.

- 1.-El administrador relaciona a cada estudiante con su familiar.
- 2.-El docente se aproxima al administrador solicitando su incorporación.
- 3.-El administrador solicita sus datos personales y los documentos necesarios.
- 4.-El administrador registra la información.
- 5.-El administrador envía la información.
- 6.-El sistema verifica la validez de los datos.
- 7.-Registra la información
- 8.-Muestra en pantalla la información entrante.
- 9.-El administrador acepta las entradas.

En la figura 6 se muestra el diagrama de secuencia del caso de uso Registrar Docente.

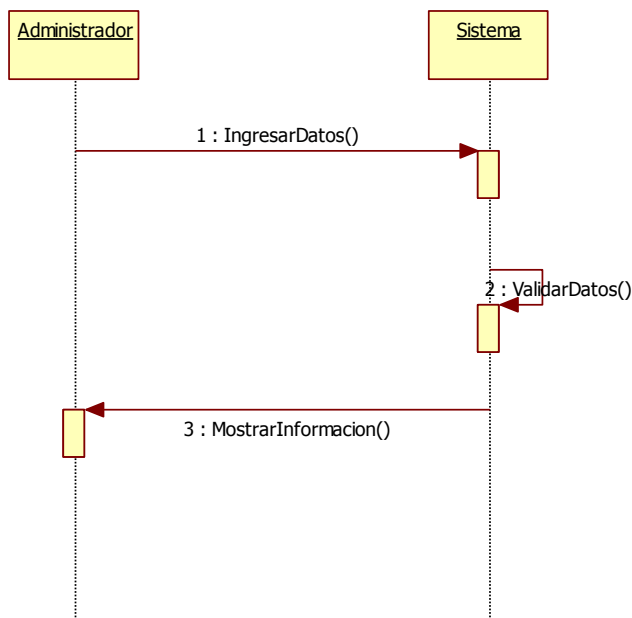


Figura 6. Diagrama de secuencia del caso de uso 5: Registrar Docente.

Caso de uso 6: Registrar Materia.

Curso normal de eventos.

- 1.-El administrador registra la información.
- 2.-El administrador envía la información.
- 3.-El sistema verifica la validez de los datos.
- 4.-Registra la información.
- 5.-Muestra en pantalla la información entrante.
- 6.-El administrador acepta las entradas.

En la figura 7 se muestra el diagrama de secuencia del caso de uso Registrar Materia.

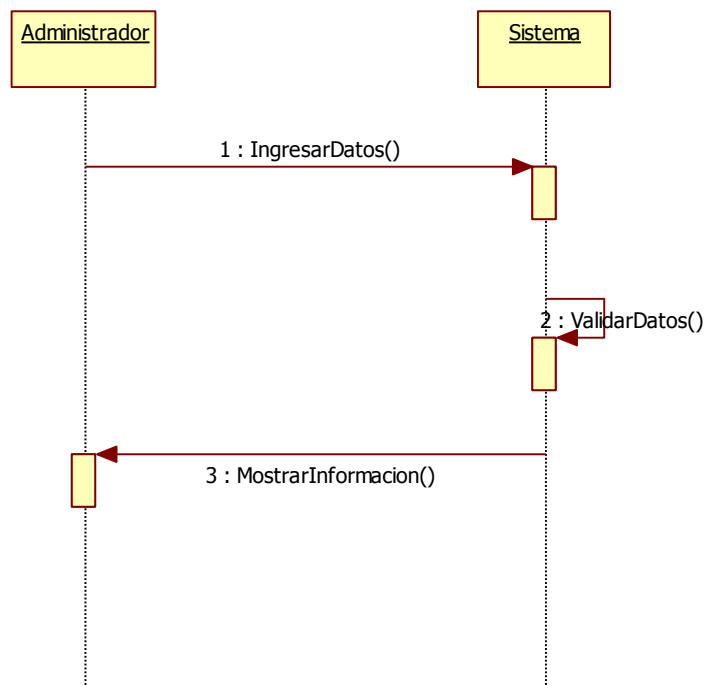


Figura 7. Diagrama de secuencia del caso de uso 6: Registrar Materia.

Caso de uso 7: Registrar Periodo.

Curso normal de eventos.

- 1.-El administrador registra la información.
- 2.-El administrador envía la información.
- 3.-El sistema verifica la validez de los datos.
- 4.-Registra la información.
- 5.-Muestra en pantalla la información entrante.
- 6.-El administrador acepta las entradas.

En la figura 8 se muestra el diagrama de secuencia del caso de uso Registrar Periodo.

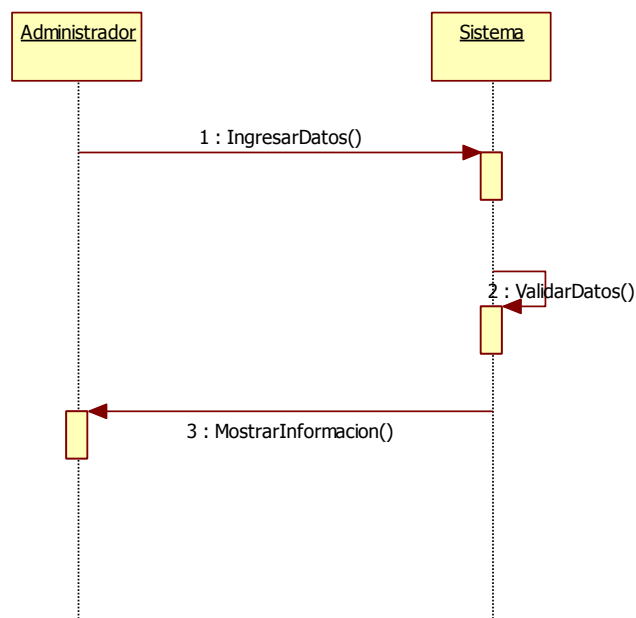


Figura 8. Diagrama de secuencia del caso de uso 7: Registrar Periodo.

Caso de uso 8: Registrar Grado.

Curso normal de eventos.

- 1.-El administrador registra la información.
- 2.-El administrador envía la información.
- 3.-El sistema verifica la validez de los datos.
- 4.-Registra la información.
- 5.-Muestra en pantalla la información entrante.
- 6.-El administrador acepta las entradas.

En la figura 9 se muestra el diagrama de secuencia del caso de uso Registrar Grado.

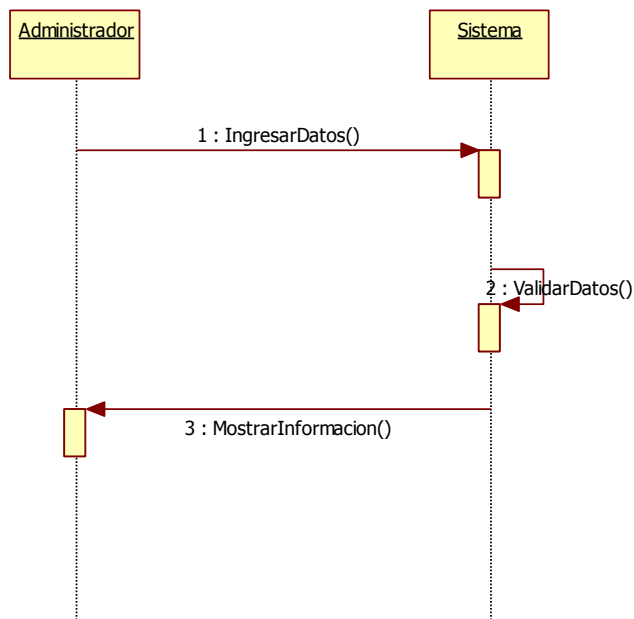


Figura 9. Diagrama de secuencia del caso de uso 8: Registrar Grado.

Caso de uso 9: Registrar Horario.

Curso normal de eventos.

- 1.-El administrador registra la información.
- 2.-El administrador envía la información.
- 3.-El sistema verifica la validez de los datos.
- 4.-Registra la información
- 5.-Muestra en pantalla la información entrante.
- 6.-El administrador acepta las entradas.

En la figura 10 se muestra el diagrama de secuencia del caso de uso Registrar Horario.

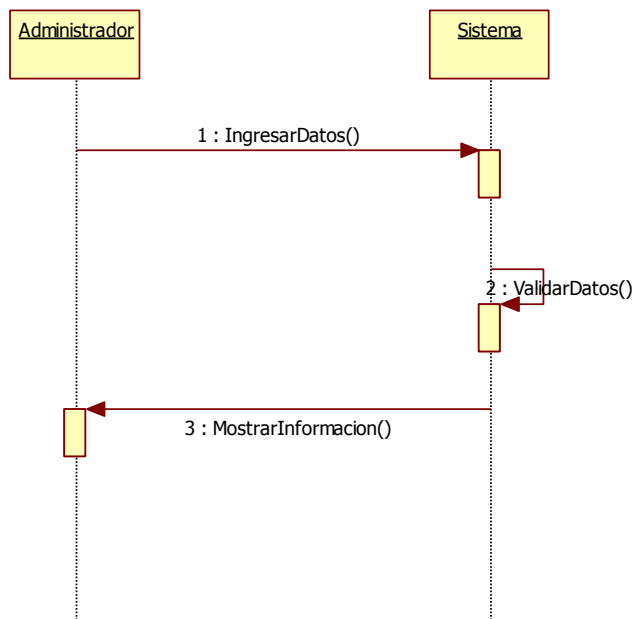


Figura 10. Diagrama de secuencia del caso de uso 9: Registrar Horario.

Caso de uso 10: Registrar Matricula.

Curso normal de eventos.

- 1.-El administrador registra la información.
- 2.-El administrador envía la información.
- 3.-El sistema verifica la validez de los datos.
- 4.-Registra la información
- 5.-Muestra en pantalla la información entrante.
- 6.-El administrador acepta las entradas.

En la figura 11 se muestra el diagrama de secuencia del caso de uso Registrar Matricula.

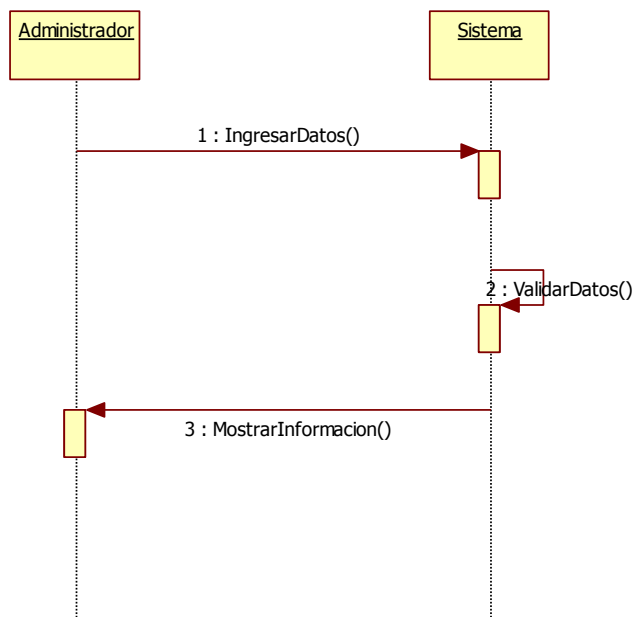


Figura 11. Diagrama de secuencia del caso de uso 10: Registrar Matricula.

Caso de uso 11: Registrar Calificaciones.

Curso normal de eventos.

- 1.- El administrador registra la información.
- 2.- El administrador envía la información.
- 3.- El sistema verifica la validez de los datos.
- 4.- Registra la información
- 5.- Muestra en pantalla la información entrante.
- 6.- El administrador acepta las entradas.

En la figura 12 se muestra el diagrama de secuencia del caso de uso Registrar Calificaciones.

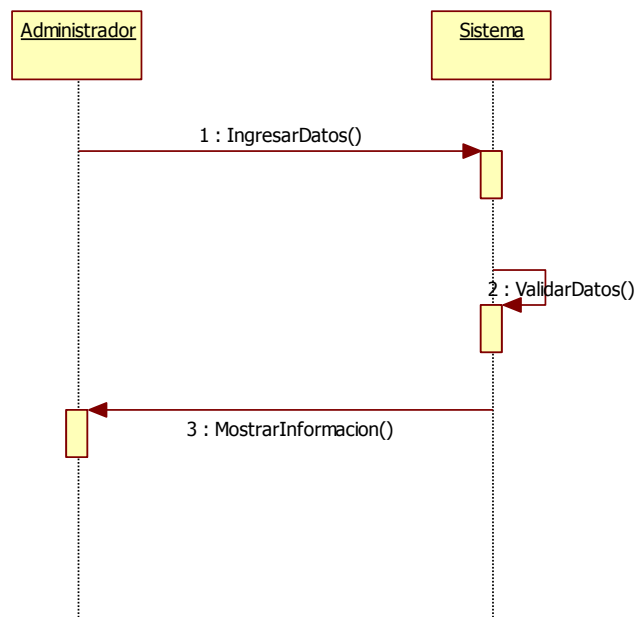


Figura 12. Diagrama de secuencia del caso de uso 11: Registrar Calificaciones.

Caso de uso 12: Consultar Calificaciones.

Curso normal de eventos.

- 1.-El estudiante ingresa al sistema.
- 2.- El sistema almacena su usuario.
- 3.- Consulta la información
- 4.- Muestra en pantalla la información.

En la figura 13 se muestra el diagrama de secuencia del caso de uso Consultar Calificaciones.

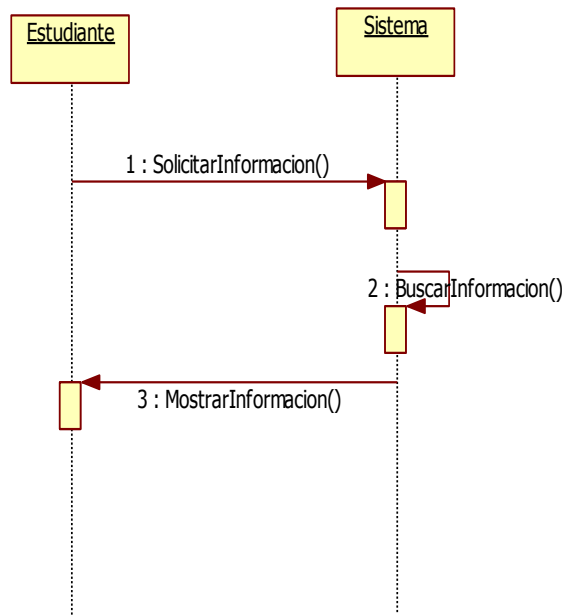


Figura 13. Diagrama de secuencia del caso de uso: Consultar Calificaciones.

Caso de uso 13: Consultar Horarios.

Curso normal de eventos.

- 1.-El estudiante ingresa al sistema.
- 2.-El sistema almacena su usuario.
- 3.-Consulta la información
- 4.-Muestra en pantalla la información.

En la figura 14 se muestra el diagrama de secuencia del caso de uso Consultar Horarios.

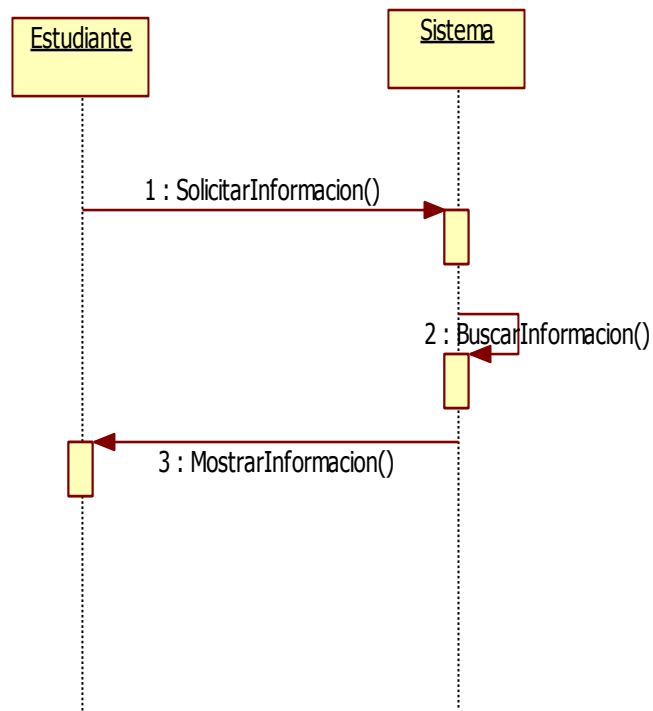


Figura 14. Diagrama de secuencia del caso de uso 13: Consultar Horarios.

3.4.2 CONTRATOS DEL SISTEMA.

Los contratos ayudan a definir el comportamiento del sistema a desarrollar, describen el efecto que sobre él tiene las operaciones.

CONTRATO: INGRESAR AL SISTEMA.

Nombre:	Introducir Identificador (identificador, contraseña)
Responsabilidades:	Verifica si el usuario puede acceder al sistema, acceder y asignar los privilegios que le corresponden según el tipo de usuario.
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: R1, R2. Caso de uso: Ingresar al Sistema
Notas:	
Excepciones:	Si el usuario no está registrado, denegar el acceso e indicar el error.
Salidas:	
Precondiciones:	El sistema conoce el identificador y contraseña de todos los usuarios.
Pos condiciones:	

Tabla 28. Contrato: Ingresar al Sistema Función: Introducir Identificador.

CONTRATOS: REGISTRAR ESTUDIANTE.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura los datos personales del estudiante.
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Estudiante.

Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El sistema genera un código único para cada estudiante. Los datos personales deben ser recolectados por el administrador
Pos condiciones:	El sistema evita errores en el ingreso de información a través del código único.

Tabla 29. Contrato: Registrar Estudiante Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar los datos personales del estudiante
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Estudiante.
Notas:	
Excepciones:	La cedula del estudiante debe ser valida
Salidas:	
Precondiciones:	Enviar todos los datos personales al sistema
Pos condiciones:	El sistema genera un nuevo usuario para el estudiante.

Tabla 30. Contrato: Registrar Estudiante Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar los datos personales del estudiante al administrador
Tipo:	Sistema
Referencias	Funciones del sistema:

Cruzadas:	Caso de uso: Registrar Estudiante.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	Los datos personales del estudiante deben estar almacenados.
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 31. Contrato: Registrar Estudiante - Función: Mostrar Información.

CONTRATOS: REGISTRAR FAMILIAR.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura los datos personales del familiar.
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Familiar.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El administrador debe recolectar toda la información acerca del familiar.

Tabla 32. Contrato: Registrar Familiar. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar los datos personales del familiar
Tipo:	Sistema

Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Familiar.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	Enviar la información necesaria sobre el parentesco familiar
Pos condiciones:	

Tabla 33. Contrato: Registrar Familiar. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar los datos personales del familiar al administrador
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Familiar.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	Los datos personales del familiar deben estar almacenados.
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 34. Contrato: Registrar Familiar. Función: Mostrar Información.

CONTRATOS: REGISTRAR PARENTESCO.

Nombre:	IngresarDatos ()
Responsabilidades:	Relacionar a cada uno de los familiares con cada estudiante
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Parentesco.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El estudiante y familiar deben estar registrados en el sistema
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 35. Contrato: Registrar Parentesco. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información del parentesco entre estudiante y familiar
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Parentesco.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	Enviar la información necesaria sobre el parentesco familiar

Pos condiciones:	
------------------	--

Tabla 36. Contrato: Registrar Parentesco. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar las relaciones familiares existentes entre estudiantes y familiares
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Parentesco.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de parentescos debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 37. Contrato: Registrar Parentesco. Función: Mostrar Información.

CONTRATOS: REGISTRAR PROFESOR.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura los datos personales del profesor de la institución
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Profesor.
Notas:	

Excepciones:	
Salidas:	
Precondiciones:	El administrador debe recolectar toda la información acerca del profesor de la institución
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 38. Contrato: Registrar Profesor. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar los datos personales del profesor
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Profesor.
Notas:	
Excepciones:	Si existe una cedula registrada en el sistema, producirá un error de registro de profesores.
Salidas:	
Precondiciones:	Enviar la información necesaria sobre profesores
Pos condiciones:	

Tabla 39. Contrato: Registrar Profesor. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar los datos personales del profesor al administrador
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Profesor.
Notas:	
Excepciones:	

Salidas:	
Precondiciones:	La información de profesores debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 40. Contrato: Registrar Profesor. Función: Mostrar Información.

CONTRATOS: REGISTRAR MATERIA.

Nombre:	IngresarDatos ()
Responsabilidades:	Capturar los datos de materias impartidas en la institución
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Materia.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El administrador debe recolectar toda la información acerca de las materias
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 41. Contrato: Registrar Materia. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre la materia
Tipo:	Sistema

Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Materia.
Notas:	
Excepciones:	Si existe un código de materia registrado, no se podrá guardar la información.
Salidas:	
Precondiciones:	Enviar la información necesaria sobre materias
Pos condiciones:	

Tabla 42. Contrato: Registrar Materia. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de materias al administrador
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Materia.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de materias debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 43. Contrato: Registrar Materia. Función: Mostrar Información.

CONTRATOS: REGISTRAR PERIODO.

Nombre:	IngresarDatos ()
---------	------------------

Responsabilidades:	Captura la información sobre los periodos lectivos
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Periodo.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El administrador debe recolectar toda la información acerca de los periodos lectivos
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 44. Contrato: Registrar Periodo. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre el periodo a registrar
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Periodo.
Notas:	
Excepciones:	Si existe un código de periodo registrado en el sistema, producirá un error de registro de periodos
Salidas:	
Precondiciones:	Enviar la información necesaria sobre periodos
Pos condiciones:	El sistema asignara un periodo en vigencia en base a las fechas de inicio y fin del mismo.

Tabla 45. Contrato: Registrar Periodo. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de periodos al administrador

Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Periodo.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de periodos debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 46. Contrato: Registrar Periodo. Función: Mostrar Información.

CONTRATOS: REGISTRAR GRADO.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura la información sobre grados o cursos de la institución
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Grado.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	El administrador debe recolectar toda la información acerca de los grados de la institución
Pos condiciones:	Los datos serán enviados al sistema para su

	validación.
--	-------------

Tabla 47. Contrato: Registrar Grado. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre el grado a registrar
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Grado.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	Enviar la información necesaria sobre los grados de la institución
Pos condiciones:	Generar código de grado.

Tabla 48. Contrato: Registrar Grado. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de grados al administrador
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Grado.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información sobre los grados debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas

	<p>por cualquier parámetro.</p> <p>La información puede ser modificada o eliminada.</p>
--	---

Tabla 49. Contrato: Registrar Grado. Función: Mostrar Información.

CONTRATOS: REGISTRAR HORARIO.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura la información sobre los horarios en curso.
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Horario.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	<p>El administrador debe recolectar toda la información acerca de los horarios en curso.</p> <p>Los periodos, materias y grados relacionados deben haber sido ingresados previamente.</p>
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 50. Contrato: Registrar Horario. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre el horario a registrar
Tipo:	Sistema
Referencias	Funciones del sistema:
Cruzadas:	Caso de uso: Registrar Horario.
Notas:	
Excepciones:	Solo se puede crear un horario para cada grado por

	periodo.
Salidas:	
Precondiciones:	Enviar la información necesaria sobre horarios.
Pos condiciones:	Se creara una matriz de información donde se almacenara los datos de materias, días y horas.

Tabla 51. Contrato: Registrar Horario. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de periodos al administrador. Mostrar matriz de horario con la información de materias, días y horas.
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Horario.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de horarios debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. Se puede agregar materias a un día y hora seleccionados.

Tabla 52. Contrato: Registrar Horario. Función: Mostrar Información.

CONTRATOS: REGISTRAR MATRICULAS.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura la información sobre matriculas de los estudiantes
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Matriculas.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información sobre estudiantes, familiares, horarios y periodos debe estar correcta y completa.
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 53. Contrato: Registrar Matriculas. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre la matricula a registrar
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Matricula.
Notas:	
Excepciones:	Si existe un código de matricula registrado en el sistema, producirá un error en el registro de información.
Salidas:	
Precondiciones:	Enviar la información necesaria sobre matriculas
Pos condiciones:	

Tabla 54. Contrato: Registrar Matriculas. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de matriculas al administrador
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Matricula.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de periodos debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada o eliminada.

Tabla 55. Contrato: Registrar Matriculas. Función: Mostrar Información.

CONTRATOS: REGISTRAR CALIFICACIONES.

Nombre:	IngresarDatos ()
Responsabilidades:	Captura la información sobre las calificaciones de los estudiantes
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Calificaciones.
Notas:	
Excepciones:	
Salidas:	

Precondiciones:	El administrador debe recolectar las calificaciones de cada una de las materias por cada grado en curso.
Pos condiciones:	Los datos serán enviados al sistema para su validación.

Tabla 56. Contrato: Registrar Calificaciones. Función: Ingresar Datos.

Nombre:	ValidarDatos ()
Responsabilidades:	Validar la información sobre el periodo a registrar
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Periodo.
Notas:	
Excepciones:	Las notas no pueden ser mayores a 10 puntos
Salidas:	
Precondiciones:	Los estudiantes deben estar matriculados en el respectivo grado
Pos condiciones:	

Tabla 57. Contrato: Registrar Matriculas. Función: Validar Datos.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de periodos al administrador
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Registrar Periodo.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de periodos debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información. La tabla de información permite realizar búsquedas por cualquier parámetro. La información puede ser modificada.

Tabla 58. Contrato: Registrar Matriculas. Función: Mostrar Información.

CONTRATOS: CONSULTAR CALIFICACIONES.

Nombre:	SolicitarInformacion ()
Responsabilidades:	Pedir información al sistema sobre las calificaciones del estudiante
Tipo:	Usuario
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Consultar Calificaciones.
Notas:	
Excepciones:	

Salidas:	
Precondiciones:	El estudiante debe estar matriculado en el periodo en vigencia
Pos condiciones:	La petición será enviada al sistema.

Tabla 59. Contrato: Consultar Calificaciones. Función: Solicitar Información.

Nombre:	MostrarInformacion()
Responsabilidades:	Mostrar la información de calificaciones al estudiante
Tipo:	Sistema
Referencias Cruzadas:	Funciones del sistema: Caso de uso: Consultar Calificaciones.
Notas:	
Excepciones:	
Salidas:	
Precondiciones:	La información de calificaciones debe estar correctamente almacenada
Pos condiciones:	Los datos son mostrados en una tabla de información.

Tabla 60. Contrato: Consultar Calificaciones. Función: Mostrar Información.

3.4.3 DIAGRAMAS DE ESTADO.

Los diagramas de estado describen gráficamente los eventos del sistema y su respectivo estado en determinado momento del tiempo. Ponemos de relieve que la realización de los diagramas de estado para este proyecto se realizo para indicar los eventos del sistema en los casos de uso.

Ingresar al Sistema.

En la figura 15 se muestra el diagrama de estados: Ingresar al Sistema.

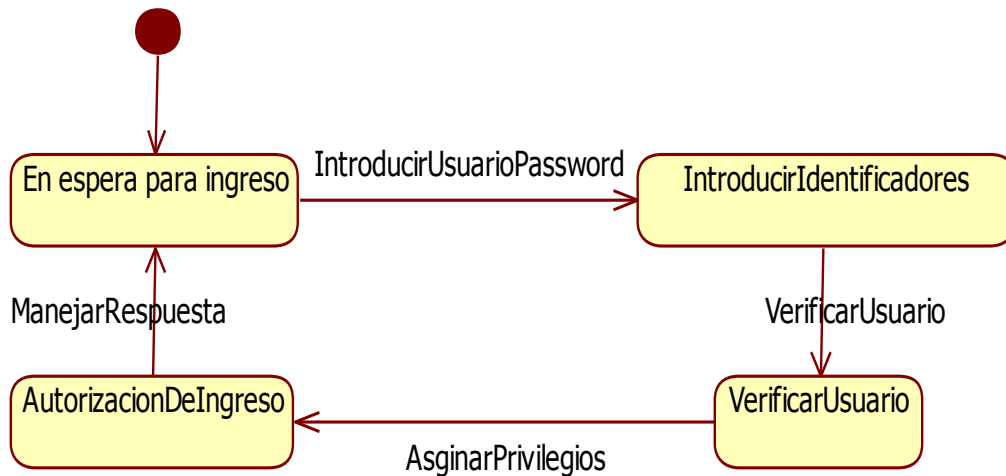


Figura 15. Diagrama de Estados: Ingresar al Sistema.

Registrar Estudiante.

En la figura 16 se muestra el diagrama de estados: Registrar Estudiante.

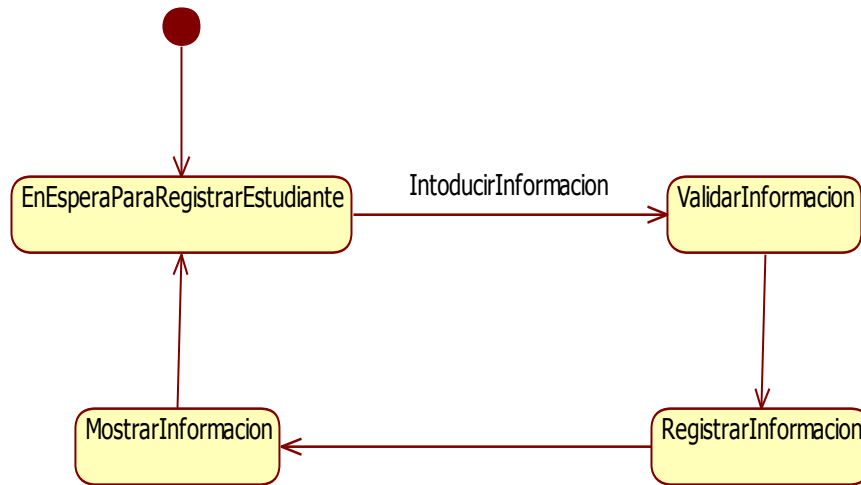


Figura 16. Diagrama de Estados: Registrar Estudiante.

Registrar Familiar.

En la figura 17 se muestra el diagrama de estados: Registrar Familiar.

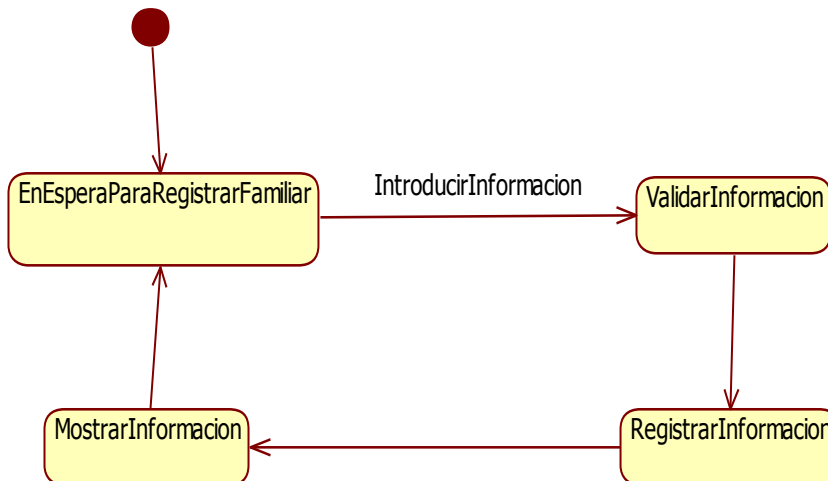


Figura 17. Diagrama de Estados: Registrar Familiar.

Registrar Parentesco.

En la figura 18 se muestra el diagrama de estados: Registrar Parentesco.

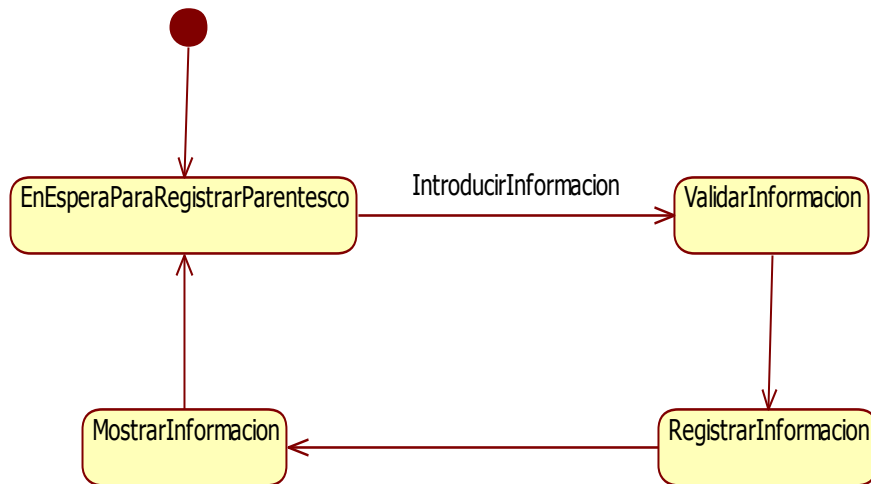


Figura 18. Diagrama de Estados: Registrar Parentesco.

Registrar Profesor.

En la figura 19 se muestra el diagrama de estados: Registrar Profesor.

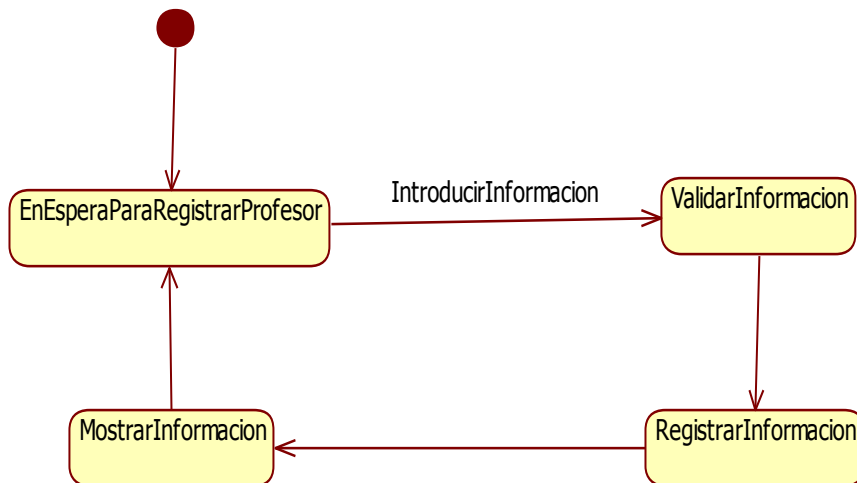


Figura 19. Diagrama de Estados: Registrar Profesor.

Registrar Materia.

En la figura 20 se muestra el diagrama de estados: Registrar Materia.

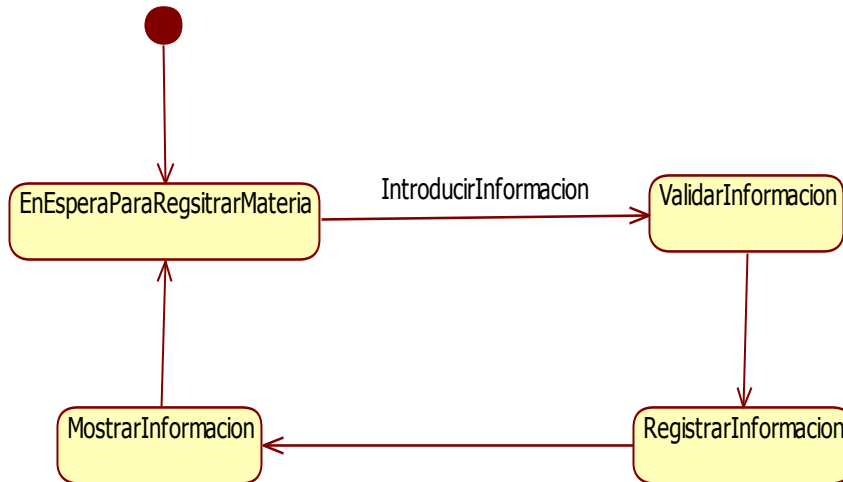


Figura 20. Diagrama de Estados: Registrar Materia.

Registra Periodo.

En la figura 21 se muestra el diagrama de estados: Registrar Periodo.

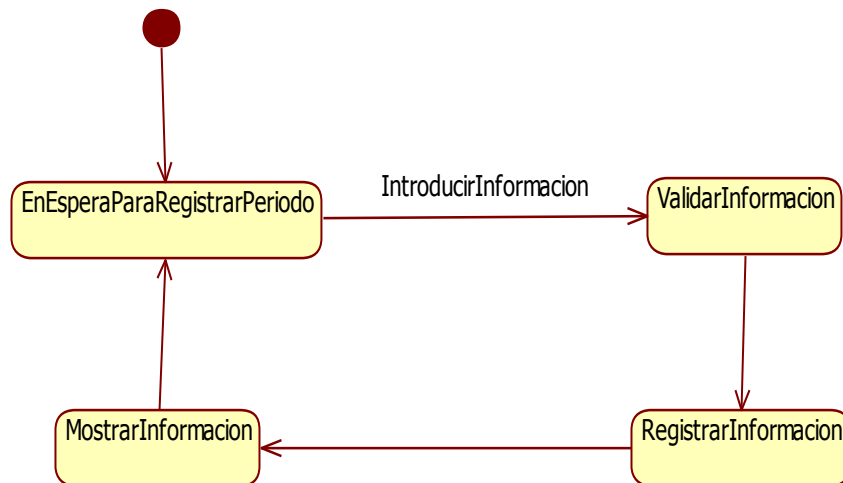


Figura 21. Diagrama de Estados: Registrar Periodo.

Registrar Grado.

En la figura 22 se muestra el diagrama de estados: Registrar Grado.

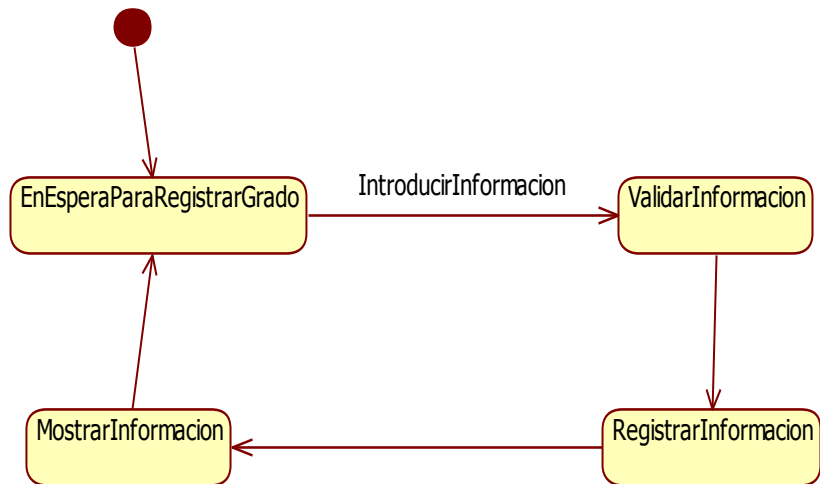


Figura 22. Diagrama de Estados: Registrar Grado.

Registrar Horario.

En la figura 23 se muestra el diagrama de estados: Registrar Horario.

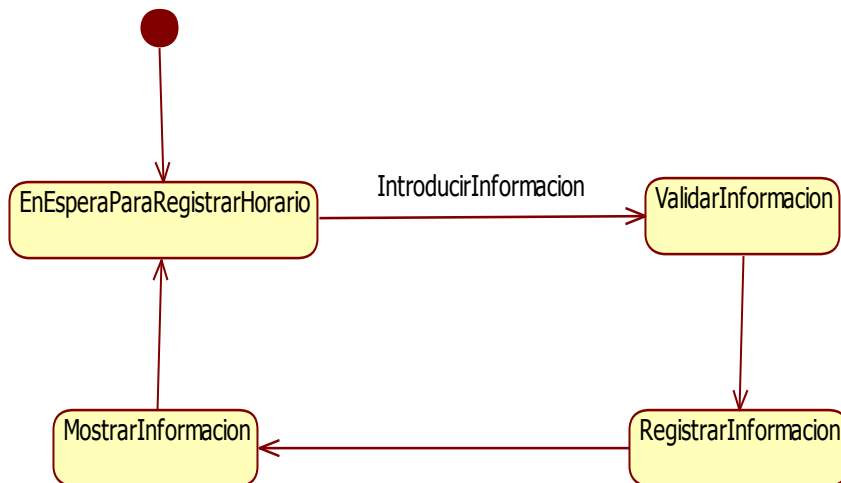


Figura 23 . Diagrama de Estados: Registrar Horario.

Registrar Matricula.

En la figura 24 se muestra el diagrama de estados: Registrar Matricula.

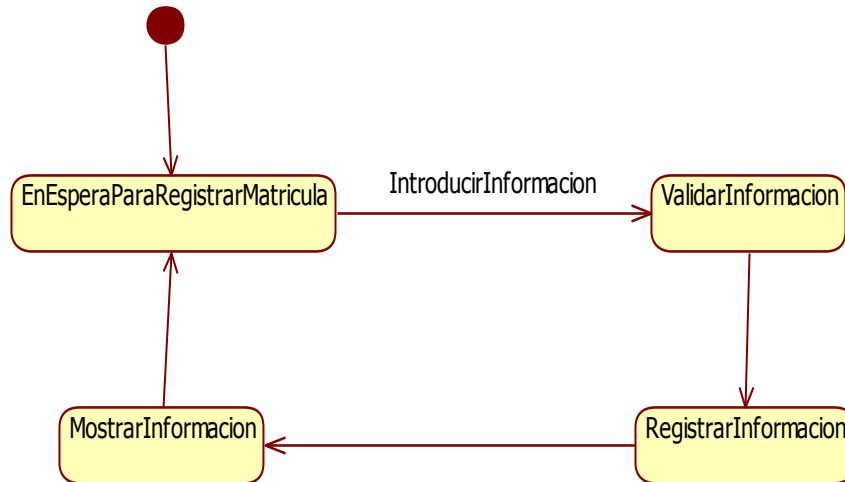


Figura 24 Diagrama de Estados: Registrar Matricula.

Registrar Calificaciones.

En la figura 25 se muestra el diagrama de estados: Registrar Calificaciones.

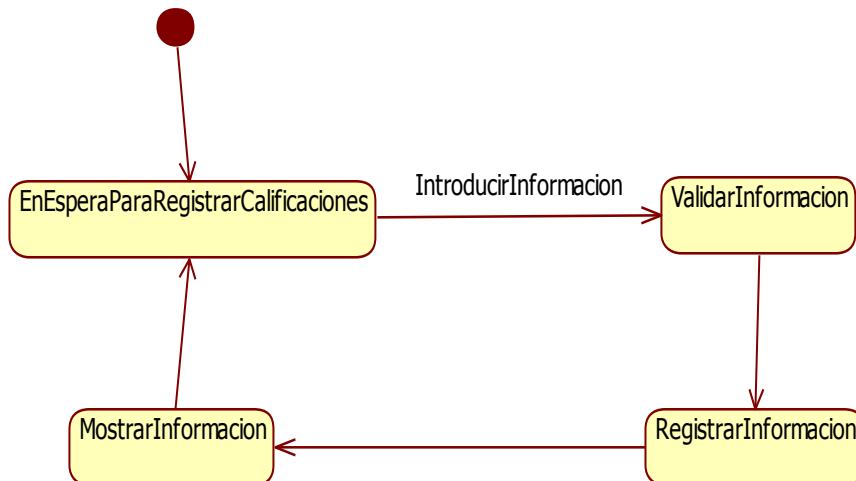


Figura 25. Diagrama de Estados: Registrar Calificaciones.

Consultar Calificaciones.

En la figura 26 se muestra el diagrama de estados: Consultar Calificaciones.

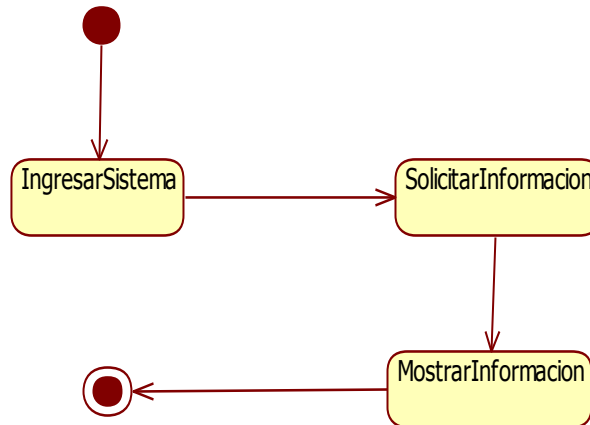


Figura 26. Diagrama de Estados: Consultar Calificaciones.

Consultar Horarios.

En la figura 27 se muestra el diagrama de estados: Consultar Horarios.

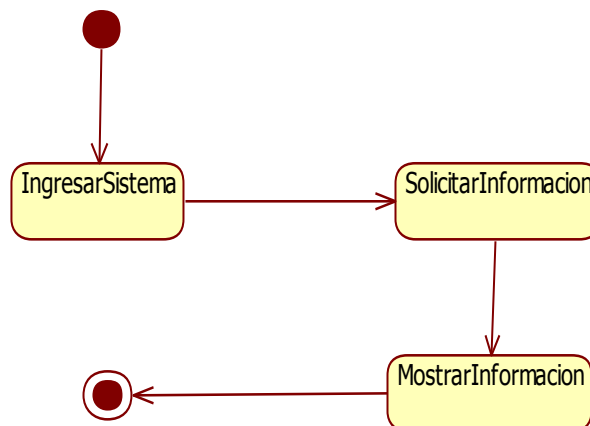


Figura 27. Diagrama de Estados: Consultar Horarios.

3.4.4 CASOS DE USO REALES.

Caso de uso 1: Ingresar al Sistema.

Caso de uso:	Ingresar al sistema.
Actores:	Administrador del sistema, administrador, estudiante.
Propósito:	Verificar la validez del usuario.
Resumen:	Un usuario ingresa a la página del sistema, ingresa su identificador único y su contraseña para poder ingresar al sistema.
Tipo:	Primario, esencial.
Referencias Cruzadas:	R1 y R2.

Tabla 61. Caso de uso real: Ingresar al Sistema.

En la figura 28 se muestra la pantalla de ingreso al sistema.



Figura 28. Pantalla de Ingreso al Sistema.

Curso normal de eventos.

Acción de actores	Respuesta del sistema
1.-Este caso de uso comienza cuando un usuario ingresa a la página de autenticación del sistema. 3.-El usuario ingresa su identificador y contraseña. 4.-Envía la información.	2.-El sistema muestra una pantalla donde debe de introducir su identificador y su contraseña. 5.-El sistema verifica la existencia del usuario en la base de datos. 6.-El sistema encuentra al usuario. 7.-Muestra en pantalla la interfaz dependiendo al tipo de usuario.

Tabla 62. Descripción del Evento: Ingresar al Sistema.

Cursos alternos.

Línea 6.- El usuario no está registrado en el sistema.

Línea 7.- El sistema muestra la pantalla de identificación del usuario.

Caso de uso 2: Registrar Estudiante.

Caso de uso:	Registrar Estudiante.
Actores:	Estudiante (iniciador), administrador.
Propósito:	Registrar la inscripción de estudiantes nuevos a la academia.
Resumen:	El estudiante se aproxima a instalaciones de la academia solicitando su incorporación, el administrador solicita sus datos personales y los documentos a presentar, posteriormente los registra.

Tipo:	Principal, esencial.
Referencias Cruzadas:	R2, R3, R4, R5, R6, R7, R8.

Tabla 63. Caso de uso real: Registrar Estudiante.

En la figura 29 se muestra la pantalla Registrar Estudiante.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

ESTUDIANTES BUSCAR: LISTAR **NUEVO ESTUDIANTE**

Codigo: A000000005 Genero: Masculino

Cedula: 0503473340 Nacionalidad: ECUATORIANA

Nombres: JESSY Vive Con: PADRES

Apellidos: LOPEZ Escuela de Procedencia: LA SALLE

Fecha de Nacimiento: 1994-06-13 Observacion: NINGUNA

Ciudad de Nacimiento: AMBATO

GUARDAR CANCELAR

Figura 29. Pantalla Registrar Estudiante.

En la figura 30 se muestra la pantalla Lista de Estudiantes.

ESTUDIANTES BUSCAR: LISTAR **NUEVO ESTUDIANTE**

✓ Datos Guardados Correctamente

Codigo	Nombres	Apellidos	Vive con	Escuela de Procedencia	Observacion	Opcion
A000000001	JUAN	PEREZ	PADRES	LA SALLE	NINGUNA	✏ ✖
A000000002	BRYAN	ALTAMIRANO	PADRES	LA SALLE	NINGUNA	✏ ✖
A000000003	CARLOS	RIVADENEIRA	PADRES	LA SALLE	NINGUNA	✏ ✖
A000000004	SEBASTIAN	LOPEZ	PADRES	LA SALLE	NINGUNA	✏ ✖
A000000005	JESSY	LOPEZ	PADRES	LA SALLE	NINGUNA	✏ ✖

1 10

Figura 30. Pantalla Lista de Estudiantes.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
--------------------	------------------------

<p>1.-El estudiante se aproxima a la secretaria solicitando su incorporación.</p> <p>2.-El administrador solicita sus datos personales y los documentos necesarios.</p> <p>3.-El administrador registra la información.</p> <p>4.-El administrador envía la información.</p> <p>8.-El administrador acepta las entradas.</p>	<p>5.-El sistema verifica la validez de los datos.</p> <p>6.-Registra la información.</p> <p>7.-Muestra en pantalla la información entrante.</p>
--	--

Tabla 64. Descripción del Evento: Registrar Estudiante.

Caso de uso 3: Registrar Familiar.

Caso de uso:	Registrar Familiar
Actores:	Administrador, estudiante.
Propósito:	Registrar la inscripción de familiares al sistema.
Resumen:	El administrador solicita datos personales de los familiares, estos se verifican y luego se registran.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R9, R10, R11, R12.

Tabla 65. Caso de uso real: Registrar Familiar.

En la figura 31 se muestra la pantalla Registrar Familiar

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

FAMILIARES BUSCAR: LISTAR NUEVO FAMILIAR

Cedula: 1706884317 Profesion: INGENIERO EN SISTEMAS

Nombres: CARLOS ANDRES Lugar de Trabajo: CENTAURO SEGUROS

Apellidos: LOPEZ Telefono Movil: 0987043594

Direccion Domicilio: FICOA Observacion: NINGUNA

Telefono Domicilio: 032809137

GUARDAR CANCELAR

Figura 31. Pantalla Registrar Familiar.

En la figura 32 se muestra la pantalla Lista de Materias.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

FAMILIARES BUSCAR: LISTAR NUEVO FAMILIAR

✓ Datos Guardados Correctamente

Cedula	Nombres	Apellidos	Direccion	Profesion	Telefono Principal	Opcion
0503452161	AUGUSTO	ALTAMIRANO	MIRAFLORES	MECANICO	032800597	✍ ✖
1706884317	CARLOS ANDRES	LOPEZ	FICOA	INGENIERO EN SISTEMAS	032809137	✍ ✖
0503452948	PATRICIO	PEREZ	FICOA	PROFESOR	032804344	✍ ✖
0504008079	CARLOS	RIVADENEIRA	LATACUNGA	ARTESANO	032555744	✍ ✖

10

Figura 32. Pantalla Lista de Familiares.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador solicita datos personales y los documentos necesarios. 2.-El administrador registra la información. 3.-El administrador envía la información. 7.-El administrador acepta las entradas.	4.-El sistema verifica la validez de los datos. 5.-Registra la información 6.-Muestra en pantalla la información entrante.

Tabla 66. Descripción del Evento: Registrar Familiar.

Caso de uso 4: Registrar Parentesco.

Caso de uso:	Registrar parentesco.
Actores:	Administrador, estudiante.
Propósito:	Registrar la inscripción de familiares al sistema.
Resumen:	El administrador registra la relación entre los estudiantes y familiares registrados en el sistema.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R13, R14, R15, R16.

Tabla 67. Caso de uso real: Registrar Parentesco.

En la figura 33 se muestra la pantalla Registrar Parentesco.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PARENTESCO BUSCAR: LISTA DE PARENTESCOS NUEVO PARENTESCO

Busqueda del Estudiante: A00000005
 Nombres del Estudiante: LOPEZ JESSY
 Busqueda del Familiar: lo
 1706884317 LOPEZ CARLOS ANDRES
 0504008079 RIVADENEIRA CARLOS

Figura 33. Pantalla Registrar Parentesco.

En la figura 34 se muestra la pantalla Lista de Parentescos.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PARENTESCO BUSCAR: LISTA DE PARENTESCOS NUEVO PARENTESCO

✓ Datos Guardados Correctamente









Familiar	Alumno	Parentesco	Opcion
PATRICIO PEREZ	JUAN PEREZ	REPRESENTANTE	 
AUGUSTO ALTAMIRANO	BRYAN ALTAMIRANO	REPRESENTANTE	 
CARLOS RIVADENEIRA	CARLOS RIVADENEIRA	REPRESENTANTE	 
CARLOS ANDRES LOPEZ	JESSY LOPEZ	REPRESENTANTE	 

Figura 34. Pantalla Lista de Parentescos.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
<p>1.-El administrador relaciona a cada estudiante con su familiar.</p> <p>2.-El administrador registra la información.</p> <p>3.-El administrador envía la información.</p> <p>7.-El administrador acepta las entradas.</p>	<p>4.-El sistema verifica la validez de los datos.</p> <p>5.-Registra la información.</p> <p>6.-Muestra en pantalla la información entrante.</p>

Tabla 68. Descripción del Evento: Registrar Parentesco.

Caso de uso 5: Registrar Docente.

Caso de uso:	Registrar Docente.
Actores:	Docente (iniciador), administrador.
Propósito:	Registrar la inscripción de profesores nuevos en la institución.
Resumen:	El docente se aproxima a instalaciones de la institución solicitando su incorporación, el administrador solicita sus datos personales y los documentos a presentar, posteriormente los registra.
Tipo:	Principal, esencial.
Referencias Cruzadas	R17, R18, R19, R20.

Tabla 69. Caso de uso real: Registrar Docente.

En la figura 35 se muestra la pantalla Registrar Profesor.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PROFESORES BUSCAR: LISTAR NUEVO PROFESOR

Cedula: 0502781230 Telefono Domicilio: 032809557

Nombres: JORGE VINICIO Telefono Movil: 0987784391

Apellidos: FLORES GUAYAQUIL Profesion: INGENIERO CIVIL

Direccion Domicilio: PALAMA Observacion: NINGUNA

GUARDAR CANCELAR

Figura 35. Pantalla Registrar Profesor.

En la figura 36 se muestra la pantalla Lista de Profesores.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PROFESORES BUSCAR: LISTAR NUEVO PROFESOR

✓ Datos Guardados Correctamente

Cedula	Nombres	Apellidos	Direccion	Profesion	Telefono Principal	Opcion
0503452948	JOSE	ENRIQUEZ	LATACUNGA	LICENCIADO EN MATEMATICAS	032855669	
0502781230	JORGE VINICIO	FLORES GUAYAQUIL	PALAMA	INGENIERO CIVIL	032809557	

1 10

Figura 36. Pantalla Lista de Profesores.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
<p>1.-El docente se aproxima al administrador solicitando su incorporación.</p> <p>2.-El administrador solicita sus datos personales y los documentos necesarios.</p> <p>3.-El administrador registra la información.</p> <p>4.-El administrador envía la información.</p> <p>8.-El administrador acepta las entradas.</p>	<p>5.-El sistema verifica la validez de los datos.</p> <p>6.-Registra la información</p> <p>7.-Muestra en pantalla la información entrante.</p>

Tabla 70. Descripción del Evento: Registrar Docente.

Caso de uso 6: Registrar Materia.

Caso de uso:	Registrar Materia.
Actores:	Administrador
Propósito:	Registrar la información de las materias en el sistema.
Resumen:	El administrador registra la información de las materias que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R25, R26, R27, R28.

Tabla 71. Caso de uso real: Registrar Materia.

En la figura 37 se muestra la pantalla Registrar Materia.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

MATERIAS BUSCAR: LISTA DE MATERIAS NUEVA MATERIA

Codigo: REA-001 Busqueda del Profesor: 1706884317

Nombre: REALIDAD NACIONAL Profesor: RIVADENEIRA JOSE VINICIO

Categoria: INDEFINIDA

Observacion: NINGUNA

Estado: Activo

GUARDAR CANCELAR

Figura 37. Pantalla Registrar Materia.

En la figura 38 se muestra la pantalla Lista de Materias.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

MATERIAS BUSCAR: LISTA DE MATERIAS NUEVA MATERIA

✓ Datos Guardados Correctamente

Nº	Codigo	Nombre	Categoria	Profesor	Observacion	Opcion
1	CNN-0001	CIENCIAS NATURALES	INDEFINIDA	JOSE ENRIQUEZ	NINGUNA	✏ ✖
2	COC-0001	COCINA ECUATORIANA	INDEFINIDA		NINGUNA	✏ ✖
3	COC-0002	COCINA INTERNACIONAL	INDEFINIDA		NINGUNA	✏ ✖
4	COM-0001	COMPUTACION	INDEFINIDA		NINGUNA	✏ ✖
5	CON-0001	CONTABILIDAD	INDEFINIDA		NINGUNA	✏ ✖
6	FIS-0001	FISICA	INDEFINIDA		NINGUNA	✏ ✖
7	LEN-001	LENGUAJE Y COMUNICACION	INDEFINIDA	JORGE VINICIO FLORES GUAYAQUIL	NINGUNA	✏ ✖
8	MAT-0001	MATEMATICA I	INDEFINIDA		NINGUNA	✏ ✖
9	MAT-0002	MATEMATICA II	INDEFINIDA		NINGUNA	✏ ✖
10	REA-0001	REALIDAD NACIONAL	INDEFINIDA	JOSE VINICIO RIVADENEIRA	NINGUNA	✏ ✖

1 2 10

Figura 38. Pantalla Lista de Materias.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 72. Descripción del Evento: Registrar Materia.

Cursos alternos.

Línea 3.- El código de la materia no se puede repetir, y la información no se registrara.

Caso de uso 7: Registrar Periodo.

Caso de uso:	Registrar Periodo.
Actores:	Administrador
Propósito:	Registrar la información de las materias en el sistema
Resumen:	El administrador registra la información de las materias que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R29, R30, R31, R32, R33.

Tabla 73. Caso de uso real: Registrar Periodo.

En la figura 39 se muestra la pantalla Registrar Periodo.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PERIODOS BUSCAR: LISTA DE PERIODOS NUEVO PERIODO

Codigo del Periodo
2012-2013

Fecha Inicio
2012/09/03

Fecha Finalizacion
2013/07/26

Observacion
NINGUNA

GUARDAR CANCELAR



Figura 39. Pantalla Registrar Periodo.

En la figura 40 se muestra la pantalla Lista de Periodos

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

PERIODOS BUSCAR: LISTA DE PERIODOS NUEVO PERIODO

✓ Datos Guardados Correctamente

Codigo	Fecha de Inicio	Fecha de Fin	Observacion	Opcion
2012-2013	2012-09-03	2013-07-26	NINGUNA	 

<< 1 >> 10

Figura 40. Pantalla Lista de Periodos.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información 5.-Muestra en pantalla la información entrante.

Tabla 74. Descripción del Evento: Registrar Periodo.

Cursos alternos.

Línea 3.- El código del periodo no se puede repetir, y la información no se registrara.

Caso de uso 8: Registrar Grado.

Caso de uso:	Registrar Grado.
Actores:	Administrador.
Propósito:	Registrar la información de los grados en el sistema.
Resumen:	El administrador registra la información de los grados que existen en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R34, R35, R36, R37.

Tabla 75. Caso de uso real: Registrar Grado.

En la figura 41 se muestra la pantalla Registrar Grado

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

GRADOS BUSCAR: LISTA DE GRADOS NUEVO GRADO

Nivel del Grado NOVENO Ubicacion SEGUNDO PISO BLOQUE A

Paralelo UNICO Observacion NINGUNA

Dirigente JORGE VINICIO FLORES

GUARDAR CANCELAR

Figura 41. Pantalla Registrar Grado.

En la figura 42 se muestra la pantalla Lista de Grados.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

GRADOS BUSCAR: LISTA DE GRADOS NUEVO GRADO

✓ Datos Guardados Correctamente

Grado	Paralelo	Dirigente	Ubicacion	Observacion	Opcion
NOVENO	UNICO	JORGE VINICIO FLORES GUAYAQUIL	SEGUNDO PISO BLOQUE A	NINGUNA	✎ ✖
OCTAVO	UNICO	JOSE ENRIQUEZ	SEGUNDO PISO BLOQUE A	NINGUNA	✎ ✖

<< 1 >> 10

Figura 42. Pantalla Lista de Grados.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información 5.-Muestra en pantalla la información entrante.

Tabla 76. Descripción del Evento: Registrar Grado.**Caso de uso 9: Registrar Horario.**

Caso de uso:	Registrar Horario.
Actores:	Administrador.
Propósito:	Registrar la información de los horarios en el sistema.
Resumen:	El administrador registra la información de los horarios que se imparten en la institución.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R38, R39, R40, R41, R42, R43.

Tabla 77. Caso de uso real: Registrar Horario.

En la figura 43 se muestra la pantalla Registrar Horario.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

HORARIOS BUSCAR: LISTA DE HORARIOS NUEVO HORARIO

Periodo: 2012-2013

Grado: NOVENO UNICO

Busqueda del Profesor: jo

0503452948 ENRIQUEZ JOSE

0502781230 FLORES GUAYAQUIL JORGE VINICIO

GUARDAR CANCELAR

Figura 43. Pantalla Registrar Horario.

En la figura 44 se muestra la pantalla Lista de Horarios.

SISTEMA ESCOLASTICO Inicio Datos Administracion Calificaciones Matriculacion Hola, admin | Salir

HORARIOS BUSCAR: LISTA DE HORARIOS NUEVO HORARIO

✓ Datos Guardados Correctamente

No.	Periodo	Grado	Profesor	Opcion
1	2012-2013	OCTAVO UNICO	JOSE ENRIQUEZ	
2	2012-2013	NOVENO UNICO	JORGE VINICIO FLORES GUAYAQUIL	
3	2012-2013	DECIMO UNICO	JOSE VINICIO RIVADENEIRA	

1 10

Figura 44. Pantalla Lista de Horarios.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 78. Descripción del Evento: Registrar Horario.

Cursos alternos.

Línea 3.- Solo se puede registrar un horario para cada grado en cada periodo.

Caso de uso 10: Registrar Matricula.

Caso de uso:	Registrar Matricula.
Actores:	Administrador
Propósito:	Registrar la información de matriculas en el sistema.
Resumen:	El administrador registra la información de matriculas de acuerdo al periodo en vigencia.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R44, R45, R46, R47, R48, R49, R50.

Tabla 79. Caso de uso real: Registrar Matricula.

En la figura 45 se muestra la pantalla Registrar Matricula.

Figura 45. Pantalla Registrar Matricula.

En la figura 35 se muestra la pantalla Lista de Matriculas.

Matricula	Folio	Alumno	Representante	Periodo	Grado	Observaciones	Opcion
001	002	JUAN PEREZ	PATRICIO PEREZ	2012-2013	OCTAVO UNICO	NINGUNA	
002	002	BRYAN ALTAMIRANO	AUGUSTO ALTAMIRANO	2012-2013	OCTAVO UNICO	NINGUN	
003	001	CARLOS RIVADENEIRA	CARLOS RIVADENEIRA	2012-2013	OCTAVO UNICO	NINGUNA	
004	001	JESSY LOPEZ	CARLOS ANDRES LOPEZ	2012-2013	OCTAVO UNICO	NINGUNA	

Figura 46. Pantalla Lista de Matriculas.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 80. Descripción del Evento: Registrar Matricula.

Cursos alternos.

Línea 3.- Solo se puede registrar la matricula si existe un periodo en vigencia.

Caso de uso 11: Registrar Calificaciones.

Caso de uso:	Registrar Calificaciones.
Actores:	Administrador.
Propósito:	Registrar la información de calificaciones en el sistema.
Resumen:	El administrador registra las calificaciones por cada grado y materia seleccionados y de acuerdo al periodo.
Tipo:	Principal, esencial.
Referencias Cruzadas:	R51, R52, R53, R54.

Tabla 81. Caso de uso real: Registrar Calificaciones.

En la figura 47 se muestra la pantalla Lista de Grados del Periodo Actual.

SISTEMA ESCOLASTICO					
Inicio		Datos		Administración	
Calificaciones		Matriculación		Hola, admin Salir	
PERIODO ACTUAL:2012-2013			BUSCAR: <input type="text"/>		LISTA DE GRADOS
Grado	Paralelo	Dirigente	Ubicacion	Observacion	Opcion
DECIMO	UNICO	JOSE VINICIO RIVADENEIRA	SEGUNDO PISO BLOQUE A	NINGUNA	
NOVENO	UNICO	JORGE VINICIO FLORES GUAYAQUIL	SEGUNDO PISO BLOQUE A	NINGUNA	
OCTAVO	UNICO	JOSE ENRIQUEZ	SEGUNDO PISO BLOQUE A	NINGUNA	

Figura 47. Pantalla Lista de Grados del Periodo Actual.

En la figura 48 se muestra la pantalla Lista de Materias por Grado.

SISTEMA ESCOLASTICO					
Inicio		Datos		Administración	
Calificaciones		Matriculación		Hola, admin Salir	
GRADO:OCTAVO UNICO			LISTA DE MATERIAS		
Codigo	Nombre de la Materia	Opcion			
CNN-0001	CIENCIAS NATURALES				
COC-0001	COCINA ECUATORIANA				
COC-0002	COCINA INTERNACIONAL				
COM-0001	COMPUTACION				
CON-0001	CONTABILIDAD				
FIS-0001	FISICA				
MAT-0001	MATEMATICA I				
SOC-0001	ESTUDIOS SOCIALES				

Figura 48. Pantalla Lista de Materias por Grado.

En la figura 49 se muestra la pantalla Lista de Estudiantes por Materia.

SISTEMA ESCOLASTICO														
Inicio			Datos			Administración			Calificaciones			Matriculación		
Hola, admin			Salir											
MATERIA: CIENCIAS NATURALES						BUSCAR: <input type="text"/>						LISTAR		
Nombres	Apellidos	1er Per.	2do Per.	3er Per.	Exa.	1er Qui.	1er Per.	2do Per.	3er Per.	Exa.	2do Qui.	Nota Final		
JUAN	PEREZ	8.00	7.00	9.00	10.00	8.40	5.00	8.00	9.00	7.00	7.27	7.83		
BRYAN	ALTAMIRANO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
CARLOS	RIVADENEIRA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
JESSY	LOPEZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		

Figura 49. Pantalla Lista de Estudiantes por Materia.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El administrador registra la información. 2.-El administrador envía la información. 6.-El administrador acepta las entradas.	3.-El sistema verifica la validez de los datos. 4.-Registra la información. 5.-Muestra en pantalla la información entrante.

Tabla 82. Descripción del Evento: Registrar Calificaciones.**Caso de uso 12: Consultar Calificaciones.**

Caso de uso:	Consultar Calificaciones
Actores:	Estudiante.
Propósito:	Consultar las calificaciones.
Resumen:	El estudiante consulta las calificaciones obtenidas durante todo el periodo lectivo.
Tipo:	Secundario.
Referencias Cruzadas:	R54.

Tabla 83. Caso de uso real: Consultar Calificaciones.

En la figura 50 se muestra la pantalla Consultar Calificaciones.

SISTEMA ESCOLASTICO											
		Inicio	Calificaciones	Horarios	Configuracion						
Hola, JUAN Salir											
LISTADO DE NOTAS											
Materia	1er Per.	2do Per.	3er Per.	Exa.	1er Qui.	1er Per.	2do Per.	3er Per.	Exa.	2do Qui.	Nota Final
CIENCIAS NATURALES	8.00	7.00	9.00	10.00	8.40	5.00	8.00	9.00	7.00	7.27	7.83
COCINA ECUATORIANA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
COCINA INTERNACIONAL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
COMPUTACION	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CONTABILIDAD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FISICA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MATEMATICA I	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ESTUDIOS SOCIALES	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figura 50. Pantalla Consultar Calificaciones.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El estudiante ingresa al sistema.	2.-El sistema almacena su usuario. 3.-Consulta la información. 4.-Muestra en pantalla la información.

Tabla 84. Descripción del Evento: Consultar Calificaciones

Caso de uso 13: Consultar Horarios.

Caso de uso:	Consultar Horarios.
Actores:	Estudiante.
Propósito:	Consultar el horario del periodo lectivo en vigencia.
Resumen:	El estudiante consulta el horario que le corresponde en base al periodo en vigencia.
Tipo:	Secundario.
Referencias Cruzadas:	R43.

Tabla 85. Caso de uso real: Consultar Horarios.

En la figura 51 se muestra la pantalla Consultar Horarios.

SISTEMA ESCOLASTICO					
Inicio		Calificaciones	Horarios	Configuracion	Hola, JUAN Salir
PERIODO: 2012-2013 GRADO:OCTAVO UNICO					
Hora	Lunes	Martes	Miercoles	Jueves	Viernes
1	CIENCIAS NATURALES	COCINA ECUATORIANA	ESTUDIOS SOCIALES	MATEMATICA I	COCINA INTERNACIONAL
2	CIENCIAS NATURALES	COCINA ECUATORIANA	ESTUDIOS SOCIALES	MATEMATICA I	COCINA INTERNACIONAL
3	COMPUTACION	COCINA ECUATORIANA	COCINA ECUATORIANA	CONTABILIDAD	COCINA INTERNACIONAL
4	COMPUTACION	COCINA ECUATORIANA	COCINA ECUATORIANA	CONTABILIDAD	COCINA INTERNACIONAL
5	ESTUDIOS SOCIALES	CIENCIAS NATURALES	MATEMATICA I	COCINA INTERNACIONAL	FISICA
6	ESTUDIOS SOCIALES	CIENCIAS NATURALES	MATEMATICA I	COCINA INTERNACIONAL	FISICA
7	CONTABILIDAD	MATEMATICA I	COMPUTACION	FISICA	CIENCIAS NATURALES
8	CONTABILIDAD	MATEMATICA I	COMPUTACION	FISICA	CIENCIAS NATURALES

Figura 51. Pantalla Consultar Horarios.

Curso normal de eventos.

Acción de actores.	Respuesta del sistema.
1.-El estudiante ingresa al sistema.	2.-El sistema almacena su usuario. 3.-Consulta la información 4.-Muestra en pantalla la información.

Tabla 86. Descripción del Evento: Consultar Horarios.

3.4.5 DIAGRAMA DE CLASES.

En la figura 52 se muestra el diagrama de clases del sistema.

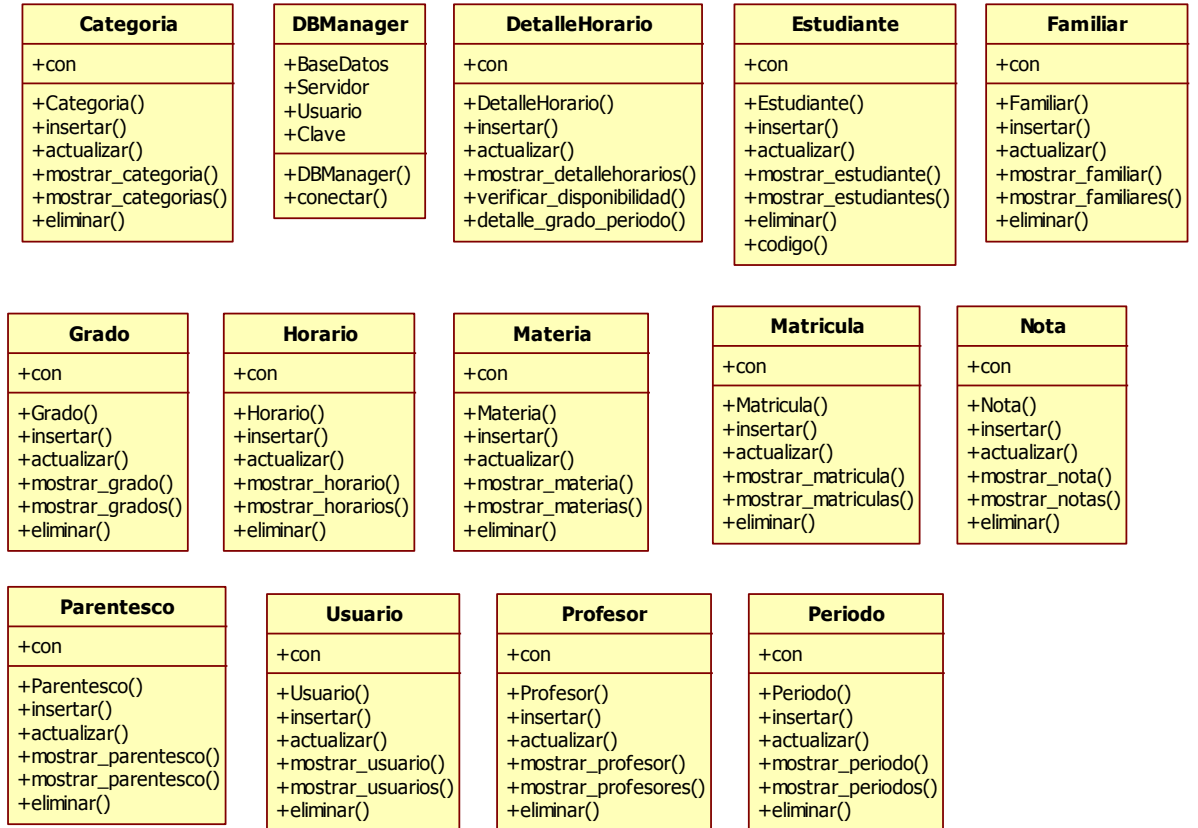


Figura 52. Diagrama de Clases.

3.4.6 MODELO ENTIDAD/RELACIÓN.

En la figura 53 se muestra el Modelo Entidad Relación.

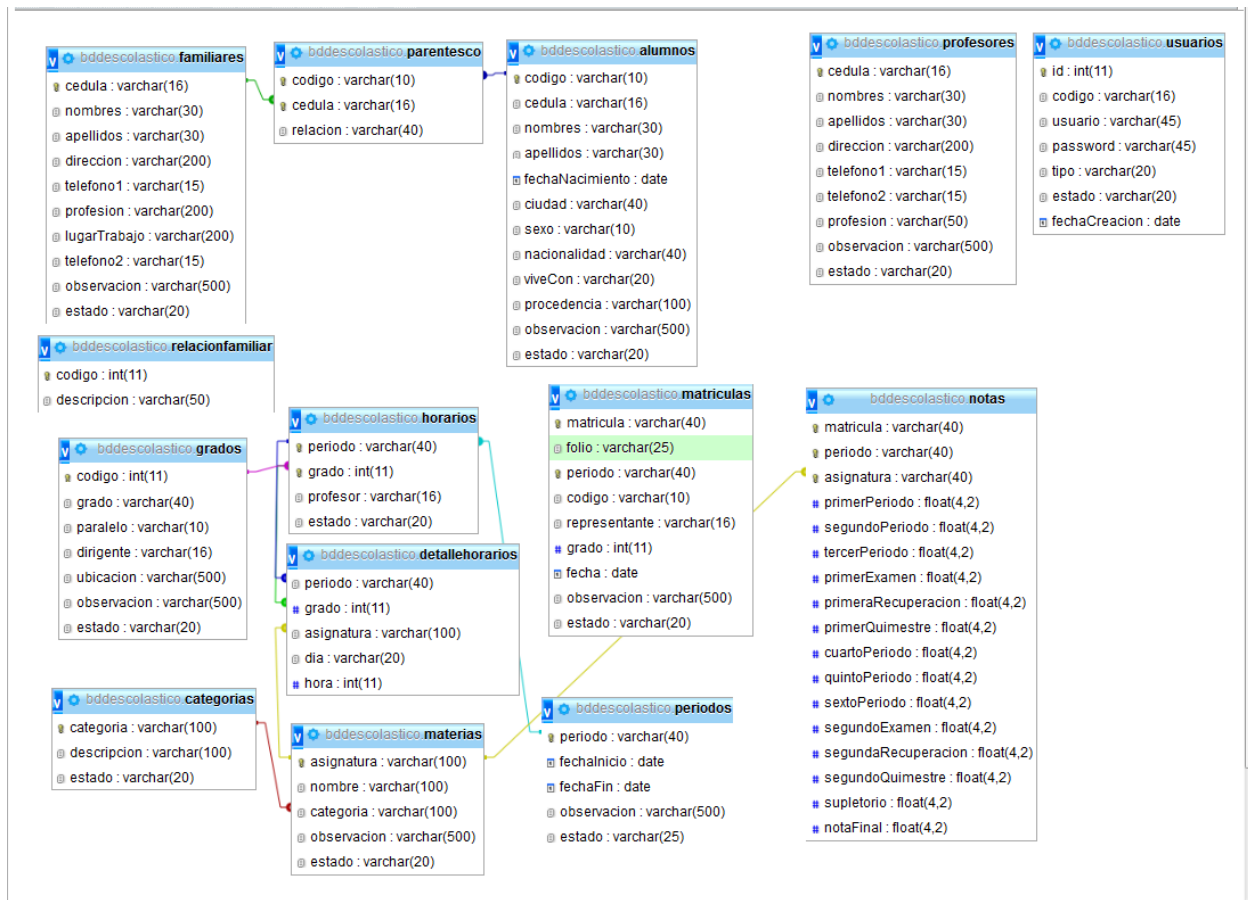


Figura 53. Modelo Entidad Relación.

3.4.7 MODELO LÓGICO RELACIONAL.

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de

Tablas donde se muestran las tablas, claves, tipo de campo.

Tabla Alumnos

Columna	Tipo	Nulo	Predeterminado
código	varchar(10)	No	
cedula	varchar(16)	Sí	NULL
nombres	varchar(30)	Sí	NULL
apellidos	varchar(30)	Sí	NULL
fechaNacimiento	date	Sí	NULL
ciudad	varchar(40)	Sí	NULL
sexo	varchar(10)	Sí	NULL
nacionalidad	varchar(40)	Sí	NULL
viveCon	varchar(20)	Sí	NULL
procedencia	varchar(100)	Sí	NULL
observación	varchar(500)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 87. Tabla Alumnos.

Tabla Categorías

Columna	Tipo	Nulo	Predeterminado
categoría	varchar(100)	No	
descripción	varchar(100)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 88. Tabla Categorías.

Tabla DetalleHorarios

Columna	Tipo	Nulo	Predeterminado
periodo	varchar(40)	Sí	NULL
grado	int(11)	Sí	NULL
asignatura	varchar(100)	Sí	NULL
dia	varchar(20)	Sí	NULL
hora	int(11)	Sí	NULL

Tabla 89. Tabla DetalleHorarios.

Tabla Familiares

Columna	Tipo	Nulo	Predeterminado
cedula	varchar(16)	No	
nombres	varchar(30)	Sí	NULL
apellidos	varchar(30)	Sí	NULL
direccion	varchar(200)	Sí	NULL
telefono1	varchar(15)	No	
profesion	varchar(200)	Sí	NULL
lugarTrabajo	varchar(200)	Sí	NULL
telefono2	varchar(15)	No	
observación	varchar(500)	Sí	NULL
estado	varchar(20)	No	

Tabla 90. Tabla Familiares.

Tabla Grados

Columna	Tipo	Nulo	Predeterminado
codigo	int(11)	No	

grado	varchar(40)	Sí	NULL
paralelo	varchar(10)	Sí	NULL
dirigente	varchar(16)	No	
ubicacion	varchar(500)	Sí	NULL
observación	varchar(500)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 91. Tabla Grados.

Tabla Horarios

Columna	Tipo	Nulo	Predeterminado
periodo	varchar(40)	No	
grado	int(11)	No	
profesor	varchar(16)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 92. Tabla Horarios.

Tabla Materias

Columna	Tipo	Nulo	Predeterminado
asignatura	varchar(100)	No	
nombre	varchar(100)	No	
categoria	varchar(100)	Sí	NULL
observación	varchar(500)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 93. Tabla Materias.

Tabla Matriculas

Columna	Tipo	Nulo	Predeterminado
---------	------	------	----------------

matricula	varchar(40)	No	
folio	varchar(25)	No	
periodo	varchar(40)	No	
codigo	varchar(10)	Sí	NULL
representante	varchar(16)	Sí	NULL
grado	int(11)	Sí	NULL
fecha	date	Sí	NULL
observación	varchar(500)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 94. Tabla Matriculas.

Tabla Notas.

Columna	Tipo	Nulo	Predeterminado
matricula	varchar(40)	No	
periodo	varchar(40)	No	
asignatura	varchar(40)	No	
primerPeriodo	float(4,2)	No	
segundoPeriodo	float(4,2)	No	
tercerPeriodo	float(4,2)	No	
primerExamen	float(4,2)	No	
primeraRecuperacion	float(4,2)	No	
primerQuimestre	float(4,2)	No	
cuartoPeriodo	float(4,2)	No	
quintoPeriodo	float(4,2)	No	
sextoPeriodo	float(4,2)	No	
segundoExamen	float(4,2)	No	
segundaRecuperacion	float(4,2)	No	

segundoQuimestre	float(4,2)	No	
supletorio	float(4,2)	No	
notaFinal	float(4,2)	No	

Tabla 95. Tabla Notas.

Tabla Parentesco.

Columna	Tipo	Nulo	Predeterminado
codigo	varchar(10)	No	
cedula	varchar(16)	No	
relacion	varchar(40)	Sí	NULL

Tabla 96. Tabla Parentesco.

Tabla Periodo.

Columna	Tipo	Nulo	Predeterminado
periodo	varchar(40)	No	
fechaInicio	date	Sí	NULL
fechaFin	date	Sí	NULL
observación	varchar(500)	Sí	NULL
estado	varchar(25)	No	

Tabla 97. Tabla Periodo.

Tabla Profesores.

Columna	Tipo	Nulo	Predeterminado
cedula	varchar(16)	No	
nombres	varchar(30)	Sí	NULL
apellidos	varchar(30)	Sí	NULL
direccion	varchar(200)	Sí	NULL

telefono1	varchar(15)	No	
telefono2	varchar(15)	No	
profesion	varchar(50)	No	
observación	varchar(500)	Sí	NULL
estado	varchar(20)	Sí	NULL

Tabla 98. Tabla Profesores.

Tabla RelacionFamiliar.

Columna	Tipo	Nulo	Predeterminado
codigo	int(11)	No	
descripcion	varchar(50)	No	

Tabla 99. Tabla RelacionFamiliar.

Tabla Usuarios.

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	
codigo	varchar(16)	Sí	NULL
usuario	varchar(45)	No	
password	varchar(45)	No	
tipo	varchar(20)	No	
estado	varchar(20)	No	
fechaCreacion	date	Sí	NULL

Tabla 100. Tabla Usuarios.

3.4.8 DIAGRAMA DE PAQUETES.

Organizar los elementos en paquetes ofrece la ventaja de separar los elementos detallados en abstracciones más amplias.

En la figura 54 se muestra el diagrama de Paquetes.

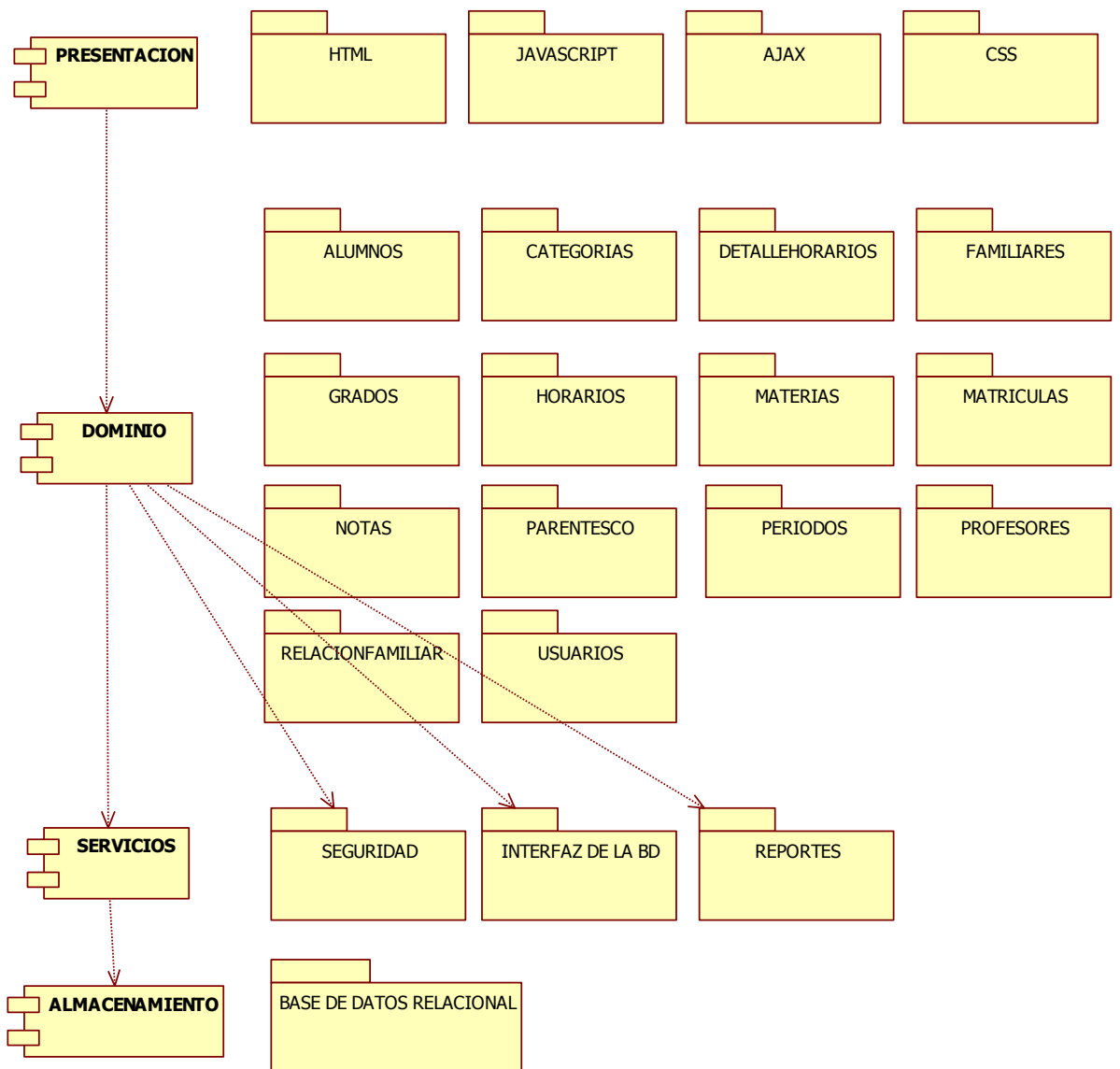


Figura 54. Diagrama de Paquetes.

3.5 FASE DE CONSTRUCCIÓN

En esta fase se desarrollan las componentes del software e incorporan al producto. El producto es el software corriendo en la plataforma adecuada, se obtiene un producto beta.

3.5.1 DIAGRAMA NAVEGACIONAL

En la figura 55 se muestra el diagrama de Navegación.

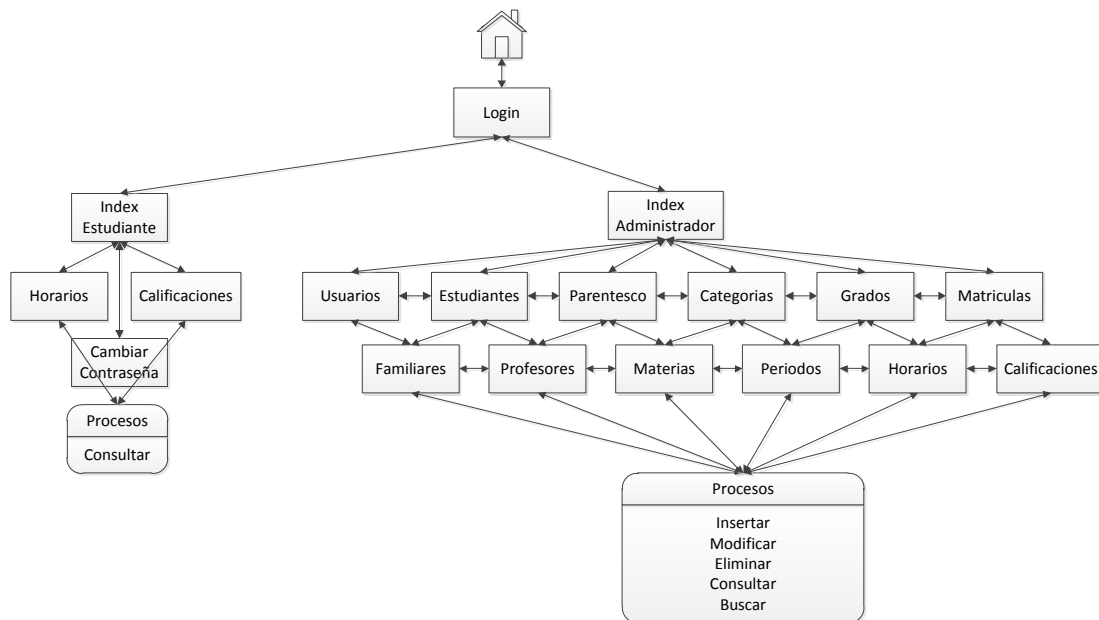


Figura 55. Diagrama de Navegación.

3.6 FASE DE TRANSICIÓN.

El objetivo de esta fase es transportar el software desarrollado a los usuarios.

3.6.1 ACTIVIDADES PARA LA IMPLEMENTACIÓN DEL SISTEMA.

- Instalar y configurar el sistema operativo.
- Instalar y configura los servidores que requiere nuestro software (Apache).
- Instalar y configurar el gestor de base de datos (MySql).
- Posteriormente se carga el Sistema de Seguimiento Académico para la Escuela de Cocina “Cuissine”.
- Se establecen los usuarios y las contraseñas en el gestor de base de datos.
- Se configuran las maquinas de los usuarios cliente dentro de la institución
- Se capacita a los administrativos de la academia y se hace entrega de los manuales de usuario.

CAPITULO 4

CALIDAD DEL SOFTWARE.

4.1 INTRODUCCIÓN.

La ISO 9126 es un estándar internacional para la evaluación del Software. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características de la siguiente manera:

Funcionalidad, el grado en que el software satisface las necesidades indicadas por los usuarios en los requisitos, y tiene los siguientes sub-atributos: adecuada, corrección, interoperabilidad, conformidad y seguridad.

Confiabilidad, un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período de tiempo establecido y tiene como sub-atributos: madurez, tolerancia a fallos y facilidad de recuperación.

Usabilidad, un conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios. Viene reflejado por los siguientes sub-atributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.

Eficiencia, conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas. Está indicado por los siguientes sub-atributos: tiempo de uso y recursos utilizados.

Mantenimiento, conjunto de atributos relacionados con el esfuerzo necesitado para modificar las especificaciones. Está indicada por los siguientes sub-atributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Movilidad, conjunto de atributos relacionados con la habilidad del software para ser transferido desde un entorno a otro.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

4.2 FUNCIONALIDAD.

4.2.1 MÉTRICA DE ADECUADA.

Nombre:	Complejidad de implementación funcional.
Propósito:	Verificar que tan completa esta la implementación funcional
Método de aplicación:	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, formula:	$x = 1 - a/b$ $a =$ número de funciones faltantes $b =$ número de funciones descritas en la especificación de requisitos
Interpretación:	$0 \leq x \leq 1$ Entre más cerca de 1 más completa
Aplicación	$x = 1 - a/b$ $a = 0$ $b = 9$ $x = 1 - 0/53$
Interpretación	$x = 1$ implica que se llegó a los objetivos deseados del proyecto en curso por lo que cumple con la adecuada.

Tabla 101. Métrica de Adecuidad

4.2.2 PUNTO FUNCIÓN.

El punto función se calcula realizando una serie de actividades comenzando por determinar los siguientes números:

- número de entradas del usuario
- número de salidas del usuario
- número de consultas de usuario
- número de archivos lógicos internos
- número de archivos de interfaz externa

Aplicando al proyecto y considerando aplicaciones enteras tendremos:

Archivos Lógicos Internos			
Nombre	DET	RET	Complejidad
Alumnos	12	1	Baja
Categorías	3	1	Baja
DetalleHorarios	4	3	Baja
Familiares	10	1	Baja
Grados	7	1	Baja
Horarios	4	3	Baja
Materias	6	3	Baja
Matriculas	9	4	Baja
Notas	17	4	Baja
Parentesco	3	3	Baja
Periodos	5	1	Baja
Profesores	9	1	Baja
Usuarios	7	1	Baja

Tabla 102. Archivos Lógicos Internos.

Archivos Interfaz Externa			
Nombre	DET	RET	Complejidad
No existen debido a que la aplicación no interactúa con otro sistema externo.			

Tabla 103. Archivos de Interfaz Externa.

EI(Entradas Internas)			
Numero	DET	FTR	Complejidad
Registrar estudiante	15	1	Baja
Asignar código al estudiante	1	1	Baja
Registrar usuario	7	1	Baja
Modificar estudiante	13	1	Baja
Eliminar estudiante	2	1	Baja
Registrar familiar	13	1	Baja
Modificar familiar	13	1	Baja
Eliminar familiar	2	1	Baja
Registrar parentesco	4	3	Media
Modificar parentesco	4	3	Media
Eliminar parentesco	2	3	Media
Registrar profesor	12	1	Baja
Modificar profesor	12	1	Baja
Registrar categorías	4	1	Baja
Modificar categorías	4	1	Baja
Eliminar categorías	3	1	Baja
Registrar materias	7	2	Media
Modificar materias	7	2	Media
Eliminar materias	2	2	Baja
Registrar periodos	6	1	Baja
Modificar periodos	6	1	Baja
Eliminar periodos	2	4	Media
Registrar grados	8	2	Media
Modificar grados	8	2	Media
Eliminar grados	2	3	Media
Registrar horarios	5	4	Alta
Registrar detalle de horarios	6	4	Alta
Modificar horarios	5	4	Alta
Registrar matriculas	10	6	Alta
Modificar matricula	10	6	Alta
Eliminar matricula	2	2	Baja
Registrar calificaciones	1	6	Media

Tabla 104. Entradas Internas.

EO(Salidas Externas)			
Numero	DET	FTR	Complejidad
Mensaje de Error de Acceso	1	1	Baja
Asignar periodo en vigencia	1	1	Baja
Detalle de horario	5	1	Baja
Lista de grados por periodo en vigencia	5	3	Alta
Lista de materias por grado y periodo en vigencia	3	3	Media
Lista de estudiantes matriculados por periodo, grado y materia	13	6	Alta

Tabla 105. Salidas Externas.

EQ(Consultas Externas)					
Numero	EI		EO		Complejidad
	DE T	FTR	DE T	FTR	
Ingreso a la aplicación	5	1	5	1	Baja
Consulta de estudiantes	6	1	6	1	Baja
Consulta de familiares	6	1	6	1	Baja
Consulta de parentescos	3	1	3	3	Media
Consulta de profesores	6	1	6	1	Baja
Consulta de categorías	3	1	3	1	Baja
Consulta de materias	5	1	5	2	Media
Consulta de periodos	5	1	5	1	Baja
Consulta de grados	5	1	5	2	Media
Controlar días y horas ocupados del detalle de horarios	5	4	7	1	Alta
Consulta de horarios	3	3	3	3	Media
Consultar representante o familiar del estudiante	3	2	4	2	Baja
Consulta de estudiantes en la matricula	3	1	3	1	Baja
Consulta de matriculas	7	1	7	5	Alta
Consulta de calificaciones	13	6	13	1	Alta

Tabla 106. Consultas Externas.

Valoración de la complejidad				
Parámetro	Complejidad	Peso	Valores	Total
Entrada	Alta	6	5	30
	Media	4	10	40
	Baja	3	17	51
Salida	Alta	7	2	14
	Media	5	1	5
	Baja	4	3	12
Fichero Lógico Interno	Alta	15	0	0
	Media	10	0	0
	Baja	7	13	91
Fichero Lógico Externo	Alta	10	0	0
	Media	7	0	0
	Baja	5	0	0
Consultas	Alta	6	3	18
	Media	4	4	46
	Baja	3	8	24
Total				331

Tabla 107. Calculo de punto función.

La relación que nos permite calcular el punto función es el siguiente:

$$\text{PF} = \text{Cuenta Total} \times (\text{Grado de confiabilidad} + \text{Tasa de error } \Sigma \text{Fi})$$

Donde:

PF: Medida de funciones

Cuenta total: Es la sumatoria del producto de factor ponderación y valores

de los parámetros.

Grado de confiabilidad: Es la confiabilidad estimada del sistema.

Tasa de error: Probabilidad subjetiva estimada del dominio de la información

ΣFi: Son valores de ajuste de complejidad.

Sin importancia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 108. Valores de ajuste de complejidad.

Escala	Sin Importancia	Incidental	Moderado	Medio	Significativo	Esencial
Factor	0	1	2	3	4	5
Comunicación de datos					x	
Procesamiento Distribuido						x
Rendimiento	x					
Configuración Equipo			x			
Volumen Transacción	x					
Entrada de Datos On-line						x
Eficiencia Usuario			x			
Actualización Online	x					
Procesos Complejos	x					
Reutilización			x			
Facilidad de Instalación		x				
Facilidad de Operación	x					
Instalación Diferentes Lugares			x			
Facilidad de Cambios		x				

Tabla 109. Características Generales del Sistema.

Con la obtención de los anteriores datos y considerando un grado de confiabilidad mínimo, calculamos:

$$PF = 331 \times (0.65 + 0.01 \times 24) \quad PF = 294.59$$

Si consideramos el máximo valor de reajuste de complejidad como $\Sigma Fi = 70$

$$PF = 331 \times (0.65 + 0.01 \times 70) \quad PF = 446.85$$

La relación obtenida entre ambos puntos es:

$$PF / PF \text{ máximo} = 294.59 / 446.85 = 0.66$$

Por lo que se concluye que la funcionalidad del sistema es de 66%.

4.3 CONFIABILIDAD.

Para determinar la confiabilidad del sistema especificamos desde el instante desde que esta comienza a funcionar, es decir desde el $t_0 = 0$, a partir de este momento se realiza las observaciones pertinentes. En son de encontrar una falla en el sistema considerando ese momento de la falla como t_1 , como el intervalo entra ambos tiempos es una variable continua se vio la necesidad de hacer uso de una función continua, que nos de la confiabilidad en términos probabilísticas:

$P(T \leq t) = F(t)$	(1)	Probabilidad de fallos
$P(T > t) = 1 - F(t)$	(2)	Probabilidad de éxito

Para el cálculo de las probabilidades se tomo la distribución exponencial, por la existencia de intervalos continuos

$$F(t) = PF e^{-\lambda t}$$

Para calcular el índice de error tomamos 8 ejecuciones en una semana durante un mes y medio se tiene reemplazando tenemos:

$$F(t) = 0.81 e^{[-1/5]*6} = 0.24$$

Siendo 0.24 la probabilidad de fallo. Por lo que el software tendrá una confiabilidad de 76%.

4.3.1 MÉTRICA DE MADUREZ.

Nombre:	Suficiencia de las pruebas
Propósito:	Cuantos de los casos de prueba necesarios están cubiertos por el plan de pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para tener una cobertura adecuada.
Medición, formula:	$x = a/b$ a = número de casos de prueba en el plan b = número de casos de prueba requerido
Interpretación:	$0 \leq x$ Entre x sea mayor mejor la suficiencia.
Aplicación	$x = a/b$ a = 13 b = 21 $x = 13/21$ $x = 0.62$
Interpretación	X = 0.62 implica que el número de pruebas realizadas es considerado como aceptable tal que sobrepasa el 50% de las pruebas pero se recomienda utilizar la totalidad de las pruebas para una mejor evaluación del proyecto.

Tabla 110. Métrica de Madurez.

4.4 USABILIDAD.

4.4.1 MÉTRICA DE FACILIDAD DE COMPRESIÓN.

Nombre:	Funciones evidentes
Propósito:	Verifica que proporción de las funciones del sistema son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, formula:	$x = a/b$ a = número de funciones (o tipos de funciones) evidentes al usuario. b = total de funciones (o tipos de funciones).
Interpretación:	$0 \leq x \leq 1$ Entre más cerca de 1 mejor
Aplicación	$x = a/b$ a = 29 b = 32 $x = 29/32$ $x = 0.91$
Interpretación	X = 0.91 implica que casi el 91% de las funciones del sistema son entendibles a los usuarios.

Tabla 111. Métrica de Facilidad de Comprensión.

4.5 FACILIDAD DE MANTENIMIENTO.

Para verificar la estabilidad del sistema es decir el índice de madurez del software (IMS), se probó con los cambios que ocurrieron en el desarrollo del software.

Para ello tenemos:

$$\text{IMS} = [\text{MT} - (\text{Fc} + \text{Fa} + \text{Fe})] / \text{MT}$$

Dónde:

MT: Número de módulos en la versión actual

Fc: Número de módulos en la versión actual que se han cambiado

Fa: Número de módulos en la versión actual que se han añadido

Fe: Número de módulos en la versión actual que se han eliminado

Considerando una ponderación de:

Intervalo	Calificación
0% ≤ IMS < 25%	Deficiente
25% ≤ IMS < 50%	Suficiente
50% ≤ IMS < 75%	Buena
75% ≤ IMS ≤ 100%	Optima

Tabla 112 Escala

Remplazando datos tenemos:

$$IMS = [10 - (1 + 2 + 0)] / 10 = 0.7$$

Tomando en cuenta las ponderaciones se concluye que 70% está en el intervalo con calificación de Buena que implica aceptablemente estable.

4.5.1 MÉTRICA DE FACILIDAD DE CAMBIO

Nombre:	Registro de cambios
Propósito:	¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?
Método de aplicación:	Registrar la proporción de información sobre cambios a los módulos.
Medición, formula:	$x = a/b$ a = número de cambios a funciones o módulos que tienen comentarios confirmados b = total de funciones o módulos modificados
Interpretación:	$0 \leq x \leq 1$ Entre más cerca de 1 más registrable, 0 indica un control de cambios deficiente o pocos cambios o alta estabilidad
Aplicación	$x = a/b$ $a = 6$ $b = 7$ $x = 6/7$ $x = 0.86$
Interpretación	X = 0.86 implica que una gran parte del código está documentado con comentarios explicativos de cómo realiza cada tarea.

Tabla 113. Métrica de Facilidad de Cambio.

CAPITULO 5

PRUEBAS Y SEGURIDAD

5.1 INTRODUCCIÓN.

Las pruebas en el desarrollo del software son realizados con la intención de descubrir errores, y se deben realizar para todos los artefactos generados durante la construcción de un producto, se aplican diferentes tipos de técnicas de prueba a cada tipo de software. Para disponer de información continua, confiable y en tiempo oportuno, se deben de considerar varios factores de seguridad. La seguridad informática debe garantizar: La disponibilidad de los sistemas de información, su recuperación rápida y completa, la integridad de la información y la confidencialidad.

5.2 PRUEBAS DEL DISEÑO ARQUITECTÓNICO.

Se concentran en las características de la arquitectura del programa, con especial énfasis en la estructura arquitectónica y en la eficiencia de los módulos y es una métrica de caja negra porque no requiere de ningún conocimiento del trabajo interno del módulo.

a) **Complejidad estructural**, se define de la siguiente manera:

$$S(i) = f_{out}^2(i)$$

Donde $f_{out}(i)$ es la expansión del módulo i .

Para la aplicación se tomo en cuenta:

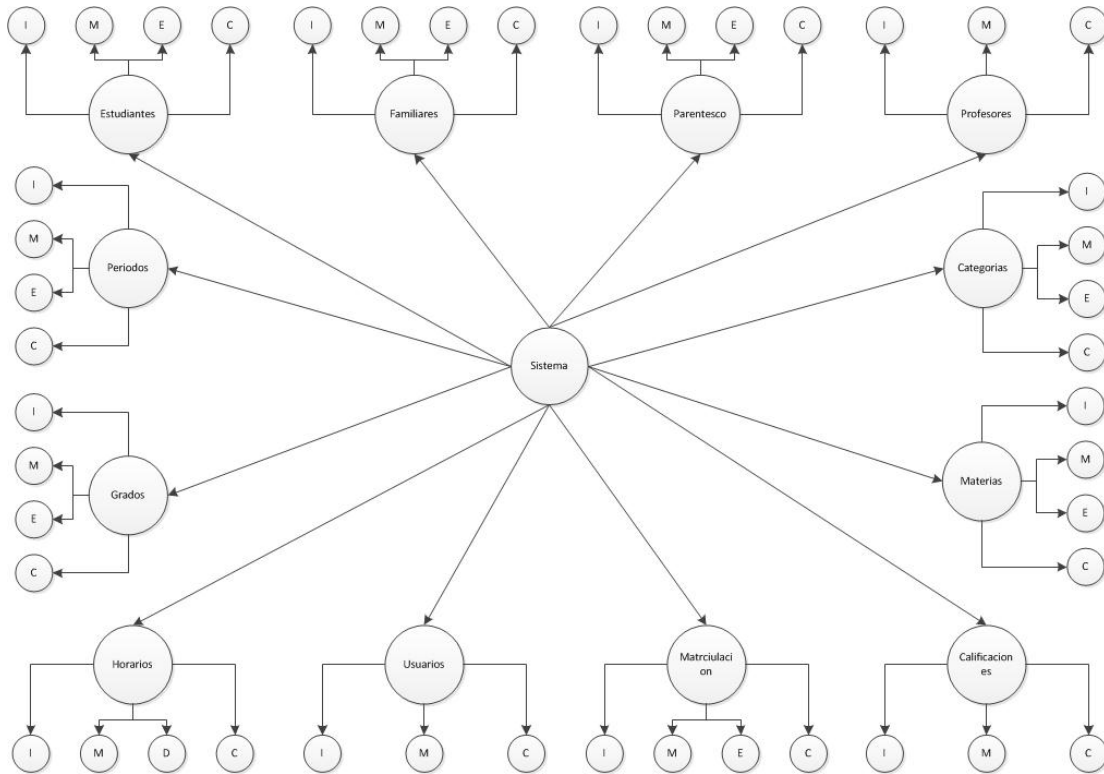


Figura 56. Grafo del sistema.

Donde $S(i) = 57$ por ser la cantidad de módulos, si el análisis comienza desde el módulo i que es el mismo sistema.

b) **Complejidad de datos**, que proporciona una indicación de la complejidad interna de un módulo i y se define:

$$D(i) = v(i) / [f_{out}(i) + 1]$$

Considerando a las variables de entrada y salida como una sola por módulo se tiene: $D(i) = 70 / (57 + 1) = 1.21$

El resultado nos dice que tiende a complicarse.

- c) **Complejidad del sistema**, se define como la suma de complejidades estructural y de datos y se define como:

$$C(i) = S(i) + D(i)$$

Remplazando tenemos:

$$C(i) = 20 + 1.21 = 21.21$$

A medida que crecen los valores de complejidad, la complejidad arquitectónica o global del sistema también aumenta. Esto lleva a una mayor probabilidad de que aumente el esfuerzo necesario para la integración y las pruebas.

5.3 SEGURIDAD.

5.3.1 SEGURIDAD FÍSICA.

La seguridad física consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial. Se refiere a los controles y mecanismos de seguridad dentro y alrededor del centro de cómputo así como los medios de acceso remoto al y desde el mismo, implementados para proteger el hardware y medios de almacenamiento de datos.

Las principales amenazas que se prevén en la seguridad física son:

- a) Desastres naturales, incendios, accidentales, tormentas e inundaciones.
- b) Amenazas ocasionadas por el hombre.

- c) Disturbios, sabotajes internos y externos deliberados.

5.3.2 SEGURIDAD DE BASE DE DATOS.

Para recuperar o almacenar información se necesita conectarse a la base de datos, enviar una consulta válida, recoger el resultado y cerrar la conexión, para realizar la consulta se utiliza el lenguaje de consultas usado comúnmente, Lenguaje de Consultas Estructurado – SQL.

- a) Asignar un súper usuario quien puede realizar altas, bajas y modificaciones a la base de datos, y para que otros usuarios puedan usarla, deben otorgarse privilegios que restringen el uso de la misma, para ello se crea diferentes usuarios para cada aspecto de la aplicación con derechos muy limitados sobre las tablas de la base de datos.
- b) Inyecciones SQL, las consultas SQL pueden burlar los controles de acceso, y de este modo evitar los chequeos estándares de autenticación y autorización, la inyección directa de comandos SQL es una técnica en la cual un atacante crea o altera comandos SQL existentes para exponer datos escondidos, o sobrescribir datos críticos, o incluso ejecutar comandos del sistema peligrosos en el servidor donde se encuentra la base de datos. Esto se consigue cuando la aplicación toma información de entrada del usuario y la combina con parámetros estáticos para construir una consulta SQL. Una solución a este tipo de problemas es revisar las entradas, PHP posee un amplio rango de funciones de validación de datos, desde los más simples encontrados en funciones sobre variables y en funciones de tipo de carácter.
- c) Realizar copias de seguridad, el gestor de Bases de datos MySQL

incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podrá poner a salvo los datos, para que, en el eventual caso de que se pierdan, se pueda recuperar.

Existen varias opciones para realizar la copia de seguridad de una base de datos de MySQL, a partir de la versión 3.23.25 y posteriores, existe la posibilidad de realizar una copia de seguridad a través de la sentencia SQL "backup table". Otra opción, es posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos, como pueden ser mysqldump ó mysqlhotcopy.

5.3.3 SEGURIDAD EN LA WEB

- a) Otorgar acceso a las aplicaciones Web sólo a personal autorizado, con privilegios mínimos que le permitan cumplir con sus funciones.
- b) Implementar políticas de contraseñas fuertes y de control de acceso a las aplicaciones Web.
- c) Se debe validar todo parámetro de entrada, incluyendo los campos de formularios teniendo en cuenta:
 - Tipo de datos (cadena, entero, real)
 - Conjunto de caracteres permitidos
 - Longitud mínima y máxima
 - Si el valor nulo es permitido
 - Si el parámetro es requerido o no
 - Si los duplicados son permitidos
 - El rango numérico

- d) El manejo inadecuado de errores puede introducir diversos problemas de seguridad en un sitio Web. Por ejemplo, la exhibición de información detallada de mensajes de error, como el contenido de variables, nombres de directorios e información sobre la base de datos, puede revelar detalles de la implementación que no deben ser expuestos bajo ninguna circunstancia a usuarios no autorizados.
- e) Almacenamiento seguro, toda información sensible por ejemplo, contraseñas almacenada por la aplicación Web, en la base de datos es cifrada con el MD5. Que es un solo sentido es decir no podrá ser visto ni por el administrador del sistema.
- f) Se implemento controles de acceso en las aplicaciones Web, para evitar la ejecución de funciones por parte de usuarios no autorizados.
- g) La aplicación Web establece sesiones para mantener el rastro del flujo de acciones de cada usuario. Para el establecimiento y mantenimiento de dichas sesiones se implementa mecanismos que garanticen la protección de las credenciales en tránsito y del identificador de sesión.

5.4 COSTO BENEFICIO.

5.4.1 INTRODUCCIÓN.

Todo proyecto de ingeniería de software debe partir con un buen plan, pero lamentablemente, la planificación es una tarea nada trivial. Uno de los aspectos que dificulta a la hora de planificar es la difícil tarea de realizar una estimación de costos y plazos realista. La diferencia en la estimación de costos entre ingeniería de software y otras disciplinas es que en ingeniería de software lo principal para las personas es el costo, y en otras disciplinas el costo de las cosas materiales depende de la actividad.

Existen técnicas para la estimación de costos, pero para ello se requiere experiencia.

El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo.

5.4.2 ANÁLISIS DE COSTO.

El Modelo Constructivo de Costes o COCOMO, es un modelo de estimación de costes de software, orientado a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente. COCOMO ayuda a estimar el esfuerzo, tiempo, gente y costos. El modelo provee tres niveles de aplicación: básico, intermedio y avanzado, basados en los factores considerados por el modelo.

Básico, es un modelo estático simplemente evaluado que calcula el esfuerzo (y costo) del desarrollo del software como función del programa expresado en líneas de código. Intermedio, calcula el esfuerzo del desarrollo del software como función del tamaño del programa y un conjunto de guías de costo que incluye una evaluación subjetiva del producto, hardware, personal y de los atributos del proyecto. Avanzado, incorpora todas las características de la versión intermedia con una evaluación del impacto de las vías de costo en cada fase del proceso de la ingeniería de software. En cada nivel de aplicación están definidos para tres tipos de proyectos de software: Modo orgánico, proyectos de software relativamente pequeños y sencillos en los que pequeños equipos con buena experiencia en la aplicación trabajan en un conjunto de requerimiento poco rígido. Modo semi – acoplado, un proyecto de software

intermedio en tamaño y complejidad en el cual equipos con distintos niveles de experiencia debe satisfacer requerimientos poco y medio rígidos. Modo acoplado, un proyecto de software que debe ser desarrollado dentro un conjunto estricto de hardware, software y de restricciones operativas.

La ecuación de COCOMO en este modo básico es:

$$E = a \times \text{KLOC}^b$$

$$D = c \times E^d$$

$$P = E / D$$

Donde E es el esfuerzo aplicado en persona por mes, D es el tiempo de desarrollo en meses, KLOC es el número de líneas estimadas para el proyecto (en miles) y P es el número de personas necesarias. Los coeficientes a, b, c y d se obtienen de la siguiente tabla:

Tipo de Proyecto	a	b	c	d
Modo orgánico	2.4	1.05	2.5	0.38
Modo semi - acoplado	3.0	1.12	2.5	0.35
Modo acoplado	3.6	1.20	2.5	0.32

Tabla 114. Tabla de los coeficientes para los diferentes proyectos.

Para el proyecto se realizaron los siguientes cálculos considerando el modo semi - acoplado:

$$a = 3.0$$

$$b = 1.12$$

$$c = 2.5$$

$$d = 0.35$$

KLOC

$$E = a \times \text{KLOC}^b = 3.0 \times 6^{1.12}$$

$$E = 22.31 \text{ persona - mes}$$

$$D = c \times E^d = 2.5 \times 22.31^{0.35}$$

$$D = 7 \text{ mes}$$

$$P = E / D = 22.31 / 7.41$$

$$P = 3 \text{ personas}$$

Considerando que el sueldo del desarrollador depende de la experiencia del mismo y es un valor muy subjetivo, se da un valor según la oferta de los programadores en el mercado de 2000.

Los costos realizados se muestran en la tabla siguiente

DESCRIPCIÓN	CANTIDAD.	COSTO MENSUAL.	CANTIDAD DE MESES.	TOTAL.
Desarrolladores	3	350.-	3	3150.-
Equipos	6	Ya existen	-	0.-
Software	-	Ya se compro	-	0.-
Adiestramiento	-	500.-	½	500.-
Otros	-	500.-	-	500.-

Tabla 115. Tablas de los costos del sistema.

Lo que lleva a que el costo total sea de 4150 dólares.

5.4.3 BENEFICIOS.

El sistema proveerá acceso y transferencia de información en tiempo real, entre los diferentes usuarios, de tal forma que la información es oportuna en el momento necesario para los mismos. Los beneficios tangibles que se pueden mencionar son:

- a) Todos los datos estarán centralizados en una sola base de datos.
- b) La información se transfiere electrónicamente a diferentes

usuarios según los privilegios que tenga.

- c) Se pueden hacer informes con la información en tiempo real.
- d) Se evita el gasto innecesario de papel reduciendo gastos operacionales.
- e) Con el software se reduce el tiempo en la productividad del personal, redundando en mejores servicios para los clientes.

Beneficios Intangibles:

La integración de los diferentes administrativos que brindan servicios similares en el área académica, facilitando la integración del sistema.

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES.

6.1 CONCLUSIONES.

Con el desarrollo del presente proyecto la institución se beneficia por contar con una base de datos que contiene información trascendental para el desempeño de sus actividades en el área académica. Además se logro implementar mecanismos de control para las matriculas considerando las materias que han sido asignadas a un horario y periodo específicos.

Habiendo logrado resolver el principal problema que tenía la institución, la demora en la obtención de información a causa de la extensa información que se almacenaban en medios tangibles provocando que las funciones de los administrativos sea tediosa.

La ESCUELA DE COCINA CUISSINE ha optimizado sus recursos y se ha beneficiado del sistema. Según estimación se requieren 2 personas para la elaboración del proyecto con facilidad y desempeño.

En base a los resultados obtenidos y después de aplicar la norma ISO 9126, podemos apreciar un desempeño bastante aceptable del sistema lo que demuestra que la metodología RUP se adapto de manera eficiente al desarrollo del presente proyecto.

6.2 RECOMENDACIONES.

La utilización de metodologías de desarrollo nos lleva a realizar software de calidad en cada fase de desarrollo a través de un buen control sobre los procesos, teniendo también una mayor transparencia del mismo, no importando las dimensiones del proyecto es siempre viable utilizar una metodología de desarrollo para producir lo que se espera en el tiempo definido y con el presupuesto calculado.

Se recomienda el buen uso de herramientas para la documentación, facilitando en gran medida el desarrollo de un proyecto, teniendo en cuenta que todas las fases previas a la implementación son realizadas por una sola persona.

BIBLIOGRAFÍA.

- [1] B. Boehm, 1979. "Ingeniería de Software Economía" Editorial Prentice Hall
- [2] Norma IEEE Glosario de Terminología de Ingeniería de Software Editorial IEEE Computer Society (junio 1983)
- [3] Jacobson, I. 1998. "Applying UML in the Unified Process" Presentación Rational Software. Presentación disponible en <http://www.rational.com/uml> como [UMLconf.zip](#) editorial Pearson Educación
- [4] Greiff W. R. Paradigma vs Metodología; El Caso de la POO (Parte II). *Soluciones Avanzadas*. Ene-Feb 1994. pp. 31-39.
- [5] Craing Larman, 1999 "UML y Patrones" Editorial Prentice Hall
- [6] JCSP Leite, 1987 "Advanced Software Engineering Proyecto Técnico Report"
- [7] Lewis G. 1994. "What is Software Engineering?" Editorial DataPro USA
- [8] Pressman Roger S. 2003 "Ingeniería del Software un enfoque práctico" Quinta edición Editorial Mc Graw Hill
- [9] Artículo No. 1 "CALIDAD DEL SOFTWARE", Juan Manuel Cueva Lovelle, conferencia 21 de octubre de 1999, Grupo GIDIS, Universidad Nacional de la Pampa cueva@lsi.uniovi.es Fecha: 2008-09-15 hrs.: 18:46 p.m.
- [10] http://es.wikipedia.org/wiki/Ingenieria_de_software Fecha: 2008-09-15 hrs.: 18:46 p.m.
- [11] Mirian Milagros. Rational Unnified Proccess. <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>
- [12] Patricio Salinas Caro. Tutorial de UML.

<http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>

[13] INTELWORKS. Redes LAN.

http://www.intelworks.net.ec/index.php?option=com_content&view=article&id=69&Itemid=1055

[14] INTELWORKS. Redes LAN.

http://www.intelworks.net.ec/index.php?option=com_content&view=article&id=70&Itemid=141

[15] KIOSKA.NET. Introducción a los Sistemas Unix.

<http://es.kioskea.net/contents/695-introduccion-a-los-sistemas-unix>

[16] Maestros del Web. Tutorial de PHP.

<http://www.maestrosdelweb.com/editorial/phpintro/>

[17] Libros Web. JavaScript.

http://www.librosweb.es/javascript/capitulo_1.html

[18] Serviweb. Desarrollo Web. *Qué es Ajax*.

<http://www.desarrolloweb.com/articulos/repaso-ajax.html>.

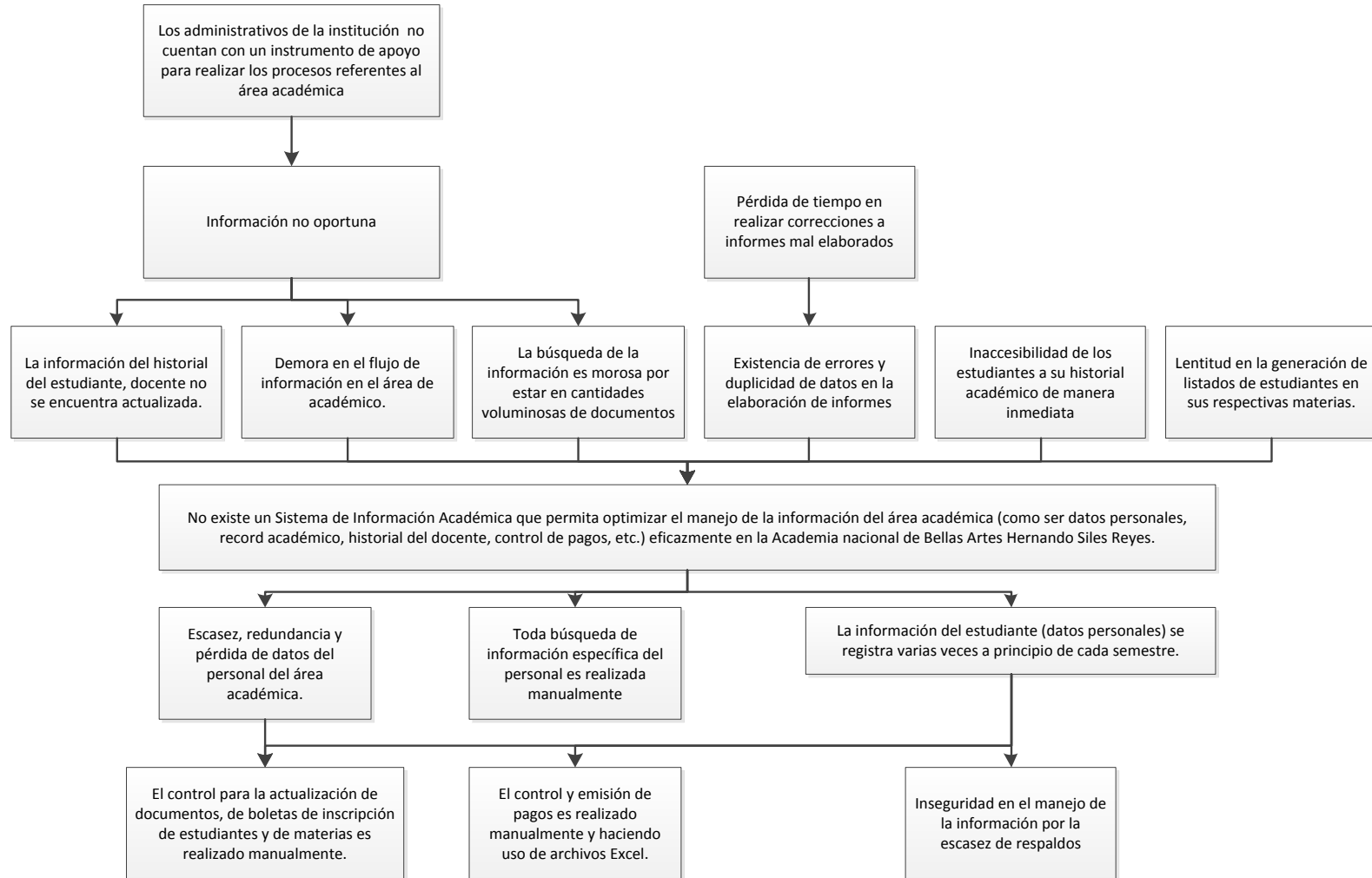
[19] Libros Web. Introducción a CSS.

http://www.librosweb.es/css/capitulo_1.html

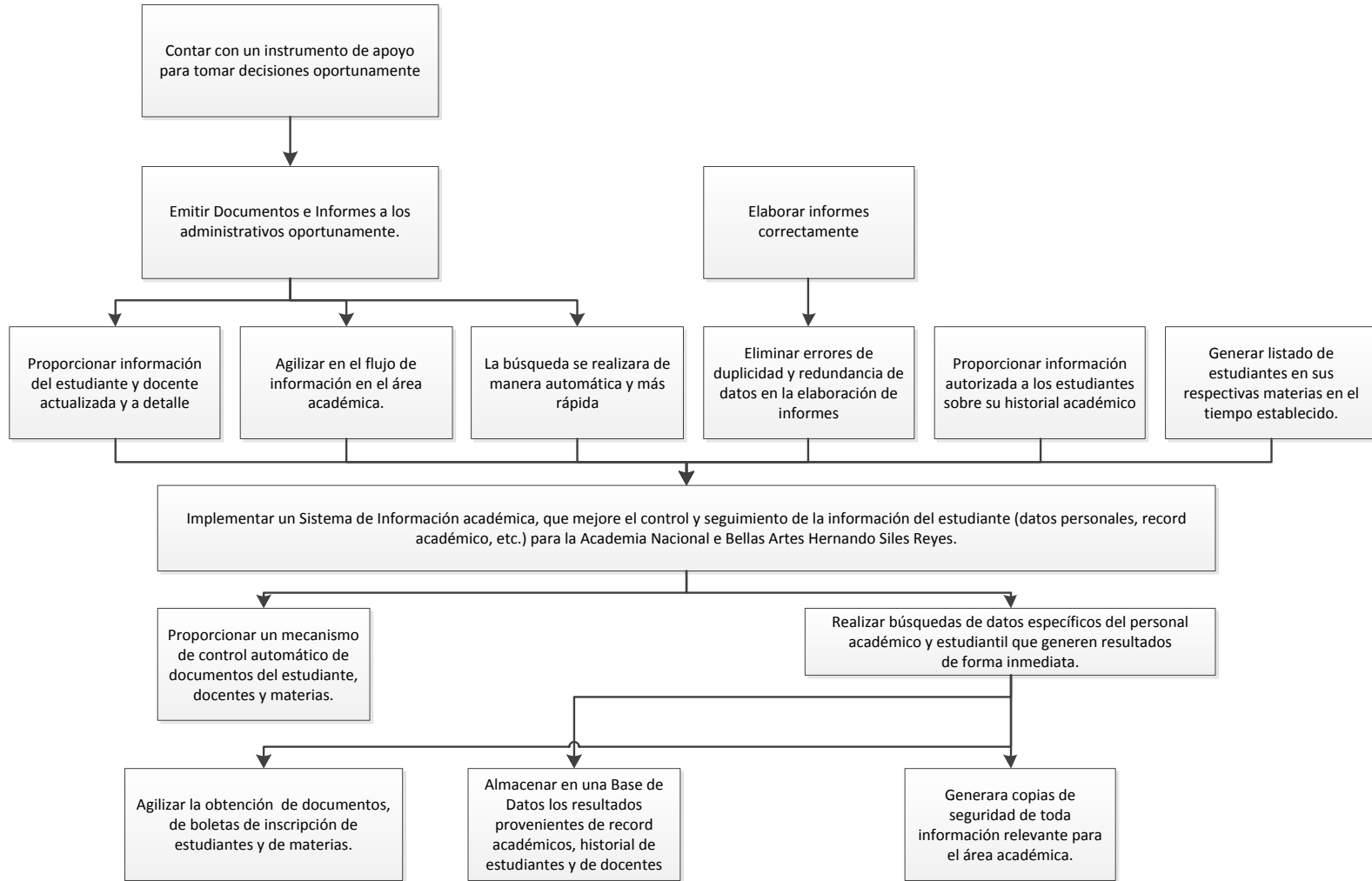
[20] Esepestudio. Qué es MySQL. <http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>.

ANEXOS

ANEXO 1: ARBOL DE PROBLEMAS



ANEXO 2: ARBOL DE OBJETIVOS



ANEXO 3: Código Fuente

Página de Ingreso a la aplicación.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=7" />
  <head>
    <title>Ingreso : Sistema Escolastico</title>
    <style type="text/css" media="all">
      @import url("css/style.css");
      @import url("css/jquery.wysiwyg.css");
      @import url("css/facebox.css");
      @import url("css/visualize.css");
      @import url("css/date_input.css");
    </style>
    <link href="templatemo_style.css" rel="stylesheet" type="text/css" />
    <?php include 'librerias_script.php'; ?>
  </head>

  <body>
    <div id="templatemo_document_wrapper">
      <div id="templatemo_wrapper">

        <div id="templatemo_menu">
          <ul>
            <li><a href="index.html">Inicio</a></li>
            <li><a href="institucion.html" >Institucion</a></li>
            <li><a href="instalaciones.html">Instalaciones</a></li>
            <li><a href="servicios.html" >Servicios</a></li>
            <li><a href="galeria.html">Galeria</a></li>
            <li><a href="miportal.php" class="current">Mi portal</a></li>
          </ul>
        </div> <!-- end of templatemo_menu -->

        <div id="templatemo_header">
          <div id="site_title">
            <h1 align="center">COLEGIO TECNICO "CUISINE"</h1>
            <h1 align="center"><a href="index.php"></a></h1>
          </div> <!-- end of site_title -->
        </div>

      </div> <!-- end of wrapper -->
    </div>
  </body>
</html>
```

```

<div class="block small center login">
  <div class="block_head">
    <div class="bheadl"></div>
    <div class="bheadr"></div>
    <h2>Login</h2>
    <ul>
      <li><a href="">Administrador</a></li>
    </ul>
  </div>      <!-- .block_head ends -->
  <div class="block_content" id="flogin">
    <?php
    if (isset($_GET["mensaje"])) {
      $mensaje = $_GET["mensaje"];
      ?>
      <div class="message info"><?php echo $mensaje; ?></div>
      <?php
    } else {
      ?>
      <div class="message info">Informacion, Ingrese sus datos de
acceso.</div>
      <?php
    }
    ?>

    <form action="validar_usuario.php" method="post">
      <p>
        <label>Usuario:</label> <br />
        <input type="text" class="text" tabindex="1" name="usuario"
id="usuario" value="" />
      </p>

      <p>
        <label>Contraseña:</label> <br />
        <input type="password" class="text" value="" tabindex="2"
name="password" id="password" />
      </p>
      <p>
        <input type="submit" class="submit" value="Acceder" />
      </p>
    </form>

  </div>
  <div class="bendl"></div>
  <div class="bendr"></div>

```

```

    </div>
</div>
<div id="templatemo_footer_wrapper">
    <div id="templatemo_footer">

        <a href="index.html">Inicio</a> | <a href="institucion.html">Insitucion</a> |
<a href="instalaciones.html">Instalaciones</a> | <a
href="informacion.html">Informacion</a> | <a href="galeria.html">Galeria</a>| <a
href="miportal.html">Mi portal</a><br /><br />

        Copyright © 2012 Luis Moscoso
    </div> <!-- end of templatemo_footer -->
</div>

</body>

</html>

```

Validar Usuario.

```

<?php
ini_set("session.use_only_cookies", "1");
ini_set("session.use_trans_sid", "0");
include("clases/conexion.php");
include 'clases/usuario.class.php';
$objUsuario = new Usuario();
$usuario = $_POST["usuario"];
$password = $_POST["password"];
$con = new DBManager;
$con->conectar();
if ($usuario != "" && $password != "") {
    $result = mysql_query("SELECT id, codigo, tipo, password, usuario FROM
usuarios WHERE usuario='$usuario'");
    if (mysql_num_rows($result) > 0) {
        $row = mysql_fetch_array($result);
        $password_db = $objUsuario->decrypt($row['password'], "cuisine2012");
        if ($password_db == $password) {
            session_set_cookie_params(0, "/", $HTTP_SERVER_VARS["HTTP_HOST"],
0);

            //usuario y contraseña válidos
            session_name("loginUsuario");
            //asigno un nombre a la sesión para poder guardar diferentes datos
            session_start();

```

```

// inicio la sesión
session_set_cookie_params(0, "/", $HTTP_SERVER_VARS["HTTP_HOST"],
0);
$_SESSION["autenticado"] = "SI";
//defino la sesión que demuestra que el usuario está autorizado
$_SESSION["ultimoAcceso"] = date("Y-n-j H:i:s");
//defino la fecha y hora de inicio de sesión en formato aaaa-mm-dd hh:mm:ss
switch ($row["tipo"]) {
    case "administrador": {
        header("Location: index_admin.php");
        break;
    }
    case "estudiante": {
        $cod = $row["codigo"];
        header("Location: index_estudiante.php?codigo=$cod;");
    }
}
$_SESSION['usuario'] = $row['usuario'];
$_SESSION['codigo']=$row['id'];
} else {
    $valor = 'Password incorrecto';
    header("Location: miportal.php?mensaje=$valor");
}
} else {
    $valor = 'Usuario no existente en la base de datos';
    header("Location: miportal.php?mensaje=$valor");
}
mysql_free_result($result);
} else {
    $valor = 'Debe especificar un usuario y password';
    header("Location: miportal.php?mensaje=$valor");
}
mysql_close();
?>

```

Index de Administrador.

```

<?php include("clases/seguridad.php") ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=7" />
    <head>
        <title>Administrador : Sistema Escolastico</title>

```

```

<style type="text/css" media="all">
  @import url("css/style.css");
  @import url("css/jquery.wysiwyg.css");
  @import url("css/facebox.css");
  @import url("css/visualize.css");
  @import url("css/date_input.css");
</style>
<?php include 'librerias_script.php'; ?>
</head>

<body>
  <div id="hld">
    <div class="wrapper">
      <?php include 'funcion_menu_admin.php'; ?>

      <div class="block">

        <div class="block_head">
          <div class="bheadl"></div>
          <div class="bheadr"></div>
          <h2>Bienvenido</h2>
        </div>

        <div class="block_content">
          <div class="grid_12" id="icondock">
            <div id="icondock" class="grid_12">
              <ul>
                <li><a href="usuarios.php" class="tip" title="Ver
usuarios">
                <br />Usuarios</a></li>
                <li><a href="estudiantes.php" class="tip"
title="Estudiantes"><br
/>Estudiantes</a></li>
                <li><a href="profesores.php" class="tip"
title="Profesores"><br
/>Profesores</a></li>
                <li><a href="materias.php" class="tip" title="Materias"><br
/>Materias</a></li>
                <li><a href="periodos.php" class="tip" title="Periodos"><br
/>Periodos</a></li>

```

```

        <li><a href="grados.php" class="tip" title="Grados"><br
        />Grados</a></li>
        <li><a href="horarios.php" class="tip" title="Horarios">
        <br />Horarios</a></li>
        <li><a href="matriculas.php" class="tip" title="Matriculas"><br
/>Matriculas</a></li>

    </ul><br clear="all" />
</div>
</div>
</div>
<!-- .block_content ends -->
<div class="bendl"></div>
<div class="bendr"></div>

</div>
<!-- .block ends -->
<div id="footer">
</div>
</div>
</body>

```

```
</html>
```

Index Estudiante.

```

<?php include("clases/seguridad.php") ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=7" />
  <head>
    <title>Usuario : Sistema Escolastico</title>
    <style type="text/css" media="all">
      @import url("css/style.css");
      @import url("css/jquery.wysiwyg.css");
      @import url("css/facebox.css");
      @import url("css/visualize.css");
      @import url("css/date_input.css");
    </style>
    <?php include 'librerias_script.php'; ?>

```

```

</head>

<body>
  <div id="hld">
    <div class="wrapper">
      <?php include 'funcion_menu_estudiante.php'; ?>

      <div class="block">

        <div class="block_head">
          <div class="bheadl"></div>
          <div class="bheadr"></div>
          <h2>Bienvenido</h2>
        </div>

        <div class="block_content">
          <div class="grid_12" id="icondock">
            <div id="icondock" class="grid_12">
              <ul>
                <li><a href="notas_estudiante.php" class="tip"
title="Notas">
                <br /><a href="calificaciones.php" class="tip"
title="Calificaciones"></a></li>
                <li><a href="horario_estudiante.php" class="tip"
title="Estudiantes">
                <br /><a href="horarios.php" class="tip"
title="Horarios"></a></li>
              </ul><br clear="all" />
            </div>
          </div>
          <!-- .block_content ends -->
          <div class="bendl"></div>
          <div class="bendr"></div>

        </div>
      <!-- .block ends -->
      <div id="footer">
      </div>
    </div>
  </div>
</body>

</html>

```