

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS COMPUTACIONALES

TEMA:

IMPLEMENTACIÓN WEB PARA LA GESTIÓN Y SEGUIMIENTO
ACADÉMICO ESTUDIANTIL DEL CENTRO EDUCATIVO “ÁLAMOS”

APLICATIVO:

SISTEMA DE GESTIÓN Y SEGUIMIENTO ACADÉMICO SAG IMPLEMENTADO
MEDIANTE EL FRAMEWORK GOOGLE WEB TOOLKIT PHP Y MYSQL

AUTOR:

Luis Fabián Rivera Onofre

DIRECTORA:

Ing. Nancy Cervantes

Ibarra – Ecuador

2012

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo LUIS FABIÁN RIVERA ONOFRE con cédula de identidad Nro. 100308327-4, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado “IMPLEMENTACIÓN WEB PARA LA GESTIÓN Y SEGUIMIENTO ACADÉMICO ESTUDIANTIL DEL CENTRO EDUCATIVO “ÁLAMOS” ”, que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

(Firma):.....

Nombre: LUIS FABIÁN RIVERA ONOFRE

Cédula: 100308327-4

Ibarra, a los 6 días del mes de enero de 2012

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto de Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento deixo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100308327-4		
NOMBRES:	LUIS FABIÁN RIVERA ONOFRE		
DIRECCIÓN:	CALIXTO MIRANDA 6-83 Y RICARDO SÁNCHEZ		
EMAIL:	riverlucho2006@hotmail.com		
TELÉFONO FIJO:	062955656	TELÉFONO MÓVIL:	098532671

DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN WEB PARA LA GESTIÓN Y SEGUIMIENTO ACADÉMICO ESTUDIANTIL DEL CENTRO EDUCATIVO “ÁLAMOS”
AUTOR (ES):	LUIS FABIÁN RIVERA ONOFRE
FECHA:AAAAMMDD	2012-01-06
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR/DIRECTOR:	ING. NANCY CERVANTES

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, LUIS FABIÁN RIVERA ONOFRE, con cédula de identidad Nro. 100308327-4, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 143.

DEDICATORIA

Ante todo a Dios por bendecir mi vida y forjar mi espíritu lleno de fortalezas y sueños, a mis padres quienes han dado todo por nuestra familia y a quienes les debo todo lo que soy, a mis hermanos quienes me apoyaron en todo momento y a todas las personas que nunca dejaron de creer en mí.

AGRADECIMIENTO

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

Gracias en especial a la Ing. Nancy Cervantes, mi directora de tesis, por su generosidad, por sus constantes aportes y sugerencias, por su tiempo. Por la disponibilidad y el conocimiento que voluntariamente donó al servicio de este trabajo de grado.

Además quiero resaltar mis agradecimientos sinceros al Msc. Juan Carlos Salas, Rector del Colegio Particular “Álamos” quien creyó en mí y me dio la oportunidad de desarrollar este proyecto de tesis.

ÍNDICE DE CONTENIDOS

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR.....	II
AUTORIZACIÓN DE USO Y PUBLICACIÓN	III
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	III
DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN.....	XIV
SUMMARY	XV
CAPÍTULO I	1
1. INTRODUCCIÓN.....	1
1.1. ALCANCE	1
1.2. ESTUDIO DE LAS HERRAMIENTAS DE DESARROLLO.....	2
1.2.1. Frameworks	2
1.2.2. Frameworks para aplicaciones Web	3
1.2.2.1. Aplicaciones Web.....	3
1.2.2.2. Frameworks para aplicaciones web	3
1.3. INTRODUCCIÓN DE GOOGLE WEB TOOLKIT (GWT).....	4
1.3.1. Conceptos Generales.....	6
1.3.1.1. Lenguaje Java.....	6
1.3.1.2. Máquina Virtual Java.....	7
1.3.1.3. JRE.....	8
1.3.1.4. JDK.....	9
1.3.1.5. AJAX.....	9
1.3.2. Google Web Toolkit	10
1.3.2.1. Funciones	10
1.3.2.2. Funcionamiento.....	12
1.3.2.3. Características	13
1.3.2.4. Arquitectura de GWT	13
1.3.2.5. Ventajas y Desventajas.....	16
CAPÍTULO II	18
2. FASE DE INICIO.....	18

2.1.	VISIÓN	18
2.1.1.	Introducción	18
2.1.1.1.	Propósito	18
2.1.1.2.	Alcance	18
2.1.1.3.	Definiciones, Siglas y Abreviaturas	18
2.1.1.4.	Referencias.....	18
2.1.2.	Posicionamiento	19
2.1.2.1.	Oportunidad de negocio.....	19
2.1.2.2.	Definición del problema.....	19
2.1.2.3.	Sentencia que define la posición del Producto	20
2.1.3.	Descripción de los interesados y usuarios	20
2.1.3.1.	Resumen de los interesados.....	21
2.1.3.2.	Resumen de los usuarios.....	22
2.1.3.3.	Entorno de usuario.....	22
2.1.3.4.	Perfiles de los Stakeholders	23
2.1.3.5.	Perfiles de usuario.....	24
2.1.3.6.	Necesidades de los interesados y usuarios	26
2.1.3.7.	Alternativas y competencia	27
2.1.4.	Vista General del Producto.....	27
2.1.4.1.	Perspectiva del producto	28
2.1.4.2.	Resumen de capacidades	29
2.1.4.3.	Costos y precios	29
2.1.4.4.	Licenciamiento e instalación.....	30
2.1.5.	Descripción del Producto	30
2.1.5.1.	Facilidad de acceso y uso.....	30
2.1.5.2.	Unificación de la información	30
2.1.5.3.	Mejor control y validación de la información.....	31
2.1.5.4.	Rangos de calidad.....	31
2.2.	PLAN DE DESARROLLO DE SOFTWARE	31
2.2.1.	Introducción	31
2.2.1.1.	Propósito	31
2.2.1.2.	Alcance	32
2.2.2.	Vista General del Proyecto.....	32
2.2.2.1.	Suposiciones y Restricciones	33

2.2.2.2. Entregables del proyecto.....	34
2.2.2.3. Plan de Desarrollo del Software.....	34
2.2.3. Organización del Proyecto	36
2.2.3.1. Participantes en el Proyecto	36
2.2.3.2. Interfaces Externas.....	37
2.2.3.3. Roles y Responsabilidades.....	37
2.2.4. Plan del Proyecto	38
2.2.4.1. Plan de las Fases	38
2.2.4.2. Calendario del Proyecto.....	40
2.2.4.3. Seguimiento y Control del Proyecto	43
CAPÍTULO III.....	45
3. CASOS DE USO.....	45
3.1. MODELOS DE CASO DE USO	45
3.1.1. Especificación Caso	45
3.1.1.1. Especificación Caso de Uso: Ingresar Alumnos	45
3.1.1.2. Especificación Caso de Uso: Ingresar Docentes.....	48
3.1.1.3. Especificación Caso de Uso: Administrar Centros Educativos	51
3.1.1.4. Especificación de Caso de Uso: Matriculación	53
3.1.1.5. Especificación Caso de Uso: Registro de Notas	55
3.1.1.6. Especificación Caso de Uso: Control de Asistencias.....	58
3.1.1.7. Especificación Caso de Uso: Generación de Certificados	61
CAPITULO IV	63
4. FASE DE CONSTRUCCIÓN	63
4.1. VISTA LÓGICA.....	63
4.1.1. Modelo Entidad-Relación	63
4.1.2. Modelo Físico.....	64
4.1.3. Diagrama Global de Paquetes	65
4.2. VISTA DE IMPLEMENTACIÓN.....	66
4.2.1. Diagramas de Actividades	66
4.2.2. Diagramas de componentes	73
4.2.3. Diagramas de arquitectura	73
4.3. FASE DE PRUEBAS	74
4.3.1. Plan de pruebas	74

4.3.2.	Desarrollo de un plan de pruebas	74
4.3.2.1.	Especificación de Caso de Prueba: Ingresar Alumnos	74
4.3.2.2.	Especificación de Caso de Prueba: Matriculación.....	75
4.3.2.3.	Especificación de Caso de Prueba: Registro de Notas.....	76
4.3.2.4.	Especificación de Caso de Prueba: Control de Asistencias	77
4.3.3.	Resultados del Plan de Pruebas	78
4.3.3.1.	Resultados de Caso de Prueba: Ingresar Alumnos.....	78
4.3.3.2.	Resultados de Caso de Prueba: Matriculación.....	78
4.3.3.3.	Resultados de Caso de Prueba: Registro de Notas.....	78
4.3.3.4.	Resultados de Caso de Prueba: Control Asistencias.....	78
CAPITULO V		79
5.	CONCLUSIONES Y RECOMENDACIONES.....	79
5.1.	CONCLUSIONES.....	79
5.2.	RECOMENDACIONES.....	80
GLOSARIO		81
REFERENCIAS.....		85
ANEXOS		
A.	ANEXO A: MANUAL DE INSTALACIÓN.....	86
A.1.	INTRODUCCIÓN.....	86
A.2.	APPSERV versión 2.5.9.....	86
A.2.1.	Prerrequisitos.....	86
A.2.2.	Instalación.....	87
A.2.3.	Después de instalar	88
A.3.	NETBEANS versión 7.0.....	89
A.4.	GWT versión 2.4	90
A.5.	HeidiSQL 5.9.....	91
B.	ANEXO B: MANUAL DE USUARIO	92
B.1.	Introducción.....	92
B.2.	Administrador funcional y usuarios del sistema	92
B.3.	Acceso a la aplicación.....	92
B.3.1.	Ingreso al sistema.....	92
B.3.2.	Autenticación de usuario	93

B.3.3.	Menú principal	93
B.3.4.	Parámetros	94
B.3.5.	Mantenimiento	98
B.3.6.	Procesos.....	100
C.	ANEXO C: DOCUMENTOS ESTANDARIZADOS	104
C.1.	FORMATO DE REPORTE MENSUAL DE CALIFICACIONES	104
C.2.	FORMATO DE REPORTE ANUAL DE CALIFICACIONES	105
C.3.	FORMATO DE ACTA – COMPROMISO DE MEJORA ACADÉMICA	106
C.4.	FORMATO DE CONTROL DE DESARROLLO DEL NIVEL ACADÉMICO DEL ESTUDIANTE	107

ÍNDICE DE FIGURAS

Figura 1.1 Arquitectura GWT.....	14
Figura 1.2 Compilador GWT Java-a-JavaScript.....	14
Figura 1.3 Navegador Web de GWT.....	15
Figura 1.4 Librería de interfaz de usuario de GWT.....	16
Figura 2.1 Perspectiva del Producto.....	28
Figura 3.1 Casos de Uso: Ingresar Alumnos.....	45
Figura 3.2 Casos de Uso Ingresar Docente.....	48
Figura 3.3 Casos de Uso Administrar Centro Educativo.....	51
Figura 3.4 Casos de Uso Matriculación.....	53
Figura 3.5 Casos de Uso Registro de Notas.....	55
Figura 3.6 Casos de Uso Control de Asistencias.....	58
Figura 3.7 Casos de Uso: Generar Certificado.....	61
Figura 4.1 Diagrama Entidad Relación.....	¡Error! Marcador no definido.
Figura 4.2 Modelado del Negocio.....	65
Figura 4.3 Registrar Alumno.....	66
Figura 4.4 Matriculación.....	67
Figura 4.5 Ingresar Notas.....	68
Figura 4.6 Registro de Asistencias.....	69
Figura 4.7 Registrar Preceptor.....	70
Figura 4.8 Registrar Alumno a Preceptor.....	71
Figura 4.9 Generación de Certificados.....	72
Figura 4.10 Diagrama de Componentes.....	73
Figura 4.11 Diagrama de Arquitectura.....	73
Figura A.1 Asistente de instalación de Appserv.....	87
Figura A.2 Asistente de instalación de NetBeans.....	89
Figura A.3 Instalar Complementos de Netbeans.....	90
Figura A.4 Añadir Complementos en NetBeans.....	91
Figura B.1 Pantalla de Ingreso al Sistema.....	92
Figura C.1 Reporte Mensual de Calificaciones.....	104
Figura C.2 Reporte Anual de Calificaciones.....	105
Figura C.3 Acta de Compromiso de Mejora Académica.....	106

ÍNDICE DE TABLAS

Tabla 2.1 Definición de la Posición del Producto	20
Tabla 2.2 Descripción de interesados y usuarios.....	21
Tabla 2.3 Costos y Precios	30
Tabla 3.1 Detalle Casos de Uso Ingresar Alumno.....	46
Tabla 3.2 Especificación Casos de Uso Registrar alumno	46
Tabla 3.3 Especificación Casos de Uso Buscar Alumno	47
Tabla 3.4 Especificación Casos de Uso Modificar Alumno	48
Tabla 3.5 Detalle Casos de Uso: Ingresar Docente	48
Tabla 3.6 Especificación Casos de Uso Registrar docente	49
Tabla 3.7 Especificación Casos de Uso Buscar Docente	50
Tabla 3.8 Especificación Casos de Uso Modificar Docente.....	51
Tabla 3.9 Detalle Casos de Uso: Administrar Centros Educativos.....	51
Tabla 3.10 Especificación Casos de Uso Crear Centro Educativo	52
Tabla 3.11 Especificación Casos de Uso Modificar Centro Educativo	53
Tabla 3.12 Detalle Casos de Uso Matriculación	53
Tabla 3.13 Especificación Casos de Uso Matriculación	54
Tabla 3.14 Detalle Casos de Uso Registrar Notas.....	55
Tabla 3.15 Especificación Casos de Uso Registrar Notas	56
Tabla 3.16 Especificación de Casos de Uso: Consultar Notas	57
Tabla 3.17 Especificación Casos de Uso: Modificar Notas.....	57
Tabla 3.18 Detalle Casos de Uso Control de Asistencias	58
Tabla 3.19 Especificación Casos de Uso: Control Asistencias.....	59
Tabla 3.20 Especificación Casos de Uso: Justificaciones	60
Tabla 3.21 Especificación Casos de Uso: Reporte Asistencias.....	60
Tabla 3.22 Detalle Casos de Uso Generar Certificados	61
Tabla 3.23 Especificación Casos de Uso: Generar Certificados	62
Tabla 4.1 Caso de Prueba: Ingresar Alumnos	75
Tabla 4.2 Caso de Prueba: Matriculación.....	76
Tabla 4.3 Caso de Prueba: Registro de Notas.....	77
Tabla 4.4 Caso de Uso: Control de Asistencias.....	77
Tabla A.1: Parámetros de configuración de Appserv.....	88

RESUMEN

La Educación, como institución, ha estado siempre generando gran cantidad de información y con la llegada de las tecnologías de la información y comunicación, las instituciones educativas tienen la obligación de utilizar las herramientas tecnológicas en la implementación de sistemas informáticos, con el afán de optimizar los procesos académicos.

Habiendo estudiado y comprendido todos estos procesos involucrados, y luego de realizar una exhaustiva observación y análisis de la práctica, se procedió a desarrollar una aplicación de software que sea capaz de automatizar la administración estudiantil en la Institución.

El desarrollo del software, incluyó la definición de los procesos de la institución para obtener un Modelo del Negocio, a partir del cual se realizó el levantamiento de los requerimientos de la aplicación utilizando un Modelo de Casos de Uso, generando posteriormente un esquema de desarrollo basado en el Proceso Unificado. Durante el diseño se describen los Modelos de Análisis, Diseño e Implementación.

Cumpliendo con los objetivos planteados y siguiendo las normas establecidas, el Sistema de Gestión y Seguimiento Académico está desarrollado con el fin de que los usuarios usen las herramientas tecnológicas actuales y se conecten al mismo a través de la web gestionando la información generada con eficiencia, confidencialidad, seguridad y comodidad.

Este sistema se ha implementado y puesto en ejecución para contribuir en el desarrollo y productividad de todos los que forman parte del Centro Educativo “Álamos”.

SUMMARY

Education as an institution has always been generating a lot of information and with the adventage of information technology and communication, educational institutions are required to use technological tools in the implementation of computer systems, with the aim of optimizing academic processes.

Having studied and understood all these processes involved, and after making a thorough observation and analysis of practice, we proceeded to develop a software application that is able to automate the management of the student at the institution.

Software development included the definition of the processes of the institution for a business model, from which the survey was conducted of the requirements of the application using a Use Case Model, subsequently generating a development plan based on the Unified Process. During the design models describing the Analysis, Design and Implementation.

Meeting the objectives and following the rules, the Management and Monitoring System in order to develop it Academically for users from using the technology available and connect to it via the web to manage the information generated with efficiency, confidentiality , safety and comfort.

This system has been implemented and applied to contribute to the development and productivity of all who are part of the Educational Centre at "Alamos."

CAPÍTULO I

1. INTRODUCCIÓN

El continuo desarrollo de la tecnología, ha llevado a los centros educativos del Ecuador a contar con sistemas informáticos que permitan mejorar la gestión de los procesos, garanticen la solidez de la información y en general, proporcionen nuevas formas de administración, gerencia y control.

En otras palabras, la disponibilidad de un sistema de información avanzado implica la adopción de un cambio en los procesos tradicionales y permite poner la información relevante a disposición de todos los miembros de la Institución a través de la Web.

Nuestro objeto de estudio se centra en el registro de matrículas, control de notas y asistencia, seguimiento de los estudiantes que académicamente se encuentren con un rendimiento bajo y ayuda en la planificación curricular docente, dicha información, en la actualidad se maneja de forma manual, limitando la entrega de información veraz y en tiempo real.

La necesidad de contar con una aplicación que permitan conocer la realidad actual en el centro educativo, es evidente e indispensable debido a que del control y procesamiento de toda esta información permitirá a la institución ser más competitiva dentro del ámbito educativo utilizando las herramientas con nuevos enfoques garantizando el proceso educativo de calidad que ofrece.

1.1. ALCANCE

Desarrollar e implementar una aplicación informática institucional que solucione los problemas de acceso a la información en tiempo real, que sirva a los docentes como herramienta de planificación curricular, que organice y controle todos los procesos académicos, apegado a los estándares¹ y lineamientos del Ministerio de Educación.

¹ **Estándares:** Que sirve como tipo, modelo, norma, patrón o referencia.

1.2. ESTUDIO DE LAS HERRAMIENTAS DE DESARROLLO

1.2.1. Frameworks

De manera general, un framework es una arquitectura de software, sobre la cual se sustenta un proyecto en términos de reutilización, permite la abstracción de clases, objetos o componentes que la conforman, tiene diferentes componentes de conexión a base de datos, controladores para la conexión directa a bases de datos como MYSQL, SQL SERVER, ORACLE o de manera general, mediante un ODBC², con diferentes métodos de acceso a datos en entornos conectado-desconectado, sin mucho esfuerzo, mejora la interfaz de desarrollo y disponibilidad de objetos, separando en capas la aplicación; adopta un modelo de programación, proporciona una estructura al código y hace que los desarrolladores escriban mejores códigos disminuyendo la cantidad de líneas y evitando redundar, haciéndolo más entendible, mantenible³, fácil de programar, convirtiendo complejas funciones en sencillas instrucciones con nuevas técnicas en el desarrollo como el uso de AJAX⁴, RSS⁵, WIKIS⁶, entre las más importantes.

Los frameworks suelen incluir:

- Soporte de programas.
- Bibliotecas.
- Software para desarrollar y unir diferentes componentes de un proyecto de desarrollo de programas.

Los frameworks son diseñados para facilitar el desarrollo de software, permitiendo a los diseñadores y programadores concentrar más tiempo y esfuerzo en identificar requerimientos de software y no tratando con los tediosos detalles de bajo nivel.

Dentro de la estructura del framework se integran varias técnicas de desarrollo ágil y de optimización de recursos software, como AJAX, entre las más importantes y que han permitido el cambio en el curso de la programación.

² **ODBC**: Open DataBase Connectivity

³ **MANTENIBLE**: Propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez que se ha presentado un evento de falla.

⁴ **AJAX**: Asynchronous Javascript and XML, Wiki (10), es una técnica de desarrollo para la Web que permite la interacción entre la aplicación y el usuario sin que éste tenga que preocuparse de que esto ocurra

⁵ **RSS**: Really Simple Syndication, es formato XML para syndicar o compartir contenido en la Web

⁶ **WIKIS**: es un sitio Web colaborativo que puede ser editado por varios usuarios

1.2.2. Frameworks para aplicaciones Web

1.2.2.1. Aplicaciones Web

En la ingeniería de software se denomina aplicación web a aquellos sistemas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información.

1.2.2.2. Frameworks para aplicaciones web

Los frameworks para aplicaciones son diseñados para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Este tipo de frameworks intentan aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web, por ejemplo, muchos framework proporcionan bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.

Con el paso del tiempo, el desarrollo de aplicaciones web es más sencillo, ya que los desarrolladores no deben perder el tiempo en averiguar cómo crear algo, puesto que gran cantidad de recursos ya existen, y es posible hacer uso de ellos, por medio de bibliotecas o de frameworks, que facilitan el trabajo de un programador.

Por medio del uso de los frameworks, se puede conseguir desarrollos más sencillos y en menor tiempo, al utilizar los recursos que éstos ofrecen.

A continuación se detalla una lista de los frameworks más usados:

- **PROTOTYPE:** es un framework Javascript para facilitar el desarrollo de aplicaciones web dinámicas.
- **MOOTOOLS:** es un framework compacto, modular y orientado a objetos, diseñado para que el desarrollador pueda hacer implementaciones avanzadas de Javascript.
- **RICO:** es una librería de uso abierto en JavaScript para crear aplicaciones amigables utilizando tecnología Ajax.
- **CAKEPHP:** está basado en los mismos principios que Ruby on Rails⁷, y se encuentra enfocado al rápido desarrollo. Su sistema de soporte creciente, simplicidad y escalabilidad hacen que sea una de los frameworks PHP más populares.
- **SYMFONY:** está enfocado a desarrolladores avanzados cuyo objeto principal son las aplicaciones de nivel corporativo. Este framework posee una gran cantidad de propiedades y puede hacer de todo; sin embargo su mayor falla es que es un poco más lento que otros frameworks PHP.

1.3. INTRODUCCIÓN DE GOOGLE WEB TOOLKIT (GWT)

Google Web Toolkit (GWT) es un framework de código abierto desarrollado por Google, lanzado en mayo del 2006 bajo la licencia Apache 2.0. con el objetivo de facilitar el desarrollo de Aplicaciones Ricas de Internet (RIAs), que permite escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores.

El corazón de GWT es un compilador de código JavaScript a partir de código Java. El compilador es capaz de generar código multibrowser⁸, aliviando la tarea del desarrollador de escribirlo por sí mismo. GWT genera código interpretable por los navegadores más importantes del mercado (Mozilla Firefox, Safari, Opera e Internet Explorer).

GWT proporciona un entorno de desarrollo basado en Java que permite construir aplicaciones AJAX utilizando dicho lenguaje. El procedimiento consiste en que primero se

⁷ **Ruby on Rails:** es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo la arquitectura Modelo Vista Controlador (MVC).

⁸ **Multibrowser:** es cuando una aplicación o una página web se puede ejecutar en todos los navegadores.

encapsula el objeto XMLHttpRequest⁹ API¹⁰, para luego minimizar los problemas entre exploradores. De esta manera, se puede construir rápido y eficientemente aplicaciones AJAX sin preocuparse demasiado acerca de la adaptación de su código para que funcione en varios navegadores.

Además, se puede aprovechar el Standard Widget Toolkit (SWT) o Swing estilo de programación, que proporciona un marco que permite combinar los widgets¹¹ en interfaces de usuario. Esta es una gran manera de mejorar la productividad y acortar sus líneas de tiempo de desarrollo, aprovechando el conocimiento del lenguaje de programación Java y la familiaridad con el marco de interfaz de desarrollo basado en eventos.

GWT proporciona además un conjunto de widgets listos para utilizar en la interfaz de usuario que se puede utilizar de inmediato para crear nuevas aplicaciones. También proporciona una forma sencilla de crear widgets innovadores mediante la combinación de las ya existentes. Para crear, depurar y probar cada unidad de aplicaciones AJAX se puede utilizar el IDE¹² de Eclipse. Así mismo, se puede construir RPC¹³ o llamadas a procedimientos remotos, que son servicios que proporcionan ciertas funcionalidades para acceder de forma asíncrona a la base de datos y a sus aplicaciones web fácilmente utilizando el marco de GWT RPC.

GWT permite integrar fácilmente con los servidores descritos en otros idiomas, así se puede mejorar las aplicaciones para proporcionar una experiencia de usuario mucho mejor mediante la utilización del Framework AJAX.

⁹ **XMLHttpRequest:** Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web, la interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

¹⁰ **API:** Application Program Interface

¹¹ **Widget:** Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets, entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

¹² **IDE:** Integrated Development Environment

¹³ **RPC:** Remote Procedure Call

1.3.1. Conceptos Generales

1.3.1.1. Lenguaje Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode¹⁴, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE¹⁵. Pero Java no se queda ahí, ya que en la industria para dispositivos móviles también hay una gran acogida para este lenguaje. La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma se refiere a la máquina virtual de Java (Java Virtual Machine).

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad para que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que el programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas.

Finalmente, se puede decir que Java brinda una solución para cada necesidad que se pudiera tener.

¹⁴ **Bytecode:** Código intermedio más abstracto que el código máquina

¹⁵ **J2EE:** Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones

Características del lenguaje

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar del ser humano. Posee características muy importantes, como:

- Es un lenguaje que se compila, generando ficheros de clases compiladas, mismas que son interpretadas por la máquina virtual de java, siendo ésta máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: El mismo código java funciona en cualquier sistema operativo que tenga instalada la máquina virtual java.
- Es un lenguaje seguro: La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, aunque el lenguaje carece de características inseguras, como por ejemplo los punteros.

La API de java permite ampliar el lenguaje para que sea capaz, por ejemplo de comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML¹⁶ dinámicas, crear aplicaciones visuales al estilo Windows, entre otras.

Para poder trabajar con java es necesario emplear un software que permita desarrollar en java. Existen varias alternativas comerciales en el mercado, como: JBuilder, Visual Age, Visual Café y un conjunto de herramientas shareware¹⁷, e incluso freeware¹⁸, que permiten trabajar con java.

1.3.1.2. Máquina Virtual Java

Una Máquina Virtual Java (en inglés *Java Virtual Machine*, JVM) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y

¹⁶ **HTML:** HyperText Markup Language

¹⁷ **Shareware:** Es una distribución de software en la que el usuario puede evaluar de forma gratuita el producto, pero con limitaciones en el tiempo de uso o en algunas de las formas de uso o con restricciones en las capacidades finales.

¹⁸ **Freeware:** Define un tipo de software que se distribuye sin costo, disponible para su uso y por tiempo ilimitado

ejecutar instrucciones expresadas en un código binario especial (el Java *bytecode*), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico. Como todas las piezas del rompecabezas Java, fue desarrollado originalmente por Sun Microsystems.

La JVM es una de las piezas fundamentales de la plataforma Java. La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde Sun Microsystems se han creado diferentes máquinas virtuales java para diferentes arquitecturas y así un programa `.class`¹⁹ escrito en un Windows que puede ser interpretado en un entorno Linux. Tan solo es necesario disponer de la máquina virtual para dichos entornos.

La máquina virtual de Java puede estar implementada en software, hardware, una herramienta de desarrollo o un Web browser.

1.3.1.3. JRE

El JRE (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java.

El usuario final usa el JRE como parte de paquetes software o plugins (o conectores) en un navegador Web.

Sun ofrece también el SDK de Java 2, o JDK (Java Development Kit) en cuyo seno reside el JRE, e incluye herramientas como el compilador de Java, Javadoc para generar documentación o el depurador. Puede también obtenerse como un paquete independiente, y puede considerarse como el entorno necesario para ejecutar una aplicación Java, mientras que un desarrollador debe además contar con otras facilidades que ofrece el JDK.

¹⁹ **.Class:** Tipos de archivos Java que contienen las instrucciones Java ya compiladas

1.3.1.4. JDK

El JDK es un súper conjunto de la JRE, y contiene todo lo que está en el JRE, además de herramientas tales como: los compiladores y depuradores necesarios para el desarrollo de applets²⁰ y aplicaciones. El Java Runtime Environment (JRE) proporciona a las bibliotecas, la máquina virtual Java y otros componentes para ejecutar applets y aplicaciones escritas en el lenguaje de programación Java.

1.3.1.5. AJAX

“AJAX, acrónimo²¹ de Asynchronous JavaScript And XML (JavaScript asíncrono²² y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA²³. 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”^[1].

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, además es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basada en estándares abiertos como JavaScript y DOM²⁴.

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.

[1]“AJAX”, 2011, <http://es.wikipedia.org/wiki/AJAX>

²⁰ **Applet**: Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

²¹ **Acrónimo**: Es la suma de los significados de las palabras que lo forman.

²² **Asíncrono**: No simultáneo o no coincidente.

²³ **RIA**: Rich Internet Applications

²⁴ **DOM**: Document Object Model

- DOM accedido con un lenguaje de scripting por parte del usuario, especialmente para implementaciones ECMAScript²⁵ como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe²⁶ en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON²⁷ y hasta EBML²⁸.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas, las cuales trabajan conjuntamente.

1.3.2. Google Web Toolkit

Google Web Toolkit (GWT) es un framework de desarrollo en Java de código abierto, que permite desarrollar y depurar aplicaciones AJAX usando el lenguaje de programación Java en el entorno de desarrollo de preferencia (refiriéndose al sistema operativo y a los IDEs).

1.3.2.1. Funciones

- **Comunicarse con el servidor mediante una sencillísima llamada RPC.**- GWT admite un conjunto indefinido de protocolos de transferencia, como JSON y XML, pero el mecanismo de llamada a procedimiento remoto (RPC) de GWT permite el establecimiento de comunicaciones Java de una forma especialmente sencilla y eficaz. Al realizar una llamada a un método remoto desde el navegador, el mecanismo RPC de GWT serializará automáticamente los argumentos, ejecutará el

²⁵**ECMAScript:** Es una especificación de lenguaje de programación basado en JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262.

²⁶ **Iframe:** Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal

²⁷ **JSON:** JavaScript Object Notation, es un formato ligero de intercambio de datos.

²⁸ **EBML:** Extensible Binary Meta Language

método adecuado en el servidor y anulará la serialización del valor de retorno del código cliente.

- **Utiliza componentes de la interfaz de usuario en varios proyectos.**- Se puede crear artilugios²⁹ reutilizables mediante la composición de otros artilugios y colocarlos después fácilmente en paneles. Para reutilizar el artilugio en otro proyecto, sólo se genera un archivo JAR.
- **Utiliza otras bibliotecas JavaScript y código JavaScript original.**- Si la biblioteca de clases de GWT no satisface las necesidades, se puede mezclar manualmente JavaScript en el código fuente Java mediante la interfaz JSNI³⁰. Por tanto, la ventaja de poder crear modelos de objetos Java Script como tipos de Java (por ejemplo, creación de código, reestructuración, sustitución de llamadas a funciones, etc.) sin tener que contar con una mayor cantidad de memoria ni sufrir una disminución de velocidad. Esta función posibilita un uso óptimo de las estructuras JSON.
- **Fácil uso del historial y del botón de retroceso del navegador.**- las aplicaciones AJAX no necesitan desgastar el botón de retroceso del navegador. GWT hace que resulte más sencillo utilizar el sitio con sólo añadir el estado al historial del botón de retroceso del navegador.
- **Elige las herramientas de desarrollo de forma productiva.**- GWT utiliza Java, por lo que se puede utilizar todas las herramientas favoritas de desarrollo con Java (Eclipse, IntelliJ, JProfiler, JUnit, NetBeans, etc.) al crear las aplicaciones AJAX. De esa forma, el desarrollador web, beneficiará el aumento de productividad derivado de la creación/solicitud de código y de la reestructuración de Java automatizada.
- **Prueba el código con JUnit.**- La integración directa de GWT con JUnit permite efectuar la comprobación por unidad tanto en un depurador como en un navegador,

²⁹ **Artilugio:** Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets.

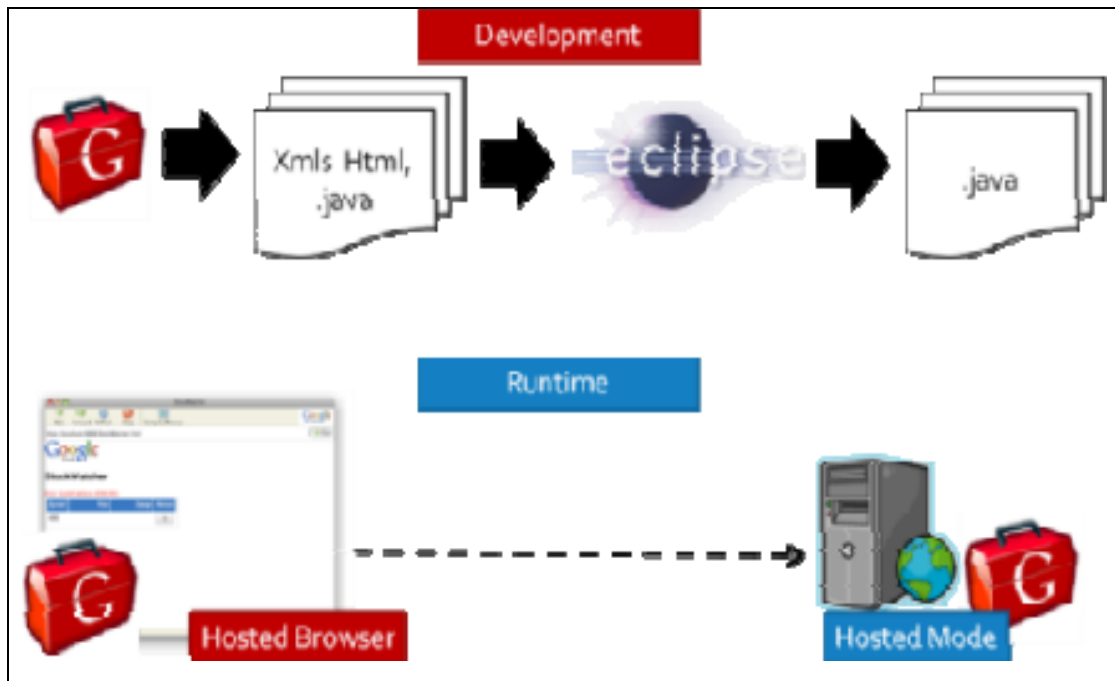
³⁰ **JSNI:** JavaScript Native Interface

e incluso comprobar por unidad las llamadas a procedimientos remotos (RPC) asíncronas.

1.3.2.2. Funcionamiento

Google Web Toolkit (GWT) permite crear aplicaciones AJAX en el lenguaje de programación Java que son compiladas posteriormente por GWT en código JavaScript ejecutable y optimizado que funciona automáticamente en los principales navegadores.

Durante el desarrollo de una aplicación, puede repetir rápidamente el mismo ciclo “editar actualizar - ver” típico de JavaScript y aprovechar la ventaja añadida de poder depurar y recorrer una a una todas las líneas de código Java. Cuando ya está todo listo para la implementación, GWT compilará el código fuente Java en archivos JavaScript optimizados independientes. Google Web Toolkit permite crear fácilmente tanto un artilugio para una página web como una aplicación completa.



Fuente: Funcionamiento de GWT [2]

Figura 1.1 Funcionamiento de GWT

[2] <http://code.google.com/intl/es-ES/webtoolkit/overview.html>

1.3.2.3. Características

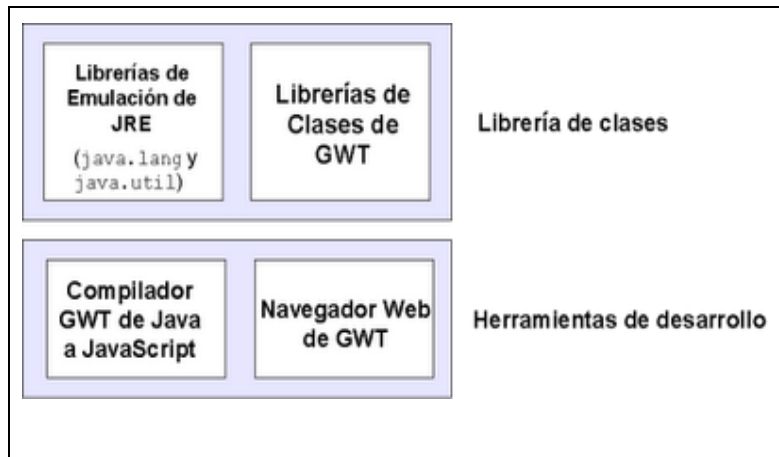
- GWT es un proyecto de código abierto, todo el código está disponible bajo la licencia Apache 2.0.
- Los componentes de la interfaz de usuario son dinámicos y reutilizables: crea un widget para construir otros. Coloca los widgets automáticamente en paneles y envía los widget propios a otros desarrolladores en archivos JAR³².
- Internacionalización: crea aplicaciones y librerías de Internacionalización rápida y fácilmente.
- Integración con JUnit³³: mediante la integración de JUnit en GWT se puede probar las aplicaciones y depurarlas en un navegador mientras se construyen.
- Compatibilidad con los navegadores: las aplicaciones en GWT serán automáticamente soportadas por navegadores como FireFox, Internet Explorer, Mozilla, Safari, y Opera sin ningún tipo de operación para la detección de los mismos, en la mayoría de los casos.
- Depuración en tiempo real: Cuando la aplicación esté lista, el código de la misma es traducido a JavaScript, pero mientras se estás desarrollando este corre sobre una máquina virtual de java. Lo que significa que en la fase de Desarrollo se tiene la posibilidad de depurar la aplicación con los avanzados sistemas de depuración y manipulación de excepciones incluidos en IDEs como Eclipse.

1.3.2.4. Arquitectura de GWT

GWT tiene cuatro componentes principales: un compilador Java-a-JavaScript, un navegador web, y dos librerías de clases:

³² **JAR:** Es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java

³³ **JUNIT:** Es un conjunto de bibliotecas utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.



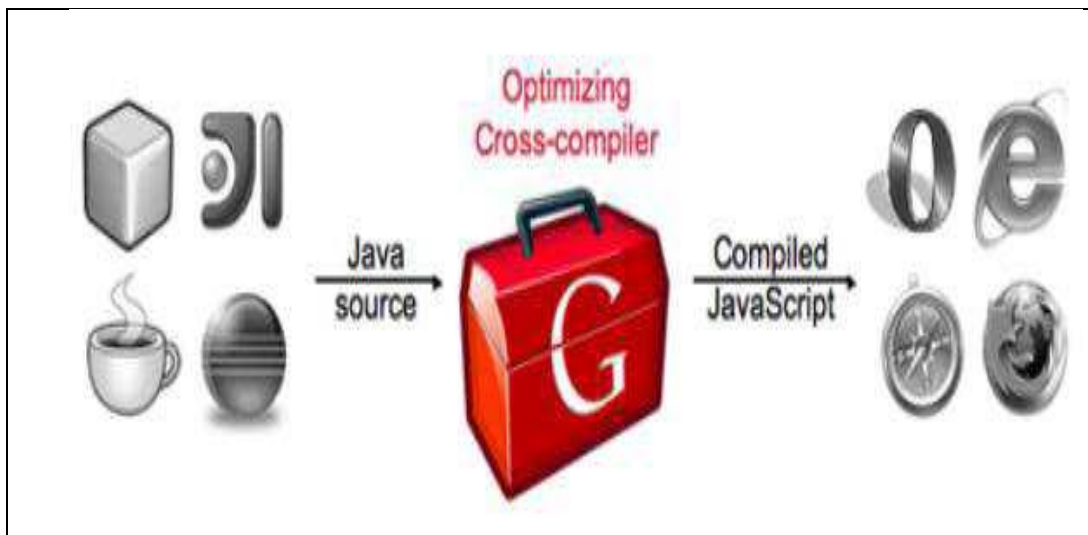
Fuente: La Arquitectura de Google Web Toolkit [3]

Figura 1.1 Arquitectura GWT

Los componentes son:

- **Compilador GWT Java-a-JavaScript**

El Compilador GWT Java-a-JavaScript traduce del lenguaje de programación Java a JavaScript. El compilador se utiliza cuando se necesita correr la aplicación en modo web.



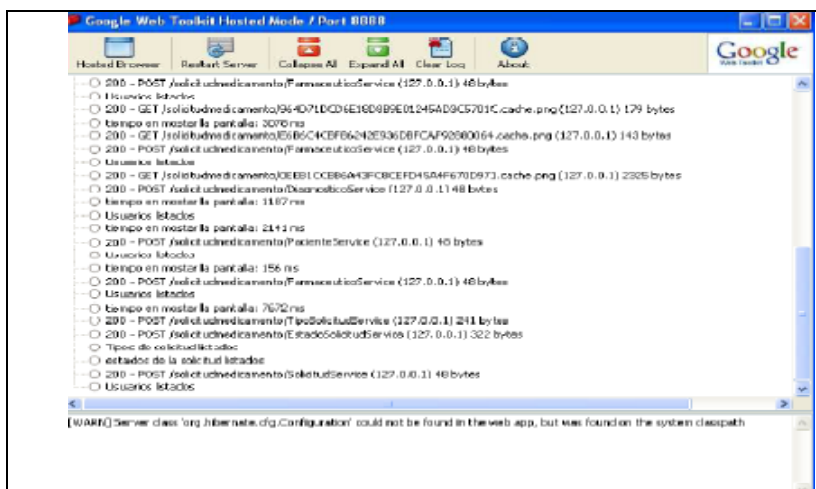
Fuente: Desarrollo de Aplicaciones Web 2.0 con GWT [4]

Figura 1.2 Compilador GWT Java-a-JavaScript

- **Navegador web de GWT**

El Navegador web de GWT permite correr y ejecutar GWT aplicaciones en modo alojado, donde lo que se está corriendo son bytecodes de Java sobre una máquina virtual sin compilarlos a JavaScript. Para lograr esto, el navegador GWT incrusta un controlador de

browser especial (un control del Internet Explorer sobre Windows o un control de Gecko/Mozilla sobre Linux) con hooks³⁴ dentro de la máquina virtual de Java.



Fuente: Propia

Figura 1.3 Navegador Web de GWT

▪ Emulación de librerías JRE

GWT contiene implementaciones en JavaScript de las librerías de clases más usadas en Java, incluyendo la mayoría de las clases del paquete `java.lang` y un subconjunto de clases del paquete `java.util`. El resto del estándar de librerías de Java no es soportado nativamente con GWT. Por ejemplo, las clases de los paquetes como `java.io` no se utilizan en aplicaciones web ya que éstas acceden a recursos en la red y al sistema de archivos local.

▪ Librería de clases de interfaz de usuario de GWT

Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que te permiten crear “widgets” para el navegador, como botones, cajas de texto, imágenes y texto. Éste es el núcleo de las librerías de interfaz de usuario para crear aplicaciones GWT.

[3] “La Arquitectura de Google Web Toolkit”, 2009, <http://esgooglewebtoolkit.blogspot.com/2007/09/la-arquitectura-de-google-web-toolkit.html>

[4] Desarrollo de Aplicaciones Web 2.0 con GWT, 2012, <http://www.slideshare.net/emiliobg/desarrollo-de-aplicaciones-web-20-con-gwt>

³⁴Hooks: Es una amplia gama de técnicas que se utilizan para alterar o aumentar el comportamiento de un sistema operativo, de aplicaciones, o de otros componentes de software mediante la interceptación de llamadas a funciones, mensajes o eventos pasados entre componentes de software.



Fuente: Propia

Figura 1.4 Librería de interfaz de usuario de GWT

1.3.2.5. Ventajas y Desventajas

Ventajas:

- No se necesita mucho conocimiento de otros lenguajes (JavaScript).
- Se utiliza el lenguaje de programación Java lo que permite tener todas las ventajas de dicho lenguaje y utilizar una gran cantidad de herramientas disponibles para el mismo.
- Los errores comunes JavaScripts son capturados por el programador durante compilación en vez de por los usuarios durante la ejecución.
- No se necesita trabajar con las incompatibilidades de los navegadores.
- Compatible con varias herramientas para desarrollar, depurar, probar las aplicaciones.
- Integración con JUnit.
- Internacionalización.
- Multiplataforma, Multinavegador.
- Permite avanzar desde un prototipo.
- Es similar a una aplicación de escritorio, con un alto dinamismo de las pantallas.
- Crea modelos de objetos JS como tipos de Java (por ejemplo, creación de código, reestructuración, sustitución de llamadas a funciones, etc.) sin tener que contar con una mayor cantidad de memoria ni sufrir una disminución de velocidad.
- Reduce el ancho de banda una vez cargada la aplicación en la caché.
- Reduce la carga en el servidor (los datos que se cargaban en la sesión del servidor ahora pasan al cliente).

- Permite la misma seguridad que otros frameworks (inclusive es más complicado inyectar datos debido al sistema de serialización que utiliza).
- Los errores comunes JavaScripts (types, type mismatches) son capturados por el programador durante compilación en vez de por los usuarios durante la ejecución

Desventajas:

- No es un enfoque estándar de HTML y JavaScript, por lo que la generación automática de HTML podría hacer más difícil el uso de herramientas estándar de HTML y CSS.
- No se utiliza directamente código JavaScript en el HTML.
- Sólo para desarrolladores Java.
- Muchos entornos Ajax permiten utilizar JavaScript en el cliente y luego elegir el lenguaje en el servidor. GWT se encuentra basado enteramente en Java.
- Enfoque inusual lo cual hace que la evaluación y administración es más difícil.

CAPÍTULO II

2. FASE DE INICIO

2.1. VISIÓN

2.1.1. Introducción

2.1.1.1. Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del Sistema de Gestión y Seguimiento Académico Estudiantil del Centro Educativo “Álamos”. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

El presente sistema se encargará de los procesos relacionados con el manejo de notas, gestión de entrevistas, calendario de actividades, registro de alumnos y profesores, y generará los reportes necesarios para el seguimiento académico de los estudiantes.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.

2.1.1.2. Alcance

Este documento de Visión se ocupa, como ya se ha apuntado, del Sistema de Gestión y Seguimiento Académico del Centro Educativo “Álamos” que es desarrollado por el tesista Luis Fabián Rivera Onofre, egresado de la Escuela de Ingeniería en Sistemas Computacionales de la Facultad de Ingenierías en Ciencias Aplicadas.

2.1.1.3. Definiciones, Siglas y Abreviaturas

Ver Glosario.

2.1.1.4. Referencias

- Glosario
- Plan de desarrollo de Software

- Metodología RUP (Rational Unified Process)
- Diagrama de Casos de Uso

2.1.2. Posicionamiento

2.1.2.1. Oportunidad de negocio

Este sistema permitirá al Centro Educativo “Álamos” automatizar los procesos académicos generados en el proceso de enseñanza aprendizaje y se podrá acceder a sus utilidades a través de la Web, de forma rápida y sencilla y sin necesidad de intermediarios lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas y amigables.

2.1.2.2. Definición del problema

El problema de	No contar con una herramienta tecnológica que permita automatizar los procesos académicos generados en la Institución. Existen procesos que se realizan manualmente. Traspapelar documentos Inexistencia de un sistema que permita dar un seguimiento académico a los estudiantes que ayude a mejorar su rendimiento.
afecta a	A todos estudiantes, docentes, padres de familia del Centro Educativo “Álamos” involucrados en los procesos que se generan en la Institución
El impacto asociados	Almacenar y organizar la información referente al registro de notas, seguimiento estudiantil, control de planificaciones, para que estos datos sean accesibles de manera oportuna y eficaz desde lugares físicamente remotos a las instalaciones de la Institución. Este es un proceso automatizado con infraestructura para Web.
Una solución adecuada sería	Automatizar el proceso, publicándolo en el internet usando una base de datos integrada al sistema académico y generar interfaces amigables y sencillas, que cuenten con las herramientas necesarias para la correcta ejecución de los procesos propios de un sistema académico.

Fuente: Propia

Tabla 2.1 Definición del Problema

2.1.2.3. Sentencia que define la posición del Producto

Para	EL Centro Educativo “Álamos” Rector Vicerrector Coordinador de Convivencia Docentes Departamentos Administrativos
Quienes	Todos los miembros de la Institución involucrados en el proceso educativo
El nombre del producto	Sistema de Gestión y Seguimiento Académico Estudiantil SAG.
Que	Contribuirá a la automatización de los procesos académicos y mejorará el rendimiento estudiantil .
no como	Los esfuerzos y procesos manuales y foráneos realizados hasta hoy.
Nuestro producto	Permite automatizar los diferentes procesos que implican el registro de notas, gestión de entrevistas, planificación docente, seguimiento estudiantil. Además proporciona un acceso rápido y actualizado de la información desde cualquier terminal con acceso a internet

Fuente: Propia

Tabla 2.1 Definición de la Posición del Producto

2.1.3. Descripción de los interesados y usuarios

Para proveer de una forma más efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes como parte del proceso de modelado de requerimientos.

Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de porqué estos requisitos son necesarios.

2.1.3.1. Resumen de los interesados

Los interesados son todas aquellas personas directamente involucradas en la definición y alcance de este proyecto. A continuación se presenta la lista de los interesados:

Nombre	Descripción	Responsabilidades
Msc. Juan Carlos Salas	Rector del Centro Educativo "Álamos"	Establecer las políticas y requerimientos para el desarrollo del proyecto. Analizar los avances y facilitar los recursos necesarios para el desarrollo del proyecto.
Lic Pablo Pillajo	Vicerrector, encargado de llevar el registro y control de notas, preceptorías	Responsable facilitar la información referente al proceso de registro de notas y seguimiento académico estudiantil
Lic Vinicio Noguera	Secretario, encargado del registro de matrículas	Responsable facilitar la información personal de los alumnos.
Egresado Luis Rivera	Analista de Sistemas	Responsable del desarrollo del proyecto, con conocimientos de entorno de desarrollo para que los prototipos puedan ser lo más cercanos posibles al producto final.

Fuente: Propia

Tabla 2.2 Descripción de interesados y usuarios

2.1.3.2. Resumen de los usuarios

Los usuarios son todas aquellas personas involucradas directamente en el uso del sistema. A continuación se presenta una lista de los usuarios:

Nombre	Descripción	Stakeholder
Administrador del sistema	Persona de la Institución de nivel administrativo	Administrar funcionalmente el sistema (gestionar los parámetros, verificar registros, gestionar cuentas y roles, dar mantenimiento al sistema frente a nuevos requerimientos).
Administrador funcional del sistema	Persona de nivel administrativo de la Institución.	Administrar funcionalmente el sistema: acceso a notas, reportes, gestión del calendario.
Coordinador	Persona de nivel administrativo de la institución	Ingresa la información de los alumnos referente a asistencia y conducta.
Docente	Persona de nivel operativo de la institución	Ingresa la información al sistema referente a calificaciones y planificaciones.
Usuario	Representantes de los estudiantes de la institución	Consultan la información académica de los estudiantes

Fuente: Propia

Tabla 2.4 Resumen de Usuarios

2.1.3.3. Entorno de usuario

Los usuarios podrán acceder al sistema mediante un módulo de autenticación debidamente validado. Trabajaran en un entorno WEB (Cliente-servidor) por lo que la

aplicación se ejecutará en un navegador de Internet y soportará cualquier plataforma.

El sistema será creado de forma que se pueda trabajar sin instalar el aplicativo en el computador.

2.1.3.4. Perfiles de los Stakeholders

Rector

Representante	Msc. Juan Carlos Salas
Descripción	Rector encargado de tomar las decisiones del Centro Educativo “Álamos”
Tipo	Administrativo
Responsabilidades	Establecer los lineamientos generales para el desarrollo del proyecto. Coordinar a nivel directivo los diferentes requerimientos que surjan en el desarrollo del sistema.
Criterio de éxito	Mantener activa la aplicación luego de ser implantada.
Grado de participación	Aprobación de la implantación del sistema
Comentarios	Mantener una relación constante con el desarrollo del proyecto. Brindar apoyo a nivel gerencial cuando sea necesario.

Fuente: Propia

Tabla 2.5 Perfil del Rector

Vicerrector

Representante	Lic Pablo Pillajo
Descripción	Vicerrector, encargado de llevar el control de notas y el seguimiento académico de los estudiantes del Centro Educativo “Álamos”
Tipo	Administrativo
Responsabilidades	Responsable de facilitar información referente al proceso de control, registro de notas.
Criterios de éxito	Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos.
Grado de participación	Responsable de establecer el calendario de registro de notas, acceso a notas, y generación de reportes.
Comentarios	Ninguno

Fuente: Propia

Tabla 2.6 Perfil del Vicerrector

Secretario

Representante	Lic. Vinicio Noguera
Descripción	Responsable de la información personal de los alumnos y docentes de la Institución
Tipo	Administrativo
Responsabilidades	Responsable de la matriculación estudiantil.
Criterio de Éxito	Cumplir con el cronograma determinado. Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos.
Implicación	Registra la información personal de los estudiantes
Comentarios	Ninguno

Fuente: Propia

Tabla 2.7 Perfil del Secretario

2.1.3.5. Perfiles de usuario

Administrador del Sistema

Representante	Administrativo de la Institución
Descripción	Persona del Área de Informática que administra el Sistema de Gestión y Seguimiento Académico Estudiantil designado por una autoridad de la Institución
Tipo	Administrador del Sistema
Responsabilidades	Gestionar accesos a usuarios Administrar los módulos de sistema Configurar los parámetros del sistema Realizar cambios en los registros.
Criterio de Éxito	Inicialización de parámetros completa Usuarios registrados
Grado de participación	Activa
Comentarios	Ninguno

Fuente: Propia

Tabla 2.8 Perfil de Usuario: Administrador del Sistema

Administrador funcional del Sistema

Representante	Administrativo de la Institución
Descripción	Persona de nivel administrativo de la Institución
Tipo	Usuario
Responsabilidades	Administrar funcionalmente el sistema: acceso notas y consultas, gestionar calendario.

Criterio de Éxito	Registro de notas en las fechas establecidas
Grado de participación	Activa
Comentarios	Ninguno

Fuente: Propia

Tabla 2.9 Perfil de Usuario: Administrador Funcional del Sistema

Coordinador

Representante	Administrativo de la Institución
Descripción	Persona de nivel administrativo de la Institución
Tipo	Persona de nivel administrativo miembro del departamento de coordinación y convivencia
Responsabilidades	Ingresar/actualizar las asistencias y conducta de los estudiantes
Criterio de Éxito	Registro exitoso
Grado de participación	Activa
Comentarios	Ninguno

Fuente: Propia

Tabla 2.10 Perfil de Usuario: Coordinador

Docente

Representante	Personal de la Institución
Descripción	Docentes que pertenecen a la Institución
Tipo	Usuario del sistema
Responsabilidades	Realizar el registro de notas mensuales
Criterio de Éxito	Registro exitoso, consulta de calificaciones exitosa
Grado de participación	Activa
Comentarios	Ninguno

Fuente: Propia

Tabla 2.11 Perfil de Usuario: Docente

Usuarios del Sistema

Representante	Representante de los estudiante
Descripción	Representante legal del estudiantes
Tipo	Usuario del sistema
Responsabilidades	Responsable de la matriculación estudiantil.
Criterio de Éxito	Realizar consultas de la información académica del estudiante.
Implicación	Registra consultas exitosas
Comentarios	Ninguno

Fuente: Propia

Tabla 2.12 Perfil de Usuario: Usuarios del Sistema

2.1.3.6. Necesidades de los interesados y usuarios

Necesidades	Prioridad	Inquietudes	Solución Actual	Solución propuesta
Un sistema que realice el control de notas, asistencia , gestión de entrevistas y realice un seguimiento académico del estudiantil del Centro Educativo	Alta	El sistema debe controlar el ingreso de notas, control de asistencias, gestionar las entrevistas, organizar la información y permitir realizar consultas cuando los usuarios lo requieran	NO EXISTE	Desarrollar un sistema que automatice el proceso y optimice los recursos.
Implementar inmediatamente el sistema con el fin de ponerlo en funcionamiento para que la institución pueda tener un control dinámico de los procesos académicos.	Alta	Permitir a los usuarios realizar consultas de la información de los estudiantes y de los docentes.	Actualmente el control se realiza mediante hojas electrónicas.	El uso del sistema será totalmente gratuito y contará con varias herramientas que faciliten la búsqueda de información.
Elaborar el sistema utilizando la tecnología web (Cliente – Servidor)	Alta	Se debe utilizar las herramientas y tecnologías actuales de diseño y desarrollo WEB.	N/A	Desarrollar el sistema utilizando las tecnologías WEB 2.0 encapsuladas dentro del framework GWT

				y PHP 5.0
La interfaz del sistema debe ser fácil de manejar, cumpliendo con todos los requerimientos establecidos.	Alta	Cumplir con todos los requerimientos de los usuarios.	N/A	Explotar al máximo las ventajas en torno a los componentes que nos ofrece GWT y PHP.

Fuente: Propia

Tabla 2.13 Necesidades de los interesados y usuarios

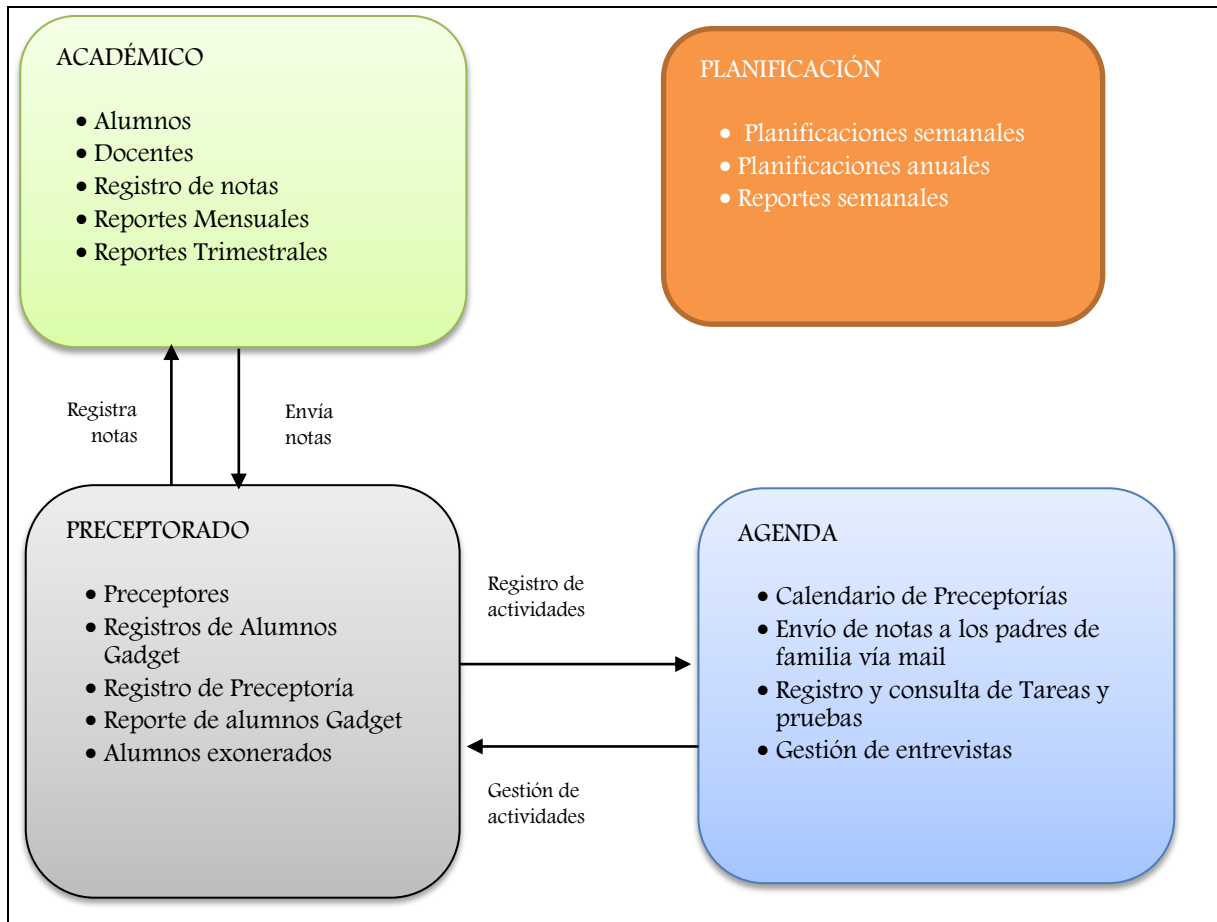
2.1.3.7. Alternativas y competencia

Adquirir un sistema desarrollado externamente para otra Institución.

2.1.4. Vista General del Producto

El producto de software a desarrollar es un sistema WEB el cual permitirá realizar un control, seguimiento y gestión de todos los procesos académicos que intervienen en la Institución, integrando a estudiantes, docentes y padres de familia. El sistema consta de cuatro módulos: un módulo Académico en donde se va a controlar y registrar toda la información que se genera en el proceso de enseñanza como son las notas mensuales, la información general de los alumnos y docentes y la generación de varios reportes indispensables. Un módulo Planificación en el cual se registrará las planificaciones anuales y semanales de los docentes y servirá como una herramienta de control y evaluación continua para las autoridades del plantel y una gran ayuda para los profesores. Un módulo Preceptorado el mismo que será una herramienta de ayuda a los docentes que son designados como Preceptores para dar el seguimiento personalizado a los alumnos pero de una manera organizada y automatizada. Finalmente una Agenda la cual servirá como una herramienta de comunicación entre padres y profesores, además será una fuente de consulta de tareas y pruebas para los alumnos.

2.1.4.1. Perspectiva del producto



Fuente: Propia

Figura 2.1 Perspectiva del Producto

2.1.4.2. Resumen de capacidades

Beneficios para el usuario	Características que lo soportan
Eficiente manejo de la información de la institución	El sistema incluye desde el registro y validación de usuarios hasta la confirmación de entrevistas y gestión académica de los estudiantes. El uso será libre y llevado a cabo por los usuarios finales de forma sistematizada y ordenada.
Control y registro de notas en tiempos definidos por las autoridades de la Institución	El sistema controlará los registros de notas por parte de los docentes calendarizando estos registros para que los representantes de los estudiantes puedan hacer consultas y tener informes detallados en tiempo real.
Se tendrá alta disponibilidad.	El acceso a la información a través de la Web permitirá a los usuarios un acceso inmediato desde cualquier punto del mundo
Facilidades para el análisis y gestión de la información.	Permitirá generar diversos tipos de reportes y funciones de consulta.

Fuente: Propia

Tabla 2.14 Resumen de Capacidades

2.1.4.3. Costos y precios

Detalle		USD	Real (USD)
Hardware	Equipos de Computación	1500	0.00
	Servidor de Aplicación Web	0.00	0.00
	Servidor de Base de Datos	0.00	0.00
	Equipo con Web Browser	700	0.00

Software	MySql 5.0	0.00	0.00
	GOOGLE WEB TOOLKIT 2.3.0	0.00	0.00
	PHP 5.0	0.00	0.00
	Notepad ++	0.00	0.00
Capacitación a los Desarrollares	Cursos y Libros, Asesoramiento	800	800
Proyecto	Papelería y Suministros de Oficina	300	300
Subtotal	(Parcial)	3300	1100
5% Imprevistos		165	55.00
Total		3465	1155

Fuente: Propia

Tabla 2.3 Costos y Precios

2.1.4.4. Licenciamiento e instalación

La propuesta de desarrollo de software está implementada en su totalidad utilizando software libre, razón por la cual no se necesita ningún tipo de licencia.

2.1.5. Descripción del Producto

2.1.5.1. Facilidad de acceso y uso

El Sistema de Gestión y Seguimiento Académico Estudiantil está desarrollado utilizando tecnología Web y las ventajas que ofrece el framework GWT y PHP de fácil implementación, acceso y uso.

2.1.5.2. Unificación de la información

Uno de los principales objetivos del sistema es determinar y presentar al usuario un formato unificado de ingreso, procesamiento y consulta de datos.

2.1.5.3. Mejor control y validación de la información

Los usuarios del sistema contarán con facilidades para la verificación de la información organizada.

2.1.5.4. Rangos de calidad

El desarrollo del Sistema de Gestión Académica Estudiantil se ajusta a la Metodología de Desarrollo de Software RUP, contemplando los parámetros de calidad que la metodología define.

2.2. PLAN DE DESARROLLO DE SOFTWARE

2.2.1. Introducción

Este Plan de Desarrollo de Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto “Implementación Web para la gestión y Seguimiento Académico Estudiantil del Centro Educativo Álamos” que resultará en la implementación del Sistema de Gestión y Seguimiento Académico Estudiantil SAG. Este documento provee una visión global del enfoque de desarrollo propuesto.

Para el proyecto utilizaremos metodología RUP. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo el proceso.

El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

2.2.1.1. Propósito

Este Plan de Desarrollo de Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto “Implementación web para la gestión y seguimiento académico estudiantil del Centro Educativo Álamos” que resultará en la implementación del Sistema de Gestión y Seguimiento Académico. Este documento provee una visión global del enfoque de desarrollo propuesto.

Para el proyecto se utilizará la metodología RUP que incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo el proceso.

El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

2.2.1.2. Alcance

En este documento se describe el plan global usado para el desarrollo del Sistema de Gestión y Seguimiento Académico del Centro Educativo Álamos. El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma individual.

Durante el proceso de desarrollo de este documento “Visión” se definen las características del producto a desarrollar, lo cual es la base para la planificación de las iteraciones. El Plan de Desarrollo del Software, se basa en la definición de requisitos obtenidos de la experiencia en la implantación de sistemas académicos, para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

2.2.2. Vista General del Proyecto

El Centro Educativo Álamos es una Institución Educativa de la ciudad de Ibarra diferenciada de las demás instituciones por su modelo de educación personalizada en la cual se integra el proceso académico a estudiantes, padres y profesores, que tiene como misión esencial contribuir al desarrollo integral de los niños y jóvenes de la ciudad de Ibarra. En los próximos 5 años, será un centro de educación orientado al desarrollo sostenible a través de una gestión de calidad académica, diversificando los vínculos con el entorno y contribuyendo al desarrollo humano de la ciudad. Con esto se crea una previsible adaptación a los nuevos sistemas de información y a la evolución tecnológica. Por ello, el Centro Educativo “Álamos” considera necesario el desarrollo de un Sistema informático que

posibilite la Gestión y Seguimiento académico estudiantil, esto como parte del proceso de automatización de todos los procesos involucrados mejorando la comunicación institucional de las distintas áreas.

El proyecto proporciona una propuesta para el desarrollo de un sistema académico con acceso a todos los miembros que conforman la Institución. Los sub-módulos identificados son los siguientes:

Académico

- Registro de alumnos y docentes
- Registro de Notas
- Consulta de Notas
- Reporte de Notas por curso
- Pase de año

Preceptorado

- Asignar alumno a Preceptor
- Registro de Preceptoría
- Registro de alumnos Gadget
- Reporte de alumnos Gadget
- Exoneraciones

Planificación

- Registro de Planificación anual
- Registro de Planificación Semanal
- Reporte de Profesores que no han entregado la Planificación

Agenda

- Calendario de Preceptorías
- Envío de Notas a Padres
- Registro de Tareas y Pruebas
- Gestión de Entrevistas

2.2.2.1. Suposiciones y Restricciones

Las suposiciones y restricciones respecto del Sistema de Gestión y Seguimiento académico, y que se derivan directamente son:

- El proyecto está completamente financiado por el Centro Educativo “Álamos” y al usar todas las herramientas de desarrollo libres, no habrá inconvenientes relacionados al costo total del proyecto ni a la agilidad con la que se deben atender los desembolsos parciales del mismo.
- El sistema será diseñado sobre plataforma Web y cumplirá con los estándares de calidad vigentes para desarrollo de software. Esto se conseguirá cumpliendo con la metodología RUP para el proceso de ingeniería de software.

De acuerdo al desarrollo del proyecto las suposiciones y restricciones se irán incrementando paulatinamente.

2.2.2.2. Entregables del proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de la unificación de procesos desde la perspectiva de artefactos, y que se propone para este proyecto.

Cabe destacar que con el desarrollo del proyecto cada todos los artefactos están sujetos a modificaciones.

2.2.2.3. Plan de Desarrollo del Software

Es el presente documento.

- **Visión**

Este documento define la visión del proyecto desde la perspectiva de los usuarios, especificando las necesidades y características del proyecto. Constituye una base en cuanto a los requisitos del sistema.

- **Glosario**

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

- **Especificaciones de Casos de Uso**

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada

utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

▪ **Prototipos de Interfaces de Usuario**

Se trata de prototipos que permiten al usuario obtener una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

▪ **Modelo de Análisis y Diseño**

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

▪ **Modelo de Datos**

Previniendo 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 Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

▪ **Modelo de Implementación**

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo

es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

- **Modelo de Despliegue**

Este modelo muestra la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

- **Casos de Prueba**

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas y los resultados esperados. Estos casos son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento específico con las instrucciones necesarias para realizarla, mismo que podrá ser automatizables mediante un script.

- **Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento

- **Producto**

Los archivos del producto empaquetado y almacenado en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión de la aplicación al final de cada iteración.

2.2.3. Organización del Proyecto

2.2.3.1. Participantes en el Proyecto

El personal del proyecto considerando las fases de Inicio, Elaboración y dos iteraciones de la fase de Construcción, estará formado por los siguientes puestos de trabajo y personal asociado:

- **Jefe de Proyecto.** Con experiencia en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y el proceso de desarrollo RUP.
- **Analista de Sistemas.** El perfil establecido es: Ingeniero en Informática con conocimientos de UML, uno de ellos al menos con experiencia en sistemas afines a la línea del proyecto.

- **Analistas – Programadores.** Luis Fabián Rivera Onofre con conocimientos en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final.
- **Ingeniero de Software.** El perfil establecido es: Ingeniero en Informática que participará realizando labores de gestión de requisitos, gestión de configuración, documentación y diseño de datos. Encargada de las pruebas funcionales del sistema, realizará la labor de Tester.

2.2.3.2. Interfaces Externas

Se define los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada subsistema y según el plan establecido.

2.2.3.3. Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que se desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario

Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.
-----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Propia

Tabla 2.16 Roles y Responsabilidades

2.2.4. Plan del Proyecto

2.2.4.1. Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar).

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	5 semanas
Fase de Elaboración	2	5 semanas
Fase de Construcción	2	14 semanas
Fase de Transición		

Fuente: Propia

Tabla 2.17 plan de Fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta

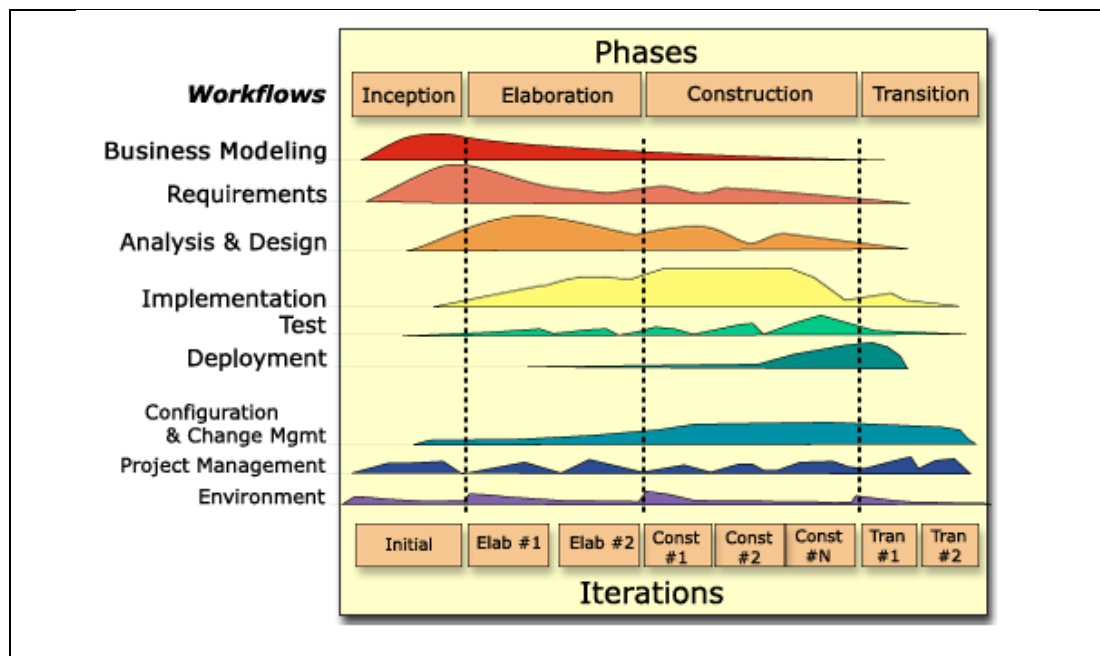
	<p>fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En este caso particular, por no incluirse las fases de revisión y entrega de todos los artefactos hasta este punto de desarrollo, también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.</p>
Fase de Construcción	<p>Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada.</p>
Fase de Transición	<p>En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.</p>

Fuente: Propia

Tabla 2.18 Plan de Fases: Hitos

2.2.4.2. Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow³⁵) en un momento determinado del desarrollo.



Fuente: RUP Etapa diseño^[5]

Figura 2.2 Descripción del Proceso Unificado

³⁵ **Workflow:** Flujo de trabajo.

^[5] "RUP Etapa diseño", 2007, <http://www.scribd.com/doc/395783/RUP-etapa-diseno>

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 7 11/07 – 15/07	Semana 20 10/10 – 14/10
Requisitos		
Glosario	Semana 2 06/06 – 10/06	Semana 4 20/06 – 24/06
Visión	Semana 2 06/06 – 10/06	Semana 6 04/07 – 08/07
Modelo de Casos de Uso	Semana 8 18/07 – 22/07	siguiente fase
Especificación de Casos de Uso	Semana 9 25/07 – 29/07	siguiente fase
Especificaciones Adicionales	Semana 8 18/07 – 22/07	siguiente fase
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 12 15/08 – 19/08	siguiente fase
Modelo de Datos	Semana 14 29/08 – 02/09	siguiente fase
Implementación		
Prototipos de Interfaces de Usuario	Semana 21 17/10 – 21/10	Semana 22 24/10 – 28/10
Modelo de Implementación	Semana 22 24/10 – 28/10	siguiente fase
Pruebas		
Casos de Pruebas Funcionales	Semana 32 26/12 – 30/12	siguiente fase
Despliegue		
Modelo de Despliegue	Semana 12	siguiente fase

	15/08 – 19/08	
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Semana 1	Semana 2
	30/05 – 03/06	06/06 – 10/06
Ambiente	Durante todo el proyecto	

Fuente: Propia

Tabla 2.19 Roles y Responsabilidades

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 7 11/07 – 15/07	aprobado
Requisitos		
Glosario	Semana 2 06/06 – 10/06	aprobado
Visión	Semana 2 06/06 – 10/06	aprobado
Modelo de Casos de Uso	Semana 8 18/07 – 22/07	Semana 9 25/07 – 29/07
Especificación de Casos de Uso	Semana 9 25/07 – 29/07	Semana 10 01/08 – 05/08
Especificaciones Adicionales	Semana 8 18/07 – 22/07	Semana 11 08/08 – 12/08
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 12 15/08 – 19/08	Revisar en cada iteración
Modelo de Datos	Semana 14 29/08 – 02/09	Revisar en cada iteración
Implementación		
Prototipos de Interfaces de Usuario	Semana 21 17/10 – 21/10	Revisar en cada iteración
Modelo de Implementación	Semana 22	Revisar en cada

	24/10 – 28/10	iteración
Pruebas		
Casos de Pruebas Funcionales	Semana 32 26/12 – 30/12	Revisar en cada iteración
Despliegue		
Modelo de Despliegue	Semana 12 15/08 – 19/08	Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Semana 1 30/05 – 03/06	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Fuente: Propia

Tabla 2.20 Roles y Responsabilidades

2.2.4.3. Seguimiento y Control del Proyecto

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por parte del jefe del mismo y por el Comité de Seguimiento y Control.

Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas

deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist³⁶ incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos³⁷ o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones.

También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que estas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá un registro del estado de cada artefacto, estableciendo una versión; el cual podrá ser modificado sólo por una Solicitud de Cambio aprobada.

³⁶ **Checklist:** Lista de verificación.

³⁷ **Mitigarlos:** Modelar, aplacar o suavizar la dureza de algo.

CAPÍTULO III

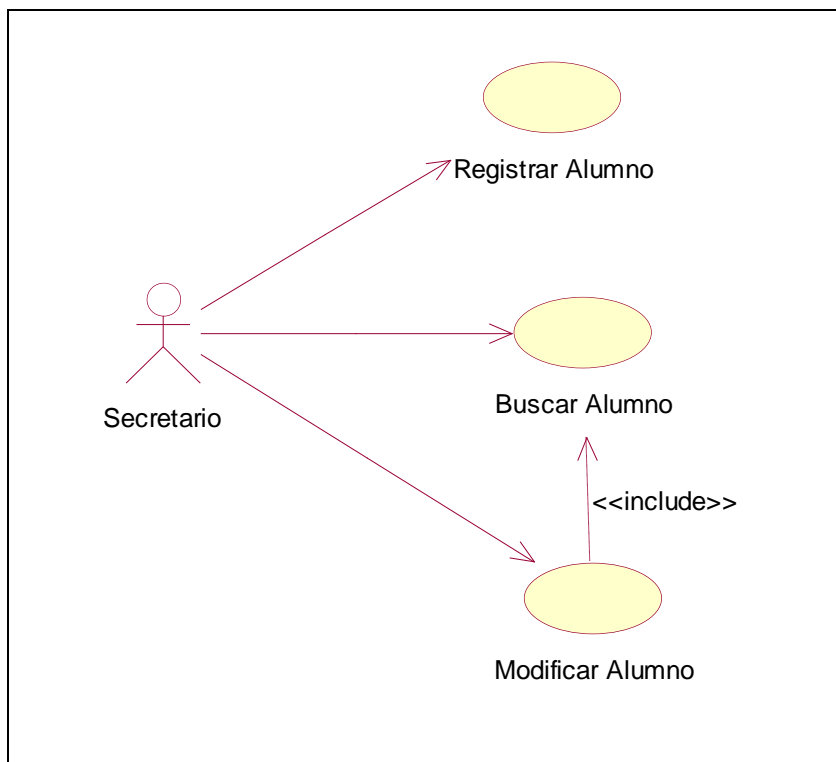
3. CASOS DE USO

3.1. MODELOS DE CASO DE USO

En esta sección se muestran los diagramas de casos de uso, que fueron identificados durante el proceso de especificación de requisitos, los cuales permiten mostrar la funcionalidad del sistema de forma detallada.

3.1.1. Especificación Caso

3.1.1.1. Especificación Caso de Uso: Ingresar Alumnos



Fuente: Propia

Figura 3.1 Casos de Uso: Ingresar Alumnos

A continuación se muestra una breve explicación de los casos de uso plasmados en el diagrama anterior.

Detalle de casos de uso ingresar alumnos	
Caso de uso	Descripción
Registrar Alumno	Se encarga de registrar los alumnos nuevos dentro del sistema.
Buscar Alumno	Realiza la búsqueda de un Estudiante en el sistema, controla que exista el Estudiante en el sistema
Modificar Alumno	Realiza la modificación de los atributos de un Estudiante

Fuente: Propia

Tabla 3.1 Detalle Casos de Uso Ingresar Alumno

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- **Registrar Alumno**

Caso de uso: Registrar Alumno	
Actor	Secretario
Descripción	Registrar los alumnos nuevos dentro del sistema
Precondición	<ul style="list-style-type: none"> • Existe un Estudiante a ser ingresado • Los datos de los alumnos deben estar registrados en el sistema.
Post-condición	<ul style="list-style-type: none"> • El Estudiante queda registrado en el sistema
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema y hace clic en el botón inicio. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Registrar Estudiante” del menú principal. 4. El sistema despliega el submenú “Registrar”. 5. El actor llena los campos del formulario desplegado. 6. El actor registra la información mediante la opción “Guardar”. 7. El sistema registra exitosamente al estudiante y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Limpiar la información de los campos e ingresar un estudiante nuevo <ol style="list-style-type: none"> 1. El actor selecciona la opción “Nuevo”. 2. El actor llena los campos con la información del nuevo estudiante 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.2 Especificación Casos de Uso Registrar alumno

- **Buscar Alumno**

Caso de uso: Buscar Alumno	
Actor	Secretario
Descripción	Realiza la búsqueda de un estudiante dentro del Sistema.
Precondición	<ul style="list-style-type: none"> Exista el Estudiante
Post-condición	<ul style="list-style-type: none"> El Estudiante es encontrado, o no existe el Estudiante en el sistema.
Flujo normal de eventos	
<ol style="list-style-type: none"> El actor entra al sistema y hace clic en el botón inicio. El sistema despliega el menú principal. El actor selecciona la opción del menú “Buscar estudiante”. El actor selecciona un criterio de búsqueda. El actor ingresa los atributos adecuados de acuerdo al criterio de búsqueda. El sistema realiza la búsqueda por atributos según criterio de búsqueda. El sistema despliega el estudiante especificado. El actor confirma que el estudiante encontrado es el buscado. 	
Flujo alternativo	
<ol style="list-style-type: none"> El actor cancela la búsqueda El actor sale del sistema. El sistema no encuentra el estudiante buscado 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.3 Especificación Casos de Uso Buscar Alumno

- Modificar Alumno**

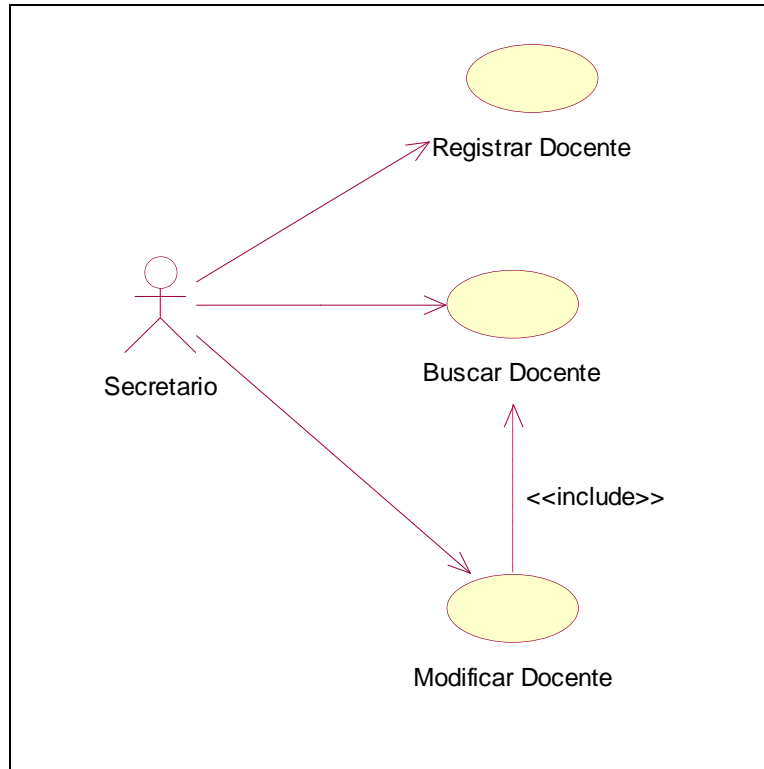
Caso de uso: Modificar Alumno	
Actor	Secretario
Descripción	Realiza la modificación de los atributos de un estudiante dentro del Sistema.
Precondición	<ul style="list-style-type: none"> Exista el Estudiante Existan atributos a ser modificados
Post-condición	<ul style="list-style-type: none"> El Estudiante modificado queda registrado en el sistema, o el Estudiante a modificar no existía en el sistema
Flujo normal de eventos	
<ol style="list-style-type: none"> El actor ejecuta el caso de uso buscar alumno. El sistema muestra la información del estudiante seleccionado. El actor modifica los atributos del estudiante. El sistema verifica los atributos modificados y muestra un mensaje de confirmación. El sistema actualiza la información del estudiante. 	
Flujo alternativo	
<ul style="list-style-type: none"> Validación de datos <ol style="list-style-type: none"> El sistema valida la información registrada. El actor corrige la información mal registrada. 	
Excepciones	

Observaciones

Fuente: Propia

Tabla 3.4 Especificación Casos de Uso Modificar Alumno

3.1.1.2. Especificación Caso de Uso: Ingresar Docentes



Fuente: Propia

Figura 3.2 Casos de Uso Ingresar Docente

A continuación se muestra una breve explicación de los casos de uso creados en el diagrama anterior.

Detalle de casos de uso ingresar docente	
Caso de uso	Descripción
Registrar Docente	Se encarga de registrar los docentes nuevos dentro del sistema.
Buscar Docente	Realiza la búsqueda de un docente en el sistema, controla que exista el docente en el sistema
Modificar Docente	Realiza la modificación de los atributos de un docente

Fuente: Propia

Tabla 3.5 Detalle Casos de Uso: Ingresar Docente

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- **Registrar Docente**

Caso de uso: Registrar Docente	
Actor	Secretario
Descripción	Registrar los docentes nuevos dentro del sistema
Precondición	<ul style="list-style-type: none"> • Existe un Docente a ser ingresado • Los datos de los Docentes deben estar registrados en el sistema.
Post-condición	<ul style="list-style-type: none"> • El Docente queda registrado en el sistema
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema y hace clic en el botón inicio. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Registrar Docente” del menú principal. 4. El sistema despliega el submenú “Registrar”. 5. El actor llena los campos del formulario desplegado. 6. El actor registra la información mediante la opción “Guardar”. 7. El sistema registra exitosamente al estudiante y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Limpiar la información de los campos e ingresar un docente nuevo <ol style="list-style-type: none"> 1. El actor selecciona la opción “Nuevo”. 2. El actor llena los campos con la información del nuevo docente a ingresar. 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.6 Especificación Casos de Uso Registrar docente

- **Buscar Docente**

Caso de uso: Buscar Docente	
Actor	Secretario
Descripción	Realiza la búsqueda de un docente dentro del Sistema.
Precondición	<ul style="list-style-type: none"> • Exista el Docente
Post-condición	<ul style="list-style-type: none"> • El Docente es encontrado, o no existe el Docente en el sistema.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema y hace clic en el botón inicio. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Buscar docente”. 4. El actor selecciona un criterio de búsqueda. 5. El actor ingresa los atributos adecuados de acuerdo al criterio de búsqueda. 6. El sistema realiza la búsqueda por atributos según criterio de búsqueda. 	

7. El sistema despliega la información del docente especificado. 8. El actor confirma que el docente encontrado es el buscado.
Flujo alternativo
1. El actor cancela la búsqueda 2. El actor sale del sistema. 3. El sistema no encuentra el docente buscado
Excepciones
Observaciones

Fuente: Propia

Tabla 3.7 Especificación Casos de Uso Buscar Docente

- Modificar Docente

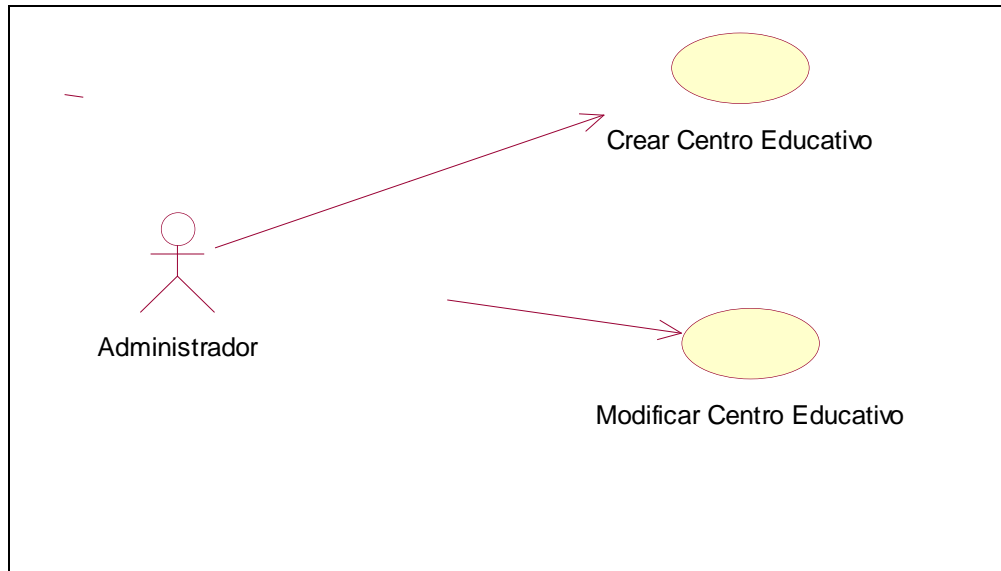
Caso de uso: Modificar Docente	
Actor	Secretario
Descripción	Realiza la modificación de los atributos de un docente dentro del Sistema.
Precondición	<ul style="list-style-type: none"> • Exista el Docente • Existan atributos a ser modificados
Post-condición	<ul style="list-style-type: none"> • El Docente modificado queda registrado en el sistema, o el Docente a modificar no existía en el sistema
Flujo normal de eventos	
1. El actor ejecuta el caso de uso buscar docente. 2. El sistema muestra la información del docente seleccionado. 3. El actor modifica los atributos del docente. 4. El sistema verifica los atributos modificados y muestra un mensaje de confirmación. 5. El sistema actualiza la información del docente.	
Flujo alternativo	
<ul style="list-style-type: none"> • Validación de datos 1. El sistema valida la información registrada. 2. El actor corrige la información mal ingresada.	
Excepciones	

Observaciones

Fuente: Propia

Tabla 3.8 Especificación Casos de Uso Modificar Docente

3.1.1.3. Especificación Caso de Uso: Administrar Centros Educativos



Fuente: Propia

Figura 3.3 Casos de Uso Administrar Centro Educativo

A continuación se muestra una breve explicación de los casos de uso creados en el diagrama anterior:

Detalle de casos de uso administrar centros educativos	
Caso de uso	Descripción
Crear Centro Educativo	Se encarga de registrar un nuevo centro educativo dentro del sistema
Modificar Centro Educativo	Realiza la modificación de los atributos de un centro educativo.

Fuente: Propia

Tabla 3.9 Detalle Casos de Uso: Administrar Centros Educativos

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- Crear Centro Educativo

Caso de uso: Crear Centro Educativo	
Actor	Administrador
Descripción	Registrar los nuevos centros educativos dentro del sistema
Precondición	<ul style="list-style-type: none"> • Existe un centro educativo a ser ingresado
Post-condición	<ul style="list-style-type: none"> • El centro educativo queda registrado en el sistema
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Crear Centro Educativo” del menú principal. 4. El actor llena los campos del formulario desplegado. 5. El actor registra la información mediante la opción “Guardar”. 6. El sistema registra exitosamente al centro educativo y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El actor selecciona la opción “Nuevo”. 2. El actor llena los campos con la información del nuevo centro educativo a ingresar. 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.10 Especificación Casos de Uso Crear Centro Educativo

- Modificar Centro Educativo

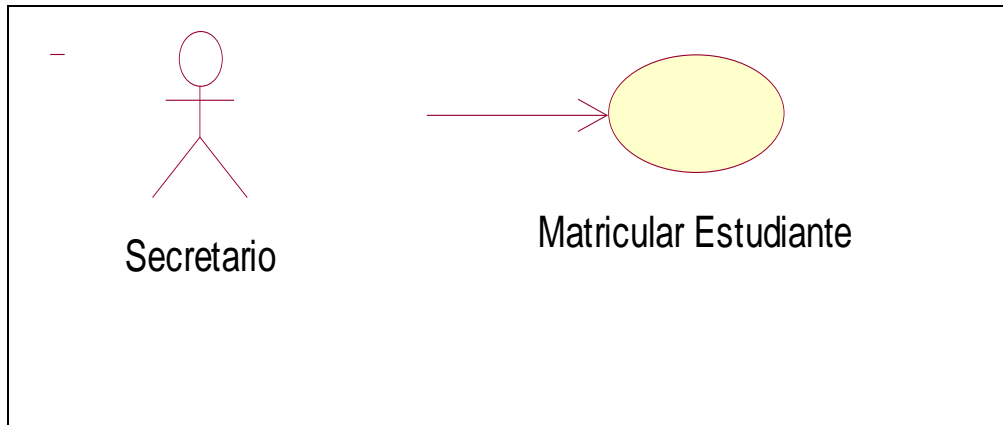
Caso de uso: Modificar Centro Educativo	
Actor	Administrador
Descripción	Permite administrar los datos del centro educativo
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema. • La información general del centro educativo se encuentra en la base de datos del sistema.
Post-condición	<ul style="list-style-type: none"> • En caso de éxito: La información del plantel es actualizada. • Caso contrario: La información del plantel no es actualizada.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Modificar Centro Educativo” del menú principal. 4. El actor ingresa la nueva información del plantel. 5. El sistema valida la información registrada. 6. El actor actualiza la información mediante la opción “Actualizar”. 7. El sistema actualiza exitosamente la información y despliega un mensaje de confirmación. 	
Flujo alternativo	

1. El sistema indica que la información no es valida 2. El sistema despliega un mensaje que la información del centro educativo no ha sido actualizada.
Excepciones
Observaciones

Fuente: Propia

Tabla 3.11 Especificación Casos de Uso Modificar Centro Educativo

3.1.1.4. Especificación de Caso de Uso: Matriculación



Fuente: Propia

Figura 3.4 Casos de Uso Matriculación

A continuación se describen los casos de usos creados en el diagrama anterior:

Detalle de casos de uso matriculación	
Caso de uso	Descripción
Matricular Estudiante	Se encarga de registrar la matrícula de los estudiantes registrados en el sistema para cada periodo académico.

Fuente: Propia

Tabla 3.12 Detalle Casos de Uso Matriculación

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

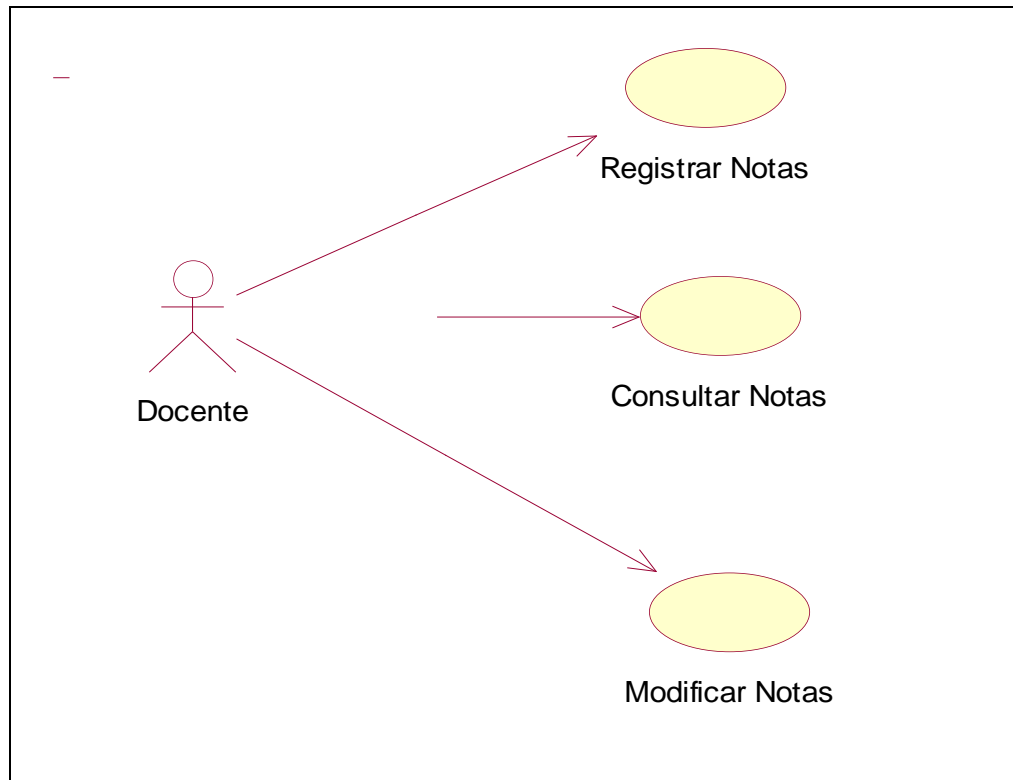
- Matriculación

Caso de uso: Matriculación	
Actor	Secretario
Descripción	Permite matricular alumnos en un nuevo año lectivo
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema. • El año lectivo se ha iniciado en el sistema. • Se encuentra activo el periodo de matrículas
Post-condición	<ul style="list-style-type: none"> • El alumno queda matriculado
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Matriculación” del menú principal. 4. El actor determina si el alumno es nuevo o antiguo en el plantel. 5. Cuando el alumno es antiguo se llama al caso de uso Buscar Alumno, el cual recibe los apellidos o los nombres del alumno como parámetros. Cuando el alumno es nuevo, ir al paso 11. 6. El sistema muestra el alumno encontrado 7. El actor registra los datos de la matrícula. 8. El sistema valida la información registrada 9. El actor registra la información mediante la opción “Guardar”. 10. El sistema registra exitosamente la información y despliega un mensaje de confirmación. 11. Si el alumno es nuevo el sistema llama al caso de uso registrar alumno. 12. El actor registra la información del nuevo alumno. 13. El actor registra los datos de la matrícula 14. El sistema registra exitosamente la información y despliega un mensaje de confirmación. 15. El actor imprime el formulario de matrícula 	
Flujo alternativo	
<ul style="list-style-type: none"> • Los datos del alumno no son encontrados <ol style="list-style-type: none"> 1. El sistema despliega un mensaje indicando que el alumno no ha sido encontrado. • Los datos del alumno no son ingresados <ol style="list-style-type: none"> 2. El sistema despliega un mensaje indicando que el alumno no ha sido matriculado. • Limpiar la información de los campos e ingresar nueva matrícula <ol style="list-style-type: none"> 3. El actor selecciona la opción “Nuevo”. 4. El actor llena los campos con la nueva información. 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.13 Especificación Casos de Uso Matriculación

3.1.1.5. Especificación Caso de Uso: Registro de Notas



Fuente: Propia

Figura 3.5 Casos de Uso Registro de Notas

A continuación se detalla los casos de uso descritos en la figura anterior:

Detalle de casos de uso registrar notas	
Caso de uso	Descripción
Registrar Notas	Se encarga de registrar las notas mensuales de los estudiantes registrados en el sistema para cada periodo académico.
Consultar Notas	Permite al docente realizar consultas de las notas registradas en el sistema.
Modificar Notas	Permite al docente modificar las notas registradas en el sistema previa autorización del administrador.

Fuente: Propia

Tabla 3.14 Detalle Casos de Uso Registrar Notas

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- Registrar Notas

Caso de uso: Registrar Notas	
Actor	Docente
Descripción	Permite registrar las calificaciones mensuales de los alumnos.
Precondición	<ul style="list-style-type: none"> • El Docente ha ingresado al sistema. • El ingreso de calificaciones este activo.
Post-condición	<ul style="list-style-type: none"> • Se han ingresado las calificaciones de los alumnos para la parcial correspondiente.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Registrar notas” del menú principal. 4. El actor selecciona el curso. 5. El actor ingresa las calificaciones de los alumnos de todas las materias que se encuentran a su cargo 6. El sistema valida la información registrada. 7. El actor registra la información mediante la opción “Aceptar”. 8. El sistema ingresa exitosamente la información y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El sistema indica que la información no es válida 2. El sistema despliega un mensaje indicando que la información ha sido registrada. 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.15 Especificación Casos de Uso Registrar Notas

- Consultar Notas

Caso de uso: Consultar Notas	
Actor	Administrador, Docente, Representante
Descripción	Permite realizar consultas de las calificaciones mensuales de los alumnos.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema.
Post-condición	<ul style="list-style-type: none"> • Se retornan los datos buscados por el actor.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Consulta de notas” del menú principal. 4. El actor ingresa los parámetros de búsqueda (nombres, apellidos o materia) 5. El sistema realiza la búsqueda por atributos según criterio de búsqueda. 6. El sistema despliega la información especificada por el actor. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Cancelar la consulta 	

1. El actor selecciona la opción “Cerrar” de la ventana desplegada. 2. El actor sale del sistema.
Excepciones
Observaciones

Fuente: Propia

Tabla 3.16 Especificación de Casos de Uso: Consultar Notas

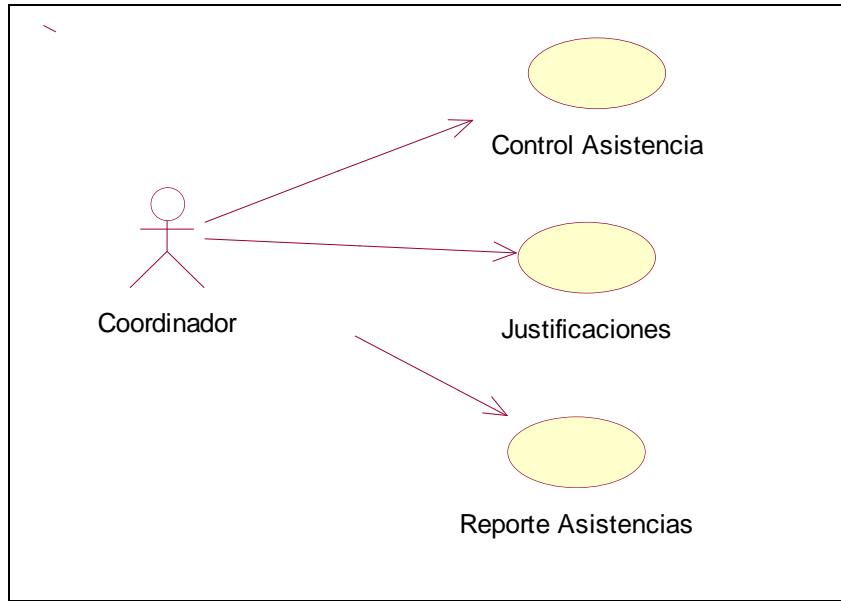
- Modificar Notas

Caso de uso: Modificar Notas	
Actor	Administrador, Docente
Descripción	Permite modificar las calificaciones mensuales de los alumnos registradas en el sistema.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema. • Modificación de notas de las materias en las cuales el actor es el responsable. • Se active la modificación de notas al actor
Post-condición	<ul style="list-style-type: none"> • Se modifican las calificaciones de un alumno.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Modificar de notas” del menú principal. 4. El actor ingresa los parámetros de búsqueda (nombres, apellidos o materia) 5. El sistema realiza la búsqueda por atributos según criterio de búsqueda. 6. El sistema despliega la información especificada por el actor. 7. El actor procede a modificar los valores de las calificaciones. 8. El sistema valida la información ingresada. 9. El actor registra la información mediante la opción “Aceptar”. 10. El sistema actualiza exitosamente la información y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Cancelar la modificación <ol style="list-style-type: none"> 1. El actor selecciona la opción “Cerrar” de la ventana desplegada. 2. El actor sale del sistema. • Error en el ingreso de datos <ol style="list-style-type: none"> 1. El sistema indica que la información no es válida 2. El sistema despliega un mensaje que la información no ha sido registrada 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.17 Especificación Casos de Uso: Modificar Notas

3.1.1.6. Especificación Caso de Uso: Control de Asistencias



Fuente: Propia

Figura 3.6 Casos de Uso Control de Asistencias

A continuación se muestra una breve explicación de los casos de uso plasmados en el diagrama anterior.

Detalle de casos de uso control de asistencias	
Caso de uso	Descripción
Control de Asistencias	Permite el registro y control de asistencias de los alumnos.
Justificaciones	Permite al coordinador justificar las inasistencias de los alumnos.
Reporte Asistencias	Permite al coordinador generar un reporte trimestral de los alumnos.

Fuente: Propia

Tabla 3.18 Detalle Casos de Uso Control de Asistencias

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- Control de Asistencias

Caso de uso: Control de Asistencias	
Actor	Administrador, Coordinador
Descripción	Permite registrar las asistencias de los alumnos en el sistema.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema.
Post-condición	<ul style="list-style-type: none"> • Se registra la asistencias de los alumnos en el sistema
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Registrar Asistencia” del menú principal. 4. El actor ingresa los datos en el sistema. 5. El sistema procede a validar la información ingresada. 6. El actor registra la información mediante la opción “Aceptar”. 7. El sistema ingresa exitosamente la información y despliega un mensaje de confirmación. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Cancelar la modificación <ol style="list-style-type: none"> 1. El actor selecciona la opción “Cerrar” de la ventana desplegada. 2. El actor sale del sistema. • Error en el ingreso de datos <ol style="list-style-type: none"> 1. El sistema indica que la información no es válida. 2. El sistema despliega un mensaje que la información no ha sido registrada 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.19 Especificación Casos de Uso: Control Asistencias

- Justificaciones

Caso de uso: Justificaciones	
Actor	Administrador, Coordinador
Descripción	Permite registrar la justificación de las inasistencias de los alumnos.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema. • Justificación de la inasistencia por parte del representante del alumno.
Post-condición	<ul style="list-style-type: none"> • Se registra la justificación de un alumno.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Justificaciones” del menú principal. 4. El actor ingresa los parámetros de búsqueda (nombres, apellidos, fecha). 5. El sistema realiza la búsqueda por atributos según criterio de búsqueda. 6. El sistema despliega la información especificada por el actor. 7. El actor procede a registrar la información. 	

<ol style="list-style-type: none"> 8. El sistema valida la información ingresada. 9. El actor registra la información mediante la opción “Justificar”. 10. El sistema actualiza exitosamente la información y despliega un mensaje de confirmación.
Flujo alternativo
<ol style="list-style-type: none"> 1. El sistema indica que la información no es válida. 2. El sistema despliega un mensaje indicando que la información no ha sido registrada. 3. El actor sale del sistema.
Excepciones
Observaciones

Fuente: Propia

Tabla 3.20 Especificación Casos de Uso: Justificaciones

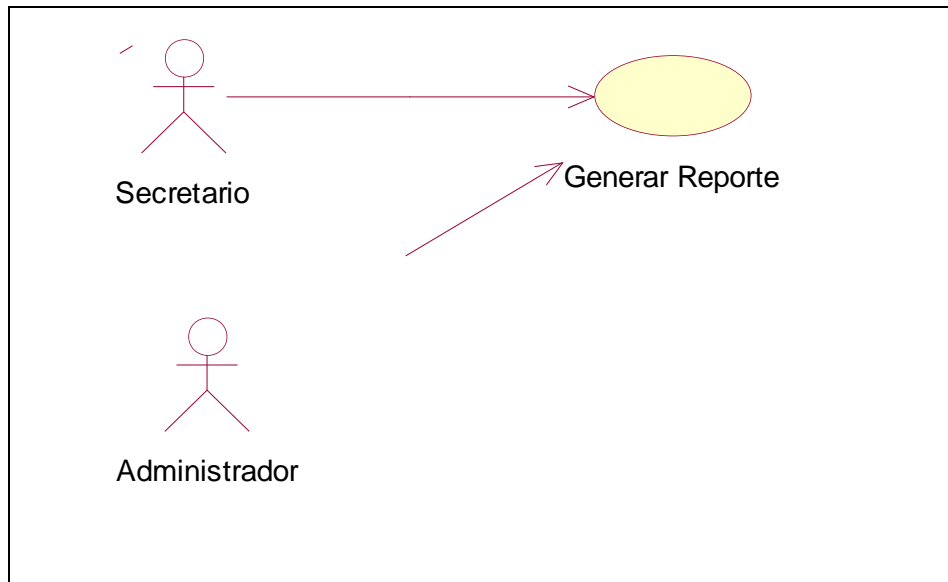
- **Reporte Asistencias**

Caso de uso: Reporte Asistencias	
Actor	Administrador, Coordinador
Descripción	Permite generar un reporte de las asistencias de los alumnos.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema. • El actor ha ingresado las inasistencias de los alumnos • El actor ha registrado las justificaciones de los alumnos
Post-condición	<ul style="list-style-type: none"> • Se genera el reporte.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Reporte Asistencias” del menú principal. 4. El usuario selecciona el reporte que desea generar. 5. Si el reporte lo requiere, el actor deberá ingresar parámetros para la generación del reporte. 6. El sistema comprueba si los parámetros ingresados son válidos. 7. El sistema genera el reporte solicitado y se lo visualiza. 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. Los parámetros recibidos no son válidos. 2. El sistema despliega un mensaje indicando que los parámetros ingresados no son válidos para el reporte 3. El actor sale del sistema 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.21 Especificación Casos de Uso: Reporte Asistencias

3.1.1.7. Especificación Caso de Uso: Generación de Certificados



Fuente: Propia

Figura 3.7 Casos de Uso: Generar Certificado

A continuación se muestra una breve explicación de los casos de uso plasmados en el diagrama anterior.

Detalle de casos de uso: Generar certificados	
Caso de uso	Descripción
Generar Certificados	Permite a un usuario generar y visualizar un reporte definido.

Fuente: Propia

Tabla 3.22 Detalle Casos de Uso Generar Certificados

Las principales especificaciones de los casos de uso descritos anteriormente se detallan a continuación:

- Generar Certificados

Caso de uso: Generar Certificados	
Actor	Administrador, Secretario
Descripción	Permite generar certificados definidos por el actor.
Precondición	<ul style="list-style-type: none"> • El actor ha ingresado al sistema.

Post-condición	<ul style="list-style-type: none"> • Se genera y visualiza el certificado
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor entra al sistema. 2. El sistema despliega el menú principal. 3. El actor selecciona la opción del menú “Generar Certificados” del menú principal. 4. El actor selecciona el certificado que desea generar. 5. Si el certificado lo requiere, el sistema pide al actor ingresar parámetros para generar el reporte. 6. El sistema comprueba si los parámetros ingresados son válidos 7. El sistema genera el reporte solicitado y se lo visualiza. 8. El actor imprime el certificado. 	
Flujo alternativo	
<ul style="list-style-type: none"> • Cancelar la generación del certificado <ol style="list-style-type: none"> 1. El actor selecciona la opción “Cerrar” de la ventana desplegada. 2. El actor sale del sistema. • Error en el ingreso de parámetros <ol style="list-style-type: none"> 1. El sistema indica que los parámetros ingresados no son válidos 	
Excepciones	
Observaciones	

Fuente: Propia

Tabla 3.23 Especificación Casos de Uso: Generar Certificados

CAPITULO IV

4. FASE DE CONSTRUCCIÓN

4.1. VISTA LÓGICA

4.1.1. Modelo Entidad-Relación

4.1.2. Modelo Físico

4.1.3. Diagrama Global de Paquetes



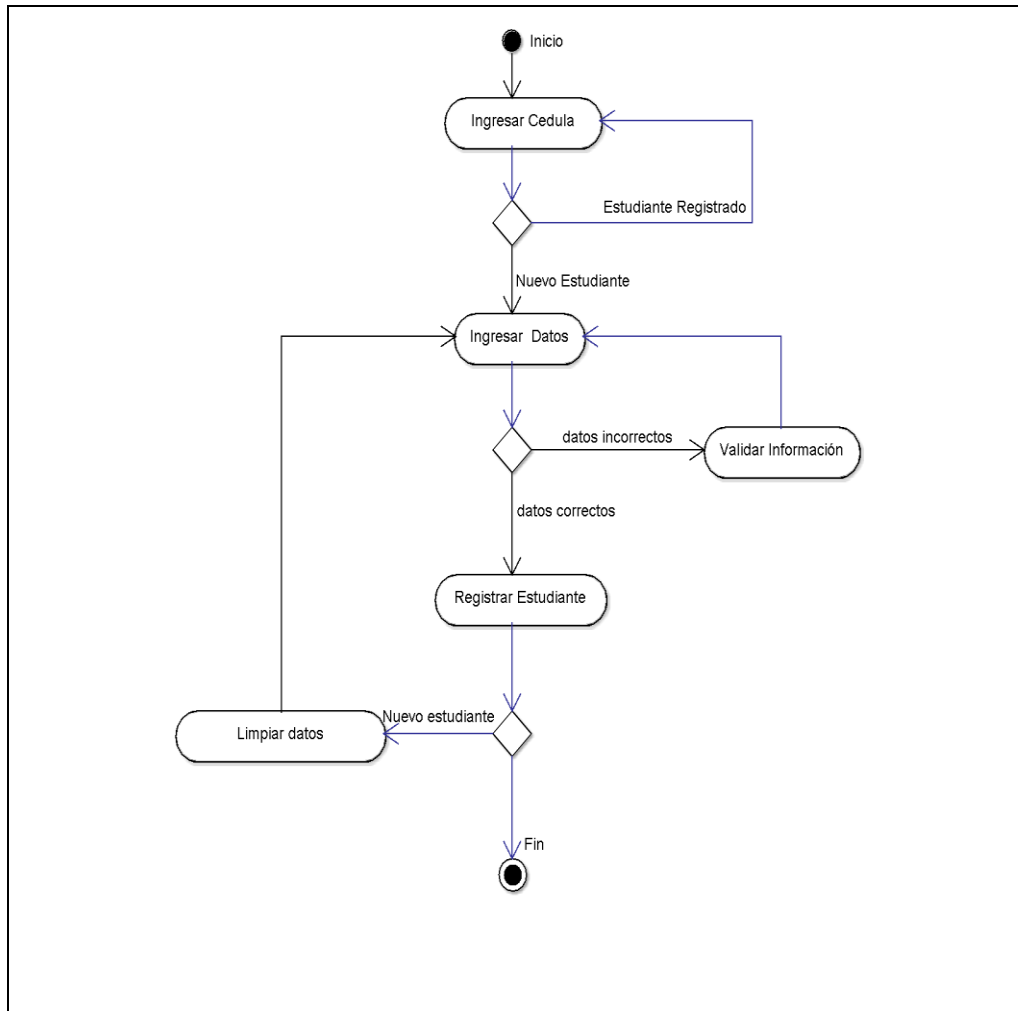
Fuente: Propia

Figura 4.1 Modelado del Negocio

4.2. VISTA DE IMPLEMENTACIÓN

4.2.1. Diagramas de Actividades

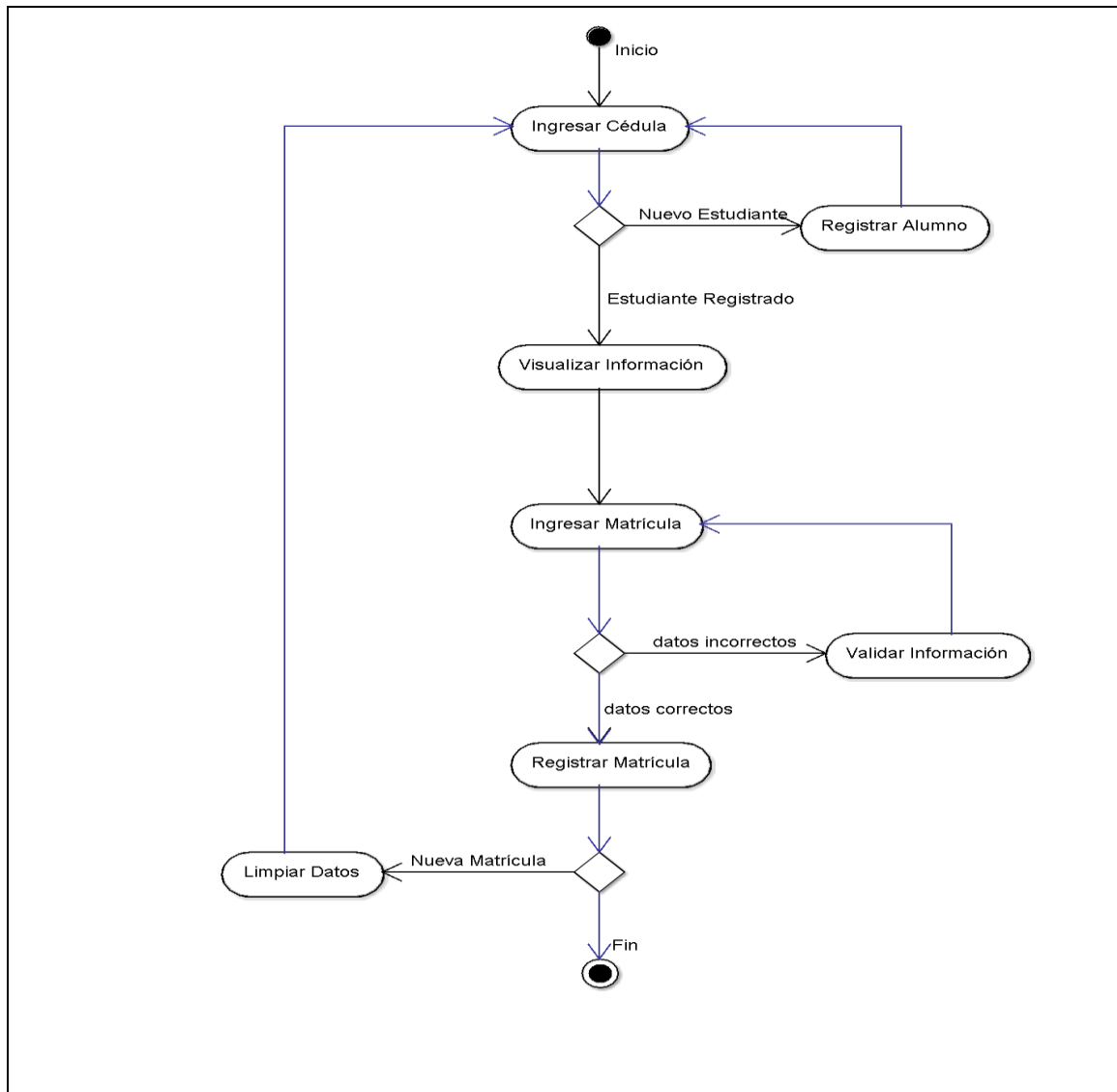
- REGISTRAR ALUMNO



Fuente: Propia

Figura 4.2 Registrar Alumno

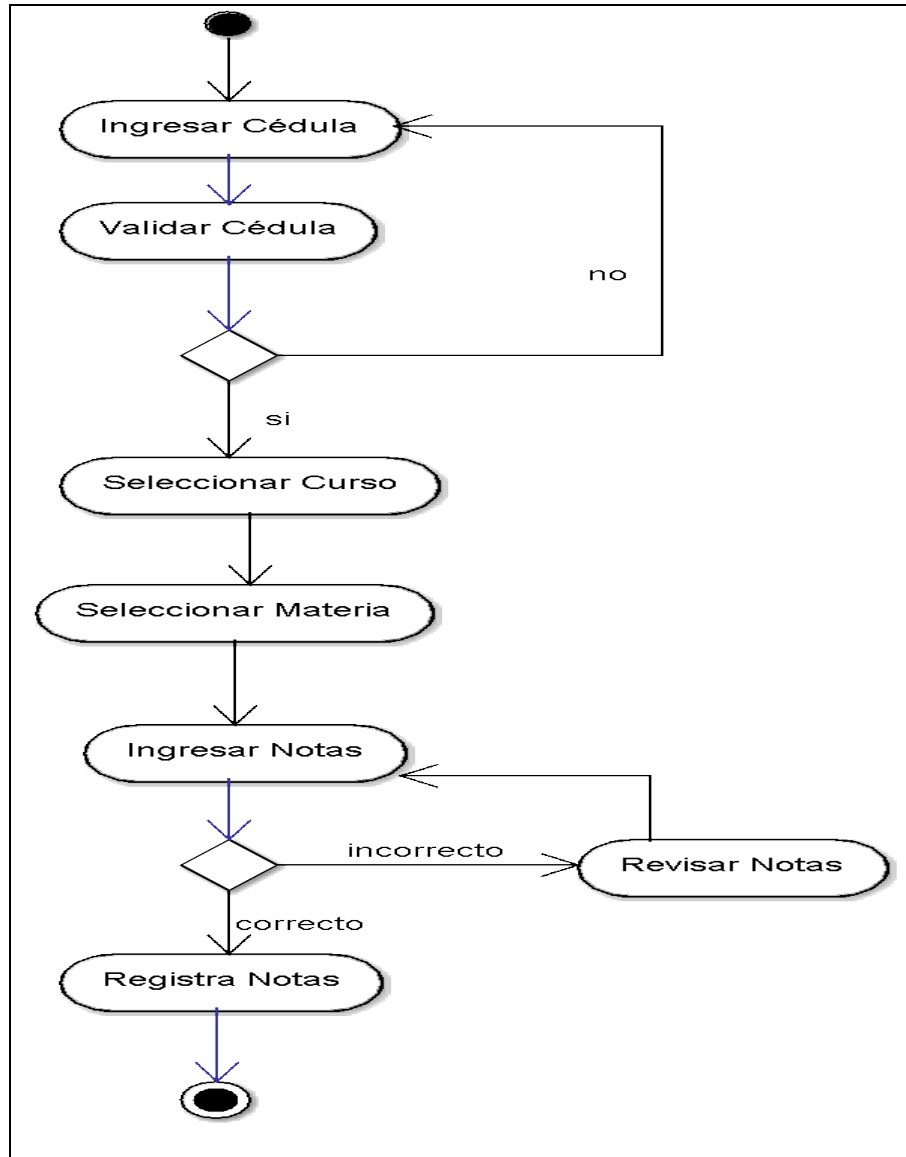
- MATRICULACIÓN



Fuente: Propia

Figura 4.3 Matriculación

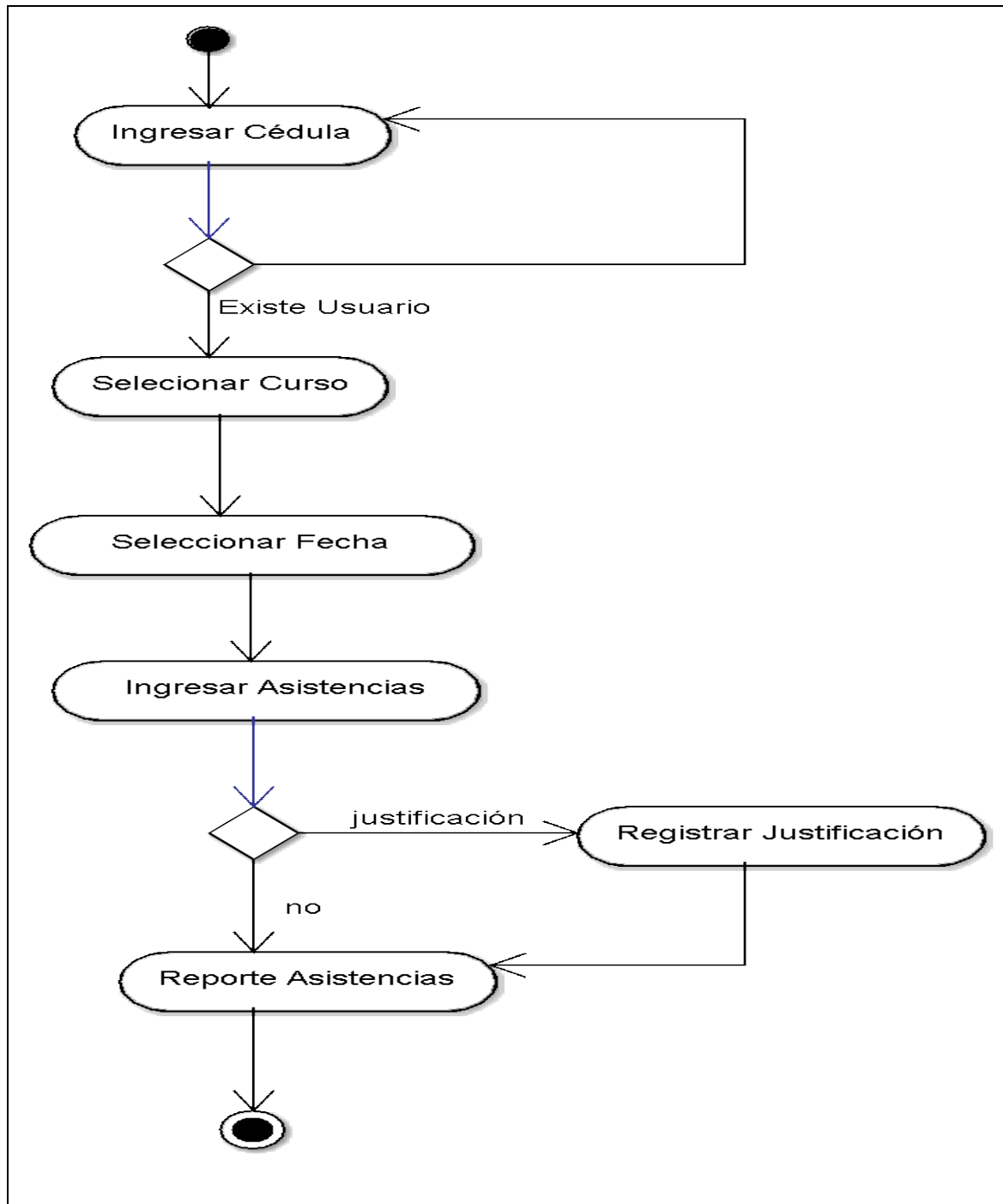
- REGISTRO DE NOTAS



Fuente: Propia

Figura 4.4 Ingresar Notas

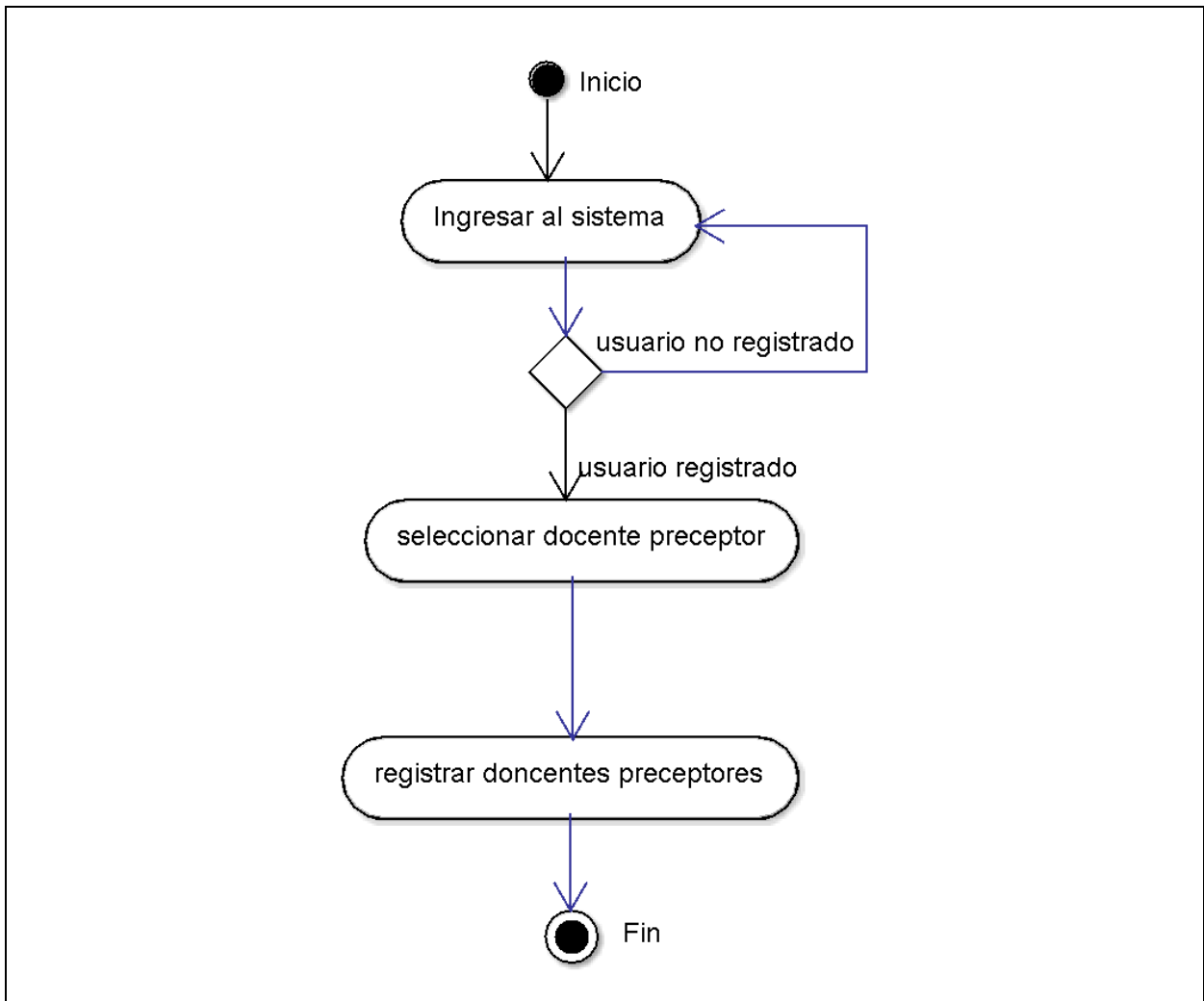
- REGISTRO ASISTENCIAS



Fuente: Propia

Figura 4.5 Registro de Asistencias

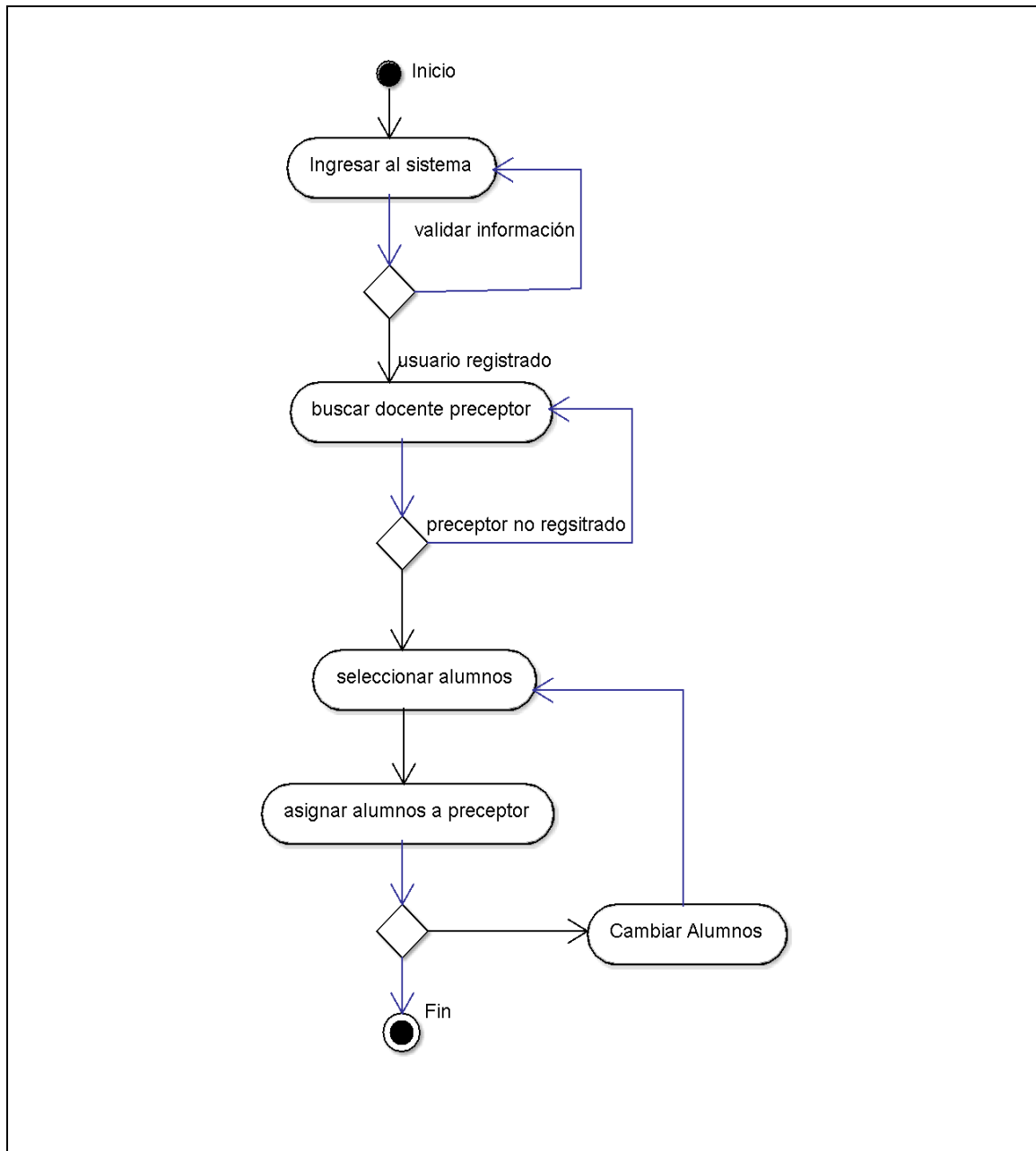
- REGISTRAR PRECEPTOR



Fuente: Propia

Figura 4.6 Registrar Preceptor

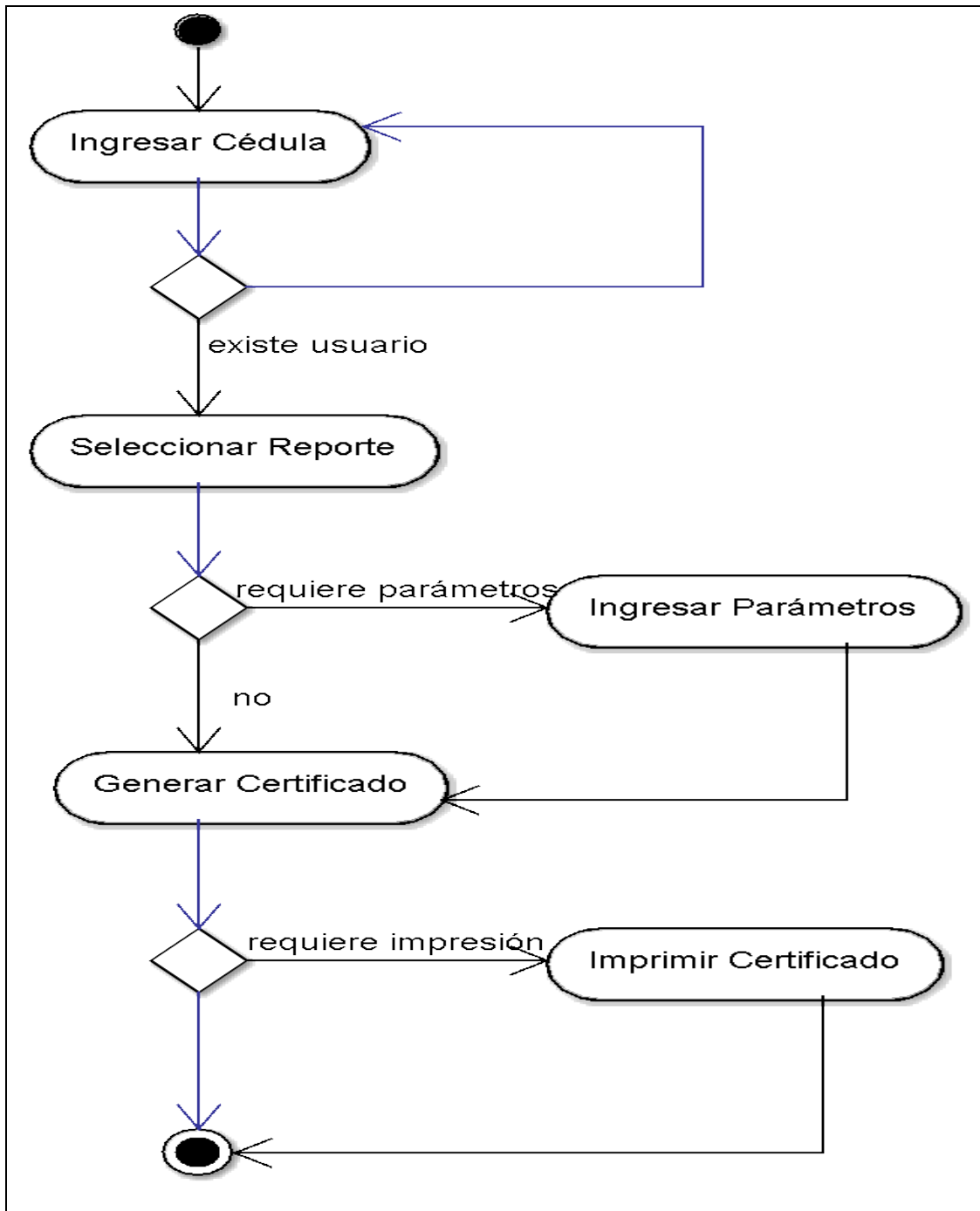
- REGISTRAR ALUMNOS A PRECEPTOR



Fuente: Propia

Figura 4.7 Registrar Alumno a Preceptor

- GENERACIÓN DE CERTIFICADOS

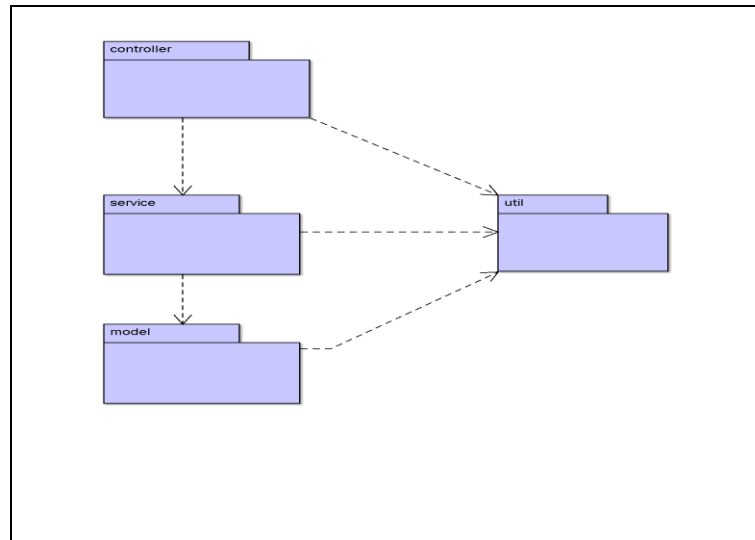


Fuente: Propia

Figura 4.8 Generación de Certificados

4.2.2. Diagramas de componentes

Se muestra la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación.

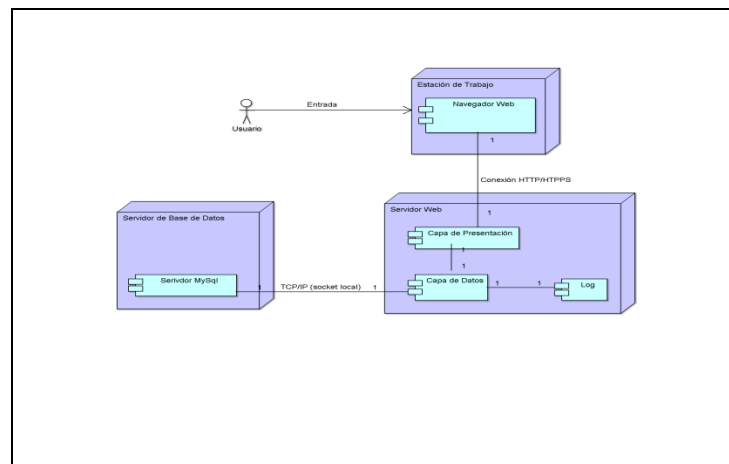


Fuente: Propia

Figura 4.9 Diagrama de Componentes

4.2.3. Diagramas de arquitectura

Se muestra la disposición de la arquitectura de la aplicación.



Fuente: Propia

Figura 4.10 Diagrama de Arquitectura

4.3. FASE DE PRUEBAS

4.3.1. Plan de pruebas

La necesidad de comprobar el correcto funcionamiento del producto hace que sea imprescindible un plan de pruebas, con el cual se procederá a realizar una serie de ensayos que permitan obtener resultados correctos y erróneos con el fin de analizar el proceso de ejecución. Con este conjunto de pruebas se podrá determinar si el programa tiene errores sobre todo en casos extremos y particulares, tanto si estos fallos se producen por la una mala implementación del programa o bien por un uso específico que realiza el usuario. El aspecto más importante para realizar la planificación de este conjunto de pruebas es abarcar con ellas todos los requisitos que debe cumplir el programa y que por tanto responda correctamente a las funcionalidades que fueron solicitados inicialmente.

Entre otros aspectos será necesario probar el desempeño computacional de la plataforma tecnológica usada; seguridad ante intentos de ataque y exactitud; corrección de su contenido y su despliegue en los diferentes programas visualizadores, entre otros aspectos.

Los casos de prueba que serán usados para probar la funcionalidad del sistema se detallan en la siguiente tabla:

Número de Prueba	Caso de prueba
1	Ingresar Alumnos
2	Matriculación
3	Registro de Notas
4	Control de Asistencias

Fuente: Propia

Tabla 4.1 Plan de Pruebas

4.3.2. Desarrollo de un plan de pruebas

4.3.2.1. Especificación de Caso de Prueba: Ingresar Alumnos

Referente al caso de Uso	Ingresar Alumnos
Nombre	Registrar un alumno en el sistema
Entradas	Datos del Alumno Nombres: Ricardo Sebastián Apellidos: Cadena Guerrero Dirección: La Victoria

	Teléfono: 2959 554 Ciudad de Nacimiento: Ibarra Fecha de Nacimiento: 06/09/2001 Sexo: Masculino Nacionalidad: Ecuatoriana Nombre Representante: Cadena Mario Cédula: 1000919884 Email: cadenita_mario@yahoo.es
Salidas	Pantalla de confirmación de alumno nuevo: Registro de Alumno
Descripción	Se indica al sistema que se va a registrar un alumno nuevo, se ingresan los datos personales del alumno, se guarda el registro.
Procedimiento de Prueba	<ul style="list-style-type: none"> · Ingrese al sistema con un usuario con perfil Secretario · Ingrese a la opción Alumno Nuevo · Ingrese los datos personales del alumno · Revise la información ingresada · Presione el botón para guardar
Resultados Esperados	El sistema guarda los datos del alumno en la base de datos. Se espera que el usuario se familiarice con el sistema y pueda ingresar y realizar el proceso completo de registro de alumnos.

Fuente: Propia

Tabla 4.1 Caso de Prueba: Ingresar Alumnos

4.3.2.2. Especificación de Caso de Prueba: Matriculación

Referente al caso de Uso	Matriculación
Nombre	Matricular Alumno que pertenece al plantel
Entradas	<p>Datos del Alumno</p> Nombres: Diego Sebastián Apellidos: Meneses Salazar Dirección: Av. Atahualpa 27-111 y los Incas Teléfono: 2650 641 Ciudad de Nacimiento: Ibarra Fecha de Nacimiento: 25/07/2004 Sexo: Masculino Nacionalidad: Ecuatoriana Nombre Representante: Meneses Diego Cédula: 1001756277 Email: meneses_diego@gmail.com
	<p>Datos de matrícula</p> Situación: Nuevo

	Número_Matricula: 10 Periodo: 2011-2012 Curso: 3er Año de Básica
Salidas	Pantalla de confirmación de la matrícula: Registro de Matrícula
Descripción	Se indica al sistema que se va a matricular un alumno que pertenece al plantel, se actualizan los datos del alumno, posteriormente se ingresan los datos de matrícula, se imprime el registro de matrícula.
Procedimiento de Prueba	<ul style="list-style-type: none"> · Ingrese al sistema con un usuario con perfil Secretario. · Ingrese a la opción Matricular alumnos. · Busque y escoja un alumno en la ventana de búsqueda. · Actualice los datos del alumno. · Ingrese los datos de matrícula. · Presione el botón para aceptar la matrícula.
Resultados Esperados	El sistema actualiza la información del alumno en la base de datos. El sistema guarda los datos de matrícula en la base de datos. Se espera que el usuario se familiarice y pueda ingresar los datos rápidamente para cumplir con el proceso de matriculación.

Fuente: Propia

Tabla 4.2 Caso de Prueba: Matriculación

4.3.2.3. Especificación de Caso de Prueba: Registro de Notas

Referente al caso de Uso	Registro de Notas
Nombre	Registrar calificaciones mensuales.
Entradas	<p>Datos previos</p> <p>La base de datos del sistema contiene previamente lo siguiente:</p> <ul style="list-style-type: none"> ○ Docente registrado en el sistema ○ Alumnos registrados en el sistema ○ Materias que el docente imparte <p>Datos a ingresar</p> <ul style="list-style-type: none"> ○ Calificaciones entre 0 y 4 puntos correspondiente a deberes ○ Calificaciones entre 0 y 8 puntos correspondiente a trabajo en clase ○ Calificaciones entre 0 y 4 puntos correspondiente a prueba mensual
Salidas	N/A
Descripción	Se registran calificaciones de las materias para todos los alumnos
Procedimiento de Prueba	<ul style="list-style-type: none"> · Ingrese al sistema con un usuario con perfil Docente. · Ingrese a la opción Registro de Notas. · Seleccione una materia para ingresar calificaciones. · Ingrese las calificaciones de la materia para el curso. · Repita el proceso para todas las materias en las cuales imparte

	<p>clases.</p> <ul style="list-style-type: none"> · Apruebe las calificaciones.
Resultados Esperados	<p>El sistema presenta las materias para cada curso.</p> <p>El sistema presenta los alumnos del curso.</p> <p>El sistema almacena las calificaciones en la base de datos.</p>

Fuente: Propia

Tabla 4.3 Caso de Prueba: Registro de Notas

4.3.2.4. Especificación de Caso de Prueba: Control de Asistencias

Referente al caso de Uso	Control Asistencias
Nombre	Registro de asistencias
Entradas	<p>Datos previos</p> <p>La base de datos del sistema contiene previamente lo siguiente:</p> <ul style="list-style-type: none"> ○ Alumnos registrados en el sistema <p>Datos a ingresar</p> <ul style="list-style-type: none"> ○ Registro de asistencias de los estudiantes. ○ Registro de justificaciones.
Salidas	N/A
Descripción	Se registran calificaciones de las materias para todos los alumnos
Procedimiento de Prueba	<ul style="list-style-type: none"> · Ingrese al sistema con un usuario con perfil Docente · Seleccione la opción Control Asistencias · Seleccione un curso · Ingrese las asistencias de los estudiantes · Apruebe las asistencias · Seleccione la opción justificaciones · Busque un alumno a justificar · Ingrese la justificación · Apruebe la justificación
Resultados Esperados	<p>El sistema presenta los alumnos del curso.</p> <p>El sistema almacena las asistencias en la base de datos.</p> <p>El sistema almacena las justificaciones en la base de datos.</p> <p>El usuario se familiarice con el proceso de control de asistencias y pueda ingresar la información correctamente.</p>

Fuente: Propia

Tabla 4.4 Caso de Uso: Control de Asistencias

4.3.3. Resultados del Plan de Pruebas

4.3.3.1. Resultados de Caso de Prueba: Ingresar Alumnos

Luego de realizar la prueba se recomienda al usuario realizar una validación de la información antes de proceder a guardar los datos en el sistema. Prueba realizada con éxito

4.3.3.2. Resultados de Caso de Prueba: Matriculación

Existió una buena interacción entre el sistema y el usuario de manera que la prueba fue realizada con éxito

4.3.3.3. Resultados de Caso de Prueba: Registro de Notas

Luego de realizar la prueba se recomienda al usuario los siguientes aspectos:

- Validar el registro de notas antes de proceder a ingresar los datos en el sistema.
- Revisar la información registrada antes de guardar la información.
- Realizar el registro de notas dentro del periodo establecido por el administrador del sistema
- Cuando se requiera modificar un registro se necesita la autorización del administrador del sistema.

Prueba realizada satisfactoriamente.

4.3.3.4. Resultados de Caso de Prueba: Control Asistencias

La prueba arrojó problemas leves superables en el registro de asistencias. Prueba superada con éxito.

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La mayoría de los planteles educativos, carecen de medios eficientes, seguros y transparentes para manejar su información académica, y para reportarla a los respectivos organismos de control. Esta necesidad es satisfecha con el desarrollo de sistemas de información, diseñados en base a los reglamentos establecidos por el Ministerio de Educación y la institución educativa.
- La tecnología Web 2.0 utilizada para el desarrollo del presente proyecto, ofrece importantes beneficios tanto al desarrollador, como al cliente o usuario final. El primero tiene entre otras ventajas la posibilidad de diseñar y programar modelos informáticos muy ricos, capaces de producir verdaderas aplicaciones empresariales accesibles desde Internet, sin un costo por licencias y soporte mínimo. El cliente por su parte, recibe un producto de calidad, y tiene la posibilidad de extenderlo, adaptarlo u optimizarlo sin las restricciones que impone el software desarrollado bajo licencias propietarias.
- Sistema de Gestión Académica SAG, por ser desarrollado en base a la metodología propuesta por el Proceso Unificado, sobre una plataforma no propietaria y tecnológicamente competitiva, satisface exitosamente los requerimientos y se proyecta como uno de los pilares para el desarrollo institucional futuro.
- El diseño del Sistema de Gestión Académica, permite que el sistema sea altamente parametrizable, por lo que se podrá adaptar a cualquier tipo de institución, obteniendo como producto un sistema personalizable que no sólo sirve para un caso

de estudio, ya que se puede adaptar a los procesos académicos que pueden diferir de una institución a otra.

5.2. RECOMENDACIONES

- Se sugiere a los usuarios revisar la documentación del Sistema de Gestión Académica SAG: manual de usuario y de instalación, acción que asegurará la correcta utilización del sistema y obtener así todos sus beneficios.
- La participación activa de los futuros usuarios de un proyecto es fundamental para alcanzar los objetivos planteados y satisfacer todos los requerimientos de la institución solicitante.
- Durante todo el proceso de desarrollo se debe mantener una buena comunicación con el cliente, de tal manera que se pueda tomar en cuenta su opinión y sugerencias, lo que permitirá obtener un producto acorde a las necesidades de dicho cliente.
- Es recomendable llevar un apropiado manejo de la documentación, lo que permitirá optimizar el mantenimiento del proyecto en caso de realizar cambios y/o actualizaciones.
- Con el objeto de fortalecer el programa de estudios de la Carrera de Ingeniería en Sistemas Computacionales de la UTN y facilitar la enseñanza/aprendizaje del desarrollo de aplicaciones empresariales, usando tecnologías modernas y dentro de cronogramas reales, se recomienda coordinar los programas académicos de las materias situadas dentro de un mismo nivel, de forma que los proyectos desarrollados por los estudiantes puedan abarcar lo aprendido en distintas asignaturas.

GLOSARIO

- **Introducción**

Un glosario contiene las definiciones de los términos propios de un proyecto o actividad. Las entradas aquí contenidas tienen por finalidad el documentar la forma y significado, que dentro del contexto del proyecto, se ha acordado dar a ciertas palabras.

Es bueno notar que el vocabulario de un proyecto, así como la lengua toda, está en constante evolución; por lo que es natural esperar que este documento sea constantemente actualizado. Se recomienda tener siempre a la mano una copia reciente.

- **Propósito**

La estandarización de términos empleados en la práctica de la Ingeniería del Software. Unificación de conceptos, significados y acepciones asociadas a los términos técnicos.

- **Alcance**

Términos notables de la Ingeniería del Software. No se trata de un diccionario informático, sino de una guía de términos, que por la frecuencia de su uso, o por su relevancia en la Ingeniería del Software, deberían ser conocidos por los profesionales relacionados con las actividades de la Ingeniería del Software.

- **Referencia**

El presente glosario hace referencia a los siguientes documentos:

- ~ Documento de Plan de Desarrollo de Software del Proyecto SAG
- ~ Documento de Visión del Proyecto SAG
- ~ Documentos de Especificaciones de Casos de Uso del Proyecto SAG
- ~ Documentos de Especificaciones de Casos de Pruebas del Proyecto SAG

- **Organización del Glosario**

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional del español.

- **Abreviaturas y Definiciones**

Abreviaturas

ODBC.- Open DataBase Connectivity (Estándar de Acceso a Bases de Datos)

AJAX.- Asynchronus Javascript and XML (JavaScript Asíncrono y XML)

RSS.- Really Simple Syndication (Sindicación Realmente Simple)

API.- Aplication Program Interface (Interfaz de Programación de Aplicaciones)

IDE.- Integrated Development Environment (Entorno de Desarrollo Integrado)

RPC.- Remote Procedure Call (Llamada a Procedimiento Remoto)

HTML.- HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

RIA: Rich Internet Applications (Aplicaciones Ricas de Internet)

EBML: Extensible Binary Meta Language (Meta Lenguaje Binario Extendible)

JSNI: JavaScript Native Interface

Definiciones

Estándares.- Que sirve como tipo, modelo, norma, patrón o referencia

Mantenible.- Propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez que se ha presentado un evento de falla.

WIKIS.- Es un sitio Web colaborativo que puede ser editado por varios usuarios

Ruby on Rails.- Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo la arquitectura Modelo Vista Controlador (MVC).

Multibrowser.- Es cuando una aplicación o una página web se puede ejecutar en todos los navegadores

DOM.- (Document Object Model - Modelo de Objetos de Documento) Especificación que determina cómo los objetos (texto, imágenes, enlaces, etc.) en una página web son representados.

JSON.- JavaScript Object Notation, es un formato ligero de intercambio de datos.

XMLHttpRequest.- Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web, la interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

Wiget.- Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets, entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

Bytecode.- Código intermedio más abstracto que el código máquina

J2EE.- Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones

Shareware.- Es una distribución de software en la que el usuario puede evaluar de forma gratuita el producto, pero con limitaciones en el tiempo de uso o en algunas de las formas de uso o con restricciones en las capacidades finales.

Freeware.- Define un tipo de software que se distribuye sin costo, disponible para su uso y por tiempo ilimitado

.Class.- Tipos de archivos Java que contienen las instrucciones Java ya compiladas

Applet.- Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

Acrónimo.- Es la suma de los significados de las palabras que lo forman.

Asíncrono.- No simultáneo o no coincidente

ECMAScript.- Es una especificación de lenguaje de programación basado en JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262.

Iframe.- Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal

Artilugio.- Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets.

JAR.- Es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java

JUNIT.- Es un conjunto de bibliotecas utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

Hooks.- Es una amplia gama de técnicas que se utilizan para alterar o aumentar el comportamiento de un sistema operativo, de aplicaciones, o de otros componentes de software mediante la interceptación de llamadas a funciones, mensajes o eventos pasados entre componentes de software.

REFERENCIAS

Referencias de Internet

[1]“AJAX”, 2011,

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

[2] “FUNCIONAMIENTO DE GWT”, 2010,

<http://code.google.com/intl/es-ES/webtoolkit/overview.html>

[3] “LA ARQUITECTURA DE GOOGLE WEB TOOLKIT”,2009,

<http://esgooglewebtoolkit.blogspot.com/2007/09/la-arquitectura-de-google-web-toolkit.html>

[4] “DESARROLLO DE APLICACIONES WEB 2.0 CON GWT”, 2012,

<http://www.slideshare.net/emiliobg/desarrollo-de-aplicaciones-web-20-con-gwt>

[5] “RUP ETAPA DISEÑO”, 2007,

<http://www.scribd.com/doc/395783/RUP-etapa-diseno>

ANEXOS

A. ANEXO A: MANUAL DE INSTALACIÓN

A.1. INTRODUCCIÓN

El presente anexo tiene como objetivo, mostrar una guía de instalación de las principales herramientas que se utilizaron para el desarrollo del proyecto.

A continuación mostraremos una lista de las herramientas que fueron utilizadas para el desarrollo del proyecto.

- Appserv versión 2.5.9, la misma que incluye:
 - Apache Web Server versión 2.2.4
 - PHP versión 5.2.3
 - MySql versión 5.0.43
- NETBEANS versión 7.0
- GWT versión 2.4
- Mozilla Firefox versión 3.6
- HeidiSQL 5.9

Cabe señalar que todas las herramientas anteriormente descritas fueron instaladas dentro de la plataforma Windows.

A.2. APPSERV versión 2.5.9

Es un proyecto de software libre bajo la licencia GNU, el mismo que instala por defecto un servidor web fácilmente configurable ya que contiene internamente el servidor Web Apache, PHP como lenguaje de script del lado del servidor, y MySql como motor de base de datos, la última versión de este software puede ser descargada de: <http://www.appservnetwork.com>

A.2.1. Prerrequisitos

Requisitos mínimos de hardware:

- Procesador PIV de 1,00 Ghz.
- Memoria 256 Mb.

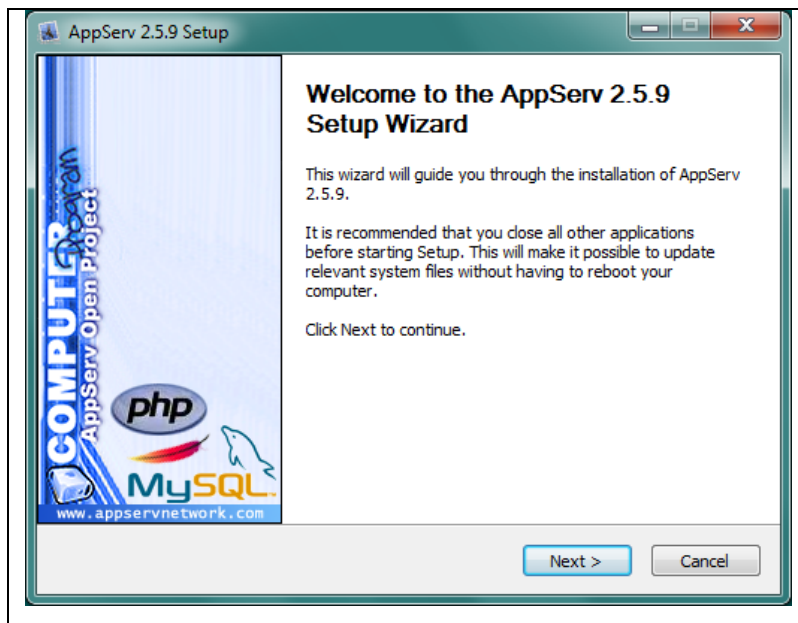
- ~ Espacio en disco 100Mb.

Requisitos mínimos de software:

- ~ Sistema operativo Windows XP en adelante.
- ~ Navegador WEB.

A.2.2. Instalación

Una vez descargado el software, abrimos el archivo ejecutable que puede estar nombrado como: appserv-win32-2.5.9.exe, una vez hecho esto nos aparecerá la pantalla principal del asistente de instalación como se muestra en la Figura A.1:



Fuente: Propia

Figura A.1 Asistente de instalación de Appserv

De aquí en adelante seguimos las instrucciones que nos muestra el asistente de instalación, ingresando los parámetros de configuración de acorde a las necesidades o al ambiente donde estemos instalando.

Para nuestro caso utilizamos la siguiente configuración:

PARÁMETRO	VALOR
Destination folder:	C:\Appserv
Appserv Package Componets	<ul style="list-style-type: none"> - Apache HTTP Server - MySql database - PHP Hypertext Preprocessor - phpMyAdmin
Server name	localhost
Administrator's email address	Riverlucho2006@hotmail.com
Apache HTTP Port	80
Root Password for MySql	root
MySql Server Settings	<ul style="list-style-type: none"> - Character Sets and Collations: UTF-8 Unicode - Old Password Support - Enable InnoDB

Fuente: Propia

Tabla A.1: Parámetros de configuración de Appserv

A.2.3. Después de instalar

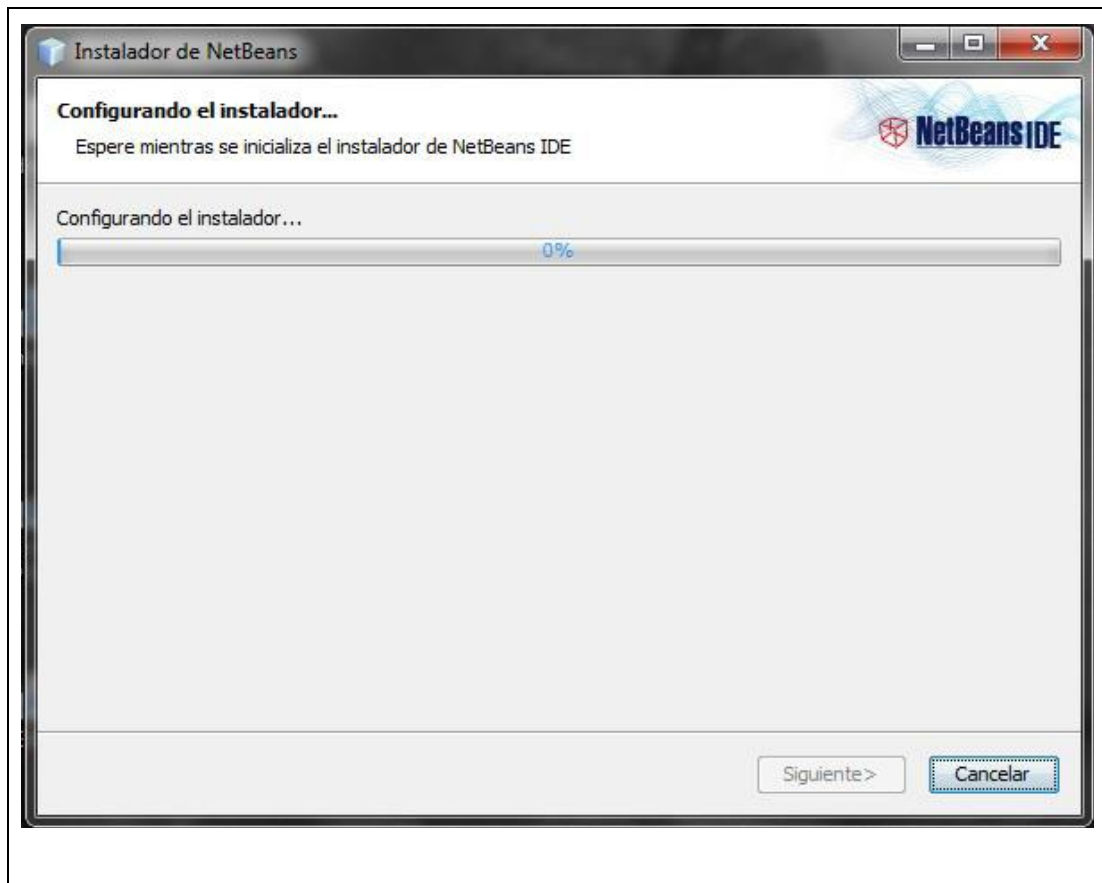
Una vez instalado el servidor los servicios, de Apache y MySql se inician automáticamente, sin embargo es pertinente revisar que estén correctamente instalados y funcionando, para esto es necesario ingresar al Panel de Control de Windows, Herramientas Administrativas, Servicios, y de la lista que aparece en la ventana de servicios se tiene que verificar que consten en la lista los servicios: “Apache 2.2” y “mysql” y que los mismos estén configurados para auto iniciarse o sea estén en modo automático.

Así mismo es necesario verificar si se ha creado correctamente el directorio de instalación donde va a funcionar el servidor web en este caso es: c:\Appserv.

A.3. NETBEANS versión 7.0

Para instalar el entorno de desarrollo de software, primero debemos obtenerlo descargándolo de la página: www.netbeans.org.

Una vez descargado el software, abrimos el archivo ejecutable que puede estar nombrado como: `netbeans-7.0-ml-windows.exe`, una vez hecho esto nos aparecerá la pantalla principal del asistente de instalación como se muestra en la Figura A.2:



Fuente: Propia

Figura A.2 Asistente de instalación de NetBeans

Una vez configurado todos los paquetes hacemos clic en el botón aceptar y luego en siguiente para comenzar con la instalación, nos aparece un cuadro con la licencia de Netbeans, aceptamos la licencia y luego en siguiente.

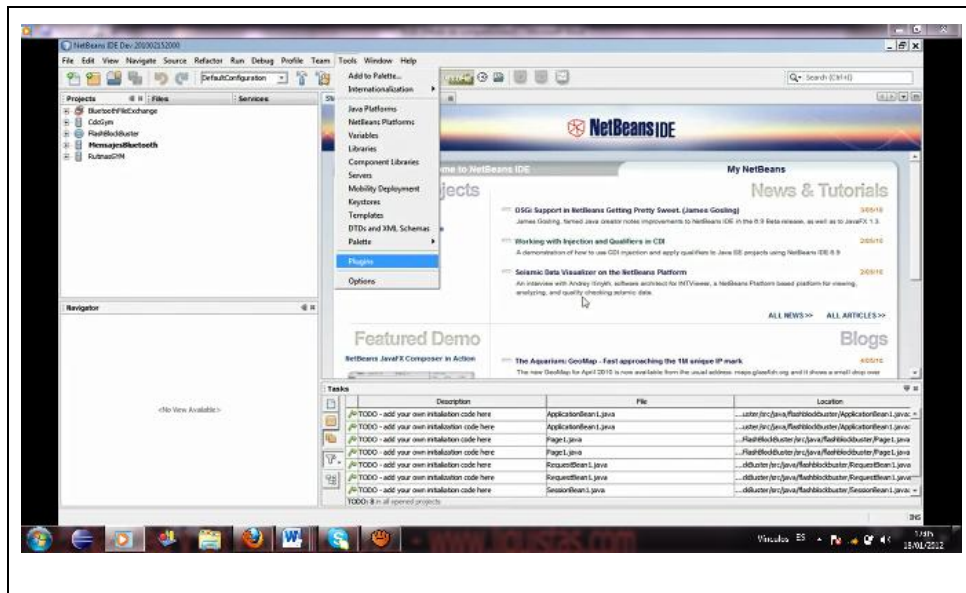
Finalmente escogemos la ruta donde queremos que se instale Netbeans y pulsamos instalar.

A.4. GWT versión 2.4

Para instalar el framework, primero debemos obtenerlo descargándolo de la página: <http://code.google.com/intl/es/webtoolkit/download.html>; escogemos la opción GWT SDK que es el complemento de GWT para Netbeans.

Una vez descargado obtendremos un archivo comprimido nombrado usualmente como: “GWT-2.4.0.zip”, este archivo lo debemos descomprimir dentro de un directorio.

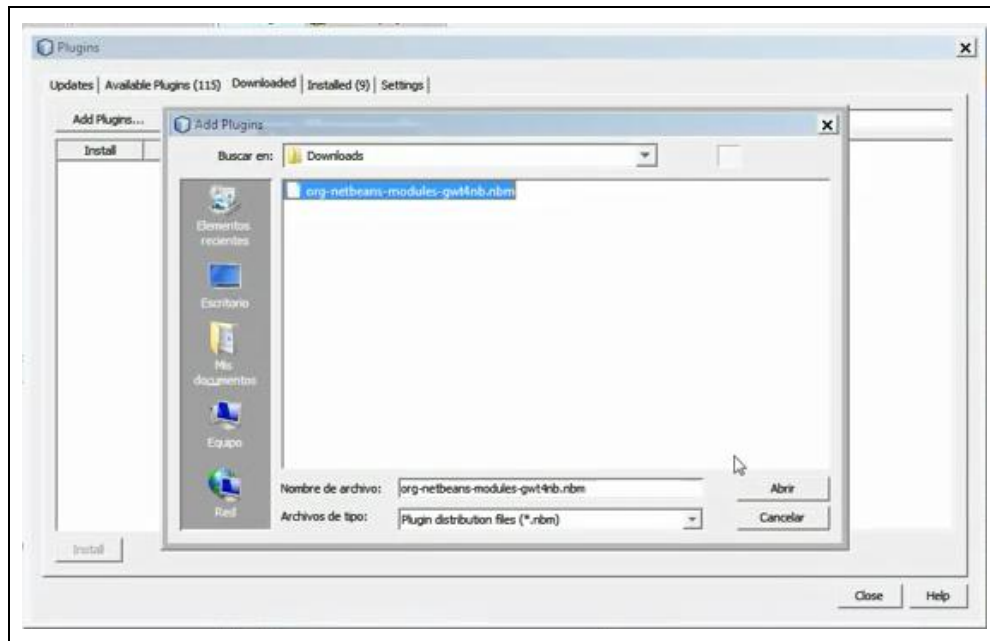
Luego abrimos NetBeans y escogemos la opción “plugins” del menú “tools” como se muestra en la figura:



Fuente: Propia

Figura A.3 Instalar Complementos de Netbeans

A continuación escogemos la opción “Add plugins” y seleccionamos la ruta en donde descomprimos el archivo de GWT llamado “org-netbeans-modules-gwt4nb.nbm” como se muestra en la figura:



Fuente: Propia

Figura A.4 Añadir Complementos en NetBeans

Una vez seleccionado el archivo pulsamos en el botón “abrir” y luego en “instalar” para comenzar con la instalación.

Finalmente nos aparece un cuadro con la licencia de GWT, aceptamos la licencia y luego en pulsamos en “siguiente”.

A.5. HeidiSQL 5.9

HeidiSql es un software que me permite administrar base de datos de MySQL, este software presenta una interfaz de fácil acceso a la información, así como en manejo y la elaboración de consultas, permite crear base de datos, tablas, procedimientos, disparadores, vistas todo esto enmarcado en una interfaz gráfica amigable y dinámica.

Este software es de código abierto y está publicado bajo la licencia GNU, para la instalación debemos primeramente descargar la última versión del archivo ejecutable de la página: <http://www.heidisql.com>, una vez hecho esto abrimos el archivo y seguimos los pasos que nos indica el asistente de instalación.

B. ANEXO B: MANUAL DE USUARIO

B.1. Introducción

El Sistema de Gestión y Seguimiento Académico Estudiantil ha sido construido para facilitar el manejo eficiente de los procesos académicos generados en el Centro Educativo “Álamos”.

El presente manual ha sido elaborado con la finalidad de ser una herramienta ágil que facilite la utilización de las opciones y funcionalidades del SAG.

B.2. Administrador funcional y usuarios del sistema

Los principales agentes del SAG son los funcionarios de las áreas de administración, docentes y representantes de los estudiantes del Centro Educativo “Álamos”, los mismos que se encargan de gestionar a través del sistema la información relacionada con las actividades del proceso académico.

B.3. Acceso a la aplicación

B.3.1. Ingreso al sistema

Mediante un navegador web se debe ingresar a la página www.alamosacademico.edu.ec, una vez hecho esto se cargara la página como se muestra en la Figura B.1.



Fuente: Propia

Figura B.1 Pantalla de Ingreso al Sistema

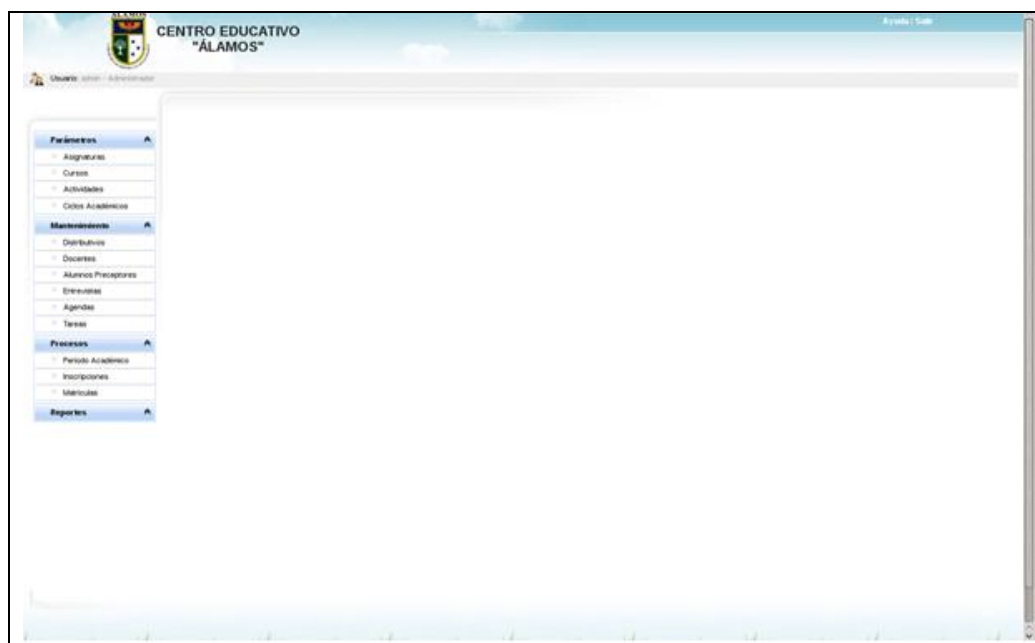
B.3.2. Autenticación de usuario

Una vez desplegada la pantalla de ingreso al sistema como nos muestra la Figura A.6, para autenticarnos como usuario debemos hacer lo siguiente:

- Llenar los campos de Usuario y Contraseña
- Dar un **enter** o hacer clic en el botón **Ingresar**

B.3.3. Menú principal

Una vez autenticados correctamente en el sistema dependiendo del rol de usuario que se haya autenticado, se nos desplegará la pantalla principal como nos muestra la Figura



Fuente: Propia

Ilustración B.2 Pantalla Principal del Sistema

Como podemos ver en la figura anterior al momento de autenticarnos nos muestra la pantalla principal y podemos desplegar el menú principal de la aplicación el mismo que consta de los siguientes módulos:

- Parámetros

Contiene los formularios de configuración del sistema.

- Mantenimiento

Contiene los formularios para realizar el mantenimiento del sistema, las tareas que realiza este módulo son: listado, edición y creación de los registros que contiene la base de datos.

- Procesos

Contiene los siguientes formularios: Periodo Académico, Inscripción y Matriculación que se usan al iniciar un periodo académico.

- Reportes

Este módulo permite realizar consultas de la información académica de la Institución y genera reportes de las consultas si el usuario lo requiere.

B.3.4. Parámetros

- Asignaturas

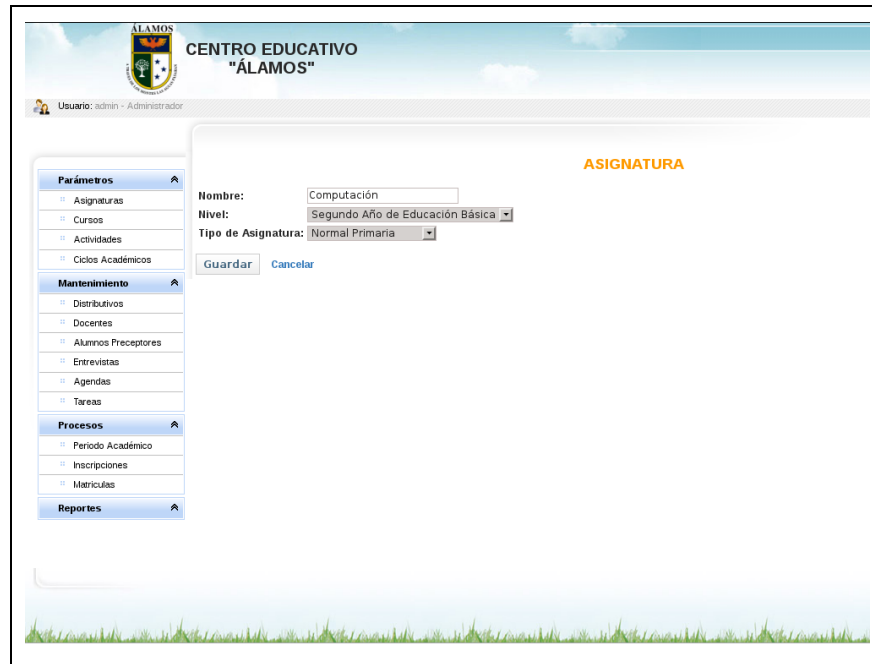
La pantalla Asignaturas muestra la información de las asignaturas que tiene la Institución y permite la creación, modificación y consulta.

- Para crear una asignatura hacemos clic en el vínculo “Crear Asignatura”, llenamos todos los campos y pulsamos el botón “guardar”, el sistema nos mostrará la siguiente pantalla:

Fuente: Propia

IlustraciónB.3 Pantalla: Crear Asignatura

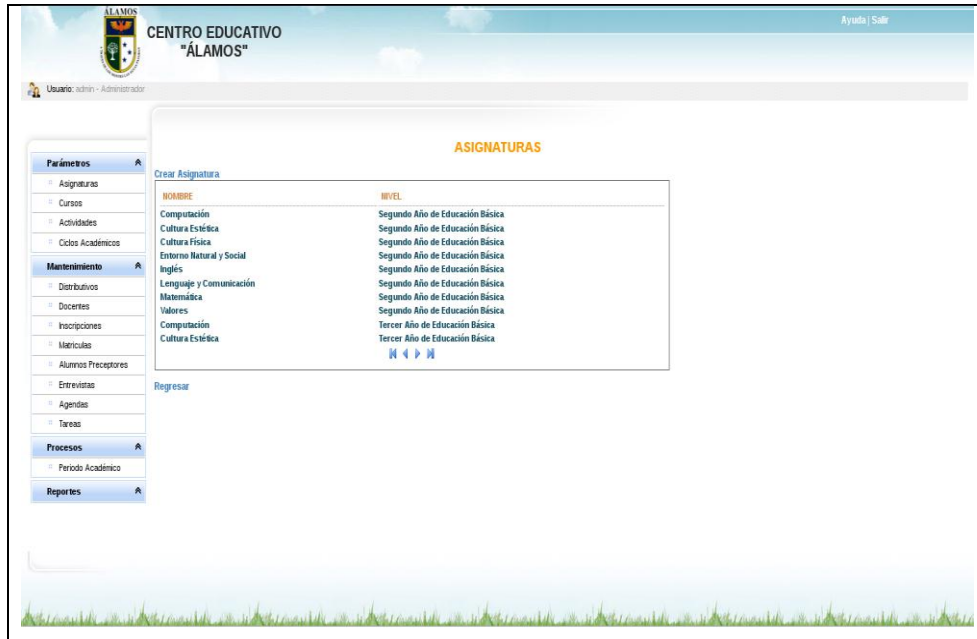
- Para editar la información de una asignatura hacemos clic sobre el nombre de la misma, el sistema nos mostrará la información de la asignatura seleccionada, luego escogemos el campo que vamos a editar, realizamos los cambios y pulsamos el botón “guardar”, el sistema nos desplegará la siguiente pantalla:



Fuente: Propia

Ilustración B.4 Pantalla: Editar Asignatura

- Para consultar las asignaturas que tiene la Institución hacemos clic en “Asignaturas” del menú principal, el sistema nos desplegará la siguiente pantalla:



Fuente: Propia

Ilustración B.5 Pantalla: Consultar Asignaturas

- Cursos

Esta opción muestra la información de los cursos que tiene la Institución, permite la creación, modificación y consulta.

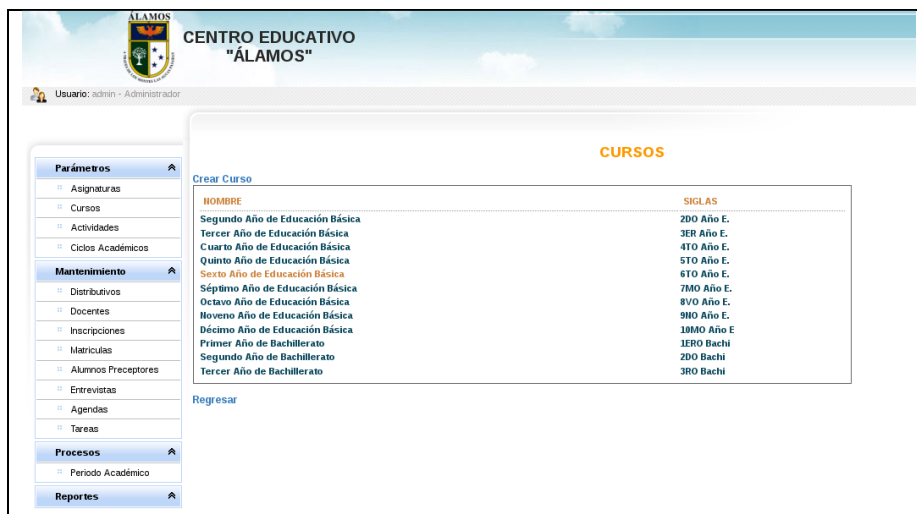
- Para crear un curso hacemos clic en el vínculo “Crear Curso”, llenamos los campos nombre, siglas y pulsamos el botón “guardar”, el sistema nos mostrará la siguiente pantalla:



Fuente: Propia

Ilustración B.6 Pantalla: Crear Curso

- Para editar la información de un curso hacemos clic sobre el nombre del mismo, el sistema nos mostrará la información del curso seleccionado, luego escogemos el campo que vamos a editar, realizamos los cambios y pulsamos el botón “guardar”.
- Para consultar los cursos que tiene la Institución hacemos clic en “Curso” del menú principal, el sistema nos desplegará la siguiente pantalla:



Fuente: Propia

Ilustración B.7 Pantalla: Consultar Cursos

- Actividades

Esta opción nos permite crear las actividades académicas de la Institución

- Para crear una actividad hacemos clic en el vínculo “Crear Actividad”, llenamos los campos nombre, siglas y pulsamos el botón “guardar”, el sistema nos mostrará la siguiente pantalla:



Fuente: Propia

Ilustración B.8 Pantalla: Crear Actividad

- Para editar la información de una actividad, hacemos clic sobre el nombre de la misma, el sistema nos muestra la información de la actividad seleccionada, escogemos el campo que vamos a editar y pulsamos sobre el botón “Guardar”

B.3.5. Mantenimiento

- Distributivos

Esta opción nos permite asignar a cada asignatura un docente para cada periodo académico.

- Para crear un distributivo hacemos clic en el vínculo “Crear Distributivo”, pulsamos en el ícono de búsqueda para seleccionar el docente y la asignatura y presionamos el botón “guardar”, el sistema nos mostrará la siguiente pantalla:



Fuente: Propia

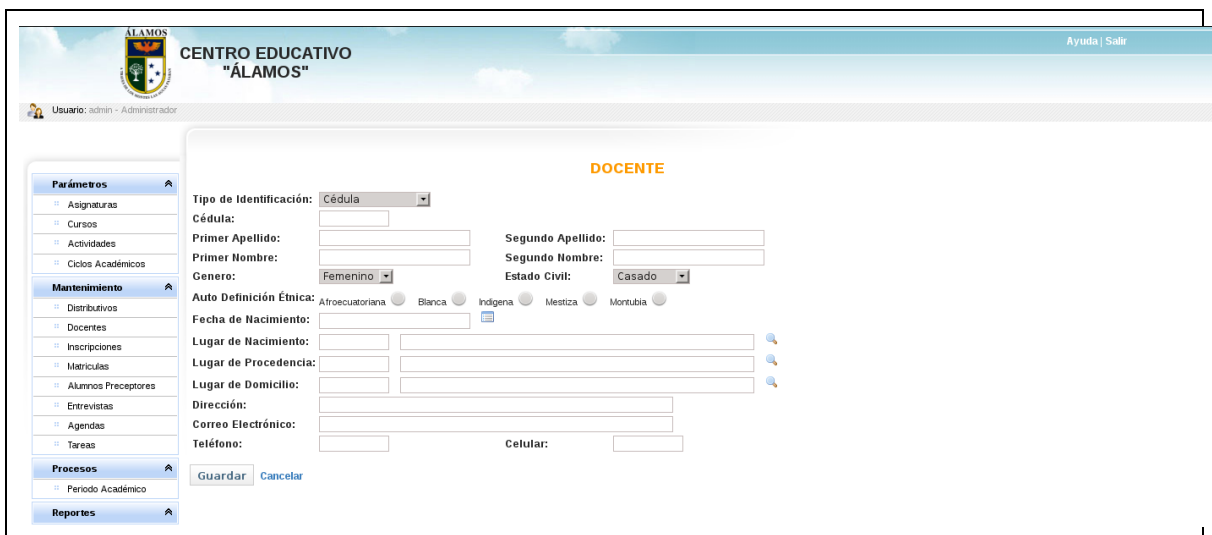
Ilustración B.9 Pantalla: Crear Distributivo

- Para editar la información de un distributivo hacemos clic sobre el nombre del docente, realizamos los cambios que necesitemos hacer y hacemos clic en “guardar”

~ Docentes

Esta opción muestra la información personal de los docentes, permite crear, consultar y editar.

- Para crear un docente hacemos clic en el vínculo “Crear Docente”, llenamos todos los campos y pulsamos el botón “guardar”, el sistema nos mostrará la siguiente pantalla:



Fuente: Propia

Ilustración B.1 Pantalla: Crear Docentes

- Para editar la información de docente hacemos clic sobre el nombre, el sistema nos desplegará la información personal del docente, a continuación escogemos el campo que vamos a editar, realizamos los cambios y pulsamos hacemos clic en “guardar”.
- Para consultar los docentes que tiene la Institución hacemos clic en “Docentes” del menú principal

- Alumnos Preceptores

Esta opción permite asignar a los docentes los alumnos a los cuales se les dará el seguimiento académico durante todo el año lectivo. El sistema nos permite asignar, modificar y consultar la información y el procedimiento para realizar todas estas acciones es similar al descrito en el punto Docentes.

- Entrevistas

Esta opción permite administrar las entrevistas que tiene cada docente con el representante de cada alumno que tiene asignado como preceptor. Al igual que las opciones anteriores el sistema nos permite crear, modificar y consultar el calendario de entrevistas y el proceso es similar al descrito en las opciones anteriores.

- Agendas

En esta opción el sistema nos permite enviar información referente al desenvolvimiento académico, comportamiento actitudinal e incidentes que sucedan dentro de las horas de clase de los estudiantes a los representantes. El procedimiento para utilizar esta opción es similar al descrito en los puntos anteriores.

B.3.6. Procesos

- Periodo Académico

En esta opción el sistema nos permite iniciar el Periodo Académico de la Institución, solo el administrador del sistema manipular esta opción.

Para iniciar un Periodo Académico hacemos clic en “Periodo Académico” del menú principal, luego escogemos “Crear Periodo Académico” y procedemos a llenar el campo descripción que es donde vamos a poner el periodo que vamos a iniciar, ejemplo 2011-2012, a continuación escogemos la fecha de inicio y finalización del periodo utilizando el

ícono “calendario”, por ultimo damos clic en “guardar”. El sistema nos desplegará la siguiente pantalla:

The screenshot displays the 'PERIODO ACADÉMICO' creation interface. At the top, it shows the logo of 'CENTRO EDUCATIVO "ÁLAMOS"' and the user 'Usuario: admin - Administrador'. The left sidebar contains a menu with the following items: 'Parámetros' (with sub-items: Asignaturas, Cursos, Actividades, Ciclos Académicos), 'Mantenimiento' (with sub-items: Distributivos, Docentes, Inscripciones, Matriculas, Alumnos Preceptores, Entrevistas, Agendas, Tareas), 'Procesos' (with sub-item: Periodo Académico), and 'Reportes'. The main content area is titled 'PERIODO ACADÉMICO' and contains a form with the following fields: 'Descripción:' (text input), 'Fecha Inicio:' (calendar icon), and 'Fecha Fin:' (calendar icon). Below the form are 'Guardar' and 'Cancelar' buttons.

Fuente: Propia

IlustraciónB.11 Pantalla: Crear Periodo Académico

~ Inscripciones

Esta opción permite al usuario registrar en el sistema a los alumnos nuevos que ingresan a la Institución

- Para inscribir un alumno hacemos clic en la opción “Inscripciones” llenamos todos los campos con la información personal del estudiante nuevo y pulsamos sobre el botón “inscribir” para terminar el proceso. El sistema nos mostrará la siguiente pantalla:

The screenshot shows a web application interface for 'CENTRO EDUCATIVO "ÁLAMOS"'. The header includes the school logo, the name 'CENTRO EDUCATIVO "ÁLAMOS"', and a user profile 'Usuario: admin - Administrador'. A sidebar menu on the left is organized into categories: 'Parámetros' (Asignaturas, Cursos, Actividades, Ciclos Académicos), 'Mantenimiento' (Distributivos, Docentes, Inscripciones, Matriculas, Alumnos Preceptores, Entrevistas, Agendas, Tareas), 'Procesos' (Periodo Académico), and 'Reportes'. The main content area is titled 'Inscribir Alumnos' and contains a form with the following fields: 'Tipo de Identificación' (Cédula), 'Cédula', 'Primer Apellido', 'Segundo Apellido', 'Primer Nombre', 'Segundo Nombre', 'Genero' (Femenino), 'Estado Civil' (Casado), 'Auto Definición Étnica' (radio buttons for Atroecuatoriana, Blanca, Indígena, Mestiza, Montubia), 'Fecha de Nacimiento', 'Lugar de Nacimiento', 'Lugar de Procedencia', 'Lugar de Domicilio', 'Dirección', 'Correo Electrónico', 'Teléfono', and 'Celular'. An 'Inscribir' button is located at the bottom of the form, and a 'Regresar' link is positioned below it.

Fuente: Propia

IlustraciónB.12 Pantalla: Inscribir Alumnos

~ *Matriculas*

Esta opción realiza el proceso de matriculación de los estudiantes. Antes de matricularse un estudiante debe estar inscrito.

- Para matricular un estudiante hacemos clic en la opción “Matriculas” del menú principal, luego hacemos clic en el vínculo Matricular, a continuación procedemos a verificar la información personal del estudiante, llenamos los datos académico y del representante, finalmente pulsamos el botón “guardar”. El sistema nos mostrará la siguiente pantalla:

ALAMOS CENTRO EDUCATIVO "ÁLAMOS" Ayuda | Salir

Usuario: admin - Administrador

Matricula

Datos Personales Datos Académicos Representante

Tipo de Identificación: Cédula

Cédula:

Primer Apellido: Segundo Apellido:

Primer Nombre: Segundo Nombre:

Genero: Femenino Estado Civil: Casado

Auto Definición Étnica: Atroecuatoriana Blanca Indígena Mestiza Mombuvia

Fecha de Nacimiento:

Lugar de Nacimiento:

Lugar de Procedencia:

Lugar de Domicilio:

Dirección:

Correo Electrónico:

Teléfono: Celular:

Guardar Cancelar

Fuente: Propia

IlustraciónB.13 Pantalla: Matriculación

C. ANEXO C: DOCUMENTOS ESTANDARIZADOS

C.1. FORMATO DE REPORTE MENSUAL DE CALIFICACIONES

Centro Educativo "Álamos"			
Ibarra - Ecuador			
Informe Mensual de Desempeño			
Estudiante:		Dirigente:	
Año de Básica:		Mes de informe:	
LITERATURA		MATEMATICA	
Deberes/4	CALIFIC.	Deberes/4	CALIFIC.
Trabajo en Clase/7	0,00	Trabajo en Clase/7	0,00
Lecciones/5	0,00	Lecciones/5	0,00
Prueba/4	0,00	Prueba/4	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00
HISTORIA		QUIMICA	
Deberes/4	CALIFIC.	Deberes/4	CALIFIC.
Trabajo en Clase/7	0,00	Trabajo en Clase/7	0,00
Lecciones/5	0,00	Lecciones/5	0,00
Prueba/4	0,00	Prueba/4	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00
ENGLISH		COMPUTACION	
Deberes/4	CALIFIC.	Deberes/6	CALIFIC.
Trabajo en Clase/7	0,00	Trabajo en Clase/7	0,00
Lecciones/5	0,00	Prueba Mensual/7	0,00
Prueba/4	0,00	Promedio Mensual	0,00
Promedio Mensual	0,00		
VALORES		DIBUJO TÉCNICO	
Trabajo en Clase/10	CALIFIC.	Deberes/6	CALIFIC.
	0,00	Trabajo en Clase/7	0,00
Prueba Mensual /10	0,00	Prueba Mensual/7	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00
ANATOMÍA		CULTURA FISICA	
Deberes/6	CALIFIC.	Deberes/6	CALIFIC.
Trabajo en Clase/6	0,00	Trabajo en Clase/7	0,00
Prueba Mensual/8	0,00	Prueba Mensual/7	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00
ECONOMÍA		TEORÍA DEL CONOCIMIENTO	
Deberes/6	CALIFIC.	Deberes/6	CALIFIC.
Trabajo en Clase/6	0,00	Trabajo en Clase/7	0,00
Prueba Mensual/8	0,00	Prueba Mensual/7	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00
FÍSICA		BIOLOGÍA	
Deberes/4	CALIFIC.	Deberes/4	CALIFIC.
Trabajo en Clase/7	0,00	Trabajo en Clase/7	0,00
Lecciones/5	0,00	Lecciones/5	0,00
Prueba/4	0,00	Prueba/4	0,00
Promedio Mensual	0,00	Promedio Mensual	0,00

Fuente: Propia

Figura C.1 Reporte Mensual de Calificaciones

C.2. FORMATO DE REPORTE ANUAL DE CALIFICACIONES

CENTRO EDUCATIVO "ÁLAMOS"
 IBARRA – ECUADOR
 Calle Piñan (PRIORATO)
 E-mail: ce_alamos@andinanet.net

REPORTE ANUAL DE CALIFICACIONES "PASE DE AÑO"

Alumno:
 Año de Básica:
 Año Lectivo:
 Viernes, 08 de Julio de 2011

ASIGNATURAS	I		II		III		Promedio	
	Punt.	Eqv.	Punt.	Eqv.	Punt.	Eqv.	Punt.	Eqv.
Lenguaje y Comunicación								
Matemática								
Estudios Sociales								
Ciencias Naturales								
Cultura Estética								
Cultura Física								
Inglés								
Computación								
Valores								
Promedio								

RESUMEN DE CONDUCTA, APROVECHAMIENTO Y ASISTENCIA

	I		II		III		Promedio	
	Punt.	Eqv.	Punt.	Eqv.	Punt.	Eqv.	Punt.	Eqv.
Aprovechamiento								
Conducta								

ASISTENCIA	I	II	III	Total
Días laborados por el alumno				
Faltas justificadas				
Faltas no justificadas				
Atrasos				
TOTAL DÍAS LABORADOS:				

En consecuencia:

es promovido al

Fuente: Propia

Figura C.2 Reporte Anual de Calificaciones

C.3. FORMATO DE ACTA – COMPROMISO DE MEJORA ACADÉMICA

ACTA - COMPROMISO DE MEJORA ACADÉMICA (PROYECTO GADGET)

En la ciudad de Ibarra a los días del mes de de 2012. En el Colegio Particular "Álamos", se reúnen: el Estudiante Sr. _____, el Representante Sr. _____, el Preceptor Sr. _____ y el Sr. Dirigente: _____.

Durante su permanencia en el Colegio se viene realizando el seguimiento académico del mencionado Estudiante, con la finalidad de dar cumplimiento a nuestro Proyecto Educativo Institucional.

Como docentes y representantes del Sr. Estudiante asumimos la responsabilidad del hecho académico, por lo que aceptamos cumplir y obligarnos a:

Docente	Padre de Familia	Estudiante
-Docente, dirigente y preceptor mantendrán informados a los padres de familia. -Docente implementará estrategias de trabajo personalizado para c/estudiante que se encuentre dentro del proyecto GADGET. Estrategias: -Utilización de un cuaderno GADGET. -Talleres de nivelación extra clases durante el periodo regular de trabajo. (7h00 – 13h30) -Utilizar adecuadamente la Agenda estudiantil, realizar un seguimiento adecuado a las notas allí registradas.	-Seguimiento y control a su hijo en tareas y tiempo dedicado al estudio en el hogar. -Evaluación e informe psicológico de un profesional externo para reforzar el realizado por el DOBE del Colegio. -Revisar, firmar y dar un correcto uso a la agenda estudiantil. - Establecer un horario de uso del tiempo libre, entrenamientos y tiempo para estudiar y descansar. - Informar al Colegio los resultados que van observando en su hijo desde el inicio del proyecto. - Asistir por lo menos una vez al mes al Colegio para conocer los avances académicos de su hijo.	- Fortalecer su responsabilidad a través de la presentación de trabajos bien hechos y dentro del tiempo establecido. - Prepararse adecuadamente para toda y cada una de las evaluaciones que se planifiquen por la Institución Educativa y por el docente de la asignatura. - Elaborar las tareas GADGET y solicitar nivelación en caso de necesitarla del profesor de la materia dentro del periodo de trabajo.

Para constancia firman:

Materia	Nombre del Profesor	Firma

 Padre de Familia

 Estudiante

 Dirigente

 Preceptor

Fuente: Propia

Figura C.3 Acta de Compromiso de Mejora Académica

C.4. FORMATO DE CONTROL DE DESARROLLO DEL NIVEL ACADÉMICO DEL ESTUDIANTE

CENTRO EDUCATIVO "ÁLAMOS"
CONTROL DE DESARROLLO DE NIVEL ACADÉMICO DEL ESTUDIANTE

ESTUDIANTE: _____
 AÑO: _____
 FECHA: _____

Materia:										Profesor:				
COMPORTAMIENTO	Ex		CUMP. DEBERES	Ex		PARTICIPACIÓN	Ex		PROBLEMAS		OBSERVACIONES		PETICIÓN PROFESOR	
	MB			MB			MB							
	B			B			B							
	R			R			R							
	I			I			I							

Materia:										Profesor:				
COMPORTAMIENTO	Ex		CUMP. DEBERES	Ex		PARTICIPACIÓN	Ex		PROBLEMAS		OBSERVACIONES		PETICIÓN PROFESOR	
	MB			MB			MB							
	B			B			B							
	R			R			R							
	I			I			I							

Materia:										Profesor:				
COMPORTAMIENTO	Ex		CUMP. DEBERES	Ex		PARTICIPACIÓN	Ex		PROBLEMAS		OBSERVACIONES		PETICIÓN PROFESOR	
	MB			MB			MB							
	B			B			B							
	R			R			R							
	I			I			I							

Materia:										Profesor:				
COMPORTAMIENTO	Ex		CUMP. DEBERES	Ex		PARTICIPACIÓN	Ex		PROBLEMAS		OBSERVACIONES		PETICIÓN PROFESOR	
	MB			MB			MB							
	B			B			B							
	R			R			R							
	I			I			I							

Fuente: Propia

Figura C.4 Control de Desarrollo del Nivel Académico