

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y
SEGUIMIENTO DE LAS ACTIVIDADES EDUCATIVAS DE LOS
ESTUDIANTES”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCION INGENIERÍA DE SISTEMAS**

Postulante : Gabriela Karina Quelali Siñani

Tutor : Lic. Nancy Orihuela Sequeiros

Revisor : Lic.: Abdias Patzi Choque

LA PAZ – BOLIVIA

2008

DEDICATORIA

A mis padres Santiago y Julia los cuales me ayudaron con su apoyo incondicional a ampliar mis conocimientos y estar más cerca de mis metas profesionales.

A mis hermanos Milenka, Jenny, Leonel y Danielita por haberme apoyado y creído en mí siempre.

Gabriela

AGRADECIMIENTOS

A Dios por dejarme llegar hasta aquí, y permitir cumplir parte de mis metas.

A mi tutor Lic. Nancy Orihuela Sequeiros por todas sus recomendaciones y colaboración en el proceso de desarrollo de este proyecto.

Al Lic. Abdías Patzi por todo el tiempo empleado en la revisión y corrección de este proyecto en calidad de revisor.

Al Ing. Jose Fernandez por su confianza en el desarrollo del proyecto.

A todas las personas, que me colaboraron desinteresadamente durante mis estudios y preparación de este proyecto en especial a Marco Luna.

Agradecer también a todos los docentes de la carrera de Informática por su apoyo y orientación en todos estos años.

RESUMEN

Ante la presencia de las nuevas tecnologías de información y comunicación las instituciones educativas se han visto comprometidas a afrontar la responsabilidad de implantar estas dentro su entorno, además se conoce que los procesos de enseñanza y aprendizaje son básicamente actos comunicativos, pues bien, las unidades educativas pretenden utilizar la enorme potencialidad de las nuevas tecnologías de información y comunicación como un canal de comunicación entre los padres de familia y la unidad educativa.

El proyecto desarrollado es un canal de comunicación que permite comunicar a las personas involucradas en el proceso educativo del estudiante de manera que el espacio y tiempo no sean factores determinantes para la comunicación, es así que este software permite conocer todas las actividades de los estudiantes en cualquier momento y desde cualquier lugar, también permite la comunicación directa entre educadores – tutores a través de envío de correos electrónicos, mensajes de texto a celulares y ofreciendo una agenda de seguimiento de actividades en línea.

Viendo las características del proyecto se hizo uso de la tecnología Web, utilizando como lenguaje de programación Java en conjunción con el Framework JSF (Java Server Faces) basado en el modelo MVC (Modelo Vista Controlador), que permite normalizar, estandarizar y acelerar el desarrollo de aplicaciones Web, y como Gestor de Base de Datos PostgreSQL.

La metodología utilizada para la construcción de este software es RUP ya que nos brinda un conjunto de actividades para transformar los requisitos de un usuario en un sistema de software.

ÍNDICE

CAPÍTULO I	MARCO REFERENCIAL	
1.1	INTRODUCCIÓN	1
1.2	ANTECEDENTES	2
1.3	ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA	2
1.4	OBJETIVOS	3
1.4.1	Objetivo General	3
1.4.2	Objetivos Específicos	3
1.5	METODOLOGÍA	4
1.6	HERRAMIENTAS	4
1.7	JUSTIFICACIÓN	4
1.7.1	Justificación Social	4
1.7.2	Justificación Económica	5
1.8	FACTIBILIDAD	5
1.9	ALCANCES	5
1.10	APORTES	5
CAPÍTULO II	MARCO TEÓRICO	
2.1	EDUCACIÓN BOLIVIANA	6
2.1.1	Sistema Educativo Nacional	6
2.1.2	Unidad Educativa	7
2.1.3	Director de la Unidad Educativa	7
2.1.3.1	Funciones del Director o del Director Encargado	7
2.1.4	Profesores	7
2.1.4.1	Funciones de los Profesores	7
2.1.5	Secretario	8
2.1.6	Padres de Familia	8
2.2	SISTEMA DE INFORMACIÓN	9
2.2.1	Componentes del Sistema de Información	9
2.2.2	Herramientas Tecnológicas	9

2.3 METODOLOGÍA RUP	10
2.3.1 Fases del Proceso Unificado de Desarrollo	14
2.3.2 UML (Lenguaje Unificado del Modelado)	15
2.3.2.1 Principales diagramas del UML	16
2.4 CALIDAD DE SOFTWARE	17
2.4.1 Definición de Calidad del Software	17
2.4.2 Métricas de Calidad	17
2.4.2.1 Características fundamentales de las métricas del software	18
2.4.3 Métrica de Calidad ISO 9126	19
2.5 HERRAMIENTAS DE DESARROLLO DE SOFTWARE	20
2.5.1 Lenguaje de Programación	20
2.5.1.1 Java	20
2.5.1.2 Framework	20
2.5.1.3 JSF (Java Server Faces)	20
2.5.2 Patrón Modelo Vista – Controlador	21
2.5.3 Sistema Gestor de Base de Datos	22
2.5.3.1 PostgreSQL	22
CAPÍTULO III MARCO APLICATIVO	
3.1 CAPTURA DE REQUERIMIENTOS	24
3.1.1 Análisis del Sistema Actual de Seguimiento	24
3.2 Análisis y Diseño	25
3.2.1 Requerimientos	25
3.2.2 Descripción general de actores	25
3.2.3 Diagramas de Casos de Uso	26
3.2.4 Descripción de casos de uso	28
3.2.5 Diagrama de Clases Inicial	30
3.2.6 Diagramas de Secuencia.	32
3.2.7 Diagramas de Actividades	36

CAPÍTULO IV IMPLEMENTACIÓN Y PRUEBAS

4.1 CONSTRUCCIÓN DEL SISTEMA	43
4.2 INTERFACES DEL SISTEMA	48
4.3 PRUEBAS	51
4.4 PUESTA EN MARCHA	57
4.5 CALIDAD DEL SOFTWARE	57
4.5.1 Punto Función	57
4.5.2 Factores de Calidad ISO 9126	59

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES	62
5.2 RECOMENDACIONES	62

BIBLIOGRAFÍA

ANEXOS

Anexo A ÁRBOL DE PROBLEMAS	A1
Anexo B ÁRBOL DE OBJETIVOS	B1

ÍNDICE DE FIGURAS

Figura 2.1	Un Proceso de Desarrollo de Software	11
Figura 2.2	Una Iteración del RUP	13
Figura 3.1	Diagrama de Caso de Uso Administrador - Sistema	26
Figura 3.2	Diagrama de Caso de Uso Docente - Sistema	27
Figura 3.3	Diagrama de Caso de Uso Estudiante - Sistema	27
Figura 3.4	Diagrama de Caso de Uso Apoderado - Sistema	28
Figura 3.5	Diagrama de Clases Inicial	31
Figura 3.6	Diagrama de Secuencia Gestionar Cursos	32
Figura 3.7	Diagrama de Secuencia Gestionar Materias	33
Figura 3.8	Diagrama de Secuencia Gestionar Docentes	33
Figura 3.9	Diagrama de Secuencia Gestionar Estudiantes	34
Figura 3.10	Diagrama de Secuencia Revisar Comunicados	34
Figura 3.11	Diagrama de Secuencia Gestionar Actividades de Curso	35
Figura 3.12	Diagrama de Secuencia Revisar Desempeño de Estudiante	35
Figura 3.13	Diagrama de Secuencia Revisar Actividades de Curso	36
Figura 3.14	Diagrama de Actividad para el Ingreso al Sistema	36
Figura 3.15	Diagrama de Actividad para la Creación de un Registro	37
Figura 3.16	Diagrama de Actividad para la Edición de un Registro	37
Figura 3.17	Diagrama de Actividad para la Eliminación de un Registro	38
Figura 3.18	Diagrama de Actividad para el Envío de Comunicados	39
Figura 3.19	Diagrama de Clases	40
Figura 3.20	Diagrama de Relacional Parte I	41
Figura 3.21	Diagrama de Relacional Parte II	42
Figura 4.1	Bean	44
Figura 4.2	ManagedBean	45
Figura 4.3	Navegador Principal .jsf	46
Figura 4.4	Archivo descriptor faces-config.xml	47
Figura 4.5	Pantalla de Ingreso al Sistema	48
Figura 4.6	Menú de Opciones- Perfil Administrador	49

Figura 4.7	Listado de Estudiantes- Perfil Administrador	49
Figura 4.8	Registro de Estudiante- Perfil Administrador	50
Figura 4.9	Edición de Estudiante- Perfil Administrador	51
Figura 4.10	Listado de Asignaturas- Perfil Administrador	52
Figura 4.11	Registro de Asignatura- Perfil Administrador	52
Figura 4.12	Listado de Asignaturas- Perfil Administrador	53
Figura 4.13	Cursos del Educador – Perfil de Educador	54
Figura 4.14	Asignaturas impartidas por el Educador – Perfil de Educador	54
Figura. 4.15	Actividades de Asignatura – Perfil de Educador	55
Figura 4.16	Registro de Actividad – Perfil de Educador	56
Figura 4.17	Listado de Actividades de un Curso – Perfil de Padre de Familia o Tutor	56

ÍNDICE DE TABLAS

Tabla 2.1	Diagramas de UML	16
Tabla 3.1	Descripción general de Actores	25
Tabla 4.1	Factores de Ponderación Punto Función	57
Tabla 4.2	Factores de Complejidad	58
Tabla 4.3	Factor de Calidad -Eficiencia	59
Tabla 4.4	Factor de Calidad -Fiabilidad	59
Tabla 4.5	Factor de Calidad -Usabilidad	60
Tabla 4.6	Factor de Calidad -Eficiencia	60
Tabla 4.7	Factor de Calidad -Movilidad	60
Tabla 4.8	Factor de Calidad -Usabilidad	61

CAPÍTULO I

MARCO REFERENCIAL

1.11 INTRODUCCIÓN

Una buena educación es la llave para mejorar la calidad de vida de las personas ya que posibilita la obtención de mejores trabajos, a mejor educación son mayores las posibilidades de obtener un nivel alto de calidad de vida.

Es por esta razón que la educación es importante desde sus inicios, y más aun cuando el estudiante es un niño o adolescente necesita de un mayor control y seguimiento de sus actividades educativas de parte de sus tutores y las personas involucradas en el proceso de enseñanza – aprendizaje del estudiante.

Para los padres de familia realizar un seguimiento correcto, oportuno y efectivo implica una inversión de tiempo que en muchos casos es imposible por distintos motivos. Esto a su vez ocasiona una pobre comunicación entre los tutores y la unidad educativa, lo cual crea grandes brechas en la consecución de este objetivo.

Los procesos de enseñanza y aprendizaje son básicamente actos comunicativos, pues bien, las unidades educativas pretenden utilizar la enorme potencialidad de las nuevas tecnologías de información y comunicación para que estos apoyen sus objetivos.

El presente proyecto pretende ser un canal de comunicación e información entre profesores, padres de familia, asesores y directores con el propósito de realizar el seguimiento y control de las actividades educativas de cada estudiante para cumplir con los objetivos propuestos por todo el conjunto de personas que se involucran en este proceso educativo.

1.12 ANTECEDENTES

Ante la presencia de las nuevas tecnologías de información y comunicación las instituciones educativas se han visto comprometidas a afrontar la responsabilidad de implantar estas nuevas tecnologías dentro su entorno.

Dentro del sistema educativo las primeras instituciones educativas que asumieron el reto fueron las universidades al cambiar la manera tradicional de educación, donde el estudiante adquiría la mayoría de sus conocimientos por medio del docente y de la bibliografía impresa que este podía consultar, con la aparición de Internet el estudiante tiene acceso a todo tipo de información de manera libre y directa. Una vez que las universidades conocieron las ventajas de las nuevas tecnologías de información y comunicación incorporaron a sus instituciones sistemas de información y control en la parte administrativa, posteriormente implementaron sistemas que les permitiese tanto a docentes como estudiantes trabajar de una manera más dinámica implementando sistemas de gestión educativa (educación a distancia, foros, Chat y otros).

En cuanto a las Unidades Educativas, estas van introduciéndose de a poco al mundo de la tecnología y reconocen que el uso de las nuevas tecnologías son útiles y necesarias para el mejoramiento del proceso de enseñanza y aprendizaje de los estudiantes.

Fuera de nuestras fronteras, colegios y escuelas implementaron algunos sistemas de gestión educativa, tal es el caso de un producto denominado SCHOOLGEN software de origen hindú que permite llevar el seguimiento de las actividades educativas de los estudiantes.

SchoolGen es un software de administración escolar en Web, donde los administradores pueden administrar sus escuelas desde cualquier lugar, los módulos que incluye son: asistencia en línea, asignaciones de tareas a estudiantes, salud, exámenes en línea, banco de preguntas, sin embargo este software no se adecua al sistema educativo de nuestro país.

1.13 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

En el contexto general del desarrollo educativo, la eficiencia del proceso de aprendizaje, se analiza a través del cumplimiento de los objetivos educativos y el adecuado rendimiento en

diferentes áreas, por esta razón los padres desempeñan un papel central en asegurarse que sus hijos obtengan la educación que necesitan y merecen.

Como se menciono anteriormente la educación es crucial en el desarrollo de una persona, por esta razón es importante que todas las personas involucradas en el proceso educativo, puedan realizar un seguimiento a las actividades educativas de los estudiantes.

Los padres de familia, profesores y directores necesitan de una comunicación fluida para obtener los resultados que estos esperan y poder corregir situaciones que perjudiquen el normal proceso educativo de los estudiantes oportunamente. Tales situaciones nos permiten hacer el siguiente planteamiento del problema:

¿Cómo conseguir la comunicación entre profesores, padres de familia y estudiantes, para llevar un mejor control y seguimiento de las actividades educativas del estudiante?

1.14 OBJETIVOS

1.14.1 Objetivo General

Desarrollar e implementar un sistema de información Web que permita el control y seguimiento sobre las actividades educativas de los estudiantes en una unidad educativa.

1.14.2 Objetivos Específicos

- Llevar un control sobre las actividades que se tienen en las distintas asignaturas y realizar el seguimiento del cumplimiento de las mismas.
- Permitir comunicación fluida entre padres de familia-profesores, padres de familia-directores, profesores-directores a través de envío mensajes sms y envío de correo electrónico.
- Lograr la participación activa de los padres de familia en el proceso educativo de sus hijos.

1.15 METODOLOGÍA

La metodología utilizada en el proyecto será la metodología RUP (*Rational Unified Process*) ya que define claramente *quién, cómo, cuándo y qué debe hacerse*; y, como su enfoque está basado en modelos utiliza un lenguaje bien definido para tal fin, el UML el cual aporta herramientas como los casos de uso, que definen los requerimientos. Además permite la ejecución iterativa del proyecto y del control de riesgos.

1.16 HERRAMIENTAS

Para la implementación de este proyecto se utilizará las siguientes herramientas tecnológicas:

- Plataforma Windows
- Lenguaje Java (JSF Framework)
- Servidor de Aplicaciones Apache Tomcat
- Sistema Gestor de Base de Datos PostgreSQL

1.17 JUSTIFICACIÓN

1.7.1 Justificación Social

El desarrollo e implementación del Proyecto beneficiará significativamente al estudiante y a todas las personas involucradas en el proceso educativo contribuyendo a:

- Comunicar a estas personas sin que el espacio ni el tiempo sean factores determinantes.
- Tomar medidas sobre cualquier factor que perjudique el proceso de enseñanza-aprendizaje del estudiante de manera oportuna.
- Llevar un seguimiento general de los estudiantes dentro la institución educativa.

1.7.2 Justificación Económica

Al ser un proyecto relativamente nuevo y al conocer las necesidades tanto de los padres de familia y unidades educativas se observa que existe demanda del proyecto y que posteriormente este pueda ser enfocado de una manera comercial.

1.18 FACTIBILIDAD

El proyecto es factible de inicio ya que al existir demanda de parte de los padres de familia e unidades educativas todo el proceso de análisis y diseño se concretara de manera normal ya que la parte interesada brindara toda la información necesaria para la implementación rápida de este proyecto.

Además que su implementación será también factible porque proyecto estará basado en tecnología Web que permite el fácil acceso desde cualquier terminal con solamente contar con el servicio de Internet.

1.9 ALCANCES

El proyecto tendrá como ámbito de trabajo la unidad educativa y el ambiente que rodea a tal institución, específicamente brindara servicios a padres de familias, estudiantes, profesores y administrativos del colegio o escuela.

1.10 APORTES

Concluida la implementación del proyecto podrá aportar en lo siguiente:

- Seguimiento y control de las actividades educativas de los estudiantes.
- Mayor participación de los padres de familia en el proceso educativo de sus hijos.
- Mayor comunicación entre padres de familia-profesores, padres de familia-directores, profesores-directores.

CAPÍTULO II

MARCO TEÓRICO

2.1 EDUCACIÓN BOLIVIANA

La educación boliviana es la más alta función del estado, porque es un derecho del pueblo e instrumento de liberación nacional y porque tiene la obligación de sostenerla, dirigirla y controlarla, a través de un vasto sistema escolar.

Es el fundamento de la integración nacional y de la participación de Bolivia en la comunidad regional y mundial de naciones, partiendo de la afirmación de nuestra soberanía e identidad.

2.1.1 Sistema Educativo Nacional

Son objetivos y políticas del Sistema Educativo Nacional:

- Garantizar la sólida y permanente formación de nuestros Recursos Humanos a través de instrumentos dinámicos, para situar a la Educación Boliviana a la altura de las exigencias de los procesos de cambio del país y del mundo.
- Organizar un Sistema Educativo Nacional capaz de renovarse y de mejorar su calidad permanente para satisfacer las cambiantes necesidades de aprendizaje y de desarrollo nacional así como para incorporar las innovaciones tecnológicas y científicas creando instrumentos de control y seguimiento y evaluación, con especial énfasis en la medición de la calidad, instrumentos de información y de investigación educativas.
- Mejorar la calidad y la eficiencia de la educación haciéndola pertinente a las necesidades de la comunidad y ampliándola en su cobertura y en la permanencia de los educandos en el sistema educativo y garantizando la igualdad de los derechos de hombres y mujeres.

De acuerdo a los objetivos planteados por el sistema educativo nacional, todos los entes involucrados juegan un papel determinante en la formación de los Recursos Humanos del país y por ende en el desarrollo de este.

2.1.2 Unidad Educativa

La Unidad Educativa es un centro de formación integral del educando, donde se desarrollan los procesos de enseñanza y aprendizaje a partir de una planificación participativa con la cooperación de director, docentes, alumnos, padres de familia.

2.1.3 Director de la Unidad Educativa

El Director es la máxima autoridad de la unidad educativa, responsable de su programación. Depende del Director de Núcleo y si este no está designado el Director Distrital.

2.1.3.1 Funciones del Director o del Director Encargado

Son funciones del Director de la Unidad Educativa las siguientes:

- Planificar, organizar, dirigir y supervisar los procesos pedagógicos.
- Planificar, organizar, dirigir y controlar las actividades administrativas de la Unidad Educativa.

2.1.4 Profesores

Los profesores son responsables de desarrollar los procesos de enseñanza para promover la educación integral de los alumnos.

2.1.4.1 Funciones de los Profesores

- Planificar, desarrollar y evaluar las actividades curriculares de sus alumnos, en coordinación con el Consejo de Profesores de un mismo ciclo.
- Registrar la asistencia de los alumnos a su cargo, informando de ello trimestralmente a la dirección.

- Elaborar los informes de aprendizaje y llenar las libretas escolares de los alumnos.
- Velar por la seguridad de los alumnos desde el ingreso hasta la salida, incluyendo las horas de recreo.
- Coordinar y mantener comunicación permanente con los padres de familia sobre el rendimiento académico y el comportamiento de sus alumnos.
- Promover y participar en el mantenimiento del orden y aseo del aula con la colaboración de sus alumnos.

2.1.5 Secretario

Son las funciones del secretario:

- Recibir, registrar, distribuir, archivar y custodiar toda la documentación de la Unidad educativa.
- Procesar y elaborar certificaciones que soliciten las autoridades, los alumnos, padres de familia o apoderados.
- Llenar y centralizar los libros de inscripción y de notas, formularios y kardex de los alumnos y profesores.

2.1.6 Padres de Familia

A efectos de la relación con la Unidad Educativa, son considerados padres de familia o apoderados aquellas personas cuyos hijos o pupilos son alumnos regulares de la Unidad educativa.

Los padres de familia tienen la obligación de:

- Justificar mediante nota escrita o personalmente los atrasos de sus hijos, así como el incumplimiento de los deberes escolares de los mismos.
- Recabar y firmar los informes de aprendizajes o las libretas escolares y otros documentos que se les envíen desde la Unidad Educativa, devolviéndolos oportunamente.
- Autorizar por escrito la participación de su hijo en actividades fuera de la unidad educativa.

2.2 SISTEMA DE INFORMACIÓN

Un **sistema de información** se define como un conjunto de procedimientos interrelacionados que forman un todo, es decir, obtiene, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización. Igualmente apoya la coordinación, análisis de problemas, visualización de aspectos complejos, entre otros aspectos. Los elementos interactúan entre sí con el fin de apoyar las actividades de las empresas, negocios u organizaciones.

Sistema de Información: es el conjunto de procesos que, operando sobre una colección de datos estructurada de acuerdo a una empresa, recopila, elabora y distribuye (parte de) la información necesaria para la información de dicha empresa y para las actividades de dirección y control correspondientes, apoyando al menos en parte, la toma de decisiones necesarias para desempeñar las funciones y procesos de negocios de la empresa de acuerdo a su estrategia.

2.2.1 Componentes del Sistema de Información

- Herramientas tecnológicas (hardware, software)
- El equipo computacional: el hardware necesario para que el sistema de información pueda operar.
- El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema, también conocidos como usuarios.
- Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

2.2.2 Herramientas Tecnológicas

Es indudable el gran impacto que el desarrollo de las nuevas tecnologías ha tenido sobre la sociedad en general, además como muchos autores han enunciado el advenimiento de la llamada sociedad del conocimiento y la información ha encontrado una plataforma de apoyo bastante adecuada con la aparición de las mismas y en especial las referidas a la producción, control, acceso, recuperación, conservación y disseminación de información.

Ahora bien, La sociedad que tradicionalmente se ha encargado de ser los custodios administradores y difusores de la información (Bibliotecas, Archivos, Centros de Documentación, información empresarial etc.) han hecho una labor importante pero no tan rápida y eficiente como un sistema electrónico de control de información que, y es importante destacarlo, manipulado por el factor humano simplemente actúa facilitándole la manera de administrar y difundir la información.

La computadora personal, el acceso a bases de datos, los bancos de datos, los discos compactos, la representación de la información multimediática, los desarrollos de software para la gestión documental que posibilitan almacenar gran cantidad de información y por último la aparición de Internet han modificado y cambiado la tradicional perspectiva de los servicios que ofrecen las unidades de información.

La introducción de estas tecnologías implica un cambio de la sociedad. Se habla de sociedad de la información o sociedad del conocimiento. Se trata de un cambio en profundidad de la propia sociedad. Las nuevas tecnologías de la información y la comunicación designan a la vez un conjunto de innovaciones tecnológicas pero también las herramientas que permiten una redefinición radical del funcionamiento de la sociedad.

La expansión de las tecnologías de la información y la comunicación basadas en la microelectrónica, la informática, la robótica y las redes de comunicaciones se está produciendo a gran velocidad en todos los ámbitos socioeconómicos y de las actividades humanas configurando la nombrada **Sociedad de Información**.

2.6 METODOLOGÍA RUP

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (Véase la Figura 2.1). El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos.



Figura 2.1 Un Proceso de Desarrollo de Software

Fuente: I. Jacobson, G. Booch, J. Rumbauch

El Proceso Unificado utiliza el UML para preparar todos los esquemas de un sistema de software. De hecho UML es una parte esencial del Proceso Unificado.

Las características primordiales del Proceso Unificado son:

- Dirigidos por Casos de Uso
- Centrado en la Arquitectura
- Iterativo Incremental

a) Dirigidos por Casos de Uso

Los Casos de Uso son una técnica de captura de requisitos. Los desarrolladores comienzan capturando los requisitos del cliente en la forma de Casos de Uso.

Normalmente, un sistema tiene muchos tipos de usuarios, cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los Casos de Uso. Un Caso de Uso es una secuencia de acciones que el sistema lleva a cabo para ofrecer algún resultado de valor para un actor.

b) Centrado en la Arquitectura

La arquitectura de un sistema es la organización o la estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

c) Proceso Iterativo e Incremental

Según [JBR - 00] el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 2.2. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

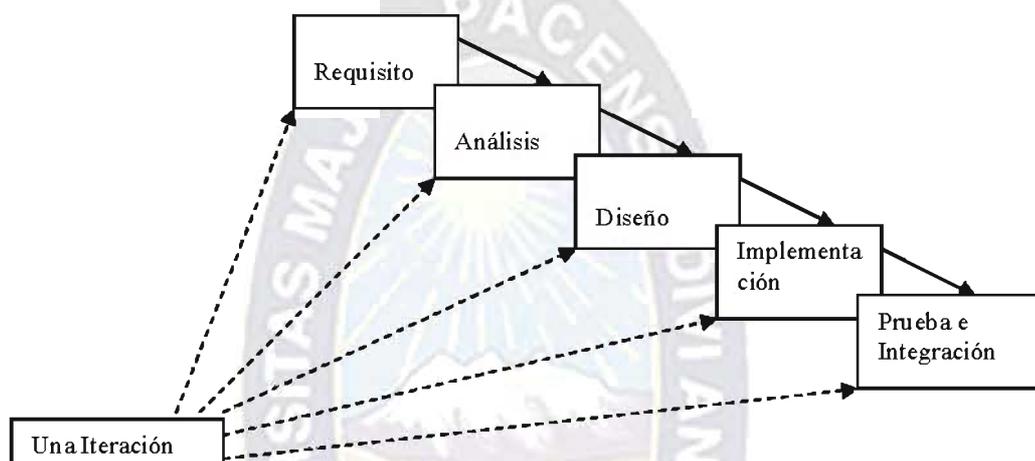


Figura 2.2: Una iteración del RUP

Fuente: Ingeniería de Software

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso.

Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

2.6.1 Fases del Proceso Unificado de Desarrollo

El ciclo de desarrollo se divide en cuatro fases; inicio, elaboración, construcción y transición; con la secuencia de modelos se visualiza lo que sucede en cada fase.

a) Inicio

Durante la fase de inicio, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales .

b) Elaboración

Durante la fase de elaboración, se especifican en detalle la mayoría de los Casos de Uso del producto y se diseña la arquitectura del sistema. La arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan el sistema entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y modelo de despliegue. Durante esta fase de desarrollo, se realizan los casos de uso más críticos que se identificaron en la fase de comienzo.

c) Construcción

Durante esta fase se crea el producto para ser entregado a la comunidad de usuarios. El grueso de los recursos requeridos se emplea durante esta fase de desarrollo. Sin embargo la arquitectura del sistema es estable, aunque los desarrolladores pueden descubrir formas mejores de estructurar el sistema.

Al final de esta fase, el producto el producto contiene todo los casos de uso que la dirección y el cliente han acordado para el desarrollo de esta versión. Sin embargo puede que no este completa libre de efectos. Muchos de estos defectos se descubren y solucionan durante la fase de transición.

d) Transición

Cubre el periodo durante el cual el producto software se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan alguna de las mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios.

2.6.2 UML (Lenguaje Unificado del Modelado)

El UML es una de las herramientas más utilizadas en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas.

El UML está constituido por un conjunto de diagramas que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema a las cuales se les conoce como un modelo. El modelo de UML de un sistema es similar a un modelo de escala de un edificio junto con la interpretación del artista del edificio. Es importante rescatar que un Modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema.

2.3.2.1 Principales diagramas del UML

Diagramas	Descripción
Diagrama de Clases	Muestra la entidades en un sistema y la forma en que tales entidades se relacionan entre si. Cada clase se representa como un rectángulo con nombre.
Diagrama de Objetos	Muestra las instancias de las clases y sus relaciones entre si. Cada Objeto se representa como un rectángulo con nombre.
Diagramas de Casos de Uso	Muestra la utilización del sistema Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. Cada caso de uso aparece como una elipse y cada actor como una figura representativa.
Diagramas de Estados	Captura el estado de un objeto durante cierto periodo. Un estado se representa como un rectángulo de vértices redondeados, y una transición entre estados como una línea que los conecta.
Diagramas de Actividad	Muestra los pasos y puntos de decisión que suceden dentro del comportamiento de un objeto, o dentro de un proceso de negocios. Cada paso es un rectángulo de vértices redondeados y cada punto de decisión es un rombo.
Diagramas de Secuencia	Representa la forma en que interaccionan los objetos entre si al paso del tiempo. Los objetos se colocan en la parte superior, y el tiempo procede desde la parte superior hacia la parte inferior del diagrama. Las flechas denotan mensajes que van de objeto a objeto.
Diagramas de Colaboración	Es otra forma de representar la manera en que los objetos trabajan en conjunto con el tiempo. Los objetos podrían estar en cualquier parte del diagrama. Los mensajes de un objeto a otro aparecen como líneas que conectan a los objetos. Cada línea está numerada de acuerdo con su ubicación en la secuencia de mensajes y muestra información que pertenece a la naturaleza del mensaje.
Diagramas de Distribución	Representa la arquitectura física de un sistema de cómputo. Puede mostrar cada equipo de cómputo y dispositivo en el sistema y los componentes que en ellos residen. El equipo de computo o nodo se representa como un cubo con los componentes individuales dentro de el.

Tabla 2.1: Diagramas de UML

Fuente: UML en 24 Horas

2.4 CALIDAD DE SOFTWARE

Todas las metodologías y herramientas tienen un único fin producir software de gran calidad.

2.4.1. Definición de Calidad del Software

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [PRE,2000]

La anterior definición sirve para hacer hincapié a tres puntos importantes.

- Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería de software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.
- Existen algunos requisitos implícitos que a menudo no se mencionan (por ejemplo: el deseo por facilitar el uso y el buen mantenimiento). Si el software se ajusta a sus requisitos explícitos pero falla en alcanzar los requisitos implícitos, la calidad del software queda en entredicho.

2.4.2 Métricas de Calidad

Las métricas sirven para valorar la calidad de los productos de ingeniería o los sistemas que se construyen.

Proporcionan una manera sistemática de valorar la calidad basándose en un conjunto de reglas claramente definidas.

2.4.2.1 Características fundamentales de las métricas del software

Se han propuesto cientos de métricas de calidad para el software, pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones que son demasiadas complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas y otras violan las nociones básicas intuitivas de lo que es realmente un software de alta calidad.

Ejiogu [EJ191] define un conjunto de atributos que deberían acompañar a las métricas efectivas del software.

Simples y fáciles de calcular. Debería ser relativamente fácil aprender a obtener la métrica y su cálculo no debería demandar un esfuerzo y cantidad de tiempo inusuales.

Empírica e intuitivamente persuasivas. La métrica debería satisfacer las nociones intuitivas del ingeniero sobre el atributo del producto en cuestión (por ejemplo, una métrica, que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).

Consistentes y objetivas La métrica debería siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información del software.

Consistentes en el empleo de tamaño El cálculo matemático de la métrica debería emplear medidas que no conduzcan extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente persuasivas.

Independientes del lenguaje de programación. Las métricas deberían basarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.

Un eficaz mecanismo para la realimentación de calidad. La métrica debería proporcionar, al desarrollador de software, información que le lleve a un producto final de mayor calidad.

2.4.3 Métrica de Calidad ISO 9126

El estándar identifica seis atributos clave de calidad:

- **Funcionalidad:** El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.
- **Confiabilidad:** Cantidad de tiempo que el software está disponible para su uso. Estará referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- **Usabilidad:** Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- **Eficiencia:** Grado en que el software hace óptimo el uso de los recursos del sistema. Viene reflejado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- **Facilidad de mantenimiento:** La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- **Portabilidad:** La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

2.5 HERRAMIENTAS DE DESARROLLO DE SOFTWARE

2.5.1 Lenguaje de Programación

2.5.1.1 Java

Java es un lenguaje de programación orientado a objetos y multiplataforma, que nos permite tener desarrollos bastante completos, ya que es un lenguaje de propósito general con el que se puede programar desde una aplicación de escritorio, una aplicación Web, aplicaciones para dispositivos móviles y otros.

2.5.1.2 Framework

Un framework, es un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

Los objetivos principales que persigue un framework son:

- Acelerar el proceso de desarrollo.
- La reutilización de código ya existente.
- La promoción de buenas prácticas de desarrollo de software como el uso de patrones (MVC).

2.5.1.3 JSF (Java Server Faces)

Java Server Faces es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador)

JSF pretende normalizar y estandarizar el desarrollo de aplicaciones Web. A continuación algunos puntos por que JSF es una tecnología muy interesante para el desarrollo de Aplicaciones Web.

- JSF trata la vista (interfaz de usuario) de una forma algo diferente a lo que estamos acostumbrados en aplicaciones Web. Sería más similar al estilo de Swing, Visual Basic o

Delphi, donde la programación de la interfaz se hace a través de componentes y basada en eventos (se pulsa un botón, cambia el valor de un campo).

-JSF es muy flexible. Por ejemplo nos permite crear nuestros propios componentes, o crear nuestros propios “render” para pintar los componentes según nos convenga.

- JSF actualmente es un framework de amplia aceptación en la comunidad de desarrolladores de aplicaciones Web en el mundo, por lo que continuamente van apareciendo nuevos componentes que se pueden integrar con esta tecnología.

2.5.2 Patrón Modelo Vista – Controlador

El patrón Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Podemos describir el patrón de la siguiente manera:

- Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchos sistemas informáticos utilizan un sistema de gestión de base de datos para gestionar los datos. En MVC corresponde al modelo.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.5.3 Sistema Gestor de Base de Datos

2.5.3.1 PostgreSQL

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre.

Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

PostgreSQL soporta conectividad para todos los lenguajes de programación conocidos y actualmente es considerada como una de los DMBS más robustos y seguros del mercado. PostgreSQL esta diseñado para ser soportado en muchas plataformas (Linux en todas sus distribuciones, AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64, Windows).



CAPÍTULO III

MARCO APLICATIVO

3.1 CAPTURA DE REQUERIMIENTOS

Antes de comenzar con la captura de requerimientos se realizara una revisión de los procesos actuales para así conocer las debilidades del sistema actual y poder fortalecerlas en el Sistema que se desarrollara.

3.1.1 Análisis del Sistema Actual de Seguimiento

Inicialmente se identifican los procesos actuales del seguimiento que realizan los padres de familia y la unidad educativa sobre los estudiantes:

- En la actualidad los padres de familia tienen horarios de consultas para averiguar el desempeño de sus hijos en las distintas materias, horarios que pueden ser variables dependiendo la disponibilidad de tiempo que tenga el Docente.
- El Estudiante cuenta con una agenda de seguimiento/asistencia, donde el docente registra cualquier tipo de observación sobre el desempeño del estudiante.
- Cada trimestre se hace llegar un Boletín de calificaciones que refleja el desempeño y aprovechamiento del estudiante.

Identificando las deficiencias de seguimiento y control que tiene el sistema actual los padres de familia y la unidad educativa pretenden utilizar la enorme potencialidad de las nuevas tecnologías de información y comunicación para que estos apoyen sus objetivos.

3.2 Análisis y Diseño

3.2.1 Requerimientos

Para identificar los requerimientos funcionales del sistema se utilizarán los casos de uso para una mejor comprensión.

En primer lugar se hará la descripción de las personas que se encuentran involucradas en los distintos procesos de seguimiento y control de actividades de los estudiantes (Ver [Tabla 3.1](#)).

3.2.2 Descripción general de actores

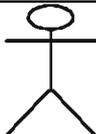
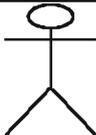
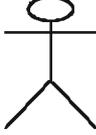
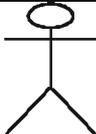
ACTOR	DESCRIPCIÓN
	Administrador es la persona encargada de registrar estudiantes, apoderados de estudiantes, docentes, cursos, horarios, asignación de cursos y asignación de horarios y comunicar actividades que se llevan en la unidad educativa.
	Docente es el responsable de desarrollar los procesos de enseñanza en las materias o asignaturas que tiene asignado, realizando a su vez un seguimiento de cumplimiento de actividades a los estudiantes, para después elaborar los informes de rendimiento.
	Apoderado es la persona cuyo hijo o pupilo es alumno regular de la unidad educativa y es responsable de justificar el cumplimiento ó incumplimiento de los deberes escolares de su hijo
	Estudiante es la persona central del Sistema sobre la que se ejerce el seguimiento.

Tabla 3.1. Descripción General de Actores

Fuente: Elaboración Propia

Una vez identificados los actores se mostrara en los siguientes diagramas de casos de uso la interacción de cada actor con el sistema.

3.2.3 Diagramas de Casos de Uso

El siguiente diagrama de casos de uso es la representación la interacción que tendrá el administrador con el sistema (Ver Figura 3.1).

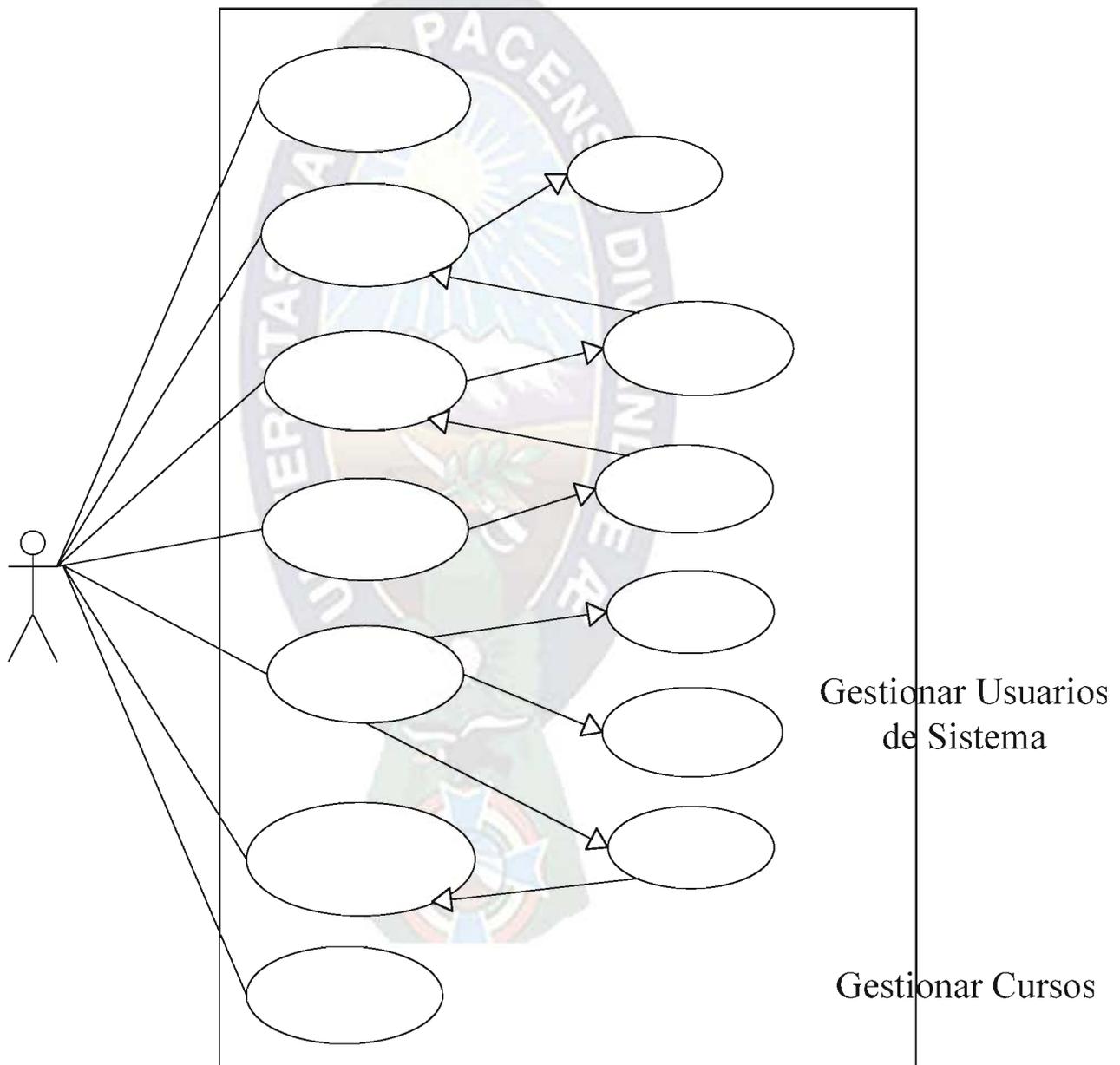


Figura 3.1: Diagrama de Caso de Uso Administrador - Sistema

Fuente: Elaboración Propia

La siguiente figura representa las interacciones del docente con el sistema (Ver Figura 3.2).

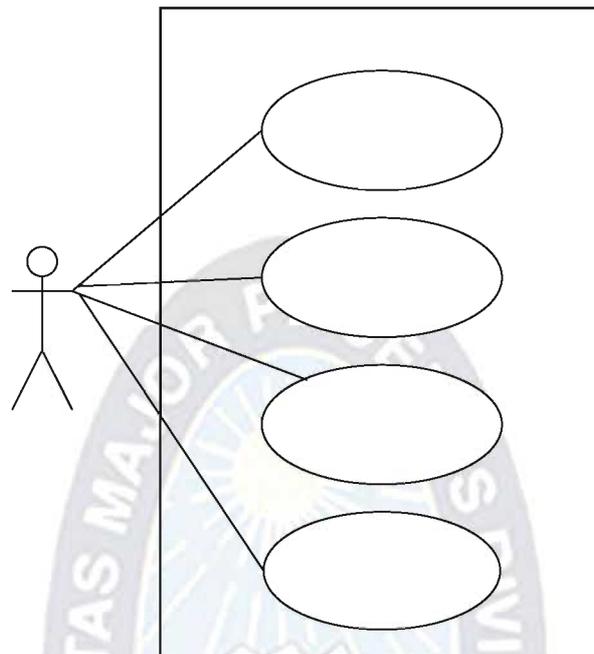


Figura 3.2: Diagrama de Caso de Uso Docente - Sistema

Fuente: Elaboración Propia

La siguiente figura de caso de uso representa las funcionalidades que tendrá el estudiante con el sistema (Ver Figura 3.3).

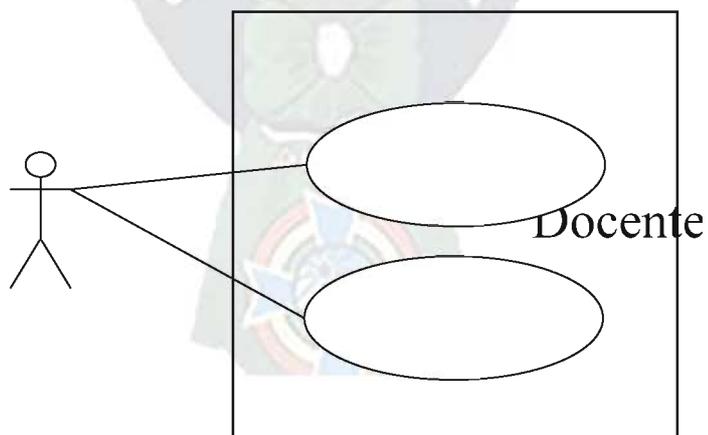


Figura 3.3: Diagrama de Caso de Uso Estudiante - Sistema

Fuente: Elaboración Propia

Y por ultimo el siguiente diagrama de caso de uso representa las funcionalidades que tendrá el apoderado con el sistema (Ver Figura 3.4).

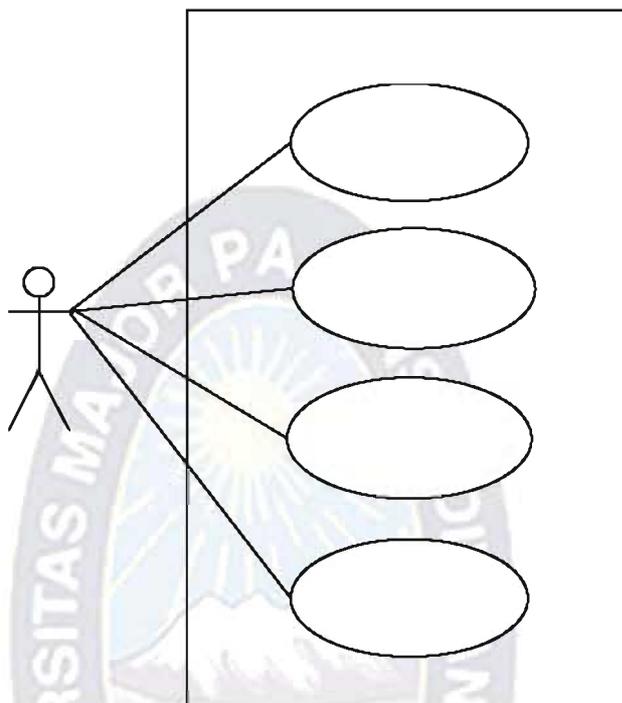


Figura 3.4: Diagrama de Caso de Uso Apoderado - Sistema

Fuente: Elaboración Propia

A continuación se describe cada caso de uso para una mejor comprensión de los mismos.

3.2.4 Descripción de casos de uso

a) **Caso de Uso 1:** Gestionar Cursos

Este caso de uso tiene como propósito ofrecer la funcionalidad de crear, editar y eliminar cursos, previa verificación de datos.

b) **Caso de Uso 2:** Gestionar Materias

Este caso de uso tiene el propósito de ofrecer la funcionalidad de crear, editar y eliminar materias, previa verificación de datos.

Apoderado de Estudiante

- c) **Caso de Uso 3:** Gestionar Docentes
El propósito de este caso de uso es ofrecer la funcionalidad de crear, editar y eliminar docentes, previa verificación de datos.
- d) **Caso de Uso 4:** Asignar Materias a Curso
Este caso de uso tiene como fin ofrecer la funcionalidad de asignar materias a un curso determinado.
- e) **Caso de Uso 5:** Asignar Docente a Materia de un Curso
Ofrece la funcionalidad de asignar y eliminar docentes a las materias de un determinado curso.
- f) **Caso de Uso 6:** Gestionar Estudiantes
Ofrece la funcionalidad para crear, editar y eliminar el registro de un estudiante además permite vincularlo con sus respectivos apoderados.
- g) **Caso de Uso 7:** Gestionar Apoderados de Estudiantes
Ofrece la funcionalidad para crear, editar y eliminar el registro de un apoderado de estudiante.
- h) **Caso de Uso 8:** Asignar Estudiante a un Curso
Este caso de uso tiene como propósito permitir la asignación y eliminación de uno o más estudiantes a un determinado curso.
- i) **Caso de Uso 9:** Habilitar Servicio de Comunicados a Estudiantes
Este caso de uso tiene como propósito habilitar el servicio de envío y recepción de comunicados al estudiante y a su respectivo apoderado.
- j) **Caso de Uso 10:** Enviar comunicados
Este caso de uso tiene como propósito permitir el registro y envío de comunicados a estudiantes, docentes y apoderados de estudiantes.

k) **Caso de Uso 11:** Gestionar actividades de curso

Este caso de uso tiene como propósito ofrecer la funcionalidad de crear, editar y eliminar actividades en la materia que imparte un docente dentro un curso.

l) **Caso de Uso 12:** Registrar observaciones de desempeño de estudiante

Este caso de uso ofrece la funcionalidad de crear, editar y eliminar notas de observación para cada estudiante y comunicarlas si se desea.

m) **Caso de Uso 14:** Revisar Comunicados

Es propósito de este caso de uso es permitir revisar comunicados que fueron dirigidos a la persona o enviados por la persona.

n) **Caso de Uso 15:** Revisar Actividades de Apoderado

Este caso de uso ofrece la funcionalidad de realizar un seguimiento de las actividades que tiene el estudiante en las diferentes materias o asignaturas..

o) **Caso de Uso 16:** Revisar Desempeño de Estudiante

Este caso de uso ofrece la funcionalidad de revisar el cumplimiento y desempeño del estudiante en las distintas materias o asignaturas.

3.2.5 Diagrama de Clases Inicial

En la figura 3.5 se describen las clases identificadas en primera instancia dentro el sistema.



Figura 3.5: Diagrama de Clases Inicial

Fuente: Elaboración Propia

Cada caso de uso descrito anteriormente será representado por un Diagrama de Secuencia para ver como interactúan los objetos entre si mismos.

3.2.6 Diagramas de Secuencia.

Los diagramas de secuencia serán mostrados desde la figura 3.6 hasta la figura 3.14

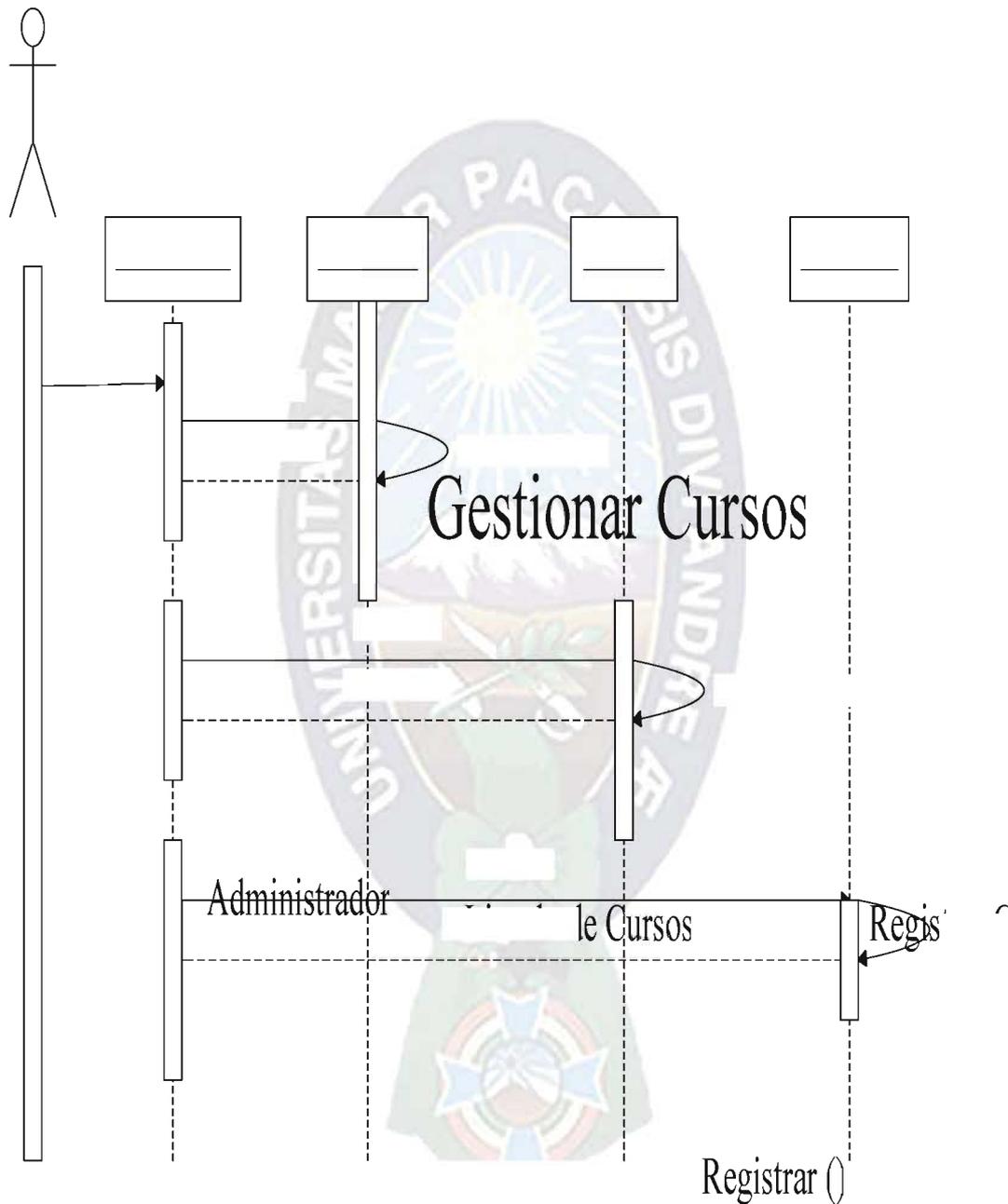


Figura 3.6: Diagrama de Secuencia Gestionar Cursos
Fuente: Elaboración Propia

Verifica

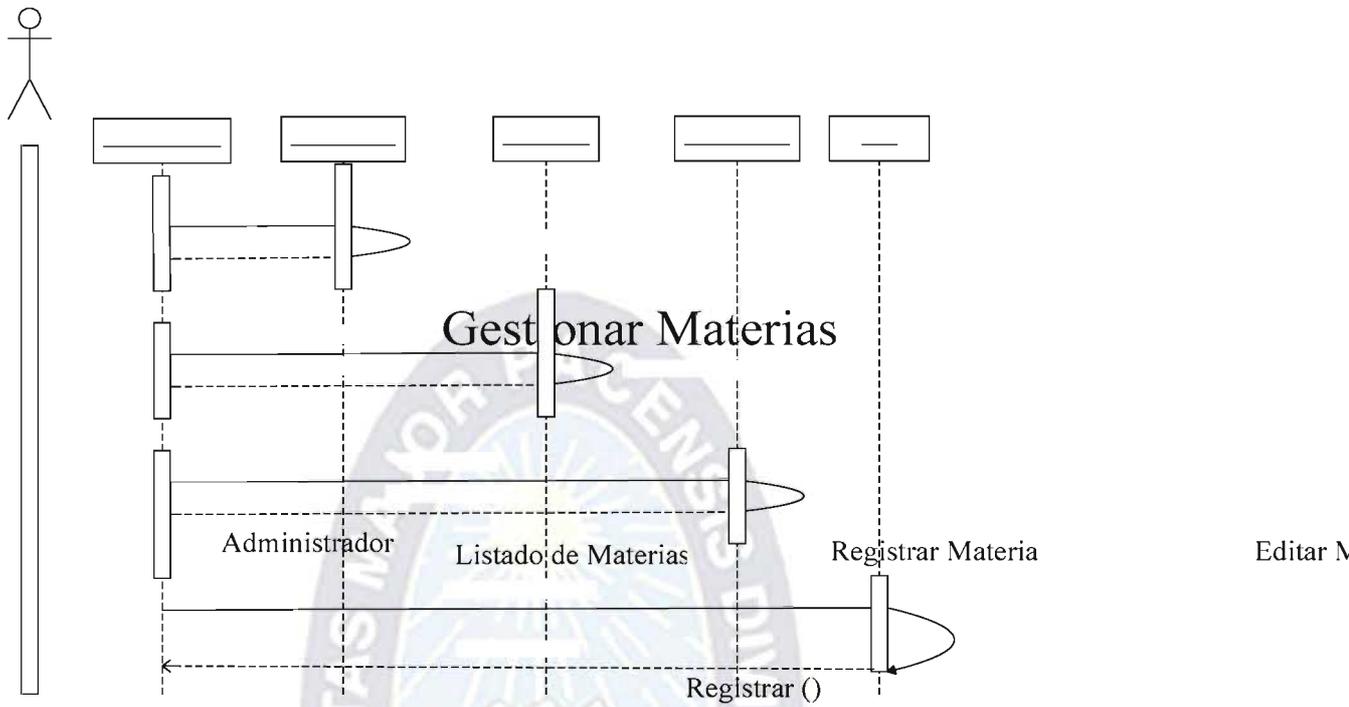


Figura 3.7: Diagrama de Secuencia Gestionar Materias

Fuente: Elaboración Propia

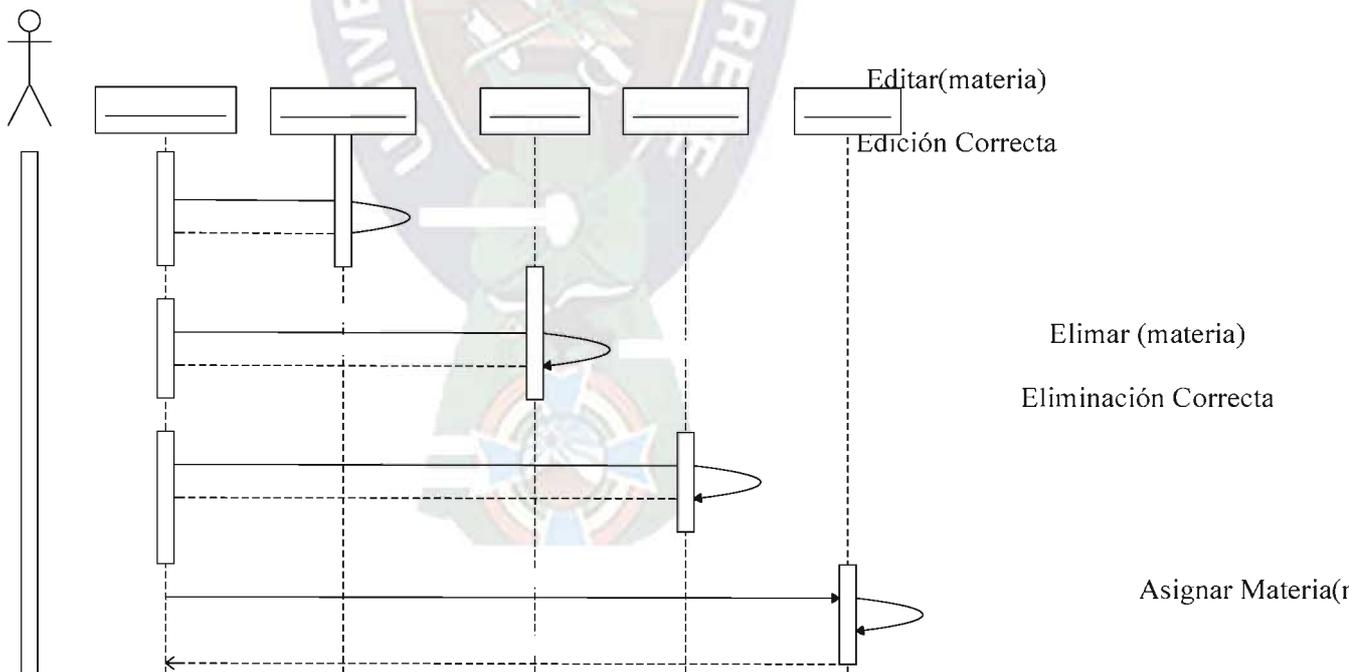


Figura 3.8: Diagrama de Secuencia Gestionar Docentes

Fuente: Elaboración Propia

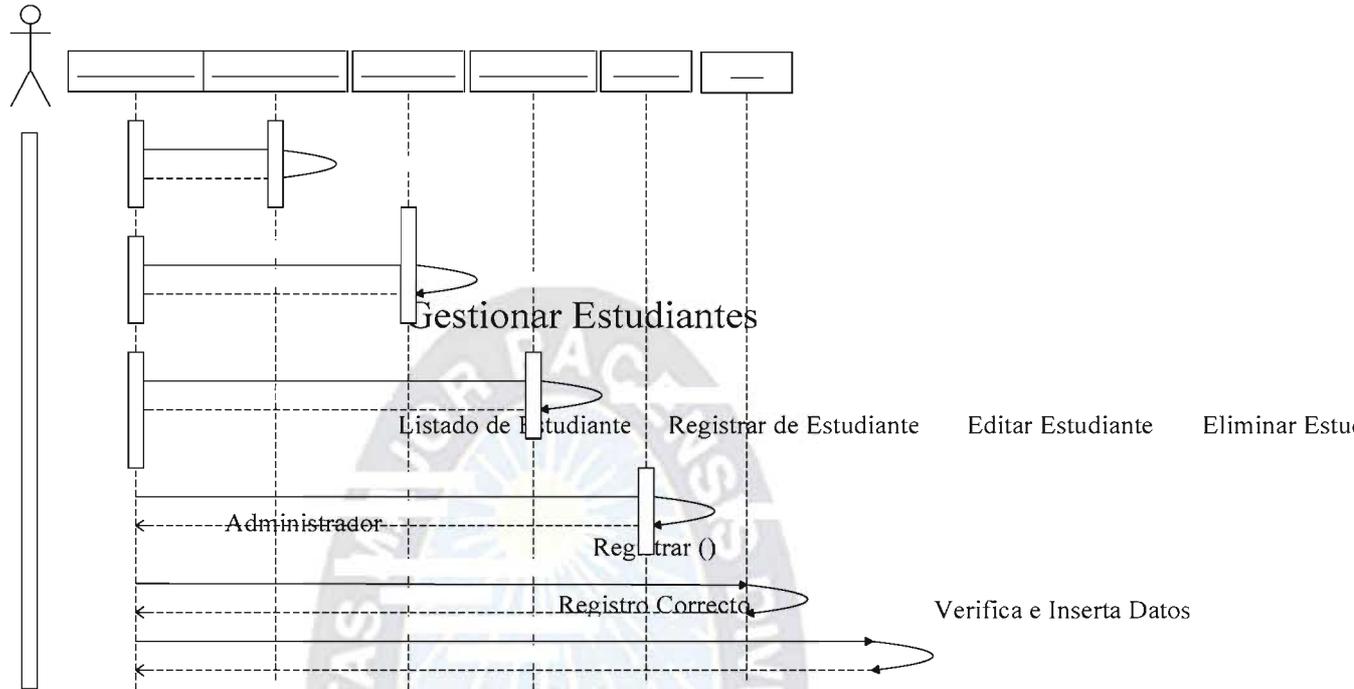


Figura 3.9: Diagrama de Secuencia Gestionar Estudiantes

Fuente: Elaboración Propia

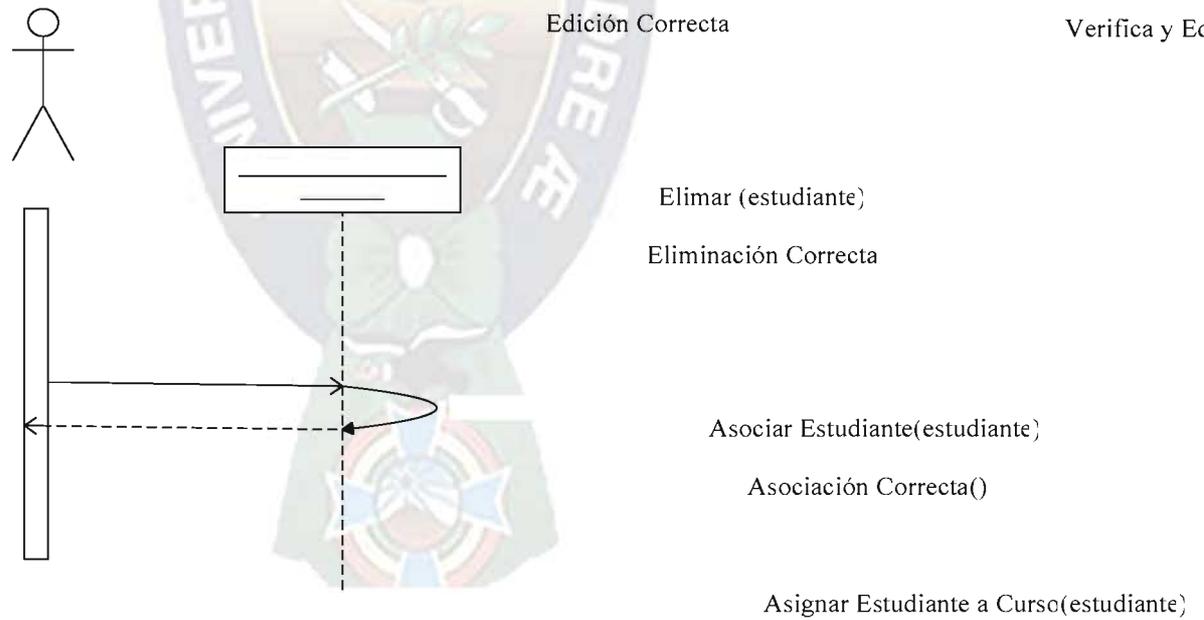


Figura 3.10: Diagrama de Secuencia Revisar Comunicados

Fuente: Elaboración Propia

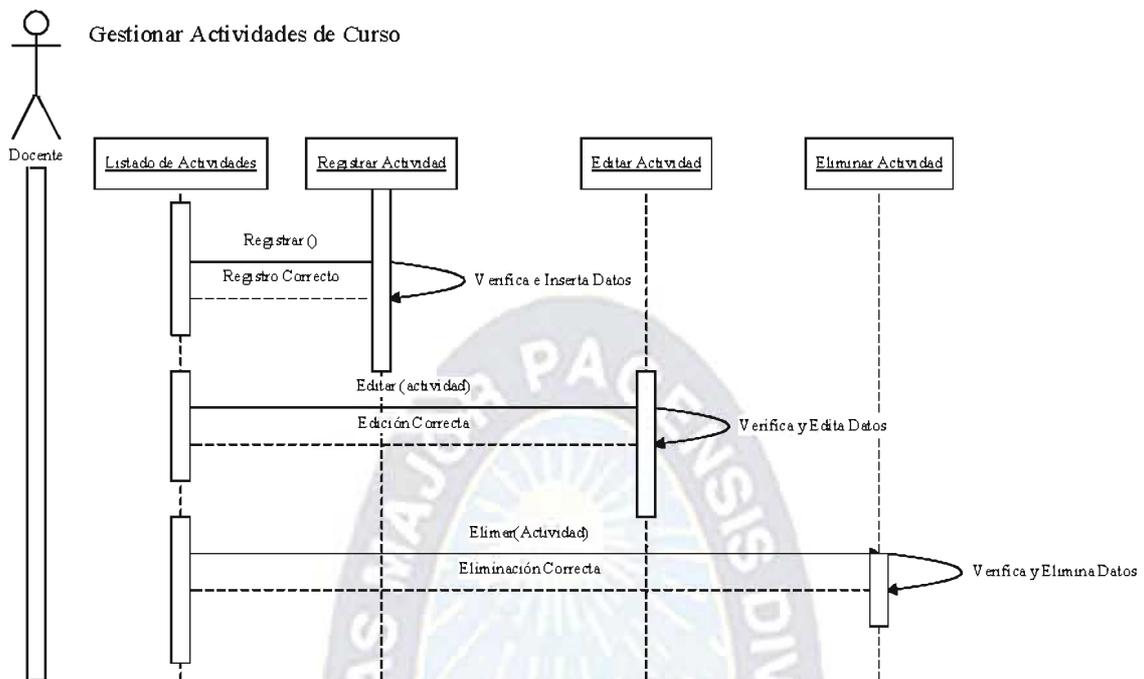


Figura 3.11: Diagrama de Secuencia Gestionar Actividades de Curso

Fuente: Elaboración Propia

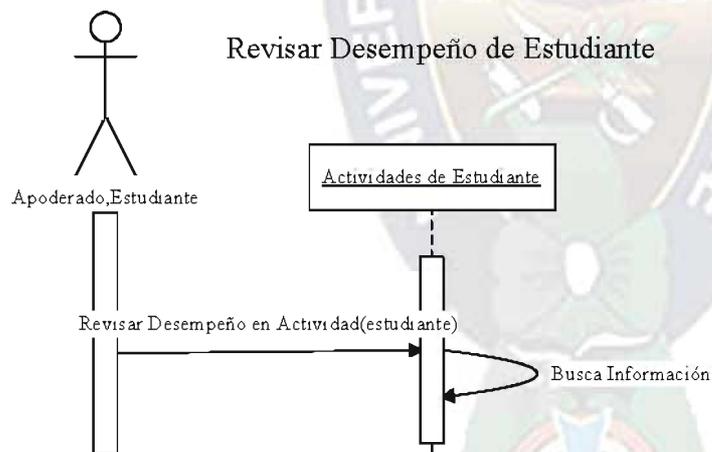


Figura 3.12: Diagrama de Secuencia Revisar Desempeño de Estudiante

Fuente: Elaboración Propia

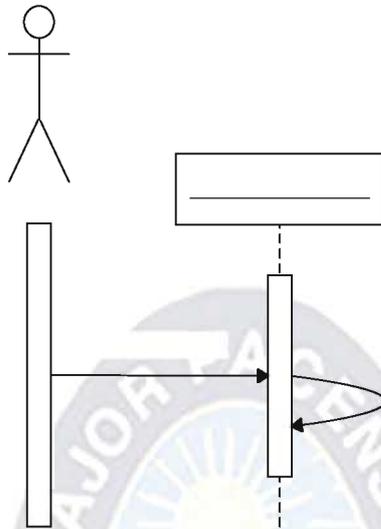


Figura 3.13: Diagrama de Secuencia Revisar Actividades de Curso

Fuente: Elaboración Propia

3.2.7 Diagramas de Actividades

Ahora se representaran algunos diagramas de actividades para conceptualizar los procesos vistos anteriormente y poder describirlos de una manera explicita para el desarrollo del sistema (Ver Figuras 19, 20,21 ,22).

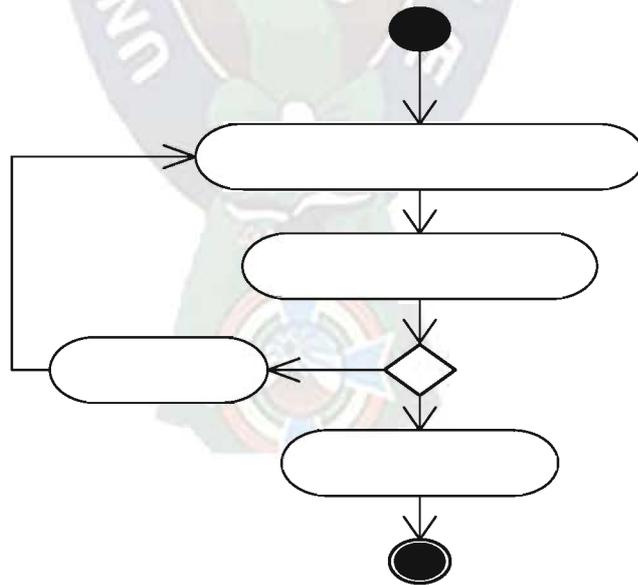


Figura 3.14: Diagrama de Actividad para el Ingreso al Sistema

Fuente: Elaboración Propia

Revisar

Activ

Docente,Apoderado,Estudiante

Revisar Tareas()

Los siguientes tres diagramas de actividades describen el proceso general de la creación, edición y eliminación de un registro.

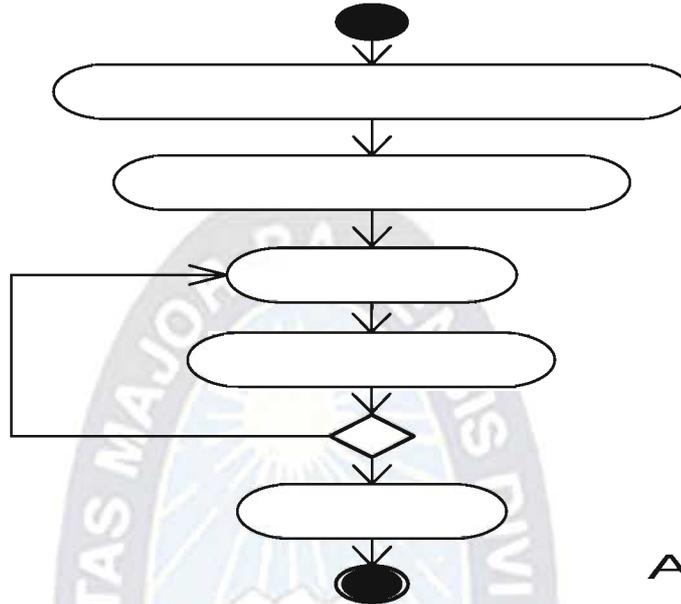


Figura 3.15: Diagrama de Actividad para la Creación de un Registro

Fuente: Elaboración Propia

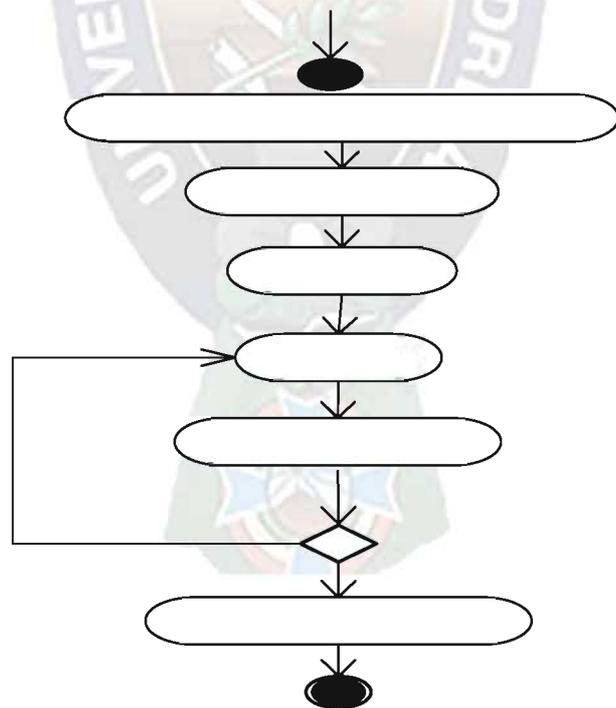


Figura 3.16: Diagrama de Actividad para la Edición de un Registro

Fuente: Elaboración Propia

Abrir la opción

Selecciona

Intro

Verific

[No
Correctos]

Gu

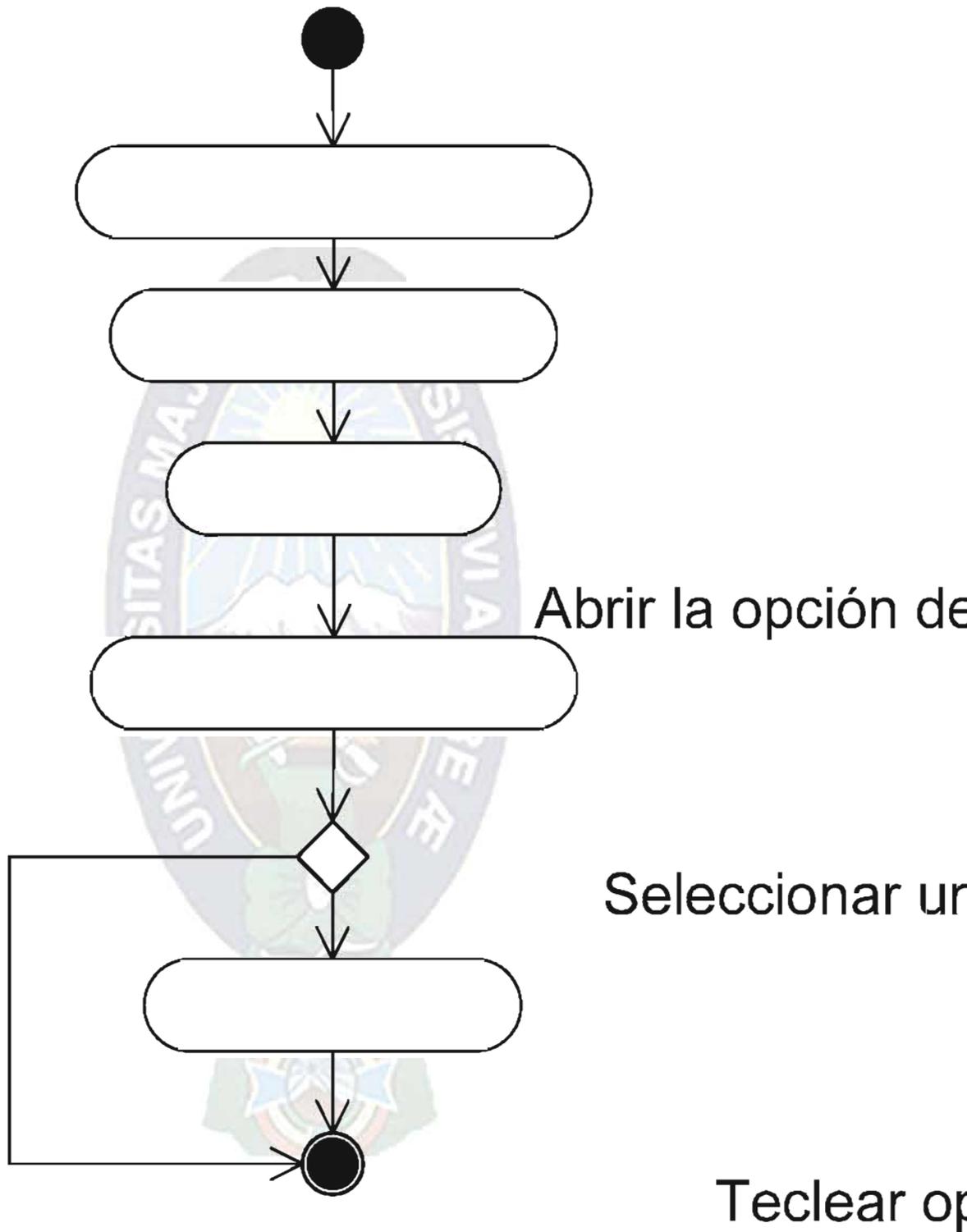


Figura 3.17: Diagrama de Actividad para la Eliminación de un Registro

Fuente: Elaboración Propia

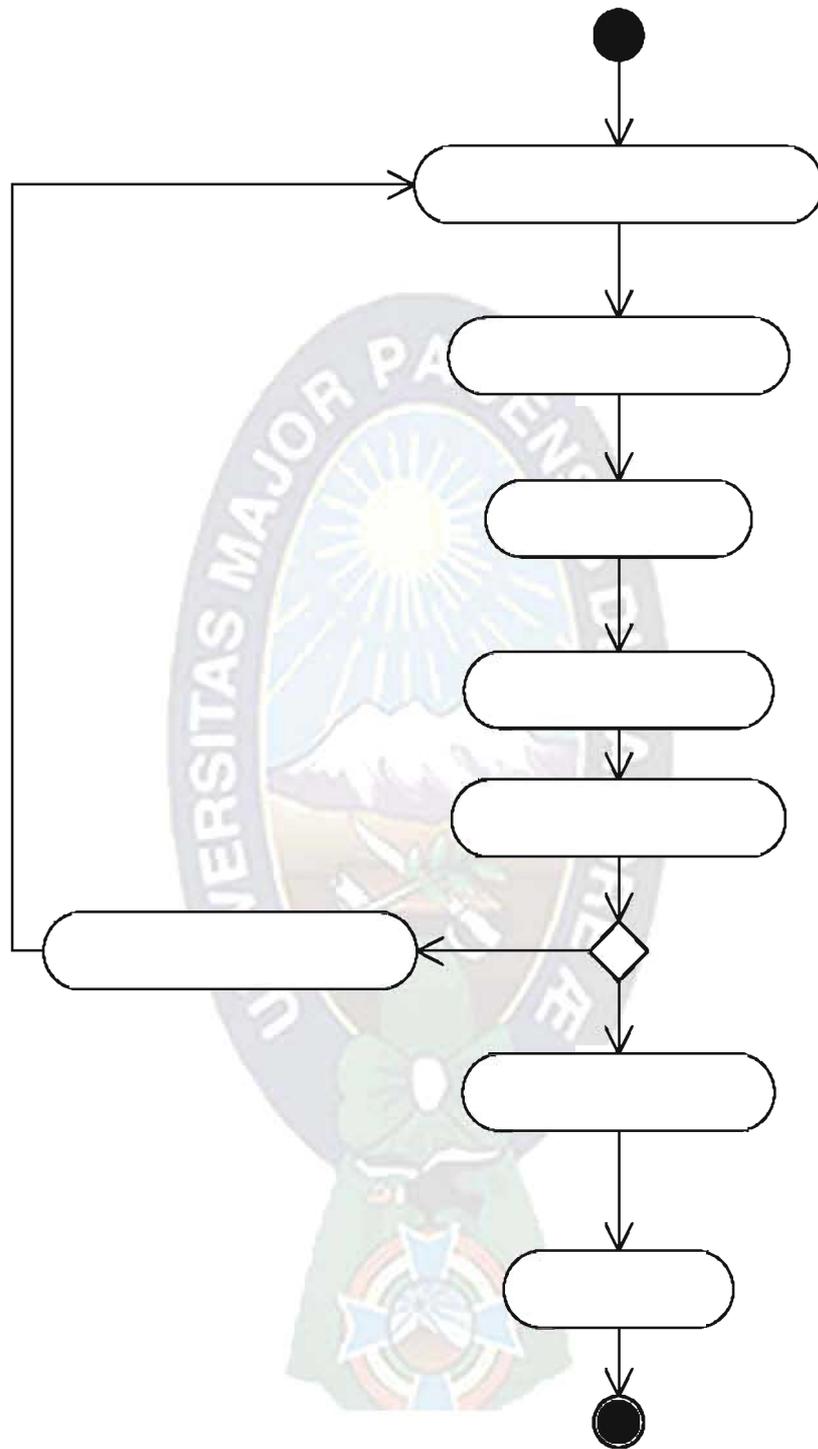


Figura 3.18: Diagrama de Actividad para el Envío de Comunicado

Fuente: Elaboración Propia

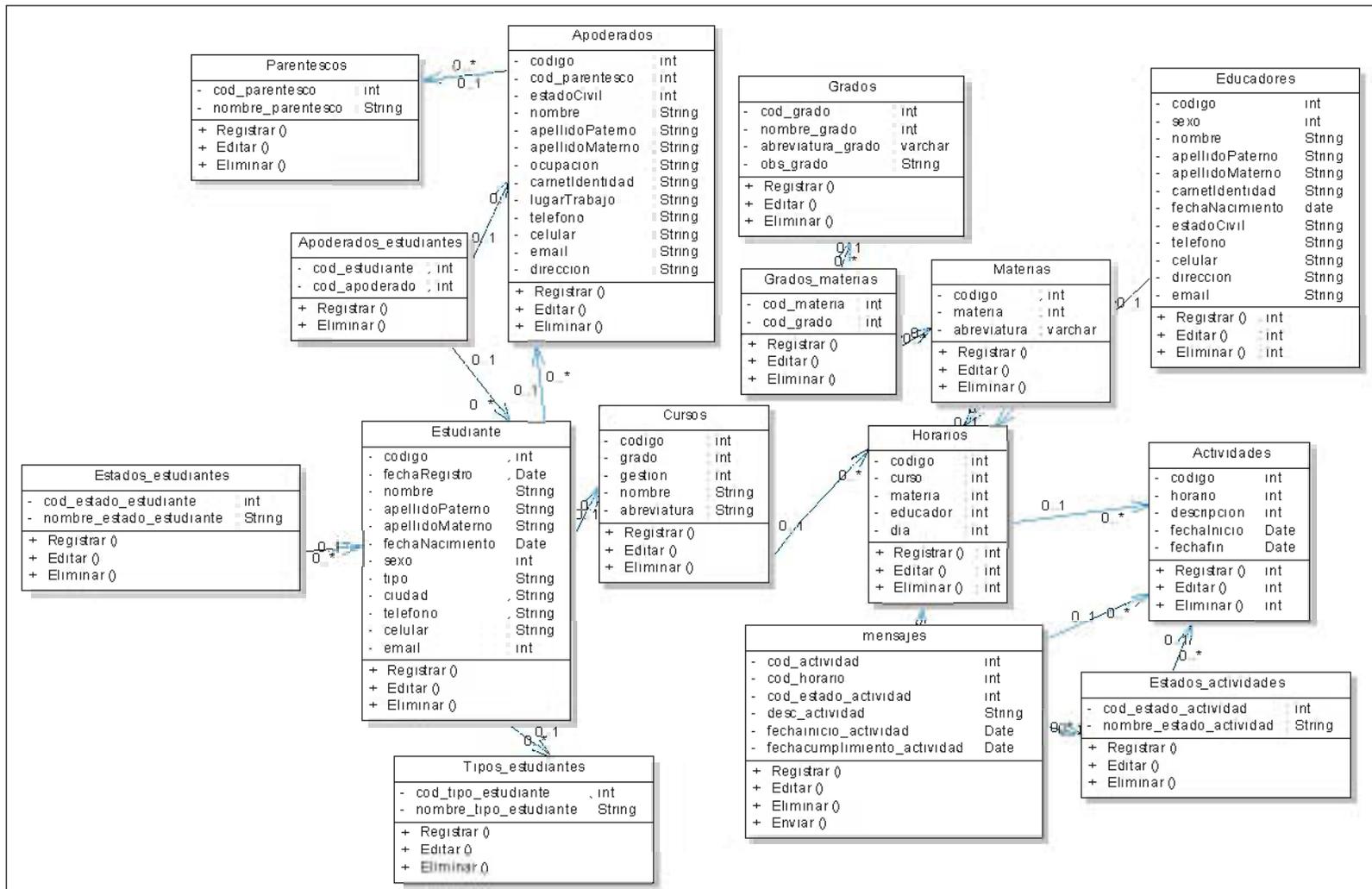


Figura 3.19: Diagrama de Clases

Fuente: Elaboración Propia

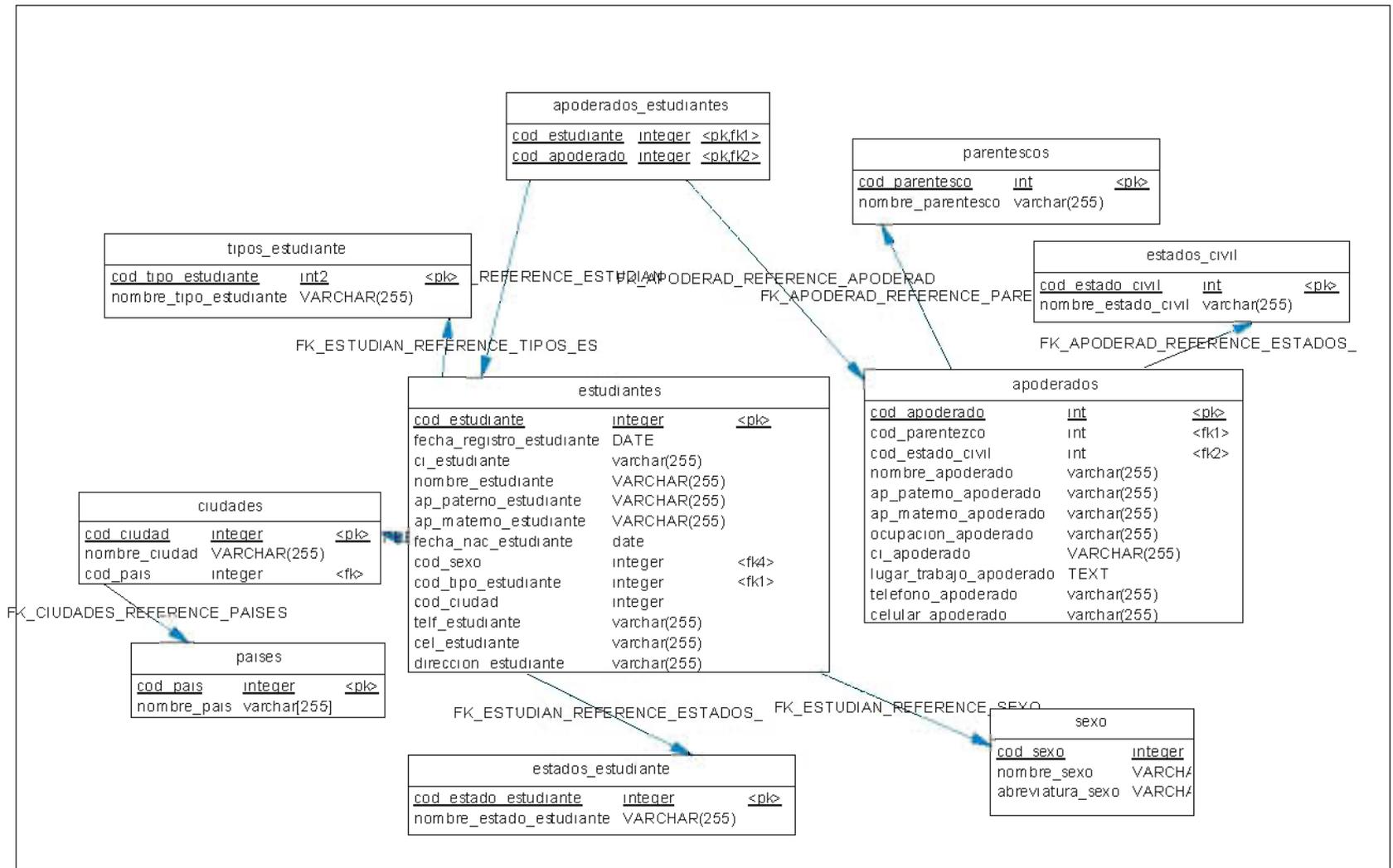


Figura 3.20: Diagrama Relacional Parte I

Fuente: Elaboración Propia

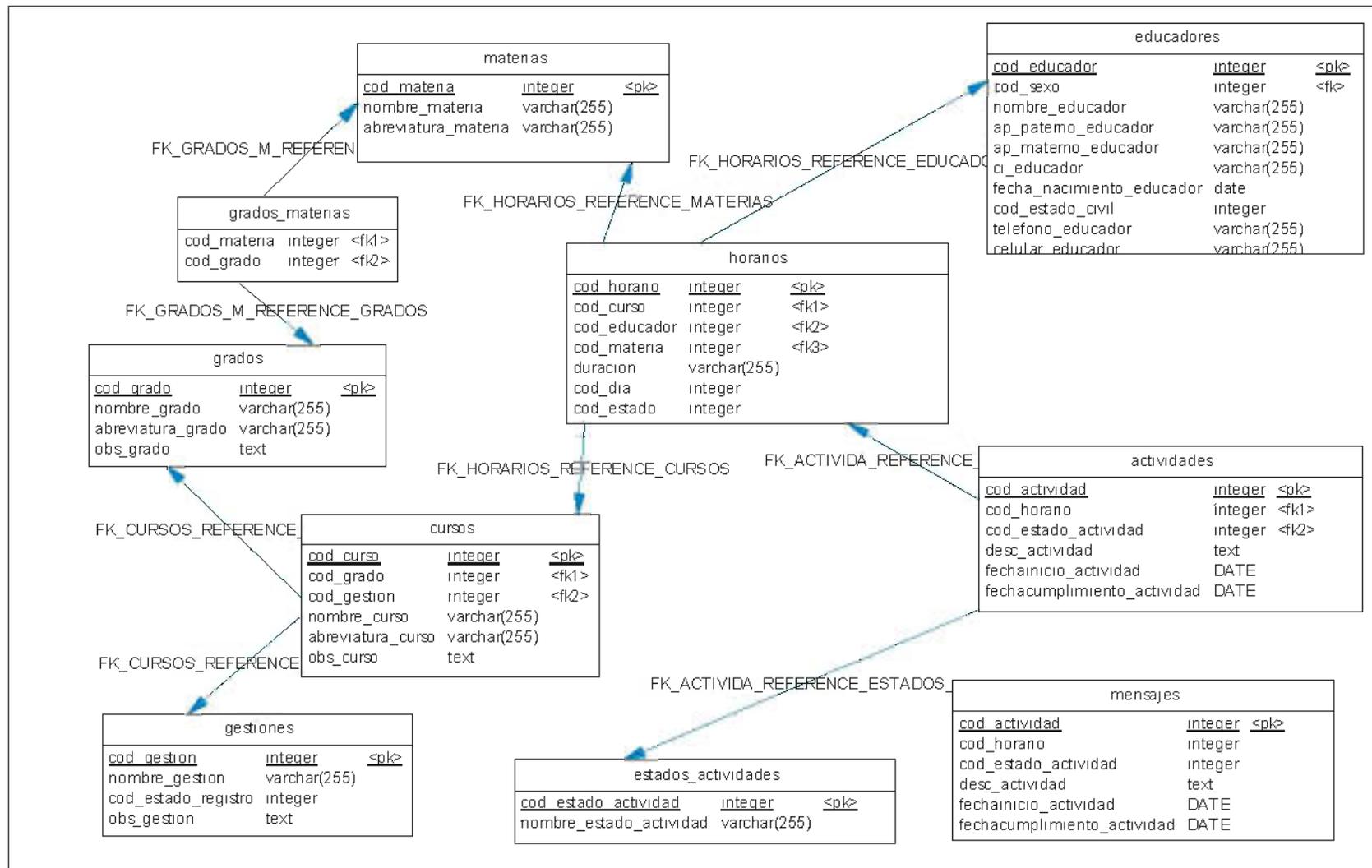


Figura 3.21: Diagrama Relacional Parte II

Fuente: Elaboración Propia

CAPÍTULO IV

IMPLEMENTACIÓN Y PRUEBAS

4.1 CONSTRUCCIÓN DEL SISTEMA

En la construcción del sistema se utilizó JSF (Java Server Faces), para ejemplificar el trabajo con este framework se describirá un ABM (altas, bajas, modificaciones) básico en el cual se mostrará la forma de trabajo con el framework.

Para tener funcionalidad sobre una estructura básica de datos, este framework trabaja con 4 Archivos.

- El Bean que viene a ser la clase con sus respectivos atributos donde se especifican todos los campos de una tabla.
- Un ManagedBean es el archivo que nos permite realizar todas las operaciones de gestión de datos como ser la adición, edición, eliminación y cualquier otro proceso que pueda existir.
- Los archivos jsf son aquellos que proporcionan la interfaz sobre la cual el usuario realiza sus peticiones al sistema esta extensión tiene sus propias etiquetas que reemplazan al html.
- El archivo de configuración faces-config.xml, donde se describen los ManagedBeans y las reglas de navegación para todos los componentes del sistema.

A continuación se ve la estructura de archivos mencionados anteriormente.

En la Figura. 4.1 se muestra el bean de una clase en particular en este ejemplo veremos la clase *materias*.

Bean
<pre>package com cofar bean, import java util Date, public class Materias extends AbstractBean (private String codMateria="", private String nombreMateria="", private String abreviaturaMateria="", private EstadoReferencial estadoRegistro=new EstadoReferencial(), public Materias() () public String getCodMateria() (return codMateria,) public void setCodMateria(String codMateria) (this codMateria = codMateria,) public String getNombreMateria() (return nombreMateria,) public void setNombreMateria(String nombreMateria) (this nombreMateria = nombreMateria,) public String getAbreviaturaMateria() (return abreviaturaMateria,) public void setAbreviaturaMateria(String abreviaturaMateria) (this abreviaturaMateria = abreviaturaMateria,) public EstadoReferencial getEstadoRegistro() (return estadoRegistro,) public void setEstadoRegistro(EstadoReferencial estadoRegistro) (this estadoRegistro = estadoRegistro,)) }</pre>
Figura. 4.1 Bean de Materias Fuente: Elaboración Propia

El siguiente fragmento de código muestra el manejo de nuestra Clase Materias..

ManagedBean
<pre>import com.cofar.bean.Materias; import com.cofar.util.Util; import java.sql.Connection; import java.sql.PreparedStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement; import java.util.ArrayList; import java.util.Iterator; import java.util.List; import javax.faces.event.ValueChangeEvent; import javax.faces.model.SelectItem; public class ManagedMaterias { public void cargarMaterias() { try { String sql="select cod_materia,nombre_materia,abreviatura_materia,cod_estado_registro" + " from materias"; if(!materiasbean.getEstadoRegistro().getCodigoEstadoRegistro().equals("") && !materiasbean.getEstadoRegistro().getCodigoEstadoRegistro().equals("3")){ sql+=" where cod_estado_registro="+materiasbean.getEstadoRegistro().getCodigoEstadoRegistro(); } sql+=" order by nombre_materia asc"; System.out.println("cargar "+sql); setCon(Util.openConnection(getCon())); Statement st=getCon().createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY); ResultSet rs=st.executeQuery(sql); rs.last(); int rows=rs.getRow(); MateriasList.clear(); rs.first(); String cod=""; for(int i=0,i<rows,i++){ Materias bean=new Materias(); bean.setCodigoMateria(rs.getString(1)); bean.setNombreMateria(rs.getString(2)); bean.setAbreviaturaMateria(rs.getString(3)); //bean.setCodigoEstadoRegistro(rs.getString(7)); cod=rs.getString(4); cod=(cod==null)?""+cod; System.out.println("st xxx "+cod); cargarEstadoRegistro(cod,bean); MateriasList.add(bean); rs.next(); } if(rs!=null){ rs.close(); st.close(); } } catch (SQLException e) { e.printStackTrace(); } } }</pre>
Figura. 4.2 ManagedBean de Materias
Fuente: Elaboración Propia

En la figura 4.3 corresponde a un fichero .jsf que interactúa directamente con el usuario.

```

.JSF

<h form id="form1" >
  <div align="center">

    <br>
    <h outputText styleClass="outputTextTitulo" value="LISTADO DE MATERIAS" /><br><br>
    <h outputText value="Estado" styleClass="outputTextTitulo"/>
    <h selectOneMenu value="#(ManagedMaterias materiasbean estadoRegistro codEstadoRegistro)" styleClass="inputText"
      valueChangeListener="#(ManagedMaterias changeEvent)" >
      <f selectItems value="#(ManagedEstadosReferenciales estadosReferenciales)" />
      <a4j support event="onChange" reRender="dataCadenaCliente" />
    </h selectOneMenu>
    <br>
    <br>

    <rich dataTable value="#(ManagedMaterias materiasList)" var="data" id="dataCadenaCliente"
      onRowMouseOut="this style backgroundColor=#FFFFFF,"
      onRowMouseOver="this style backgroundColor=#CCDFFA","
      headerClass="headerClassACliente"
      columnClasses="tituloCampo"
      rows="10">
      <h column>
        <f facet name="header">
          <h outputText value="" />
        </f facet>
        <h selectBooleanCheckbox value="#(data checked)" />
      </h column>
      <h column>
        <f facet name="header">
          <h outputText value="Materia" />
        </f facet>
        <h outputText value="#(data.nombreMateria)" />
      </h column>
      <h column>
        <f facet name="header">
          <h outputText value="Abreviatura" />
        </f facet>
        <h outputText value="#(data.abreviaturaMateria)" />
      </h column>
      <h column>
        <f facet name="header">
          <h outputText value="Estado" />
        </f facet>
        <h outputText value="#(data.estadoRegistro.nombreEstadoRegistro)" styleClass="tituloCampo" />
      </h column>
    </rich dataTable>
    <ws dataScroller forDataTable="dataCadenaCliente" />
    <br>

    <h commandButton value="Registrar" styleClass="btn" action="#(ManagedMaterias Guardar)" />
    <h commandButton value="Editar" styleClass="btn" action="#(ManagedMaterias actionEditar)" onclick="return
editarItem('form1 dataCadenaCliente');" />
    <h commandButton value="Eliminar" styleClass="btn" action="#(ManagedMaterias actionEliminar)" onclick="return
eliminarItem('form1 dataCadenaCliente');" />
  </div>
  <!--cerrando la conexion-->
  <h outputText value="#(ManagedMaterias closeConnection)" />
</h form>

```

Figura. 4.3: Navegador Principal .jsf

Fuente: Elaboración Propia

Por último se ve el archivo faces-config.xml donde se describen las reglas de navegación

```
faces-config.xml

<!-- ***** Materias ***** -->

<managed-bean>
  <managed-bean-name>ManagedMaterias</managed-bean-name>
  <managed-bean-class>com.cofar.web.ManagedMaterias</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

<!-- ***** Fin Materias ***** -->

<!-- ***** Materias ***** -->

<navigation-case>
  <from-outcome>
    actionAgregarMaterias
  </from-outcome>
  <to-view-id>/materias/agregar_materia.jsp</to-view-id>
</navigation-case>

<navigation-case>
  <from-outcome>
    cancelarMaterias
  </from-outcome>
  <to-view-id>/materias/navegador_materias.jsp</to-view-id>
</navigation-case>

<navigation-case>
  <from-outcome>
    navegadorMaterias
  </from-outcome>
  <to-view-id>/materias/navegador_materias.jsp</to-view-id>
</navigation-case>

<navigation-case>
  <from-outcome>
    ActionEditarMaterias
  </from-outcome>
  <to-view-id>/materias/modificar_materia.jsp</to-view-id>
</navigation-case>

<navigation-case>
  <from-outcome>
    eliminarMaterias
  </from-outcome>
  <to-view-id>/materias/eliminar_materias.jsp</to-view-id>
</navigation-case>
</navigation-rule>
```

Figura. 4.4. Archivo de Configuración para Reglas de Navegación

Fuente: Elaboración propia

4.2 INTERFACES DEL SISTEMA

En la Figura. 4.5 se tiene la pantalla de ingreso al sistema, en la cual solicita el usuario y contraseña para su ingreso.



Figura 4.5 Pantalla de Ingreso al Sistema

Fuente: Elaboración Propia

Una vez que el usuario ingresa al sistema se despliega las opciones del menú de acuerdo al perfil que tenga el usuario:

- Administrador
- Profesor / Educador
- Apoderado
- Estudiante

El ejemplo siguiente se encuentra con el perfil de Administrador.

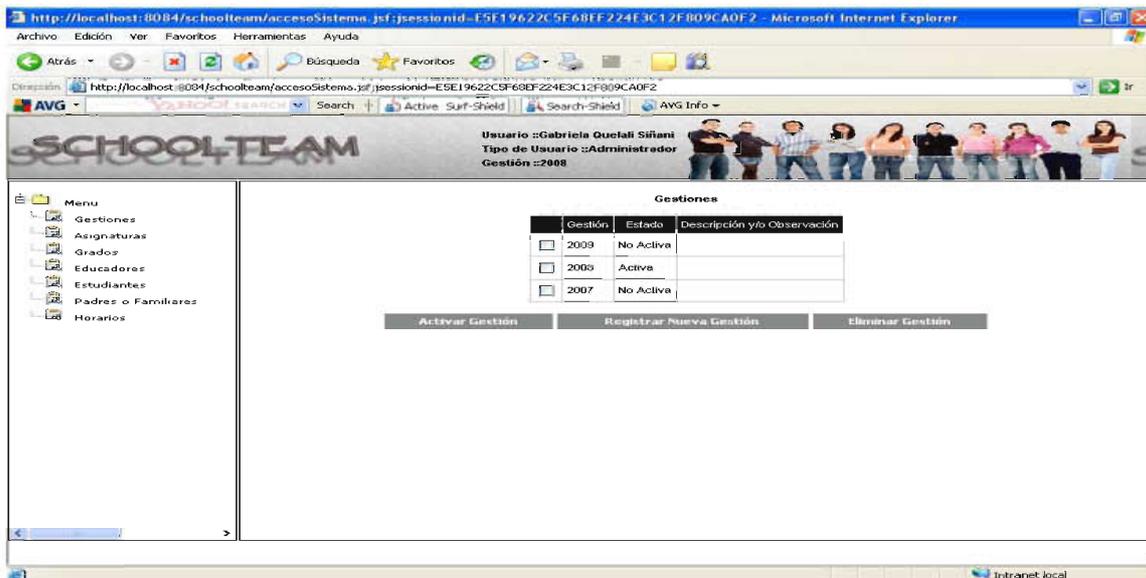


Figura. 4.6 Menú de Opciones – Perfil Administrador

Fuente: Elaboración Propia

Para acceder a las opciones del menú disponemos de un árbol ubicado en la parte izquierda de la pantalla que es construido de acuerdo al perfil con el que se ingresa, en el ejemplo se ingreso con el perfil de Administrador. Ver Figura. 4.7.

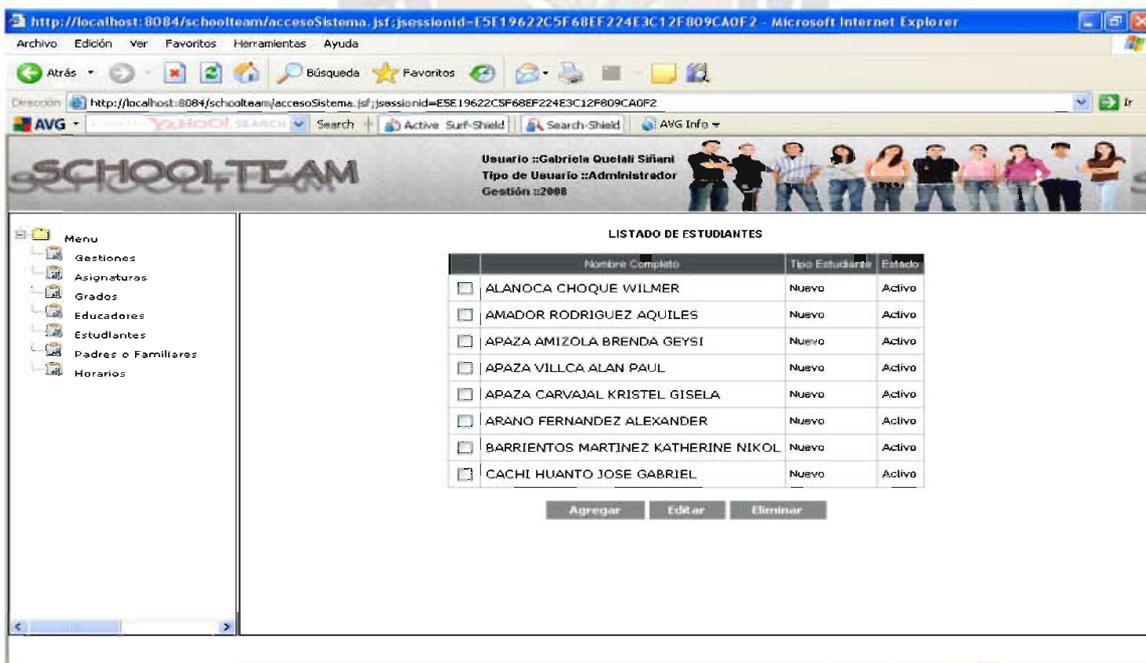


Figura. 4.7 Listado de Estudiantes – Perfil Administrador

Fuente: Elaboración propia

En el listado de Estudiantes, cuenta con las siguientes opciones: agregar, editar y eliminar.

- Agregar. Se utiliza esta opción para la creación de un nuevo Estudiante, esto es realizado a través de la opción **Registrar** del Listado de Estudiantes. Ver Figura. 4.8
- Editar. Esta opción sirve para la edición de datos del Estudiante, esta acción es realizada a través de la opción **Editar** del Listado de Estudiantes. Ver Figura. 4.9
- Eliminar. La opción eliminar nos permite dar de baja del sistema a uno o más estudiantes, esto lo realizamos a través de la opción **Eliminar** del Listado de Estudiantes.

http://localhost:8084/schoolteam/accesoSistema.jsf;jsessionid=E5E19622C5F68EF224E3C12F809CA0F2 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://localhost:8084/schoolteam/accesoSistema.jsf;jsessionid=E5E19622C5F68EF224E3C12F809CA0F2 Ir

AVG

SCHOOLTEAM Usuario ::Gabriela Quelali Siñani
Tipo de Usuario ::Administrador
Gestión ::2008

REGISTRO DE ESTUDIANTE

Introduzca Datos

Nombre Apellido Paterno Apellido Materno

Sexo Fecha Nacimiento (dd/mm/aaaa) Lugar de Nacimiento

Carnet de Identidad Teléfono Celular ENTEL

Email Dirección

Referencia de Emergencia

Observaciones

Datos para la Institución

Tipo de Estudiante Estado de Estudiante Grado

Curso Nombre de Usuario Contraseña

Estado de Servicio

Guardar Cancelar

NetBeans IDE 5.5.1 - schoolteam intranet local

Figura. 4.8 Registro de Estudiante – Perfil Administrador

Fuente: Elaboración Propia

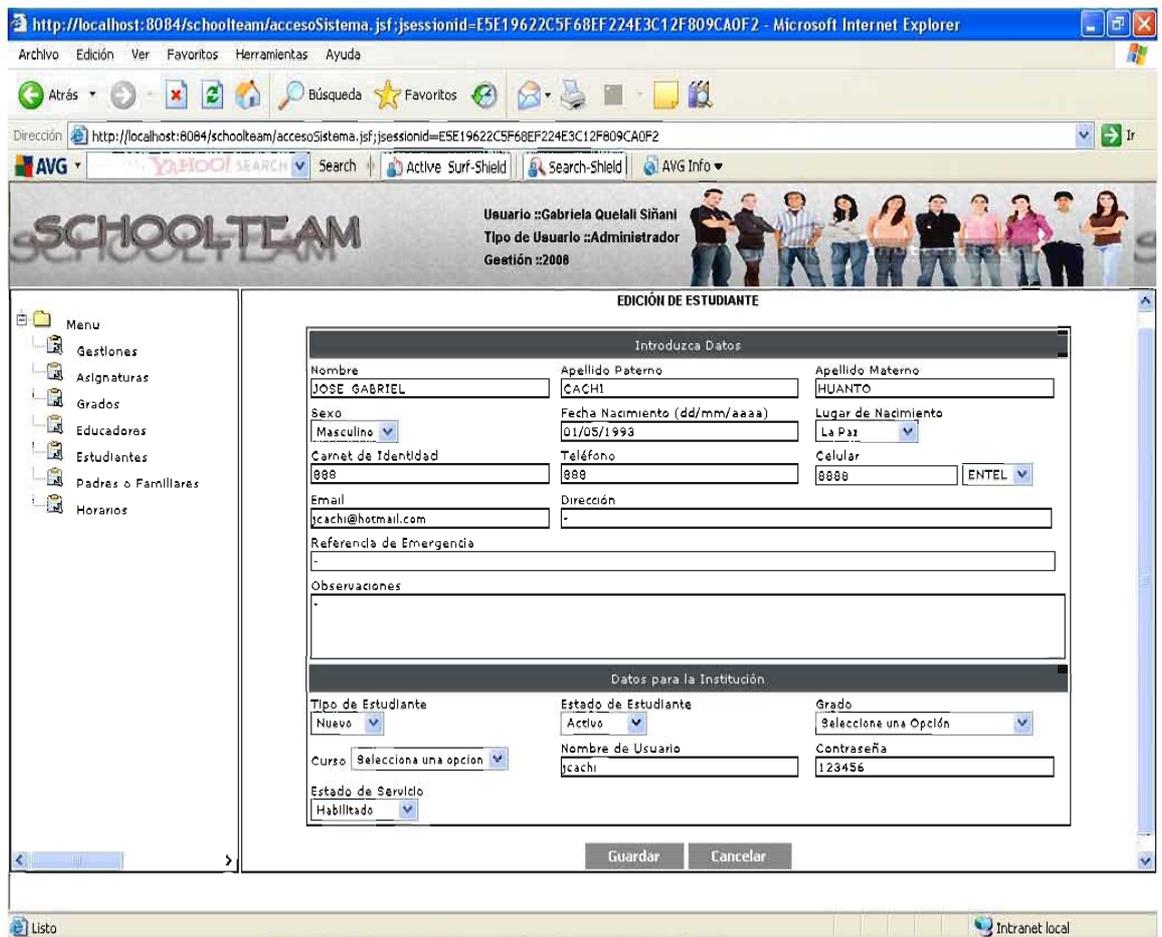


Figura. 4.9 Edición de Estudiante – Perfil Administrador

Fuente: Elaboración Propia

4.3 PRUEBAS

Para verificar que los procesos del sistema se efectúan correctamente utilizaremos las pruebas de caja negra.

-Prueba 1. Registro de Asignatura o Materia

En la Figura 4.10 se muestra el listado de materias o asignaturas que se imparten en la unidad educativa.

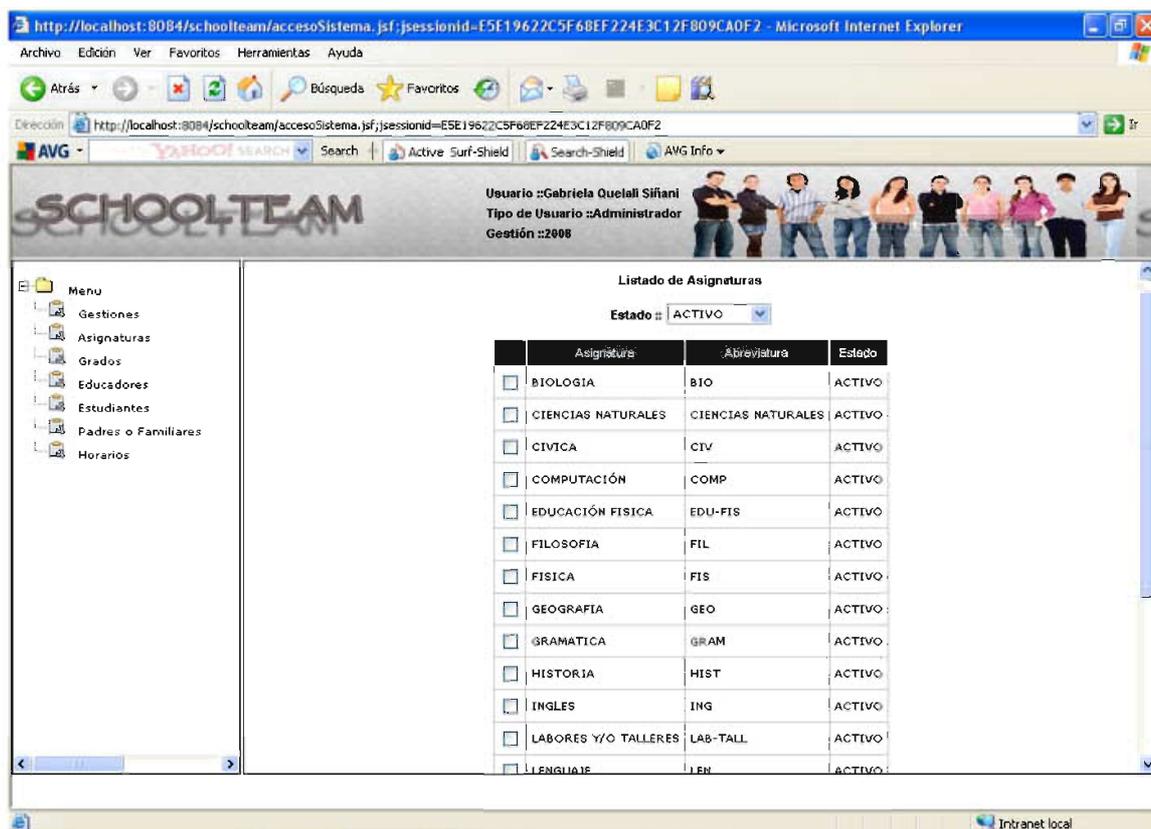


Figura. 4.10 Listado de Asignaturas – Perfil Administrador

Fuente: Elaboración Propia

Se realizara la prueba de registro de materia o asignatura.

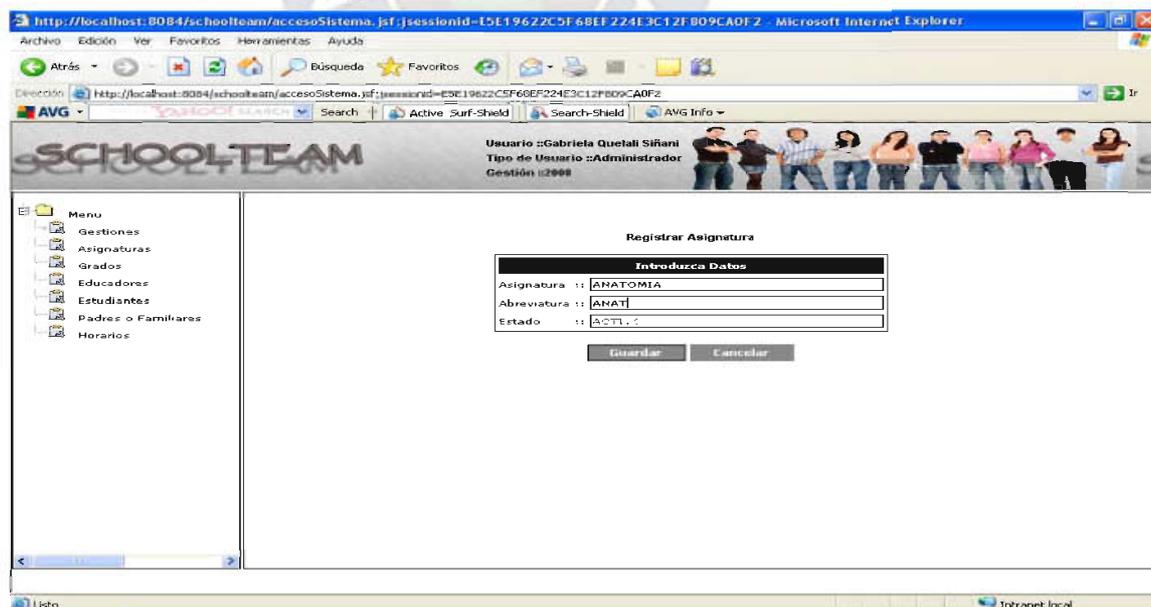


Fig. 4.11 Registro de Asignatura – Perfil Administrador

Fuente: Elaboración Propia

Una vez que se ingresan los datos en el registro se procede a guardar los datos, una vez ocurrido el sistema nos envía al navegador principal de materias donde podemos revisar si efectivamente lo que introducimos quedo registrado en la base de datos.

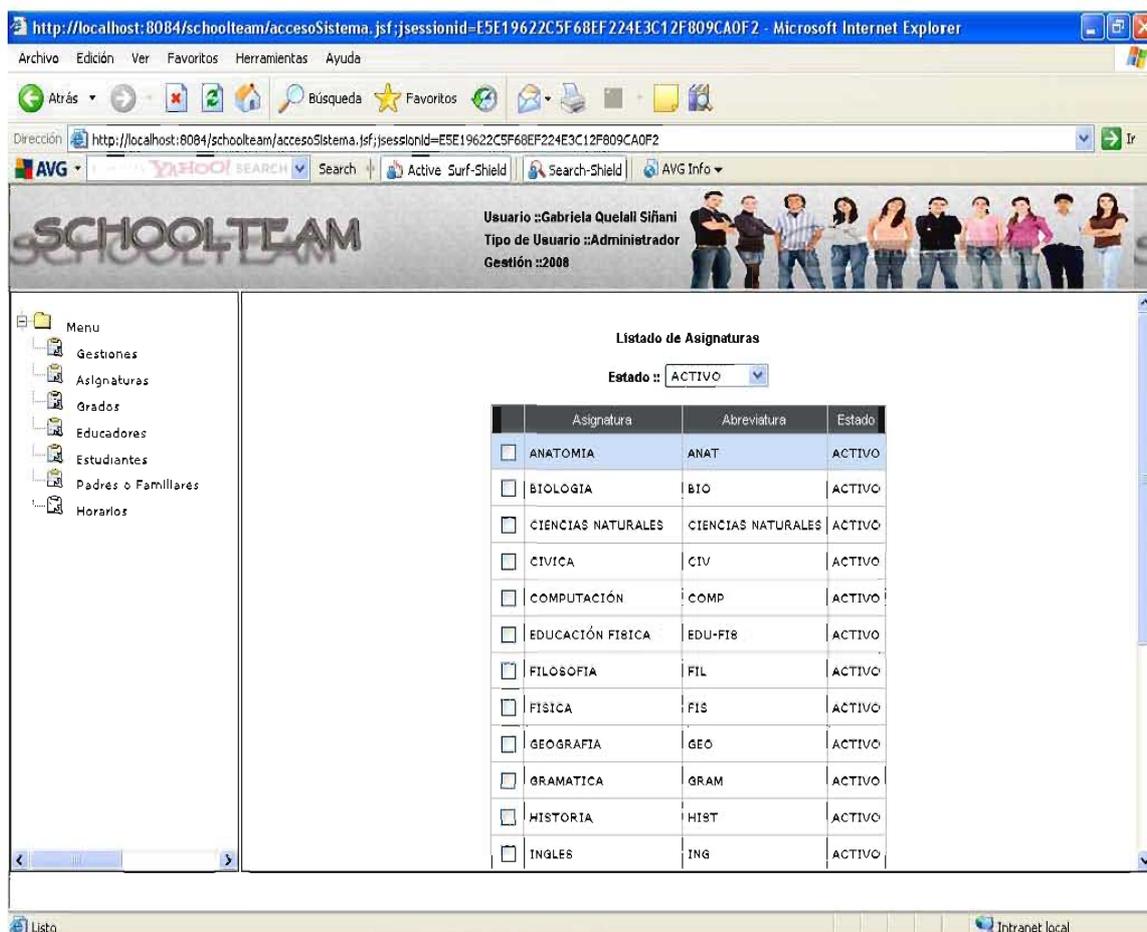


Figura. 4.12 Listado de Asignaturas – Perfil de Administrador

Fuente: Elaboración Propia

En la figura 4.12 podemos revisar que efectivamente se guardaron los datos que introducimos.

-Prueba 2. Asignación y Revisión de Actividades de un Curso

Para esta prueba se ingresa con el Perfil de Educador, el educador cuenta con la opción Asignar Actividades donde se le despliega los cursos en los que tiene asignado asignaturas que impartir (Ver Figura 4.13).

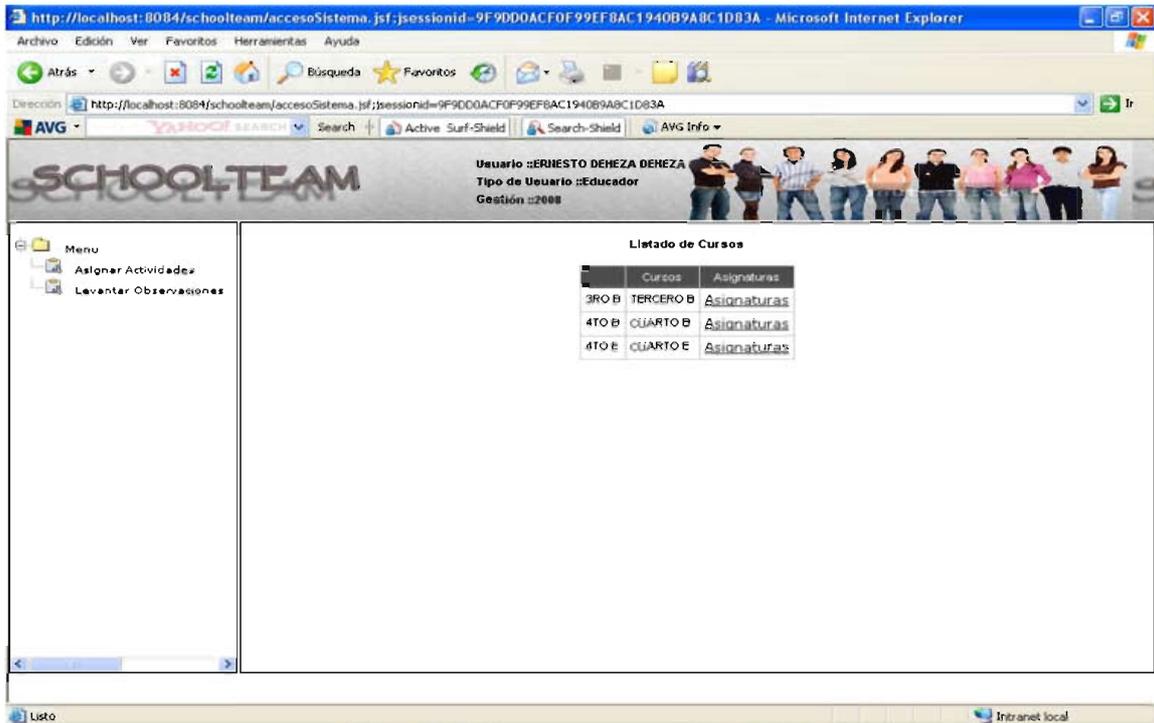


Figura. 4.13 Cursos del Educador – Perfil de Educador

Fuente: Elaboración Propia

En la parte derecha del listado de cursos se tiene la opción asignaturas, ingresando al link tenemos el listado de las asignaturas de las que el Educador es responsable en un determinado Curso (Ver Figura 4.14).

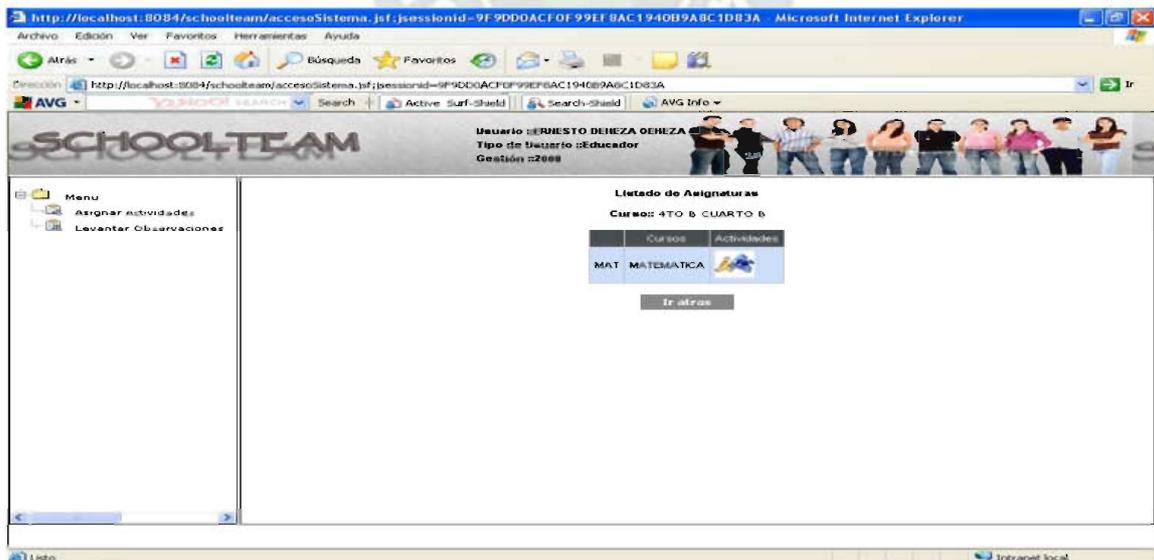


Figura. 4.14 Asignaturas impartidas por el Educador – Perfil de Educador

Fuente: Elaboración Propia

Como se ve en la pantalla anterior, se tiene un link de actividades, ingresando al link se tiene un listado de las actividades propias del curso y de la asignatura seleccionada.

The screenshot shows a web browser window displaying the 'SCHOOLTEAM' application. The user is identified as ERNESTO DEHEZA DEHEZA, an Educator, with a management year of 2008. The page title is 'Listado de Actividades' for the course '4TO. B. CUARTO B' and subject 'MAT-MATEMATICA'. A table lists two activities:

	Fecha Asignación	Fecha Límite	Actividad	Estado	Observar Estudiantes
<input type="checkbox"/>	2008-07-02	2008-07-09	Examen de Casos de Factorización	Confirmado	Observar Estudiantes
<input type="checkbox"/>	2008-07-01	2008-07-04	estudiar casos de factorizacion	Cerrada	Observar Estudiantes

Below the table are buttons for 'Ir atras', 'Agregar', 'Editar', and 'Eliminar'. A left sidebar contains a 'Menu' with options 'Asignar Actividades' and 'Levantar Observaciones'.

Figura. 4.15 Actividades de Asignatura – Perfil de Educador

Fuente: Elaboración Propia

En la pantalla anterior se tiene las opciones de Agregar, Editar, Eliminar que permiten gestionar las actividades de una determinada asignatura dentro un curso, el educador registra todas las actividades que realiza en la asignatura que imparte, para que estas puedan ser revisadas por el estudiante y padre de familia, para así ejercer el seguimiento y control de las actividades del estudiante, a continuación se hace el registro de una nueva actividad la cual deberá verse por el padre de familia que tiene a su hijo en el curso asignada la actividad.

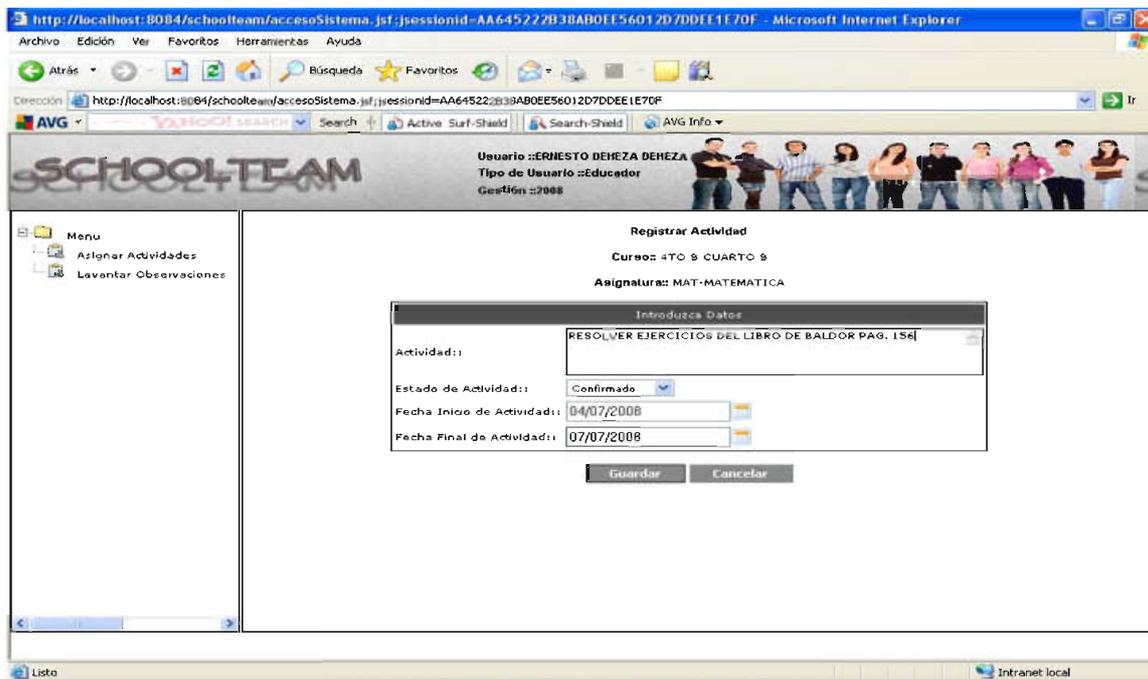


Figura. 4.16 Registro de Actividad – Perfil de Educador

Fuente: Elaboración Propia

A continuación se ingresa con el perfil de Padre de Familia o Tutor, para verificar si efectivamente él puede revisar y conocer las actividades del curso de su hijo (Ver Figura 4.16).

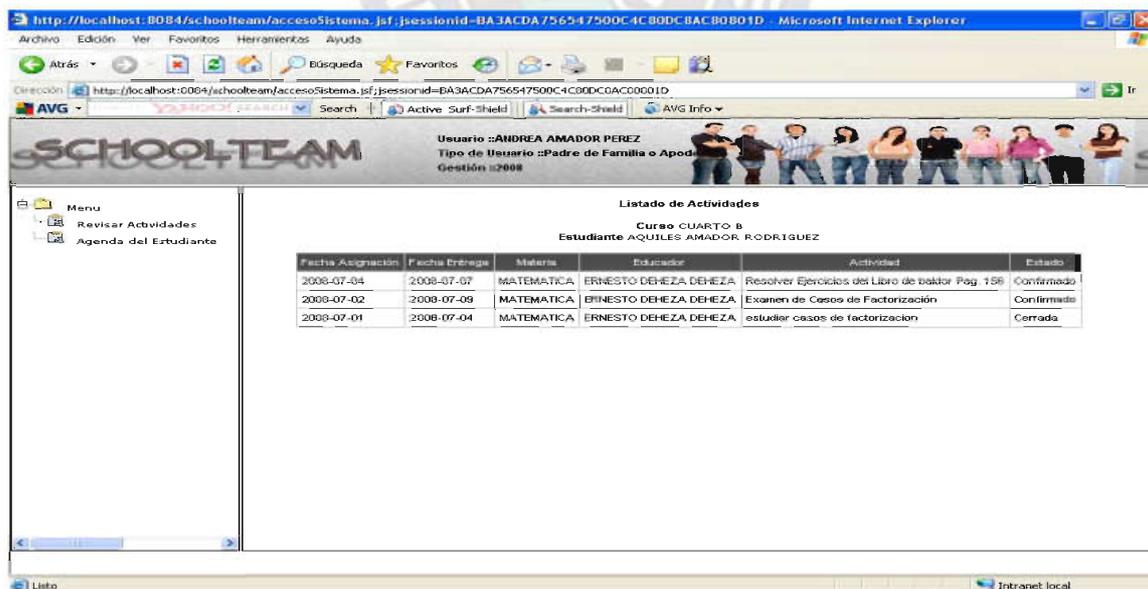


Figura. 4.17 Listado de Actividades de un Curso – Perfil de Padre de Familia o Tutor

Fuente: Elaboración Propia

4.4 PUESTA EN MARCHA

Para la puesta en marcha del presente proyecto se realizó una descripción de los requisitos mínimos de implementación.

- 1 computador donde residirán el servidor de aplicaciones y base de datos.
- 1 computador por curso, ya que el óptimo funcionamiento del sistema consiste en el registro y la comunicación de todas las actividades suscitadas en el aula.
- Capacitación al personal que operará el sistema.

4.5 CALIDAD DEL SOFTWARE

Para establecer los términos de calidad que cumple el software desarrollado, se aplicará la métrica del punto función.

4.5.1 Punto Función

En primer lugar obtendremos los factores de ponderación

Parámetro de medición	Cuenta	Factor de Multiplicación	Total
Número de entradas de usuario	20	4	80
Número de salidas de usuario	34	5	170
Número de peticiones de usuario	20	4	80
Número de archivos	2	7	14
Número de Interfaces externas	1	5	5
Total:			349

Tabla 4.1: Factores de Ponderación Punto Función

Fuente: Elaboración Propia

Ahora señalaremos los factores de complejidad asociados al punto función, donde evaluaremos estos factores en un rango de 0 a 5 de la siguiente manera:

0: Sin influencia

1: Incidental

2: Moderado

3: Medio

4: Significativo

5: Esencial

FACTORES DE COMPLEJIDAD	
¿Requiere el sistema copias de seguridad y recuperación fiables?	5
¿Se requiere comunicación de datos?	4
¿Existen funciones de procesamiento distribuido?	0
¿Es crítico el rendimiento?	4
¿Será ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?	3
¿Requiere el sistema entrada de datos interactivo?	4
¿Requiere la entrada de datos interactivo que las transiciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?	3
¿Se actualizan los archivos maestros en forma interactiva?	1
¿Son complejos las entradas, salidas, peticiones o archivos?	2
¿Es complejo el procesamiento interno?	3
¿Se ha diseñado el código para ser reutilizable?	5
¿Están incluidas en el diseño la conversión y la instalación?	5
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5

Tabla 4.2 : Factores de Complejidad

Fuente: Elaboración Propia

PFSA (Puntos de función sin ajustar) = 349

Total grado de influencia (TDI) = 54

Factor de Ajuste (VAF) = $0.65 + (0.01 \times \text{TDI}) = 1.19$

PFA = PFSA * VAF = $349 * 1.1$

4.5.2 Factores de Calidad ISO 9126

Evaluaremos el Proyecto aplicando la norma ISO 9126.

Parte 1: Funcionalidad	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Idoneidad	No aplica.
Exactitud	$X = 1$
Interoperabilidad	$X = 0.8$
Seguridad	$X = 0.9$
Cumplimiento de normas	No aplica.
Funcionalidad = $f = (1 + 0.8 + 0.9) / 3 = 0.9$	

Tabla 4.3: Factor de Calidad - Funcionalidad

Fuente: Elaboración Propia

Parte 2: Fiabilidad	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Madurez	$X = 0.85$
Recuperabilidad	$X = 0.80$
Tolerancia a fallos	$X = 0.95$
Fiabilidad = $f_i = (0.85 + 0.80 + 0.95) / 3 = 0.87$	

Tabla 4.4: Factor de Calidad - Fiabilidad

Fuente: Elaboración Propia

Parte 3: Usabilidad	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Aprendizaje	$X = 0.9$
Comprensión	$X = 0.9$
Operatividad	$X = 0.95$
$Usabilidad = u = (0.9+0.9+0.95) / 3 = 0.91$	

Tabla 4.5: Factor de Calidad - Usabilidad

Fuente: Elaboración propia

Parte 4: Eficiencia	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Comportamiento en el tiempo	$X = 0.9$
Comportamiento de recursos	$X = 0.8$
$Eficiencia = e = (0.9+0.8) / 2 = 0.85$	

Tabla 4.6: Factor de Calidad - Eficiencia

Fuente: Elaboración Propia

Parte 5: Mantenimiento	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Estabilidad	$X = 1$
Facilidad de análisis	$X = 0.7$
Facilidad de cambio	$X = 1$
Facilidad de pruebas	$X = 0.8$
$Mantenimiento = m = (1+0.7+1+0.8) / 4 = 0.875$	

Tabla 4.7: Factor de Calidad - Mantenimiento

Fuente: Elaboración propia

Parte 6: Movilidad	Interpretación: $0 \leq X \leq 1$, donde 1 es el mejor valor que pueda tener la métrica.
Capacidad de instalación	$X = 1$
Capacidad de reemplazamiento	$X = 1$
Adaptabilidad	$X = 1$
$Movilidad = m_o = (1+1+1) / 3 = 1$	

Tabla 4.8: Factor de Calidad - Movilidad

Fuente: Elaboración propia

De acuerdo a los resultados obtenidos en las tablas 4.1 – 4.6 podemos establecer la calidad global del sistema en base a estos 6 parámetros medidos por la norma ISO 9126.

$$\begin{aligned} \text{Calidad Global (CG)} = & (\text{Funcionalidad (f)} + \\ & \text{Fiabilidad (fi)} + \\ & \text{Usabilidad (u)} + \\ & \text{Eficiencia (e)} + \\ & \text{Mantenimiento (m)} + \\ & \text{Movilidad (mo)}) / 6 \end{aligned}$$

$$CG = (f + fi + u + e + m + mo) / 6$$

$$CG = (0.9 + 0.87 + 0.91 + 0.85 + 0.875 + 1) / 6$$

$$CG = 5.345 / 6 = 0.90$$

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Después de la implementación del sistema se concluye lo siguiente:

- Los padres de familia y/o tutores manifestaron un alto interés por conocer el cumplimiento de las actividades educativas de sus hijos en todo momento, podemos deducir esta situación por el alto índice de lectura de actividades.
- Se observa una participación activa entre las personas que forman parte del proceso de enseñanza – aprendizaje de los estudiantes, ya que a través del sistema ellos pueden manifestar cualquier situación que crean importante para la consecución de los objetivos dentro de este proceso.
- Por lo mencionado anteriormente, apuntamos a que los objetivos planteados en este proyecto llegan a cumplirse ya que con el sistema se logran romper las barreras de tiempo y distancia que no permitían realizar un seguimiento adecuado sobre las actividades educativas de los estudiantes.
- El uso de un Framework nos permite normalizar, estandarizar el desarrollo de una aplicación Web, además ayuda a acelerar el proceso de desarrollo y permite la reutilización de código ya existente.

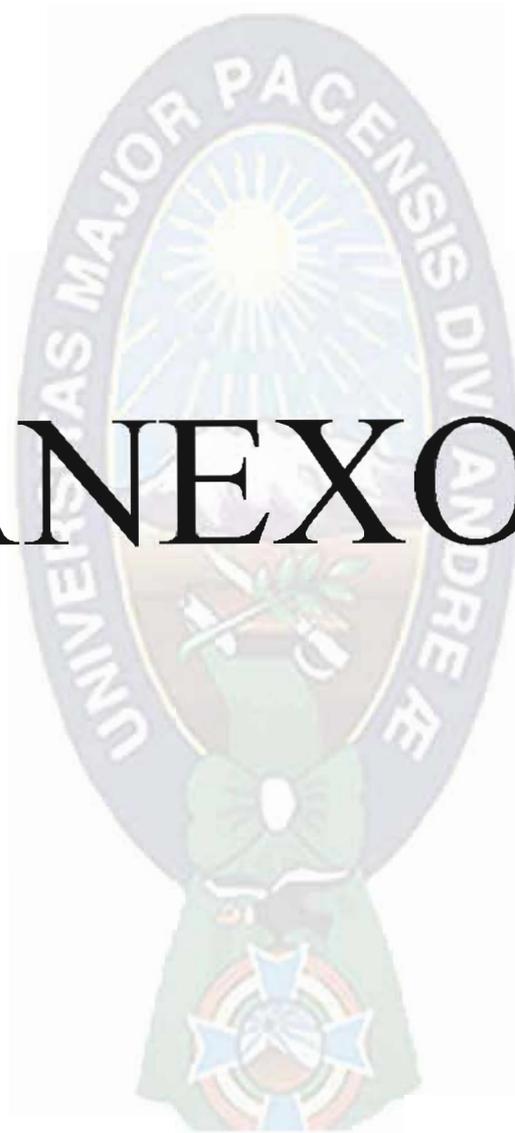
5.2 RECOMENDACIONES

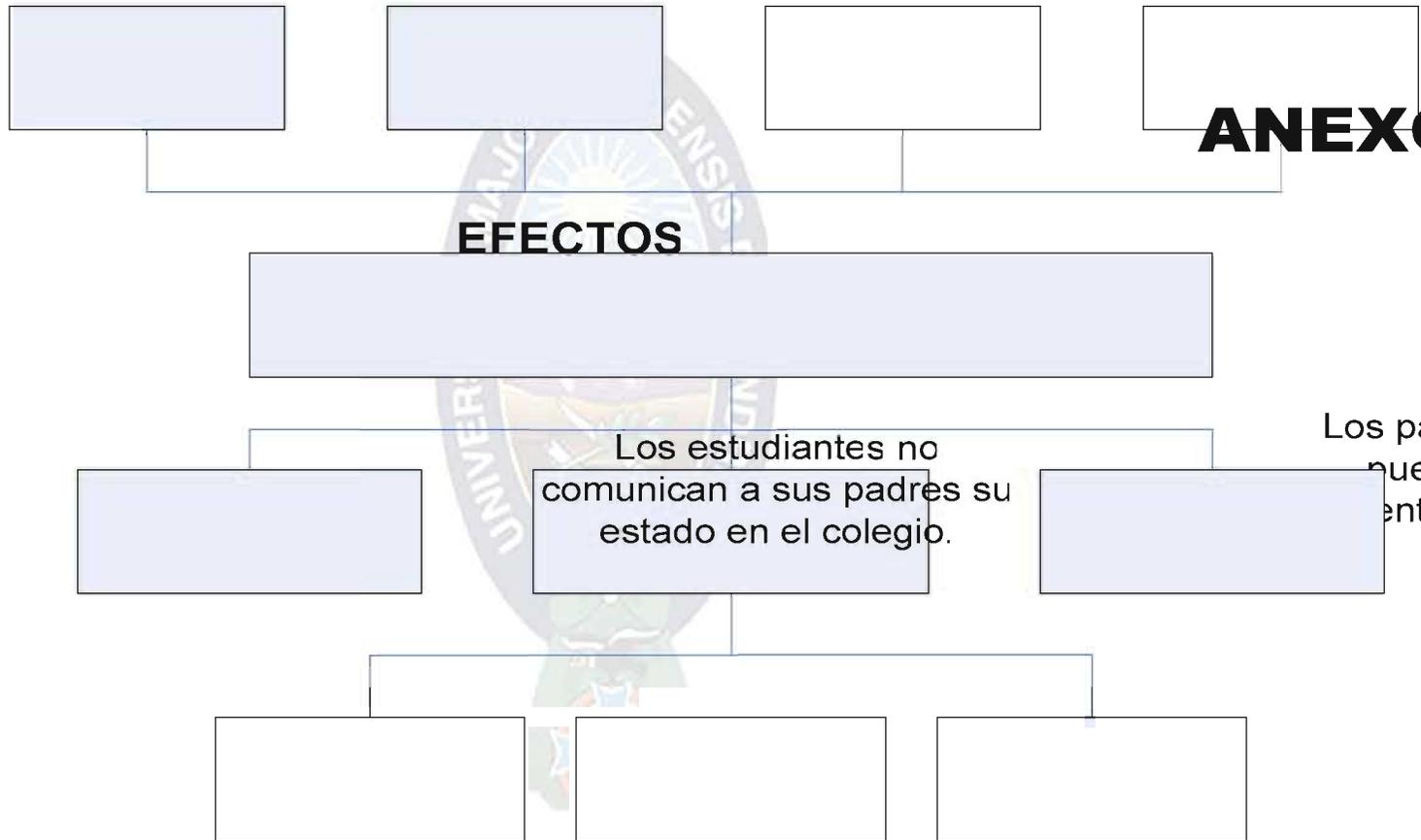
El sistema para un óptimo funcionamiento requiere de una introducción de datos dinámica de parte del educador, es así que para este efecto la institución educativa donde este proyecto será implementado debe definir políticas específicas que normen esta situación.

BIBLIOGRAFÍA

- [PRE-2000] “INGENIERIA DEL SOFTWARE UN ENFOQUE PRÁCTICO”, Roger S. Pressman, Editorial McGraw Hill 2000.
- [UML-2000] “APRENDIENDO UML EN 24 HORAS”, Joseph Schmuller, Editorial
- [MET INV-1997] “INTRODUCCIÓN A LA METODOLOGÍA DE LA INVESTIGACIÓN”, R. B. Avila Acosta, Editorial WH, 1997
- [IVA-1999] “EL PROCESO UNIFICADO DE DESARROLLO DEL SOFTWARE”, I. Jacobson, G. Booch, J. Rumbauch, Edición Wesley, 1999
- [TEC INF ENS-2004] “TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN APLICADAS A LA ENSEÑANZA”, Antonio R. Bartolomé
- [WIK-2008] Wikipedia, la enciclopedia libre
Disponible en: <http://es.wikipedia.org/wiki/Portada>
- [SCG-2008] SCHOOLGEN
Disponible en: www.schoolgen.com
- [FRM-2008] FRAMEWORKS
Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf,
- [JSF - 2007] JAVA SERVER FACES
Disponible en: <http://www.desarrolloweb.com/articulos/2380.php> ,JSF
- [JSF - 2008] JAVA SERVER FACES
Disponible en: <https://javaserverfaces.dev.java.net/>

ANEXOS





ANEXO A - ÁR

EFFECTOS

Los estudiantes no comunican a sus padres su estado en el colegio.

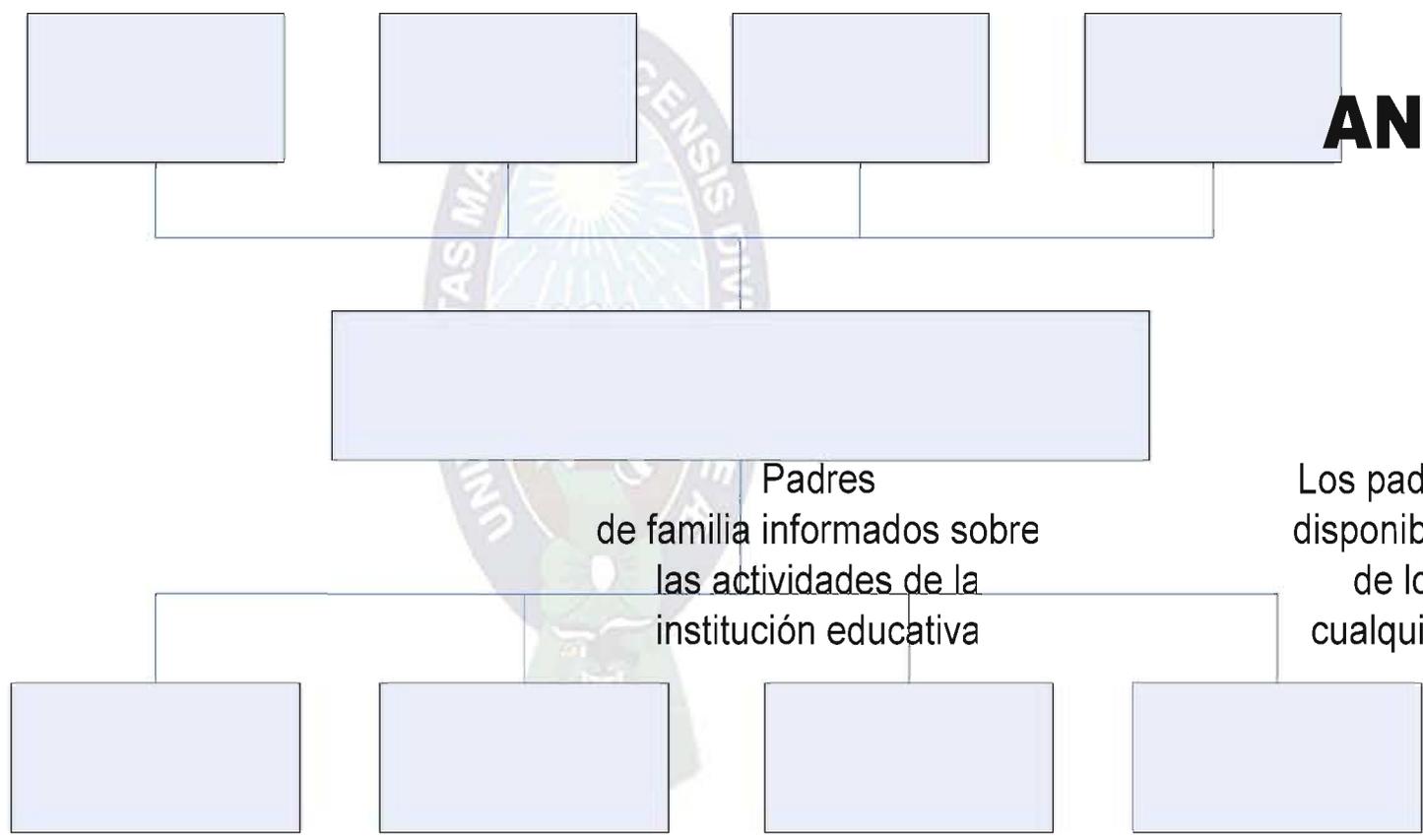
Los padres de familia pueden asistir a las entrevistas con los profesores.

PROBLEMA CENTRAL

Poca comunicación para

A-1

ANEXO B -



Los padres de familia tienen disponibles las calificaciones de los estudiantes en cualquier momento y lugar

B-1

OBJETIVO GENERAL

75
Comunicación fluida y activa