

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



**PROYECTO DE GRADO**

**SISTEMA DE INFORMACIÓN ACADÉMICA  
CARRERA DE LINGÜÍSTICA E IDIOMAS  
UNIVERSIDAD PÚBLICA DE EL ALTO**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POSTULANTE : ANGEL HERNÁN TICONA VILLA  
TUTOR : LIC. EFRAIN SILVA SANCHEZ  
REVISOR : M.SC. CARLOS MULLISACA CHOQUE**

**La Paz –Bolivia**

**2011**

## *Dedicado:*

*A mí querida madre Pascuala por el sacrificio, esfuerzo y confianza con que apoya mis sueños.*

*A mis hermanas: Beatriz y Lourdes por su apoyo incondicional.*

## **AGRADECIMIENTOS**

A Dios por permitirme disfrutar de esta vida tan maravillosa y concluir con este objetivo particularmente especial.

Al Lic. Efrain Silva Sanchez Docente Revisor, por su tiempo, paciencia, aporte de ideas, consejos y experiencia en este proyecto.

Al M.Sc. Carlos Mullisaca Choque Docente Revisor, por la confianza depositada en mi persona, por los consejos que me permitieron no desanimarme, sobre todo por su gran amistad y motivación.

A Norberto Condori Yujra, por sus ideas y sugerencias sin condiciones.

Al personal de la Carrera de Lingüística e Idiomas de la Universidad Pública de El Alto, por permitirme implementar el presente proyecto, y en particular a la responsable de Kardex Patricia Quispe y su Director.

A todos los docentes de la Carrera de Informática, que aportaron con sus conocimientos en mi formación superior.

Finalmente, a la Universidad Mayor de San Andrés por cobijarme en sus aulas por todos estos años, sin duda la mejor universidad.

**Muchas Gracias...**

tvangelvilla2010@hotmail.com

## RESUMEN

Hoy en día la información es un recurso de gran importancia para toda organización, pública y/o privada que trabaja con enormes volúmenes de datos. Las instituciones de carácter educativo no son la excepción, razón por lo cual, se debe prestar especial atención al manejo de datos e información que se genera en las mismas.

La Universidad Pública de El Alto (UPEA), esta experimentando un crecimiento significativo en la cantidad de estudiantes matriculados en sus distintas carreras. El presente trabajo hace énfasis en la parte académica de la Carrera de Lingüística e Idiomas dependiente de la UPEA. Los procesos académicos de; inscripción de estudiantes nuevos, regulares, registro de notas y emisión de documentos son manipulados y procesados en forma manual provocando retraso en el trabajo de los funcionarios del área académica, sin embargo este proyecto pretende solucionar los inconvenientes anteriores implementando el sistema de información académica para la carrera de lingüística e idiomas.

Para el desarrollo del sistema, se ha utilizado la metodología Procesos Unificado Ágil (AUP), y sus respectivas herramientas de desarrollo que propone. En la implementación, se utilizó la plataforma Windows, Php para la programación, Apache como servidor, MySql para la base de datos y JQuery para aplicación interactiva, PhpExcel y Zpdf para la emisión de reportes.

# ÍNDICE

<b>CAPÍTULO I</b>	<b>PÁG.</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
1.1. ANTECEDENTES.....	1
1.1.1. ANTECEDENTE INSTITUCIONAL.....	2
1.1.2. ANTECEDENTE DE PROYECTOS SIMILARES.....	3
1.2. OBJETO DE ESTUDIO .....	4
1.3. PLANTEAMIENTO DEL PROBLEMA.....	5
1.3.1 LISTADO DE PROBLEMAS.....	5
1.4. OBJETIVOS.....	6
1.5.1 OBJETIVO GENERAL.....	6
1.5.2 OBJETIVOS ESPECÍFICOS.....	6
1.5. JUSTIFICACIÓN.....	7
1.5.1. JUSTIFICACIÓN TÉCNICA.....	7
1.5.2. JUSTIFICACIÓN ECONÓMICA.....	8
1.5.3. JUSTIFICACIÓN SOCIAL.....	8
1.6. ALCANCES.....	9
1.7. METODOLOGÍA.....	9
1.8. APORTES.....	10

## **CAPÍTULO II**

<b>FUNDAMENTO TEÓRICO</b> .....	<b>12</b>
2.1 MARCO METODOLÓGICO.....	12
2.1.1. INGENIERIA DE SOFTWARE.....	12
2.1.2. METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE.....	13
2.1.3. AGILE UNIFIED PROCESS (AUP).....	17
2.1.3.1. FASES DE LA METODOLOGÍA AUP.....	17
2.1.3.1.1. FASE DE INICIACIÓN.....	18
2.1.3.1.2. FASE DE ELABORACIÓN.....	19
2.1.3.1.3. FASE DE CONSTRUCCIÓN.....	20
2.1.3.1.4. FASE DE TRANSICIÓN.....	21
2.1.3.2. DISCIPLINAS AUP.....	22
2.1.3.2.1. MODELADO DEL NEGOCIO.....	22
2.1.3.2.2. IMPLEMENTACIÓN.....	23
2.1.3.2.3. PRUEBAS.....	24
2.1.3.2.4. DESPLIEGUE.....	25
2.1.3.2.5. GESTIÓN DE CONFIGURACIÓN.....	26

2.1.3.2.6.	GESTIÓN DE PROYECTO.....	28
2.1.3.2.7.	ENTORNO.....	29
2.1.4.	LENGUAJE DE MODELADO UNIFICADO (UML).....	29
2.1.4.1.	DIAGRAMA DE CASOS DE USO.....	31
2.1.4.2.	DIAGRAMA DE CLASES.....	32
2.1.4.3.	DIAGRAMA DE COMPONENTES.....	33
2.1.4.4.	DIAGRAMA DE ESTADOS.....	33
2.1.4.5.	DIAGRAMA DE SECUENCIA.....	34
2.1.4.6.	DIAGRAMA DE ACTIVIDADES.....	35
2.1.4.7.	DIAGRAMA DE DESPLIEGUE.....	35
2.1.5.	INGENIERIA WEB.....	36
2.1.6.	METODOLOGÍA UWE (UML-BASED WEB ENGINEERING).....	36
2.1.6.1.	MODELO DE CASOS DE USO.....	37
2.1.6.2.	MODELO CONCEPTUAL.....	38
2.1.6.3.	MODELO NAVEGACIONAL.....	38
2.1.6.4.	MODELO DE PRESENTACIÓN.....	42
2.1.7.	SEGURIDAD DE LA FORMACIÓN.....	42
2.1.7.1.	SEGURIDAD FÍSICA.....	43
2.1.7.2.	SEGURIDAD LÓGICA.....	43
2.1.7.3.	SEGURIDAD CONDUCTUAL.....	44
2.1.8.	ARQUITECTURA TRES CAPAS.....	44
2.1.9.	CALIDAD DEL SISTEMA.....	45
2.1.9.1.	FUNCIONALIDAD.....	46
2.1.9.2.	CONFIABILIDAD.....	46
2.1.9.3.	USABILIDAD.....	46
2.1.9.4.	EFICIENCIA.....	46
2.1.9.5.	FACILIDAD DE MANTENIMIENTO.....	47
2.1.9.6.	PORTABILIDAD.....	47
2.1.10.	ESTIMACIÓN DE COSTOS.....	47
2.2.	MARCO REFERENCIAL.....	48
2.3.	MARCO CONCEPTUAL.....	50
2.4.	MARCO LEGAL.....	52
2.4.1.	EVALUACIÓN.....	52
2.4.2.	ADMISION ESTUDIANTIL.....	53
2.4.3.	MODALIDAD EXAMEN DE CLASIFICACIÓN.....	54
2.4.5.	MODALIDADES DE GRADUACIÓN.....	55
2.4.6.	CONVALIDACIÓN, HOMOLOGACIÓN Y COMPENSACIÓN.....	55
2.5.	MARCO TECNOLÓGICO.....	56
2.5.1.	PLATAFORMA.....	56
2.5.2.	SERVIDOR APACHE.....	57
2.5.3.	PHP.....	57
2.5.4.	MYSQL.....	57
2.5.5.	AJAX.....	58
2.5.6.	JQUERY.....	58
2.5.7.	CSS.....	59

2.5.7. GENERACIÓN DE PLANILLAS.....	59
2.5.8. GENERACIÓN DE REPORTES CON PDF.....	60
2.5.9. MAGICDRAW.....	60
2.10. DISEÑO DE INTERFAZ.....	60

## CAPÍTULO III

<b>MARCO APLICATIVO.....</b>	<b>63</b>
3.1. FASE DE INICIACIÓN.....	63
3.1.1. SITUACIÓN ACTUAL.....	63
3.1.1.1. ORGANIGRAMA DE LA INSTITUCIÓN.....	63
3.1.1.2. SISTEMA LÓGICO.....	69
3.1.1.3. PROCESOS ACADÉMICOS.....	70
3.1.1.4. IDENTIFICACIÓN DE RIESGOS.....	75
3.2. FASE DE ELABORACIÓN.....	76
3.2.1. MODELADO DEL NEGOCIO.....	76
3.2.1.1. REQUERIMIENTOS.....	76
3.2.1.2. IDENTIFICACIÓN DE ACTORES.....	78
3.2.1.3. CASOS DE USO.....	78
3.2.1.4. DESCRIPCIÓN DE CASOS DE USO.....	79
3.2.1.5. GLOSARIO DE TERMINOS.....	83
3.2.1.6. DIAGRAMA DE ACTIVIDADES.....	84
3.2.2. MODELO CONCEPTUAL.....	86
3.2.3. MODELO NAVEGACIONAL.....	87
3.2.4. MODELO DE PRESENTACIÓN.....	92
3.2.5. ARQUITECTURA DEL SISTEMA.....	96
3.3. FASE DE CONSTRUCCIÓN.....	97
3.3.1. DIAGRAMA DE SECUENCIA.....	97
3.3.2. DIAGRAMA DE CLASES.....	101
3.3.3. DIAGRAMA DE COMPONENTES.....	102
3.3.4. DIAGRAMA DE DESPLIEGUE.....	103
3.3.5. ITERACIONES DEL SISTEMA.....	103
3.3.5.1. PRIMERA ITERACIÓN.....	104
3.3.5.2. SEGUNDA ITERACIÓN.....	106
3.3.5.3. TERCERA ITERACIÓN.....	111
3.4. FASE DE TRANSICIÓN.....	113
3.4.1. PRUEBA DEL SISTEMA.....	113
3.4.2. POLÍTICAS DE SEGURIDAD.....	114
3.4.3. RESTRICCIONES DEL SISTEMA.....	116
3.4.4. CAPACITACION.....	116

## **CAPÍTULO IV**

<b>EVALUACIÓN DEL SISTEMA.....</b>	<b>118</b>
4.1 FACTORES DE CALIDAD.....	118
4.1.1. FUNCIONALIDAD.....	118
4.1.2. CONFIABILIDAD.....	121
4.1.3. USABILIDAD.....	122
4.1.4. EFICIENCIA.....	123
4.1.5. FACILIDAD DE MANTENIMIENTO.....	123
4.1.6. PORTABILIDAD.....	124
4.2. COSTO / BENEFICIO.....	125
4.2.1. ANALISIS DE COSTOS.....	125
4.2.1.1. COSTO DE DESARROLLO DEL SOFTWARE.....	125
4.2.1.2. COSTO DE IMPLEMENTACIÓN DEL SISTEMA.....	127
4.2.1.3. COSTO DE ELABORACIÓN DEL PROYECTO.....	128
4.2.2. ANALISIS DE BENEFICIOS.....	128

## **CAPÍTULO V**

<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>130</b>
5.1 CONCLUSIONES.....	130
5.2 RECOMENDACIONES.....	131

<b>BIBLIGRAFÍA.....</b>	<b>132</b>
-------------------------	------------

## **ANEXOS**

ANEXO A	Árbol de problemas
ANEXO B	Árbol de objetivos
ANEXO C	Marco lógico
ANEXO D	Instrumentos de diagnostico
ANEXO E	Cronograma de actividades
ANEXO F	Casos de uso
ANEXO G	Diagramas de actividad
ANEXO H	Diagramas de navegación
ANEXO I	Diagramas de presentación
ANEXO J	Diagramas de secuencia
ANEXO K	Pantallas del sistema

## **DOCUMENTOS**

## ÍNDICE DE FIGURAS

FIGURA		PÁG.
Figura 2.1	Tecnología estratificada.....	12
Figura 2.2	Ciclo de vida AUP.....	17
Figura 2.3	Modelo iterativo – incremental.....	21
Figura 2.4	Actividades de administración de configuración.....	27
Figura 2.5	Modelo UML.....	30
Figura 2.6	Diagrama de casos de uso.....	32
Figura 2.7	Diagrama de clases.....	32
Figura 2.8	Diagrama de Componentes.....	33
Figura 2.9	Diagrama de estados.....	34
Figura 2.10	Diagrama de secuencia.....	34
Figura 2.11	Diagrama de Actividades.....	35
Figura 2.12	Diagrama de despliegue.....	36
Figura 2.13	Modelo de aplicación UWE.....	37
Figura 2.14	Clase de un modelo conceptual.....	38
Figura 2.15	Clase de Navegación.....	39
Figura 2.16	Clase de navegación notación larga y corta.....	40
Figura 2.17	Clase visita guiada y visita guiada en notación corta.....	40
Figura 2.18	Clase consulta (A) y consulta en notación corta (B).....	41
Figura 2.19	Clase Menú (A) y su notación corta (B).....	41
Figura 2.20	Arquitectura 3-capas.....	45
Figura 2.21	Arquitectura de PhpExcel.....	59
Figura 3.1	Organigrama de la institución.....	64
Figura 3.2	Registrar nuevo estudiante.....	71
Figura 3.3	Proceso de inscripción.....	72
Figura 3.4	Caso de uso actual - Record académico.....	74
Figura 3.5	Diagrama de casos de uso del sistema.....	79
Figura 3.6	Diagrama de actividad - Registro de estudiante.....	84
Figura 3.7	Diagrama de actividad – Inscripción.....	84
Figura 3.8	Diagrama de actividad - Generación de planillas.....	85
Figura 3.9	Diagrama de actividad – Cargar planilla con notas.....	85
Figura 3.10	Modelo conceptual.....	86
Figura 3.11	Diagrama de navegación – Estudiantes.....	87
Figura 3.12	Diagrama de navegación – Documentos.....	88
Figura 3.13	Diagrama de navegación – Inscripción.....	89
Figura 3.14	Diagrama de navegación – Notas.....	90
Figura 3.15	Diagrama de navegación – Administración.....	91
Figura 3.16	Diagrama de presentación - Pagina principal.....	92
Figura 3.17	Diagrama de presentación - Registro de estudiantes.....	93
Figura 3.18	Diagrama de presentación – Inscripción.....	94
Figura 3.19	Diagrama de presentación - Generar planilla.....	95
Figura 3.20	Diagrama de presentación - Cargar planilla.....	95
Figura 3.21	Arquitectura del sistema.....	96
Figura 3.22	Diagrama de secuencia - Acceso al sistema.....	97

Figura 3.23	Diagrama de secuencia – Registro de estudiantes.....	98
Figura 3.24	Diagrama de secuencia – Inscripción.....	99
Figura 3.25	Diagrama de secuencia – Crear Planilla.....	100
Figura 3.26	Diagrama de secuencia – Cargar Planilla.....	100
Figura 3.27	Diagrama de clases del sistema.....	101
Figura 3.28	Diagrama de componentes del sistema.....	102
Figura 3.29	Diagrama de despliegue del sistema.....	103
Figura 3.30	Pantalla de inicio de sesión.....	104
Figura 3.31	Menú principal del sistema.....	104
Figura 3.32	Módulo estudiantes – Registrar estudiante.....	105
Figura 3.33	Módulo Inscripción.....	106
Figura 3.34	Módulo Inscripción – Lista materias.....	106
Figura 3.35	Módulo inscripción - Inscribir a examen de clasificación.....	107
Figura 3.36	Módulo inscripción – Retiro y/o Adición.....	108
Figura 3.37	Módulo inscripción – Boleta de inscripción.....	108
Figura 3.38	Módulo Notas.....	109
Figura 3.39	Módulo de notas – Transcribir notas.....	109
Figura 3.40	Módulo notas – Crear planilla de notas.....	110
Figura 3.41	Módulo notas – Vista de planilla.....	110
Figura 3.42	Módulo notas - Cargar planilla de notas.....	111
Figura 3.43	Pantalla – Emisión de documentos.....	111
Figura 3.44	Módulo documentos – Record académico.....	112
Figura 3.45	Pantalla – Certificado de notas.....	112
Figura 3.46	Prueba de caja negra.....	113

## ÍNDICE DE TABLAS

TABLA		PÁG.
Tabla 2.1	Metodologías tradicionales y ágiles.....	16
Tabla 2.2	Actividades de disciplina modelado.....	22
Tabla 2.3	Actividades de disciplina implementación.....	24
Tabla 2.4	Actividades de disciplina prueba.....	25
Tabla 2.5	Actividades de disciplina despliegue.....	26
Tabla 2.6	Actividades de disciplina gestión de proyecto.....	28
Tabla 2.7	Actividades de disciplina entorno.....	29
Tabla 2.8	Valores de multiplicidad.....	33
Tabla 2.9	Comparación de proyectos similares.....	49
Tabla 3.1	Planes de estudio de la carrera.....	70
Tabla 3.2	Formas de ingreso.....	70
Tabla 3.3	Riesgos del sistema.....	75
Tabla 3.4	Requisitos básicos del sistema.....	77
Tabla 3.5	Descripción general de actores.....	78
Tabla 3.6	Descripción caso de uso - Registrar estudiante.....	80
Tabla 3.7	Descripción caso de uso – Inscripción.....	80
Tabla 3.8	Descripción caso de uso - Examen de clasificación.....	81
Tabla 3.9	Descripción caso de uso - Retiro / Adición.....	81
Tabla 3.10	Descripción caso de uso notas – Crear planilla.....	82
Tabla 3.11	Descripción caso de uso notas – Cargar planilla.....	82
Tabla 3.12	Glosario de términos.....	83
Tabla 3.13	Cronograma de entregables.....	103
Tabla 4.1	Cálculo de puntos de función no ajustados.....	120
Tabla 4.2	Valores de ajuste de complejidad.....	120
Tabla 4.3	Test de evaluación de usabilidad.....	122
Tabla 4.4	Conversión de puntos de función a KLDC.....	125
Tabla 4.5	Constantes de COCOMO.....	126
Tabla 4.6	Costo de licencias.....	127
Tabla 4.7	Costo de equipos.....	127
Tabla 4.8	Costo de elaborar proyecto.....	128
Tabla 4.9	Análisis de beneficios.....	129

# CAPÍTULO I

## INTRODUCCIÓN

Los cambios acelerados en la tecnología han revolucionado el manejo de la información acorde a las necesidades de una sociedad en transformación. Hoy en día, la información es un recurso de gran importancia para toda organización, pública y/o privada que trabaja con enormes volúmenes de datos. Las instituciones de carácter educativo no son la excepción, ya que las mismas reciben una creciente cantidad de nuevos estudiantes cada gestión, por lo cual se debe prestar especial atención al manejo de la información que se genera en dichas instituciones.

La Universidad Pública de El Alto (UPEA) recientemente creada, esta experimentando un crecimiento significativo en la cantidad de estudiantes matriculados en sus distintas carreras, los mismos requieren nuevas formas de atención a sus requerimientos en cuanto a los procesos de carácter académico, infraestructura y otros. El presente trabajo hace énfasis en la parte académica de la Carrera de Lingüística e Idiomas dependiente de la UPEA, los mismos son manipulados y procesados en forma manual, en las etapas de: inscripción de estudiantes nuevos, regulares, registro de notas y emisión de documentos. Estos procesos se repiten cada vez que se inicia y cierra la gestión.

Un instrumento de valía es la Informática que sustituye en parte el trabajo manual por el automatizado. La UPEA en su carrera de Lingüística e Idiomas requiere mejorar sus procesos de gestión académica utilizando tecnologías de información y comunicación, debido a la gran cantidad de información que genera en forma semestral por el ingreso de nuevos estudiantes y los ya existentes.

Los procesos de gestión académica que actualmente maneja la institución son lentos y morosos, porque son administradas en forma manual y poco sistematizadas. Razón por la cual nace la idea de proyecto de grado “**Sistema de Información Académica para la Carrera de Lingüística e Idiomas de la Universidad Pública de El Alto**”, que permitirá agilizar los procesos de administración académica en sus distintas etapas ya mencionadas.

## **1.1. ANTECEDENTES**

### **1.1.1. ANTECEDENTE INSTITUCIONAL**

La Carrera de Lingüística e Idiomas de la UPEA, se encuentra ubicado en la ciudad de El Alto zona Villa Esperanza S/N del distrito IV. La misma fue instituida durante la creación de la UPEA por ley 2115 del 5 de septiembre de 2000; por entonces contaba con 53 inscritos, solamente se consolida el año 2001. Sin embargo, el año 2002 la carrera se divide en dos: Villa esperanza y Villa Dolores. Los estudiantes de Villa Esperanza deciden refundar como carrera de Lingüística, consiguiendo su reconocimiento en el Honorable Consejo Universitario. Después de los hechos de octubre se vuelven a unir en una sola, que es como funciona actualmente.

La autonomía universitaria finalmente fue conquistada el 12 de noviembre del año 2003 según ley 2556. La Carrera se consolida definitivamente en las jornadas Académicas llevada a cabo el año 2007 denominándose Lingüística e Idiomas de la Universidad Pública de El Alto [ESTATUTO,2009].

## **MISIÓN**

La carrera de Lingüística e Idiomas tiene la misión de:

Formar profesionales íntegros, con conocimiento científico, tecnológico en la Ciencia de la Lingüística e Idiomas; con conciencia crítica y reflexiva con valores éticos, morales y altamente competentes para responder a la vida intracultural, intercultural y plurilingüe del Estado Plurinacional de Bolivia en pro de un desarrollo sostenible[PLAN,2010].

## **VISIÓN**

La carrera de Lingüística e Idiomas tiene la visión de:

Impulsar el avance de las ciencias del Lenguaje desarrollando el nivel científico, tecnológico y creativo de las lenguas oficiales del Estado Plurinacional de Bolivia y otros idiomas aplicando la teoría y la práctica para el progreso lingüístico, sociolingüístico y sociocultural a favor de los intereses de la sociedad boliviana.

El congreso de carrera es la máxima autoridad que dirige la carrera. La dirección es la autoridad ejecutiva de la carrera de lingüística e idiomas, a objeto de administrar, coordinar, desarrollar estrategias y mecanismos de ejecución emanadas de la asamblea docente estudiantil y consejo de carrera.

### **1.1.2. ANTECEDENTE DE PROYECTOS SIMILARES**

En la carrera de Informática de La Universidad Mayor de San Andrés se desarrollaron varios trabajos similares al presente, sin embargo cada uno tiene su propia particularidad acorde al contexto y requerimiento de cada institución. En ese sentido los trabajos similares que destacamos son:

- “Sistema de seguimiento académico carrera de arquitectura universidad pública de el alto”, del postulante: Víctor Hugo Calcina López, cuyo objetivo del sistema es implementar un sistema de información académica vía web que permita centralizar la información de seguimiento académico, para la Universidad Publica de El Alto. La metodología que adopto para el desarrollo del sistema es Object Oriented Software Engineering de Jacobson, en cuanto al servidor se utilizó Apache, Php para programación y para la base de datos MySql. [CALCINA,2007].
- “Sistema de Gestión Académica para el Instituto Normal Superior Simón Bolívar”, del postulante: Diego Omar Chambi Lima, cuyo objetivo es desarrollar e implementar el Sistema de Gestión Académica para el Instituto Normal Simón Bolívar. La metodología que empleo es UML de Larman, para la programación Php, MySql como gestor de base de datos y Apache como servidor [CHAMBI,2007].
- “Sistema de Administración Académica vía Web caso: Escuela de Idiomas”, del postulante: Francisco Carpio Calle, cuyo objetivo fue Desarrollar e Implementar un Sistema de Administración Académica para la escuela de Idiomas del Ejercito. El método utilizado es UWE (UML-Based Web Engineering) , en cuanto al lenguaje de programación fue desarrollada en Php, Mysql en gestión de base de datos y Apache como servidor [CARPIO,2008].

## 1.2. OBJETO DE ESTUDIO

Sistema de Información académica para la carrera de Lingüística e Idiomas de la Universidad Publica de El Alto (UPEA).

### 1.3. PLANTEAMIENTO DEL PROBLEMA

Después de aplicar instrumentos de investigación preliminar como: la observación y entrevistas en la carrera de Lingüística e Idiomas, se pudo evidenciar que los funcionarios manejan los procesos de inscripción, retiro y adición de materias y emisión de documentos en forma manual y en otros casos se utiliza Word y Excel. Los mismos tienen efectos como: retardo en emisión de informes, realización de historias académicas utilizando actas impresas y anotando las notas en materias presentadas por estudiantes y otros (Aunque la carrera cuenta con laboratorio de computación). Para mayor información sobre el relevamiento de información véase anexo D.

Por otro lado, los aspectos que incrementan las falencias mencionadas son:

#### 1.3.1. LISTADO DE PROBLEMAS

- ✓ Los procesos de seguimiento académico son realizados: utilizando los libros de actas, la inscripción no verifica prerequisites al igual que el retiro y/o adición de materias, paralelos con más estudiantes inscritos que otros.
- ✓ El manejo simultáneo de varios planes de estudio dificulta la emisión de documentos como historias académicas. Se trabaja con 8 planes de estudios al mismo tiempo.
- ✓ El almacenamiento de la información no es adecuado, ya que se registran en hojas de cálculo Excel separadas y no centralizadas, provocando redundancia en el manejo de datos.
- ✓ Se genera gran cantidad de documentos como fotocopias de CI y Matricula en cada gestión en los procesos de inscripción, retiro y/o adición que se solicita al estudiante, los mismos son adjuntados a los documentos que ya existen en kardex.

En consecuencia se ha optado por el siguiente problema de investigación:  
**¿El sistema de información académica para la carrera de lingüística e idiomas, permitirá mejorar los procesos de registro, inscripción y seguimiento académico, mostrando una información eficiente y oportuna?**

## **1.4. OBJETIVOS**

### **1.4.1. OBJETIVO GENERAL**

Desarrollar e implementar un Sistema de Información Académica para la carrera de Lingüística e Idiomas de la Universidad Pública de el Alto, que apoye los procesos de centralización y organización de la información académica, para brindar resultados oportunos y eficientes.

### **1.4.2. OBJETIVOS ESPECÍFICOS**

- ✓ Establecer características principales del sistema académico, justificando su desarrollo e implementación del mismo acorde a una metodología.
- ✓ Desarrollar un módulo de registro de estudiantes nuevos, que permita ingresar datos más importantes del mismo.
- ✓ Desarrollar un módulo de inscripción de estudiantes nuevos y antiguos.
- ✓ Implementar un módulo de retiro y/o adición de materias, acorde al pensum vigente de la carrera.
- ✓ Desarrollar un módulo de manejo de notas, generando planillas de notas y cargado de los mismos.

- ✓ Implementar el módulo de consultas mas frecuentes.
- ✓ Desarrollar un módulo para generar reportes y estadísticas para la toma de decisiones por parte de las autoridades.
- ✓ Evaluar el sistema de acuerdo a las normas establecidas en el desarrollo de software.

## 1.5. JUSTIFICACIÓN

La función de la carrera de Lingüística e Idiomas, es importante en el proceso de formación de nuevos profesionales, para el cual requiere el uso de medios y recursos adecuados, uno de ellos es la implementación de tecnologías de la información y comunicación que coadyuve en procesos que administra kardex. La misma la justificaremos desde tres puntos de vista: técnica, económica y social.

### 1.5.1. JUSTIFICACIÓN TÉCNICA

La carrera de Lingüística e Idiomas cuenta con recursos computacionales para la implementación del sistema académico. Los mismos no tienen funciones específicas que administren base de datos, en la parte de software se tienen instalados en plataforma Windows, con programas como: Office 2007, PDF y otros programas, los mismos que nos permitirán la implementación de la aplicación.

En la parte de Hardware la carrera dispone de equipos de computación, las cuales tienen las siguientes características:

Área	Cantidad de equipos	Característica	Observación
Kardex	3	Core 2 y Pentium	Funcionando
Despacho	1	Core 2	Funcionando
Laboratorio	30	Pentium IV	Funcionando

Por otro lado, se tiene impresoras láser que ya están en uso. Las mismas permiten un desempeño de red, mejorando así el proceso de interacción de estudiantes con el sistema.

Para la parte de desarrollo del sistema se utilizará como lenguaje de programación Php5, como gestor de base de datos MySql y Apache como Servidor Web por sus características de gratuidad.

### **1.5.2. JUSTIFICACIÓN ECONÓMICA**

La información es un recurso de valía en toda institución, más aun si esta es manejada en grandes volúmenes de datos, en la medida, en que las mismas sean automatizadas implica ahorro en tiempo, reducción en material de escritorio, reduce el trabajo manual y así mejorando el servicio a la población estudiantil de la carrera.

Por otro lado, mencionamos que el sistema será desarrollado bajo la premisa de software libre, que implica la no erogación de gastos en licencias de uso. Por otra parte la institución cuenta con cableado de red lo cual facilitara su implementación.

### **1.5.3. JUSTIFICACIÓN SOCIAL**

La carrera de Lingüística e Idiomas con el transcurrir del tiempo recibe a nuevos estudiantes, en un promedio de 90 estudiantes por gestión aunque en este último año ha recibido más de 250 estudiantes. La información de los mismos es manipulada en forma manual y hojas de cálculo Excel. En ese sentido el sistema que se propone coadyuvará a la institución, mostrando confiabilidad en el manejo de datos de los estudiantes, reducirá el trabajo excesivo del personal de kardex, reducirá los tiempos de respuesta a la comunidad mejorando la atención en procesos como: inscripción, retiro y/adición, consultas más frecuentes y emisión de reportes para la toma de decisiones en forma oportuna.

## 1.6. ALCANCES

El presente proyecto contempla los siguientes módulos:

- Registro de estudiantes nuevos en la base de datos.
  - ✓ Registro semestral de estudiantes
  - ✓ Impresión de boletas de registro.
- Habilitación de usuarios del sistema con distintos niveles de acceso.
  - ✓ Registro de funcionarios como usuarios del sistema
  - ✓ Habilitar a estudiantes como usuarios del sistema
  - ✓ Habilitar docentes como usuarios del sistema
- Inscripción de estudiantes en las materias y paralelos por gestión.
  - ✓ Registro de estudiante en paralelos y materias correspondientes
  - ✓ Emitir boleta de inscripción
- Registro y control de notas.
  - ✓ Registrar notas de estudiantes.
  - ✓ Listado de estudiantes con notas.
  - ✓ Generación y cargado de planilla de notas.
- Registro y Asignación de docentes a paralelos por materia.
- Emisión de historias académicas.
- Consultas y reportes mas frecuentes.

## 1.7. METODOLOGÍA

Para la elaboración del presente proyecto se tomará en consideración una serie de métodos y técnicas, los mismos se detallan a continuación:

### a) Propuesta de perfil

- ✓ Planteamiento del problema
- ✓ Marco lógico

## b) Metodología de desarrollo

En el desarrollo del proyecto se considera el uso de la metodología AUP(Agile Unified Process), en las diferentes etapas del desarrollo del proyecto considerando:

- ✓ Identificación de requerimientos
- ✓ Análisis del sistema
- ✓ Diseño del sistema
- ✓ Desarrollo del sistema
- ✓ Implementación

## c) Tecnología a emplearse

- ✓ Para el manejo de la base de datos se utiliza MySql.
- ✓ Para la programación se emplea Php.
- ✓ Apache como Servidor Web.
- ✓ Fireworks para la edición de botones e imágenes.
- ✓ Html5 para la edición de contenidos.
- ✓ Microsoft Excel para la generación de listas y planilla de notas.
- ✓ PDF para la generación de reportes.
- ✓ JQuery y JavaScript para apoyar al diseño de interfaz.
- ✓ MagicDraw y MagicUwe herramientas case para apoyar el modelado de la aplicación.

## 1.8. APORTES

El presente proyecto de grado tiene como principal aporte la implementación del sistema de información académica para la carrera de Lingüística e Idiomas, permitiendo un registro, seguimiento y control adecuado.

Además será una herramienta que proporcione una información oportuna para la toma de decisiones por parte de las autoridades y planificación de nuevos emprendimientos propios de la institución con otra visión tecnológica.

Aplicación de otras metodologías como AUP, para construir sistemas de información de manera ágil.



## CAPÍTULO II

### FUNDAMENTO TEÓRICO

Para alcanzar objetivos planteados en el capítulo anterior, en esta etapa describiremos el sustento del proyecto que se realiza. Las mismas describen las herramientas necesarias para su análisis, diseño y posterior implementación.

Este capítulo es dividido en: marco metodológico, marco referencial, marco conceptual, marco legal y marco tecnológico. Esta forma de configuración está acorde a los requerimientos del tutor de proyecto.

#### 2.1. MARCO METODOLÓGICO

##### 2.1.1. INGENIERÍA DE SOFTWARE

Una definición más completa según la IEEE referida por Pressman para ingeniería de software es “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo y mantenimiento del software”. La ingeniería de software es una tecnología estratificada como en la figura 2.1.

**Figura 2.1:** Tecnología estratificada



Fuente: [PRESSMAN,2005]

Cualquier enfoque de calidad debe apoyarse sobre un compromiso con la calidad. El proceso, define el marco de trabajo para un conjunto de áreas clave, las cuales forman los métodos técnicos donde se obtienen productos de trabajo como son: modelos, documentos, datos, informes, etc. asegurando la calidad y el cambio de gestión adecuadamente.

Las Herramientas de ingeniería proporcionan el soporte automatizado o semiautomatizado para el proceso y los métodos.

Los Métodos proporcionan los “como” técnicos para construir software. Los métodos abarcan tareas que incluyen la comunicación el análisis de requerimientos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte.

### **2.1.2. METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE**

Las metodologías ágiles tienen como objetivo esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que pudieran surgir a lo largo del proyecto.

Mencionemos alguna de las metodologías ágiles destacadas [AMARO,2007]:

- Extreme Programming (XP)
- Scrum
- Agile Modeling Adaptive Software Development (ASD)
- Crystal Clear
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD)
- Lean Software Development (LSD)
- Agile Unified Process (AUP)
- Software Development Rhythms

## Microsoft Solutions Framework (MSF)

Se hace referencia a ellos como metodologías ágiles desde que se publicó el manifiesto ágil en 2001.

### MANIFIESTO ÁGIL

Dicho manifiesto valora los siguientes aspectos:

- ✓ **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- ✓ **Desarrollar software que funciona más que conseguir una buena documentación.** La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- ✓ **La colaboración con el cliente más que la negociación de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- ✓ **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

El manifiesto anterior se basa en doce principios para alcanzar la agilidad. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto a metas a seguir y organización del mismo. Según [Pressman,2005] dichos principios son:

- i. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- ii. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- iii. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- iv. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- v. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- vi. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- vii. El software que funciona es la medida principal de progreso.
- viii. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- ix. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- x. La simplicidad es esencial.
- xi. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- xii. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

## METODOLOGÍAS TRADICIONALES VS ÁGILES

Una metodología se convierte en tradicional cuando es utilizada frecuentemente por desarrolladores de software.

Según la filosofía de desarrollo se pueden clasificar las metodologías en dos grupos: las metodologías tradicionales, que se basan en una fuerte planificación durante todo el desarrollo, y las metodologías ágiles, en las que el desarrollo de software es incremental, cooperativo, sencillo y adaptado. Dichas características son ilustradas en la tabla 2.1.

**Tabla 2.1:** Metodologías tradicionales y ágiles

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.

Fuente: [CURSO,2009]

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta interesante, es decir, para cada proyecto, la problemática sería definir cada una de las prácticas, y en el momento preciso definir parámetros para saber cual usar.

### 2.1.3. AGILE UNIFIED PROCESS (AUP)

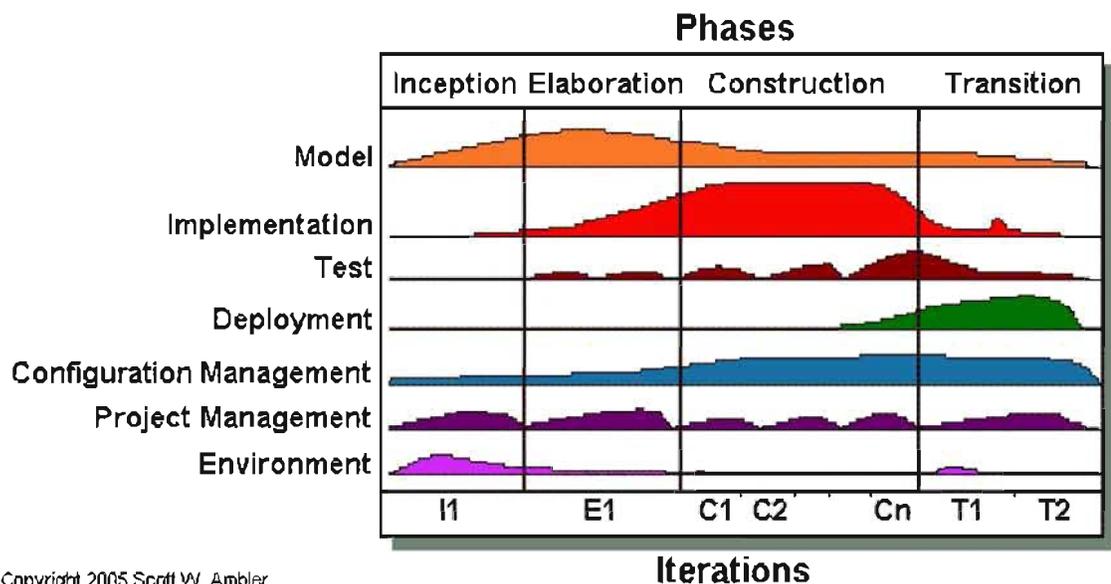
“El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP)” [Wiki,2011]. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

Las cuatro fases de AUP propuesta por Scott W. Ambler son:

#### 2.1.3.1. FASES DE LA METODOLOGÍA AUP

AUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 2.2 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto AUP.

Figura 2.2: Ciclo de vida AUP



Copyright 2005 Scott W. Ambler

Fuente: [AMBLER,2005]

### 2.1.3.1.1. FASE DE INICIACIÓN

El objetivo de esta fase es identificar los objetivos y el alcance inicial del proyecto. Una arquitectura potencial para el sistema, obtener fondos y aceptación por parte de las personas involucradas. Las actividades que destaca Scott W. Ambler para esta fase son:

- ✓ **Definir el alcance del proyecto.** Esto incluye la definición, a un alto nivel, de qué es lo que hará el sistema y definir qué es lo que el sistema no va a hacer. Aquí se establecen los límites desde dónde el equipo operará. Esto suele tomar la forma de una lista de características de alto nivel y / o el punto de casos de uso.
- ✓ **Estimación de costos y calendario.** En un nivel alto, el calendario y el costo del proyecto son estimados. Las estimaciones generales son realizadas en iteraciones de fases posteriores, más específicamente es implementado en las fases tempranas de elaboración. Como en todas las planificaciones, estas tareas que van a ser completadas en un futuro cercano y son detalladas con más precisión y con una gran confianza mientras que las tareas bajo la línea son entendidas para ser estimadas con un que no es posible programar todo un proyecto, en su pistoletazo de salida con cualquier grado aceptable de confianza con un margen de error más grande. Esto ha sido reconocido por la mayoría de las industrias de que no es posible programar un proyecto completo de un sólo con algún grado de aceptable de desacuerdo. Lo mejor que se puede hacer es planificar para el corto plazo y la precisar a largo plazo lo mejor que se pueda.
- ✓ **Definición de Riesgos.** Los riesgos del proyecto son primeramente definido aquí. La administración del riesgo es importante en proyectos de AUP. La lista de riesgos es una compilación en vivo que cambiará en el

tiempo cuando los riesgos serán identificados, mitigados, evitados y / o materializados o exterminados. El control de riesgos del proyecto, como los riesgos de más alta prioridad, manejan la programación de las iteraciones. Los riesgos más altos, por ejemplo, son dirigidos en iteraciones más tempranas que los riesgos de menor prioridad.

- ✓ **Determinar la factibilidad del proyecto.** Su proyecto debe tener sentido desde la perspectiva técnica, operacional y del negocio. En otras palabras, debe ser capaz de crearlo, una vez desplegado debe ser capaz de correrlo, y debe tener un sentido económico para hacer estos aspectos. Si su proyecto no es viable, este debe ser cancelado.
- ✓ **Preparar el entorno del proyecto.** Esto incluye el área de trabajo para el equipo. Solicitar el personal que se necesitará, obteniendo hardware y software que será requerido inmediatamente y compilar una lista de hardware y software que será necesitado después.

#### 2.1.3.1.2. FASE DE ELABORACIÓN

El principal objetivo de esta fase es probar la arquitectura del sistema a desarrollar. El punto es asegurar que el equipo de trabajo puede desarrollar un sistema que satisfaga los requisitos, y la mejor manera de hacerlo es la construcción de extremo a extremo del esqueleto de trabajo del sistema conocido como "prototipo de la arquitectura". Esto es en realidad un concepto pobre porque mucha gente piensa en deshacerse de los prototipos. En cambio, su significado es software funcional de alto nivel, el cual incluye varios casos de uso de alto riesgos para mostrar que el sistema es técnicamente factible [AMBLER,2005]

Es importante señalar que los requisitos no se especifican por completo en este punto. Se detallan sólo lo suficiente como para entender los riesgos de la arquitectura y para asegurar que exista una comprensión de los alcances de cada

requerimiento para que la planificación posterior se puede llevar a cabo. Los riesgos de la Arquitectura son identificados y priorizados durante la Elaboración.

Durante esta fase, el equipo también se está preparando para la fase de construcción. Como el equipo gana una mano en la arquitectura del sistema, ellos comienzan con la creación del ambiente propicio para la Construcción mediante la compra de hardware, software y herramientas. Las personas son dirigidas desde la perspectiva de la Administración del Proyecto; los recursos son solicitados o contratados. Los planes de comunicación y la colaboración se finalizan.

Para salir o cerrar la fase de Elaboración el equipo tiene que pasar el hito de la Arquitectura del Ciclo de Vida. Los principales puntos que se abordan con este hito es que el equipo ha demostrado que tienen un prototipo de trabajo de extremo a extremo que muestra que el equipo tiene una estrategia viable para construir el sistema y que los interesados están dispuestos a seguir financiando el proyecto.

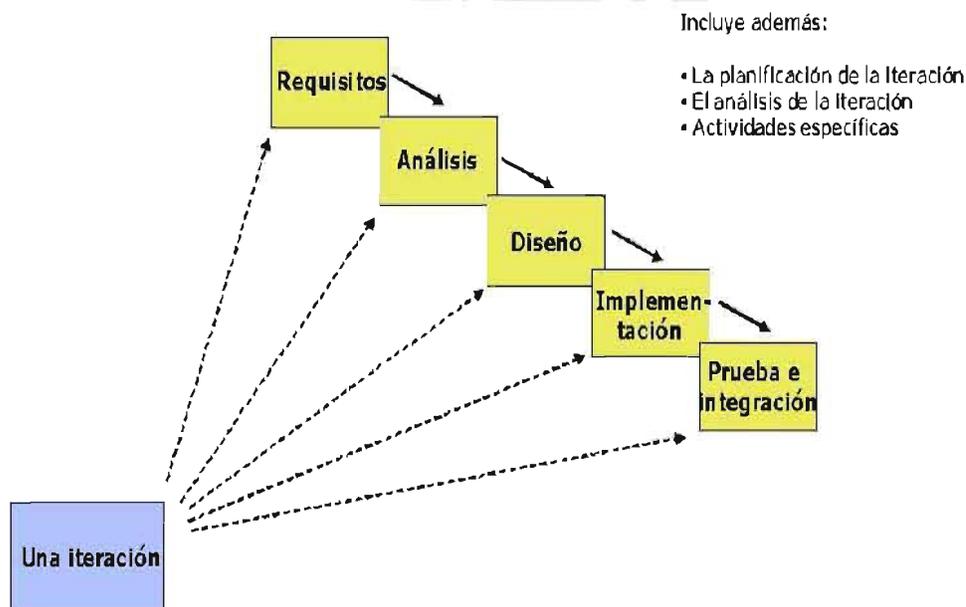
#### **2.1.3.1.3. FASE DE CONSTRUCCIÓN**

El objetivo de esta fase es construir el software operativo de forma incremental que cumpla con las necesidades de prioridad más altas de las personas involucradas en el negocio. Previo a la codificación, se tendrá que definir el modelo de análisis y diseño de la solución, para lo cual se usarán los diagramas de clases de análisis y los diagramas de clases de diseño con sus correspondientes descripciones y especificaciones.

Para esta fase se establecerán iteraciones de acuerdo a lo establecido en la fase de iniciación y a la planificación del proyecto. En paralelo con la codificación, se implementarán las pruebas necesarias para probar cada funcionalidad, de manera que una vez terminada una iteración, las funcionalidades implementadas puedan ser corroboradas y en caso de que sea necesario, se hagan las correcciones pertinentes.

Según Jacobson 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 Proceso Unificado de Rational (RUP), es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Véase figura 2.3.

**Figura 2.3:** Modelo iterativo - incremental



Fuente: [KRUTCHEN,2000]

#### 2.1.3.1.4. FASE DE TRANSICIÓN

El objetivo de esta fase es validar y desplegar el sistema en el entorno de producción. Deben hacerse pruebas extensivas a lo largo de esta fase, incluyendo las pruebas beta. Una buena afinación del proyecto tiene lugar aquí incluyendo las modificaciones a los defectos significativos.

El tiempo y esfuerzo necesarios en la Transición varía de un proyecto a otro. Sistemas internos son generalmente más simples de desplegar que sistemas externos. Los sistemas de alta visibilidad pueden requerir pruebas beta extensiva

por grupos pequeños antes de liberarse a la población más grande. La liberación de un nuevo sistema de marca puede traer consigo la compra y configuración de hardware mientras se actualiza un sistema existente que también puede traer una conversión de datos y una coordinación exhaustiva con la comunidad de usuarios.

### 2.1.3.2. DISCIPLINAS AUP

Las disciplinas son actividades que el equipo de desarrolladores debe realizar para construir, validar y entregar un software que satisfaga las necesidades de los involucrados.

#### 2.1.3.2.1. MODELADO DEL NEGOCIO

El objetivo de esta disciplina es entender el negocio de la organización, tratar el dominio del problema e identificar una solución viable para tratar el dominio del problema. Las actividades más relevantes de esta etapa son descritas en la tabla 2.2, la misma es organizada por cada fase de la metodología AUP.

**Tabla 2.2:** Actividades de la disciplina modelado

FASES	ACTIVIDADES
Iniciación	<ul style="list-style-type: none"> <li>✓ Explore el uso de <b>casos de uso</b>.</li> <li>✓ Identifique los <b>procesos del negocio</b> creando <b>diagramas de flujo de datos</b>.</li> <li>✓ Identifique las entidades principales del negocio y sus relaciones trabajando</li> <li>✓ Identifique las principales <b>reglas del negocio</b> y <b>requerimientos técnicos</b>.</li> <li>✓ Inicie un <b>glosario</b> de términos importantes técnicos y del negocio.</li> <li>✓ Comprenda la estructura orgánica a través del modelado de la organización.</li> <li>✓ De la lista de requisitos ¿Cuál priorizar?</li> <li>✓ Proponga un <b>modelo de despliegue</b> inicial.</li> </ul>
Elaboración	<ul style="list-style-type: none"> <li>✓ <b>Identificar riesgos técnicos</b>. Sus necesidades de trabajo, casos de uso y requerimientos técnicos.</li> <li>✓ <b>Modelado de la Arquitectura</b>. Como usted construye el prototipo de la arquitectura necesitará <b>modelar por lluvia de ideas</b>.</li> <li>✓ <b>Prototipado de interfaces de usuario</b>. Trabaje las principales pantallas que podría cambiar durante la construcción.</li> </ul>

<b>Construcción</b>	<ul style="list-style-type: none"> <li>✓ <b>Participación activa de interesados y modelado inclusivo</b> que usan técnicas y herramientas simples que son críticas para su negocio.</li> <li>✓ Si lo desea, puede profundizar detalles de sus <b>casos de uso</b>, quizás utilizando <b>diagramas de flujo</b> o <b>diagramas de actividad UML</b></li> <li>✓ Explore las <b>reglas del negocio</b> y los <b>requerimientos técnicos</b> en la misma forma.</li> <li>✓ Analice sistemas actuales o una <b>base de datos</b> (Situación <b>Análisis actual</b>).</li> <li>✓ En lugar de las descripciones de casos de uso, de reglas del negocio y de requerimientos técnicos, usted puede encontrar más efectivo simplemente escribir <b>casos de prueba de aceptación</b>.</li> <li>✓ Debido a que la interfaz de usuario es el sistema, es muy probable que descubra que prefieren centrarse en las pantallas e informes en lugar de mirar el desarrollo de otros productos de su trabajo.</li> <li>✓ Mantenga su <b>glosario del proyecto</b>.</li> </ul> <p><b>Es probable que desee crear:</b></p> <ul style="list-style-type: none"> <li>✓ <b>Diagrama de secuencia de UML</b>. Este diagrama representa la lógica dinámica en el código fuente.</li> <li>✓ <b>Modelo de despliegue</b>. Típicamente crear algún ordenamiento del diagrama de resumen representado en la arquitectura del sistema de despliegue/ red. Disposición física de de los equipos.</li> <li>✓ <b>Diagramas de estilo libre</b>. Comúnmente creados en pizarras y luego son borrados cuando no se necesitan. Diagramas de estilo libre son la técnica más común de dibujado.</li> <li>✓ <b>Documentos de resumen del sistema</b> típicamente incluyen diagramas de estilo libre.</li> <li>✓ <b>Diagrama de Clases UML</b>. Use una herramienta de modelado que le permita generar código fuente. Su diagrama de clases debe estar basada en un modelo de dominio.</li> <li>✓ <b>Modelo de amenazas de seguridad</b>. Si problemas de seguridad preocupan, entonces se trabajara un modelo que mitigue este aspecto.</li> <li>✓ <b>Modelo físico de datos</b>. Este es probablemente el modelo de diseño más importante, que debería considerar el uso de una herramienta CASE para desarrollar y mantener en el tiempo, sobre todo una herramienta que genera DDL código. Es posible tomar un <b>enfoque ágil para el modelado de datos</b>.</li> </ul>
<b>Transición</b>	<ul style="list-style-type: none"> <li>✓ <b>Modelado de lluvia de ideas</b>. Para tratar de entender las causas principales de un defecto.</li> <li>✓ <b>Finalice la documentación de resumen del sistema</b>. El mejor momento para finalizar su documentación general del sistema es cuando el sistema esté realmente establecido (ahora es cuando se debería de colocar todos los diagramas de estilo libre y bocetos de pizarras en una buen diagrama usando una herramienta de dibujo).</li> </ul>

Fuente: [AMBLER,2005]

### 2.1.3.2.2. IMPLEMENTACIÓN

El objetivo de esta disciplina es transformar el modelo en código ejecutable y realizar un nivel básico de pruebas, en particular pruebas unitarias. Las mismas actividades figuran en la tabla 2.3, según fases de la metodología.

Tabla 2.3: Actividades de disciplina implementación

FASES	ACTIVIDADES
Iniciación	<ul style="list-style-type: none"> <li>✓ <b>Prototipado técnico.</b> Es posible que tenga que "picar" un pequeño aspecto de un requisito con el fin de entender lo suficiente, para <b>estimar</b> el esfuerzo requerido. Estos prototipos son típicamente pequeños.</li> <li>✓ <b>Prototipado de Interfaces de Usuario.</b> El prototipo de UI (Interfaz de usuario), usualmente arroja en este punto, lo que será usado para convencer al usuario para que entienda las necesidades del usuario.</li> </ul>
Elaboración	<p><b>Probar la arquitectura.</b> Identificar la arquitectura potencial y luego probar que esta arquitectura funcione a través del desarrollo de la arquitectura del prototipo extremo a extremo para su sistema, y a la vez mitigando gran parte de los riesgos técnicos en su proyecto.</p>
Construcción	<ul style="list-style-type: none"> <li>✓ <b>Primeras pruebas.</b> Obtenga un acercamiento de la base del <b>Desarrollo Dirigido por Pruebas (TDD)</b> para todos los aspectos de la aplicación.</li> <li>✓ <b>Construya constantemente.</b> Creaciones diarias son un buen comienzo, pero idealmente usted quiera construir su sistema cada vez que el <b>código fuente</b> cambie.</li> <li>✓ <b>Evolución de la lógica de dominio.</b> Implemente su lógica de negocio en clases negocio/dominio.</li> <li>✓ <b>Evolucionar las interfaces de usuario.</b> Hacer su software tan usable como sea posible siguiendo las estrategias de diseño de interfaces de usuario y usabilidad.</li> <li>✓ <b>Evolucionar el esquema de datos.</b> Su esquema de datos debe ir evolucionado en conjunto con su dominio y código de interfaces de usuario.</li> <li>✓ <b>Desarrollo de interfaces de activos existentes.</b> Necesitará acceder frecuentemente a la funcionalidad dentro de los sistemas existente (servicios web, procedimientos almacenados, etc.).</li> <li>✓ <b>Generar el script de conversión de datos.</b> Deberá a menudo acceder a las fuentes de datos actuales.</li> </ul>
Transición	<p><b>Corregir defectos.</b> Concéntrese en la corrección de defectos encontrados como resultado de las pruebas.</p>

Fuente: [AMBLER,2005]

### 2.1.3.2.3. PRUEBAS

Realizar una evaluación objetiva para asegurar calidad. Esto incluye encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que se cumplen los requisitos. La tabla 2.4 muestra el resumen de actividades a realizar en cada fase.

Tabla 2.4: Actividades de disciplina prueba

FASES	ACTIVIDADES
Iniciación	<ul style="list-style-type: none"> <li>✓ <b>Planificación inicial de pruebas.</b> El objetivo principal es identificar cuántas pruebas necesita hacer, quien será el responsable, el nivel de participación requerido por los usuarios, y los tipos de herramientas y los entornos necesarios.</li> <li>✓ <b>Exanimación inicial de los productos de trabajo del proyecto.</b> Hacia el final de esta fase el plan de proyecto, la visión y mucho más deben estar disponibles.</li> <li>✓ <b>Exanimación inicial de modelos.</b> A un alto nivel, el <b>modelo de requerimientos inicial</b>, y quizás un modelo de arquitectura inicial, deben ser producidos por su esfuerzo de <b>modelado</b>.</li> </ul>
Elaboración	<ul style="list-style-type: none"> <li>✓ <b>Validación de la Arquitectura.</b> Usted debe tomar un enfoque de <b>desarrollo controlado por pruebas (TDD)</b> para construir su prototipo técnico el cual compruebe la arquitectura de su sistema.</li> <li>✓ <b>Evolucione su modelo de pruebas.</b> Su equipo deberá desarrollar un <b>paquete de pruebas de regresión</b>, compuesta por la unidad de pruebas de su <b>desarrollo controlado por pruebas (TDD)</b> en la <b>implementación</b>, su aceptación de pruebas de su <b>modelado</b>, y las pruebas de su sistema (funcionamiento, integración, carga, .. pruebas).</li> </ul>
Construcción	<ul style="list-style-type: none"> <li>✓ <b>Pruebas de software.</b> Además de las unidades de prueba de los desarrolladores deberá hacer pruebas de instalación del script de <b>despliegue o liberación</b>, sistema de pruebas de esfuerzos tales como la carga / pruebas de tensión y las pruebas de función, y sus pruebas de aceptación de usuario. Lo más común es que promueva su código en un ambiente de pruebas de pre-producción</li> <li>✓ <b>Evolucione su modelo de pruebas.</b></li> </ul>
transición	<ul style="list-style-type: none"> <li>✓ <b>Validación del sistema.</b> Usted se concentrará en las "<b>grandes pruebas</b>" de actividades tales como las del sistema, las de integración y las de aceptación, y las pruebas piloto/beta.</li> <li>✓ <b>Validación de la documentación.</b> Su documentación de sistema (vista general del sistema, usuarios, soporte, y documentación de operaciones), y sus materiales de capacitación necesitará validarlos.</li> <li>✓ <b>Analice su modelo de pruebas.</b> Va a tener que seguir ejecutando de paquete de pruebas de regresión y actualizarlo tanto como necesite, hasta que su sistema esté listo para ser <b>desplegado</b> en producción.</li> </ul>

Fuente: [AMBLER,2005]

#### 2.1.3.2.4. DESPLIEGUE

El objetivo de esta disciplina es planificar el despliegue del sistema y ejecutar el mismo para poner el sistema a disposición de los usuarios finales, es decir; que módulos serán desplegados prioritariamente, quienes operaran y donde se desplegara. Las mismas se resumen en la tabla 2.5.

Tabla 2.5: Actividades de disciplina despliegue

FASES	ACTIVIDADES
Iniciación	<ul style="list-style-type: none"> <li>✓ <b>Identificar el rango liberación potencial.</b> Definir el rango de liberación potencial en las primeras fases del proyecto, lo ayudará en los esfuerzos de planeación.</li> <li>✓ <b>Comience con un plan de entregables de alto nivel.</b> Este esfuerzo se debe enfocar en planificar los entregables de su sistema, identificando el rango de liberación potencial.</li> </ul>
Elaboración	<p><b>Actualizar el plan de implementación.</b> Una parte importante de definir la arquitectura es definir la configuración de los entregables del sistema, pueda ser una configuración cliente/servidor para usuarios internos, una interfaz basada en HTML para usarla en Internet, y una versión "stand-alone" usuarios únicos para quienes no tengan conexión. Además, la organización de los componentes del software y hardware actuales, es decir como están. La misma ayuda a identificar los diferentes tipos de instalación que se deben realizar, y a la vez ofrece una imagen global de proceso de implementación.</p>
Construcción	<ul style="list-style-type: none"> <li>✓ <b>Desarrollar el script de instalación.</b> Como desarrolla el sistema debe también escribir y probar el <b>script de instalación</b> necesario para entregarlo en la pre producción del ambiente de pruebas.</li> <li>✓ <b>Desarrollar notas del entregable.</b> Sus <b>notas de entregables</b> deben resumir los avances que posee el entregable actual del sistema.</li> <li>✓ <b>Desarrollar documentación inicial.</b> Además de entregar software funcional, también se debe entregar la <b>documentación del sistema (operaciones, soporte, visión general, y la documentación al usuario)</b>. Usted no debe dedicar mucho tiempo a la documentación por el momento, porque su sistema está en desarrollo.</li> <li>✓ <b>Actualice su plan.</b> De manera como progresa el desarrollo del sistema, debe avanzar su plan de implementación.</li> <li>✓ <b>Implementar el sistema en entornos de pre-producción.</b> Debe entregar regularmente el sistema en un ambiente de pre-producción para efectuar pruebas y llevar a cabo el control de calidad necesario, así como realizar demostraciones a los involucrados.</li> </ul>
Transición	<ul style="list-style-type: none"> <li>✓ <b>Concluir el proceso de implementación.</b> Para concluir este proceso debe definir una línea base de entrega como referencia, las actividades de la <b>administración de la configuración</b>, y realizar una "última" revisión de software.</li> <li>✓ <b>Finalizar la documentación.</b> La mayor parte de la <b>documentación del sistema</b> es generalmente realizada durante esta etapa, debido a que la funcionalidad del sistema se estabiliza.</li> <li>✓ <b>Anunciar la implementación.</b> Se debe anunciar el calendario de implementación de manera anticipada e incluyendo las fechas estimadas de capacitación e instalación.</li> <li>✓ <b>Capacitar el personal.</b> Capacitar los clientes o usuarios de su proyecto, así como a la administración, equipo de operaciones y equipo de soporte, es siempre parte importante de la implementación.</li> <li>✓ <b>Puesta en producción.</b> En este punto se debe realizar cualquier conversión o migración de datos, y puede ser todo de una vez, un trabajo por lotes o una conversión gradual de los datos, conforme lo requieran los usuarios. Usted puede decidir ejecutar su nuevo sistema en paralelo con su actual sistema varias semanas, para asegurarse de que funciona realmente en producción.</li> </ul>

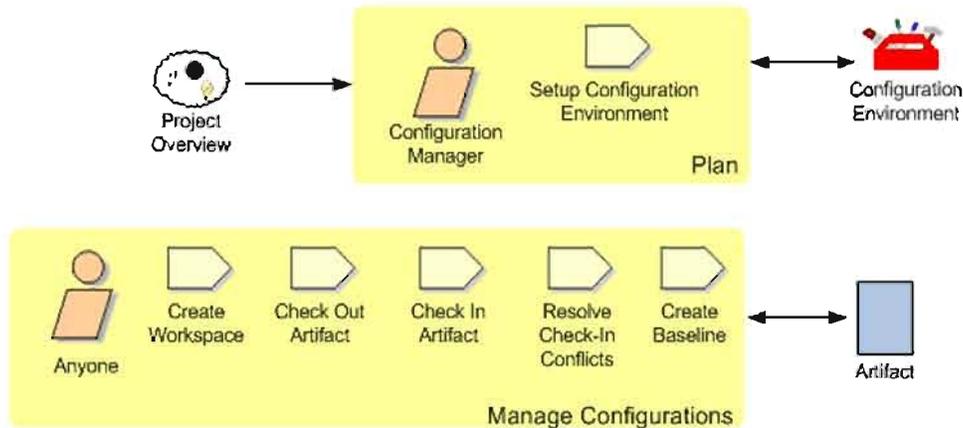
Fuente: [AMBLER,2005]

### 2.1.3.2.5. GESTIÓN DE CONFIGURACIÓN

El objetivo de esta disciplina es la gestión de acceso a los artefactos del proyecto. Esto no sólo incluye el seguimiento de las versiones de los artefactos

sino también controlar y gestionar los cambios en ellos. La misma podemos ver en la figura 2.4.

**Figura 2.4:** Actividades de administración de configuración



**Fuente:** [AMBLER,2005]

Las actividades más significativas que destacamos en gestión de configuración (CM por sus siglas en inglés) esta disciplina son:

- ✓ La estructura de directorios apropiada, la cual debe seguir los lineamientos corporativos, necesita ser creada para el equipo del proyecto.
- ✓ Los miembros del equipo del proyecto necesitan tener acceso al folder o directorios del proyecto y cualquier otro software cliente instalados en sus ordenadores
- ✓ Los miembros del equipo del proyecto también necesitan ser entrenados con los conceptos básicos de CM y las herramientas necesarios.
- ✓ Su repositorio de CM necesitará ser instalado si este aún no se ha instalado.
- ✓ Cada uno debe poner su trabajo bajo el control de CM en una base regular, verificar las entradas y salidas en cada caso, resolver conflictos de actualización cuando se requiera, y presupuestar los productos del proyecto cuando las versiones más actualizadas estén aprobadas.

### 2.1.3.2.6. GESTIÓN DE PROYECTO

El objetivo de esta disciplina es dirigir las actividades que tienen lugar dentro del proyecto. Esto incluye gestionar riesgos, dirigir a las personas y coordinar las personas y sistemas fuera del alcance del proyecto para asegurar que se entrega a tiempo y dentro del presupuesto. Las mismas se detallan en la tabla 2.6, sobre administración de proyecto.

Tabla 2.6: Actividades de disciplina gestión de proyecto

FASES	ACTIVIDADES
Inicio	<ul style="list-style-type: none"> <li>✓ <b>Inicie conformando el equipo.</b> Personas con habilidades de <b>modelado</b> para requerimientos iniciales y personas técnicas para identificar una arquitectura potencial.</li> <li>✓ <b>Crear relaciones con sus involucrados del proyecto.</b> El soporte a los usuarios y la <b>participación</b> es crítica para su éxito.</li> <li>✓ <b>Determinar la factibilidad financiera, técnica, operacional, y política.</b></li> <li>✓ <b>Desarrollar un cronograma de alto nivel para todo el proyecto.</b></li> <li>✓ <b>Desarrollar un plan detallado de iteración para la siguiente iteración.</b></li> <li>✓ <b>Manejo de riesgo.</b> Identificar riesgos y desarrollar estrategias para enfrentarlos.</li> <li>✓ <b>Obtener financiamiento y apoyo de los involucrados.</b> Mostrar que usted entiende el alcance, su viabilidad, los riesgos, y que tiene un plan viable para proceder.</li> <li>✓ <b>Cerrar esta fase.</b> Cuando los involucrados formalicen el proyecto.</li> </ul>
Elaboración	<ul style="list-style-type: none"> <li>✓ <b>Construya el equipo.</b> Personas con habilidades de análisis, desarrollo e implementación.</li> <li>✓ <b>Proteger el equipo.</b></li> <li>✓ <b>Obtener recursos.</b> Su equipo necesita financiación, instalaciones (por ejemplo salas y cubículos), hardware, software, y así sucesivamente para hacer su trabajo.</li> <li>✓ <b>Manejo del riesgo.</b> Continúe los esfuerzos de administración del riesgo.</li> <li>✓ <b>Actualice el plan del proyecto.</b> Continúe las actividades planeadas.</li> <li>✓ <b>Cerrar esta fase.</b> Demostrar que su arquitectura funciona y que se está enfrentando correctamente los principales riesgos del proyecto.</li> </ul>
Construcción	<ul style="list-style-type: none"> <li>✓ <b>Administre el equipo.</b> Continúe desarrollando el equipo manténgase protegiéndolos y proveyéndoles los recursos que necesitan.</li> <li>✓ <b>Manejo del riesgo.</b> Continúe los esfuerzos de administración del riesgo.</li> <li>✓ <b>Actualizar su plan de proyecto.</b> Debe considerar las necesidades de sus equipos de operación y soporte, capacitación del usuario final, y el plan de pruebas al sistema piloto.</li> <li>✓ <b>Cerrar esta fase.</b> Mostrar que su equipo ha desarrollado un sistema que está potencialmente listo para implementarse en producción.</li> </ul>
Transición	<ul style="list-style-type: none"> <li>✓ <b>Administrar el equipo.</b> Incluir encargados de pruebas e implementadores, que monitoreen los avances.</li> <li>✓ <b>Cerrar esta fase.</b> Mostrar que su sistema ha pasado las pruebas y es aceptable para los involucrados.</li> <li>✓ <b>Iniciar el próximo ciclo del proyecto.</b> Los sistemas se desarrollan y se ponen en producción de manera incremental.</li> </ul>

Fuente: [AMBLER,2005]

### 2.1.3.2.7. ENTORNO

El objetivo de esta disciplina es apoyar el resto del esfuerzo asegurando que el proceso, la orientación (normas, estándares y guías) y las herramientas (software, hardware, etc.) adecuadas estén disponibles para el equipo cuando son necesarias. Las mismas podemos verlos en la tabla 2.7.

Tabla 2.7: Actividades de disciplina entorno

FASES	ACTIVIDADES
Iniciación	<ul style="list-style-type: none"><li>✓ <b>Configure el entorno de trabajo.</b> Esta será una tarea permanente ya que hay gente que se añade al equipo en el tiempo.</li><li>✓ <b>Identifique la categoría del proyecto.</b> Muchas organizaciones desarrollan varias versiones de sus procesos de software, por ejemplo uno para equipos pequeños, uno para reemplazar sistemas legales, otro para sistemas de plataforma comercial, etc.</li></ul>
Elaboración	<ul style="list-style-type: none"><li>✓ <b>Evolucionar el entorno de trabajo.</b> Necesitará evolucionar su entorno instalando nuevas herramientas, o remover las herramientas que ya no necesitan.</li><li>✓ <b>Ajuste de los procesos de materiales.</b> Se debe ajustar <b>AUP</b> para cumplir con las necesidades del equipo. Esto puede incluir materiales del proceso de AUP (por ejemplo esta página), se debe elegir entre escribir un documento corto que no se quiere hacer, o se debe simplificar eligiendo la cosa correcta en el momento correcto.</li></ul>
Construcción	<ul style="list-style-type: none"><li>✓ <b>Apoyar al equipo.</b> Miembros del equipo necesitaran ayuda para utilizar y / o la configuración de diversas herramientas, plantillas de documentación y seguir la <b>guía</b> de su empresa.</li><li>✓ <b>Establecer el ambiente de capacitaciones.</b> Entrenar al usuario, personal de soporte y el personal de operación. La capacitación debe requerir espacios de entrenamiento y versiones de entrenamiento del sistema disponibles, frecuentemente en la fase de <b>Transición</b>.</li></ul>
Transición	<ul style="list-style-type: none"><li>✓ <b>Configuración de las operaciones y soporte de los entornos.</b> Personal de soporte, y operación, para simular reportes de defectos en una forma segura.</li><li>✓ <b>Recobrar licencias de software.</b> A medida que su proyecto llega a la conclusión puede ser necesario desinstalar las licencias de software los equipos que ya no necesitan el software para que las licencias puedan estar disponibles a los demás dentro de su organización.</li></ul>

Fuente: [AMBLER,2005]

Por lo expuesto, las disciplinas AUP utilizan UML en etapas como el modelado por tanto, describimos las partes más sobresalientes.

### 2.1.4. LENGUAJE DE MODELADO UNIFICADO (UML)

El UML se define como “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software...”[Larman,1999]. Es un

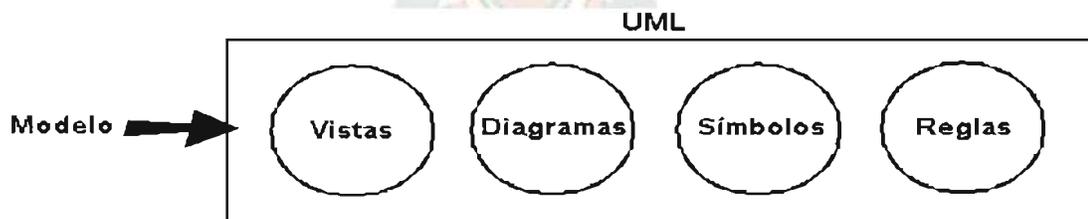
sistema dotacional destinados a los sistemas de modelado que utilizan conceptos orientados a objetos.

La notación UML, se ha convertido desde finales de los 90 en un estándar para modelar con tecnología orientada a objetos todos aquellos elementos que configuran la arquitectura de un sistema de información y, por extensión, de los procesos de negocio de una organización. De la misma manera que los planos de un arquitecto disponen el esquema director a partir del cual levantamos un edificio, los diagramas UML suministran un modelo de referencia para formalizar los procesos, reglas de negocio, objetos y componentes de una organización.

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un lenguaje de modelado, la misma es ilustrada en la de la figura 2.5, un lenguaje de modelado consiste de vistas, diagramas, símbolos y reglas que indican cómo utilizar los elementos de UML para modelar requerimientos.

**Figura 2.5: Modelo UML**



**Fuente:** [JRB,2000]

## Vistas

Las vistas muestran diferentes aspectos del sistema modelado. Una vista no es una gráfica, pero sí una abstracción que consiste en un número de diagramas y todos esos diagramas juntos muestran una "fotografía" completa del sistema.

## Diagramas

Los diagramas son las gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que son utilizados en combinación para proveer todas las vistas de un sistema: diagramas de caso de uso, de clases, de objetos, de estados, de secuencia, de colaboración, de actividad, de componentes y de distribución [JRB,2000].

## Símbolos

Los conceptos utilizados en los diagramas son los elementos de modelo que representan conceptos comunes orientados a objetos, tales como clases, objetos y mensajes, y las relaciones entre estos conceptos incluyendo la asociación, dependencia y generalización.

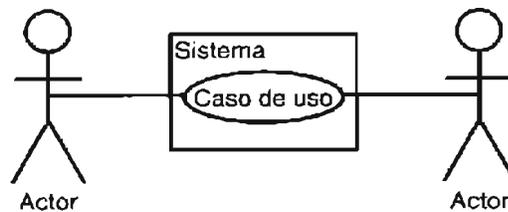
## Reglas

Proveen comentarios extras, información o semántica acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

### 2.1.4.1. DIAGRAMA DE CASOS DE USO

Los diagramas de casos de uso modelan la funcionalidad del sistema según perciban los usuarios externos llamados actores. A su vez este explica gráficamente un conjunto de casos de uso, actores y sus relaciones: es decir, muestra quien puede hacer qué y las relaciones que existen entre acciones. Como se muestra en la figura 2.6.

**Figura 2.6:** Diagrama de casos de uso

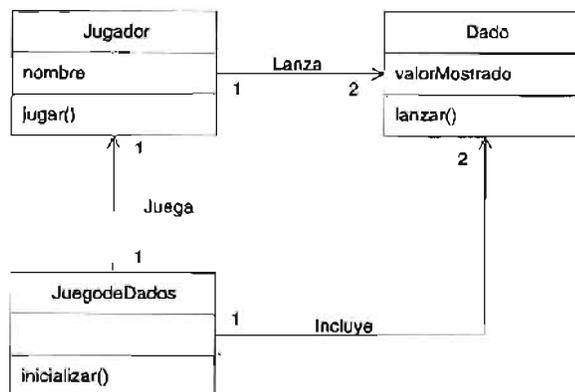


Fuente: [SCHMULLER,2001]

#### 2.1.4.2. DIAGRAMA DE CLASES

El diagrama de clases muestra las clases que componen el sistema y cómo se relacionan entre sí. Un diagrama de Clases es una representación gráfica de la vista estática, que muestra una colección de elementos declarativos del modelo, son fronteras naturales a la hora de formar diagramas. Ver figura 2.7.

**Figura 2.7:** Diagrama de clases



Fuente: [LARMAN,1999]

#### Asociación de clases

Una asociación representa una relación entre clases, la misma puede suceder de muchas formas. Cada extremo de la relación tiene un valor de multiplicidad, que indica cuantos objetos de ese lado de la asociación esta relacionadas con un objeto del lado contrario.

**Tabla 2.8:** Valores de multiplicidad

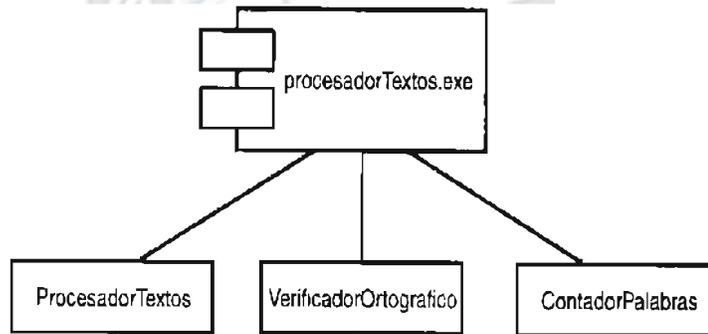
Relación	Multiplicidad
*	Cero o mas; muchos
1...*	Uno o mas
1...40	De uno a cuarenta
5	Exactamente 5
3,5,8	Exactamente tres, cinco u ocho

**Fuente:** [LARMAN,1999]

### 2.1.4.3. DIAGRAMA DE COMPONENTES

Muestra la organización y las dependencias entre un conjunto de componentes. Se usan para agrupar clases en componentes o módulos. A continuación se observa los elementos para el diseño del respectivo diagrama en la figura 2.8.

**Figura 2.8:** Diagrama de Componentes

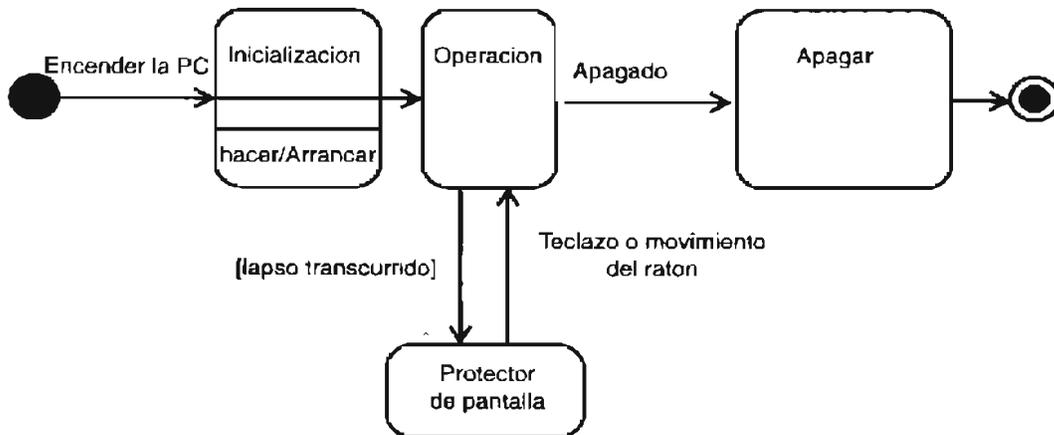


**Fuente:** [SCHMULLER,2001]

### 2.1.4.4. DIAGRAMA DE ESTADOS

Los objetos en los sistemas modifican sus estados como respuestas a sucesos y al tiempo. El diagrama de UML captura estos cambios de estado, es decir para analizar los cambios de estado de los objetos. Muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos [RJB,2000]. La figura 2.9, muestra los eventos en el encendido de un ordenador personal.

Figura 2.9: Diagrama de estados



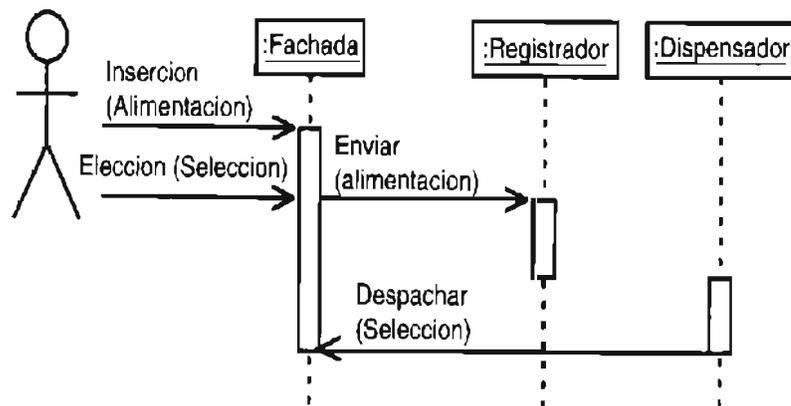
Fuente: [SCHMULLER,2001]

#### 2.1.4.5. DIAGRAMA DE SECUENCIA

Los diagramas de secuencia muestran un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema. Los diagramas se centran en los eventos que trascienden las fronteras del sistema y que fluyen de los actores a los sistemas [Larman1999].

La figura 2.10 muestra un diagrama de secuencia del caso de uso "comprar gaseosa".

Figura 2.10: Diagrama de secuencia

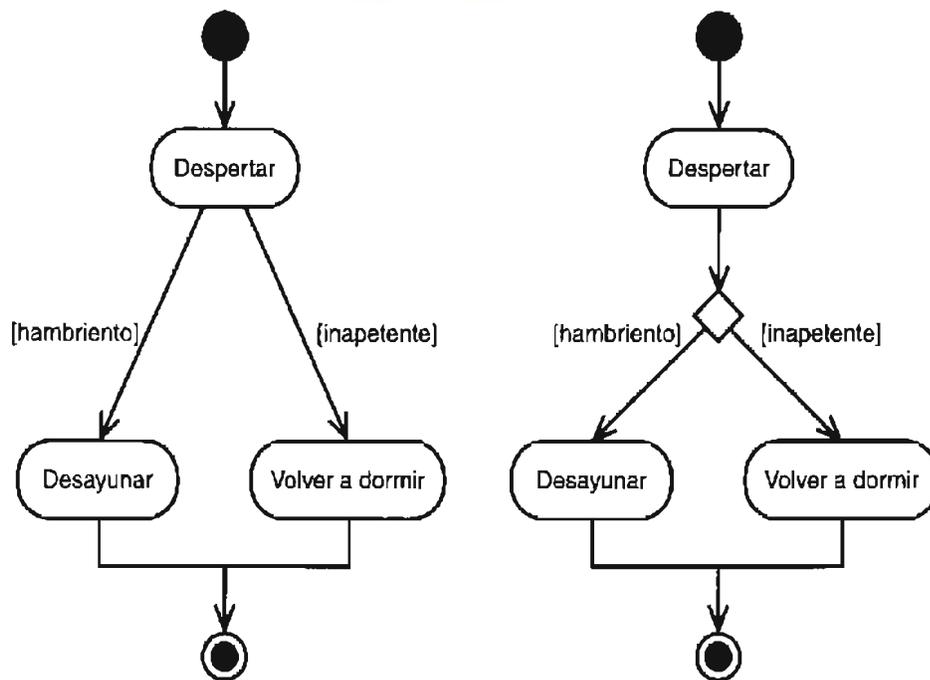


Fuente: [SCHMULLER,2001]

#### 2.1.4.6. DIAGRAMA DE ACTIVIDADES

Un diagrama de actividad es un caso especial del diagrama de estados, simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos [RJB,2000]. La misma podemos en la figura 2.11.

Figura 2.11: Diagrama de Actividades

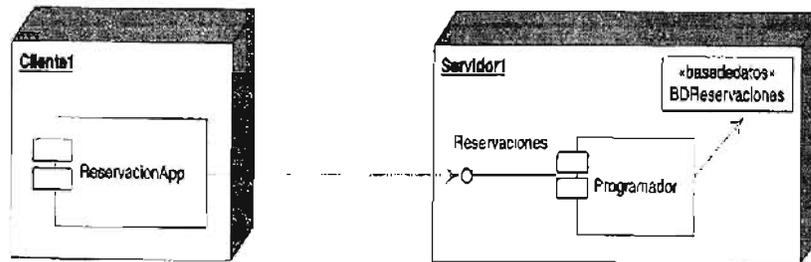


Fuente: [SCHMULLER,2001].

#### 2.1.4.7. DIAGRAMA DE DESPLIEGUE

Muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo. Se utiliza para identificar Sistemas de Cooperación: Durante el proceso de desarrollo el equipo averiguará de qué sistemas dependerá el nuevo sistema y que otros sistemas dependerán de él. La Figura 2.12 ilustra un ejemplo de diagrama de despliegue.

**Figura 2.12:** Diagrama de despliegue



Fuente: [LARMAN,1999]

### 2.1.5. INGENIERIA WEB

Los sistemas y aplicaciones basadas en la web ofrecen un complejo contenido y funcionalidad a una amplia población de usuarios. Ingeniería Web (Iweb) es el proceso con el que se crean aplicaciones Web (webApps) de alta calidad. La ingeniería Web toma prestado muchos conceptos y principios fundamentales de ingeniería de sistemas. Además el proceso de ingeniería Web acentúa actividades técnicas y administrativas similares [PRESSMAN,2005].

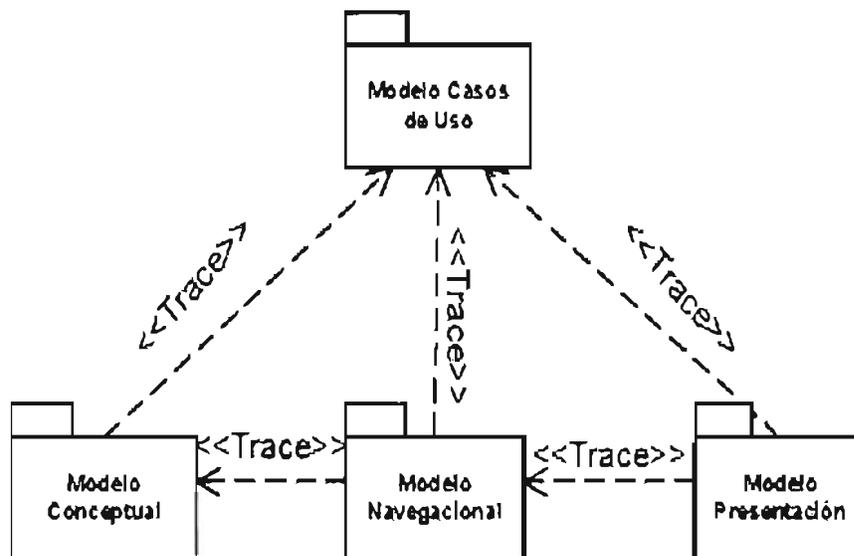
La ingeniería Web aplica sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitosos de sistemas y aplicaciones basados en la Web de alta calidad.

### 2.1.6. METODOLOGÍA UWE (UML-BASED WEB ENGINEERING)

La metodología UWE (Ingeniería Web Basada en UML) es una propuesta desarrollada por Nora Koch basada en estándares UML y en el proceso unificado para modelar aplicaciones web. La misma esta formada por una notación para especificar el dominio y un modelo para llevar a cabo el desarrollo del proceso de modelado.

Las aplicaciones Web tienen características especiales, que van desde el entorno en que se operan hasta los requerimientos de usuario. El UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizarán en las vistas especiales para el modelado de Aplicaciones Web. Además, permite que un diseñador Web pueda hacer uso de otras técnicas UML agregando otras vistas de la aplicación, es decir que UWE no limita el número de vistas de la aplicación. Véase la figura 2.13.

Figura 2.13: Modelo de aplicación UWE



Fuente: [KOCH,2001].

#### 2.1.6.1. MODELO DE CASOS DE USO

El modelo de casos de uso nos permite describir la especificación de requerimientos funcionales de la aplicación Web, identificando a los usuarios (actores) e indica las interacciones de estos con el sistema. Esto da lugar a los diagramas de caso, medio sistemático e intuitivo para capturar requisitos funcionales.

### 2.1.6.2. MODELO CONCEPTUAL

La construcción de un modelo lógico–conceptual del dominio se lleva a cabo de acuerdo a los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizan en la aplicación Web, es decir, los objetos que son relevantes para la realización de una actividad o que son el resultado de una de ellas.

Este modelo especifica cómo se encuentra relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web. Un diagrama de clases en UML se utiliza para representar gráficamente el modelo conceptual, como en la figura 2.14.

**Figura 2.14:** Clase de un modelo conceptual

<b>NOMBRE DE LA CLASE</b>
<b>Atributos</b>
<b>Operaciones</b>
<b>Variantes</b>

**Fuente:** [KOCH,2001]

### 2.1.6.3. MODELO NAVEGACIONAL

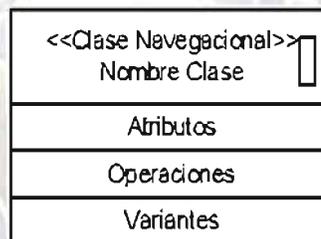
El modelo de navegación especifica, que objetos pueden ser visitados mediante la navegación a través de la aplicación Web y las asociaciones entre ellos. Los modelos de navegación son representados por los diagramas de clases estereotipadas. Este modelo de UWE es el más importante, pues con el se pueden representar elementos estáticos, a la vez se pueden incorporar lineamientos semánticos de referencia para las funcionalidades dinámicas de una aplicación Web. El modelo esta comprendido por el modelo de espacio navegacional y el modelo de estructura navegacional.

## Modelo de espacio de navegación

El modelo de espacio de navegación define los caminos de navegación (asociaciones de navegación directa) entre los diferentes objetos de la navegación. El modelo del espacio navegacional se construye con las clases de navegación y asociaciones entre las mismas y son representadas por un diagrama de clase en UML. Los elementos de este modelo son: Clase de navegación, asociación de navegación, índice, vuelta guiada, consulta, menú.

**Clase de Navegación:** Es una clase cuya instancia puede ser visitada por el usuario durante la navegación, la misma debe tener el mismo nombre que la correspondiente clase del modelo conceptual. Ver figura 2.15.

**Figura 2.15:** Clase de Navegación



Fuente: [KOCH,2001]

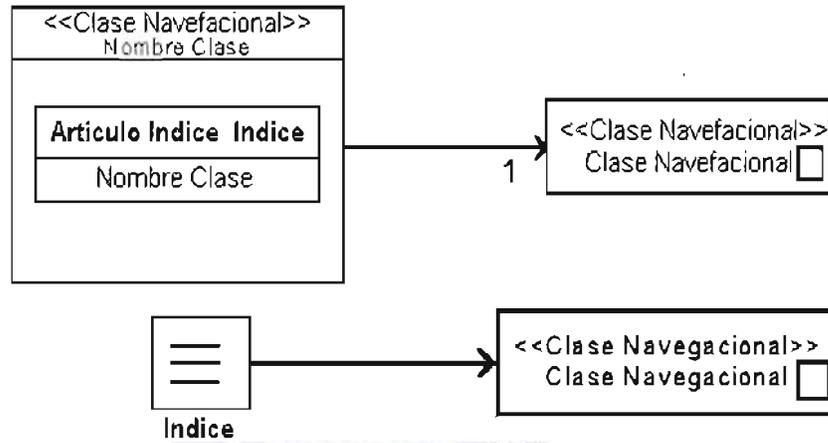
**Asociación de navegación:** Estos representan la navegabilidad directa entre las clases de navegación inicial y la clase de navegación final, en este modelo las asociaciones se representan por una flecha.

## Modelo de estructura de navegación

El Modelo de estructura de navegación detalla la estructura de acceso que se usan en la navegación como son los índices, visitas guiadas, preguntas y menús.

**Índices:** Un índice permite acceso directo a las instancias de una clase de navegación, en la figura 2.16 vemos la notación larga y corta.

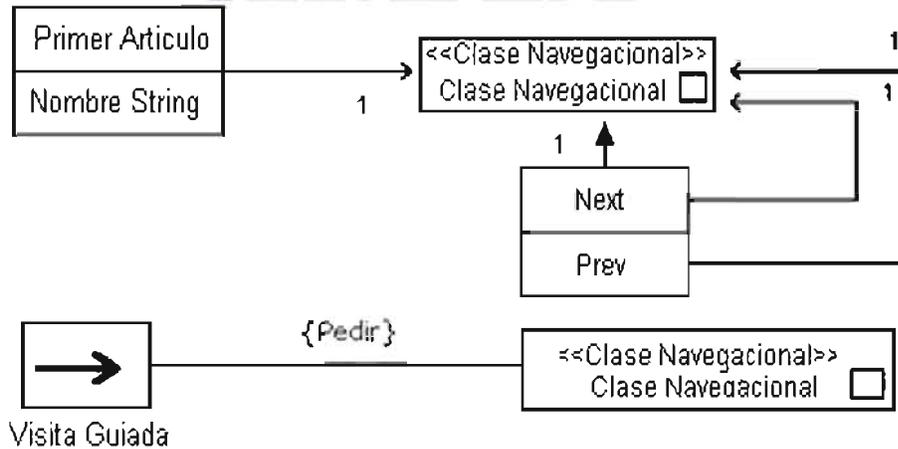
**Figura 2.16:** Clase de navegación notación larga y corta



Fuente: [Koch,2001]

**Visita Guiada:** Estos elementos proporcionan acceso secuencial a instancias de una clase de navegación. Las vueltas guiadas pueden ser controladas por el sistema o el usuario. Ver figura 2.17.

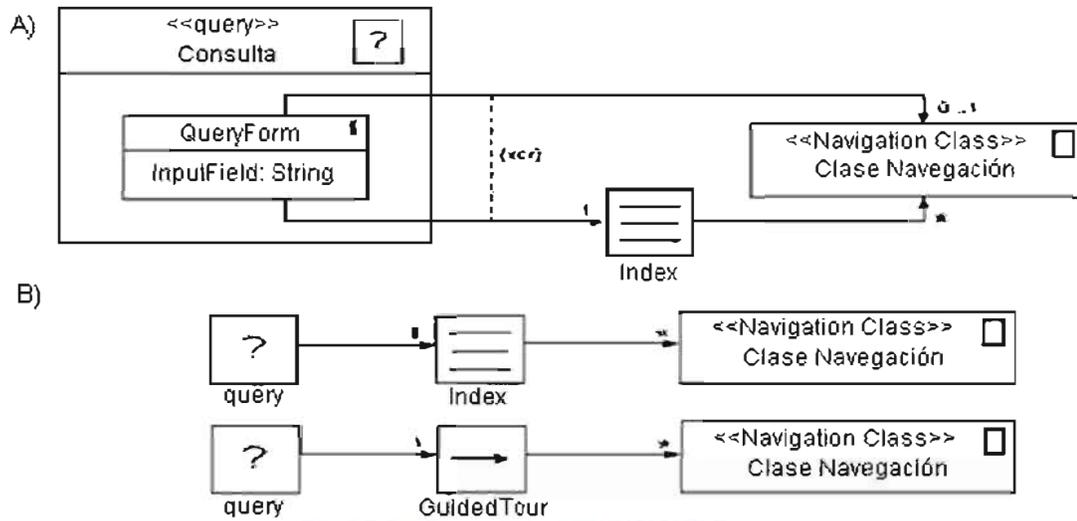
**Figura 2.17:** Clase visita guiada y visita guiada en notación corta



Fuente: [KOCH, 2001].

**Consulta:** Una consulta (Query) se encuentra modelada por una clase que tiene como una serie de preguntas como atributo. Esta consulta puede surgir de una operación de selección. Ver figura 2.18.

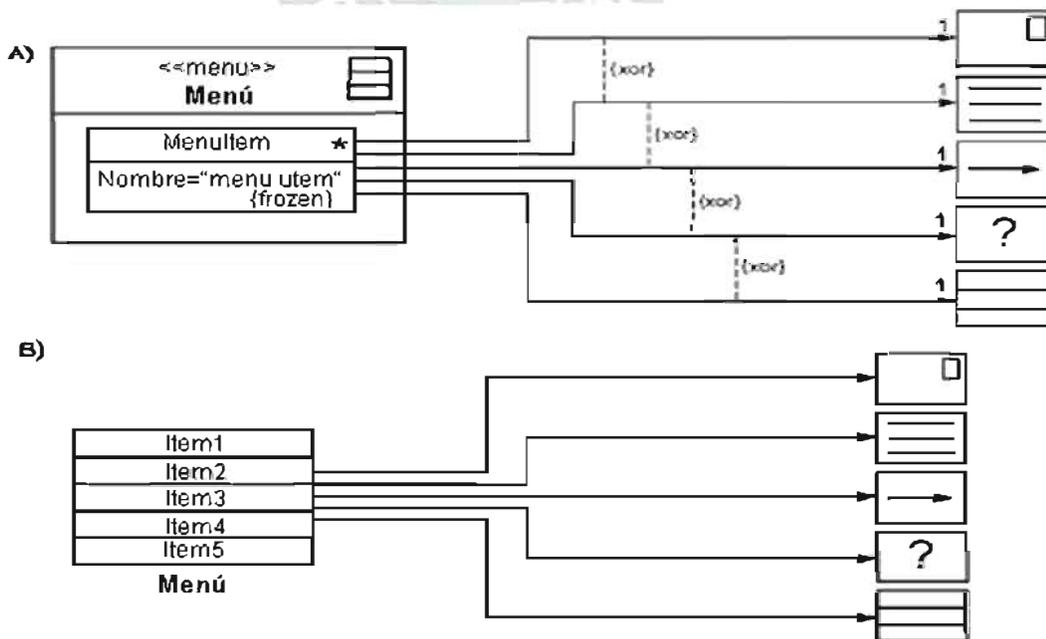
**Figura 2.18:** Clase consulta (A) y consulta en notación corta (B)



Fuente: [KOCH, 2001].

**Menú:** es un índice de un conjunto de elementos heterogéneos, tal como un índice, una visita guiada, una consulta u otro menú. Cada elemento tiene un nombre y un enlace propio a las instancias de una clase de navegación o a los propios elementos del acceso. Véase figura 2.19.

**Figura 2.19:** Clase Menú (A) y su notación corta (B)



Fuente: [KOCH, 2001].

#### **2.1.6.4. MODELO DE PRESENTACIÓN**

El modelo de presentación proporciona una vista abstracta sobre la interfaz de usuario (IU) de una aplicación web. El diseño de presentación soporta la construcción de un modelo de presentación basado en el modelo de estructura de navegación e información adicional, se recolecta durante el análisis de requerimientos. El modelo de presentación consiste en un conjunto de vistas que muestran el contenido y la estructura de los nodos simples, es decir como cada nodo es presentado al usuario y como el usuario puede interactuar con ellos [KOCH,2001].

El objetivo es visualizar la organización de la estructura de la aplicación Web de un forma mas intuitiva que con el modelo de estructura de navegación. Este paso consiste en modelar las fases de la presentación mostrando donde se presentaran al usuario los objetos de navegación y los elementos de acceso, por ejemplo en el marco de la ventana se muestra el contenido y que será reemplazado cuando se accione un enlace. En otras palabras el modelo de presentación nos ayuda a diseñar formularios de interacción del sistema con el usuario, para el cual utiliza estereotipos de diseño de formulario.

#### **2.1.7. SEGURIDAD DE LA INFORMACIÓN**

La información es un recurso fundamental en una institución, los datos de que se almacenaran como resultado del seguimiento a los estudiantes, por lo tanto su seguridad es vital, para el cual es necesario ver los niveles de seguridad que existe en la institución. La seguridad es responsabilidad de todos aquellos que están en contacto con el sistema, y sólo es tan buena como la conducta más indefinida o la política en la organización. La seguridad tiene tres aspectos interrelacionados: físico, lógico y conductual. Los tres deben trabajar juntos si la calidad de seguridad permanece alta [KENDALL,2005].

### **2.1.7.1. SEGURIDAD FÍSICA**

Se refiere a la seguridad de los componentes físicos del sistema, puede incluir acceso controlado a las salas de cómputo por medio de signos legibles por la máquina o un registro de entrada y salida del sistema por un humano, usando cámaras de televisión de circuito cerrado para supervisar las áreas de la computadora y frecuentemente apoyando los datos y almacenando los respaldos en un área a prueba de fuego o a prueba de agua.

- ✓ No todas las personas pueden acceder a las computadoras de kardex, sin previa autorización.
- ✓ El servidor debe ser manejado por el personal autorizado para la administración de base de datos.

Además, se debe asegurar para que un usuario típico no pueda mover los equipos y se debe garantizar el suministro ininterrumpido de energía eléctrica.

### **2.1.7.2. SEGURIDAD LÓGICA**

Se refiere a medidas de acceso a los recursos y a la información y al uso correcto de los mismos, son restricciones enfocadas a minimizar el uso no autorizado del sistema. Los controles lógicos en la mayoría de los casos son contraseñas o códigos de autorización de alguna clase. Cuando se usan, permiten al usuario entrar al sistema o a una parte particular de una base de datos con una contraseña correcta.

Sin embargo, las contraseñas se manejan de manera descuidada en muchas organizaciones. “Los empleados han escuchado por casualidad gritar una contraseña en las oficinas atestadas, grabar las contraseñas para sus pantallas de despliegue y compartir las contraseñas personales con empleados autorizados que han olvidado las suyas” [KENDALL,2005].

Para mejorar el manejo de las mismas utilizaremos el algoritmo de encriptación existe MD5.

## **ALGORITMO MD5**

En el caso de datos realmente escondidos, si su representación original no se necesita, los resúmenes criptográficos pueden llegar a considerarse también. El ejemplo clásico de gestión de resúmenes criptográficos es el almacenamiento de secuencias MD5 de una contraseña en una base de datos, en lugar de la contraseña misma.

### **2.1.7.3. SEGURIDAD CONDUCTUAL**

Las expectativas conductuales de una organización están implícitas en sus manuales de política e incluso en letreros anunciados en los carteles de anuncios. Sin embargo, la conducta que los miembros de la organización asimilan también es crítica para el éxito de los esfuerzos de seguridad.

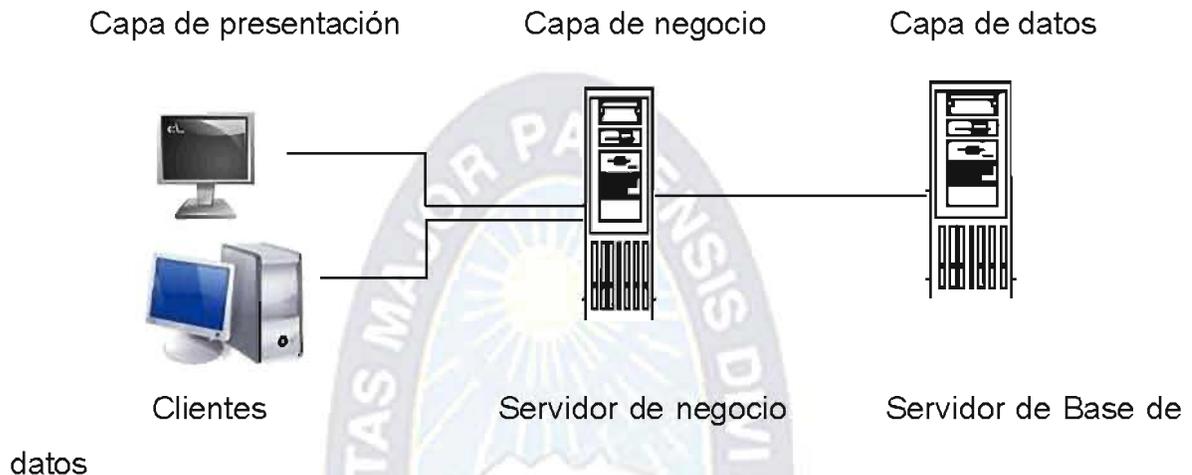
La seguridad puede empezar con la identificación de empleados que eventualmente tendrán acceso a las computadoras, datos e información, para asegurar que sus intereses son consistentes con los intereses de la organización y que entienden por completo la importancia de llevar a cabo los procedimientos de seguridad. Se deben escribir, distribuir y actualizar las políticas con respecto a la seguridad para que los empleados estén totalmente conscientes de las expectativas y responsabilidades.

### **2.1.8. ARQUITECTURA TRES CAPAS**

El sistema utiliza la arquitectura tres-capas, las mismas están compuestas por capas de: presentación, lógica y de datos. La capa de presentación interactúa y realizan peticiones del cliente a los componentes de la capa intermedia. Los

componentes de la capa intermedia manipulan la lógica del negocio haciendo peticiones a la base de datos. Véase figura 2.20.

**Figura 2.20:** Arquitectura 3-capas



**Fuente:** Elaboración propia

Se utilizará el protocolo TCP/IP estándar en redes para la comunicación de los clientes con el servidor.

### **2.1.9. CALIDAD DEL SISTEMA**

El termino calidad es “un concepto confuso condicionado por el contexto de la situación, y multidimensional” [PIATTINI,2003]. Por otro lado Pressman define calidad como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares explícitamente documentados y con las características implícitas que se espera en todo software desarrollado.

La estándar ISO-9126 que identifica cinco factores clave de calidad de software, los mismos coinciden en con los definidos para la evaluación de calidad de aplicaciones web. Las mismas consideran factores de: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad.

#### **2.1.9.1. FUNCIONALIDAD**

Este atributo valora el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global. La funcionalidad es el grado en que el software satisface las necesidades que indican los siguientes sub-atributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

#### **2.1.9.2. CONFIABILIDAD**

Se establece, hasta donde se puede esperar que un programa lleve a cabo su función con la exactitud requerida. En términos estadísticos como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico. Este atributo mide la cantidad de tiempo que el software está disponible para usarlo según los siguientes sub-atributos: madurez, tolerancia a las fallas, y facilidad de recuperación.

#### **2.1.9.1. USABILIDAD**

La usabilidad es el esfuerzo necesario para aprender a operar con el sistema, preparando los datos de entrada e interpretar los datos de salida (resultados) de un programa. La facilidad con que se usa el software de acuerdo con los siguientes sub-atributos: Facilidad de comprensión, facilidad de aprendizaje y operatividad.

#### **2.1.9.4. EFICIENCIA**

Es el grado en el que el software emplea de manera óptima los recursos del sistema, como lo indica los siguientes sub-atributos: comportamiento en el tiempo y comportamiento de los recursos.

### **2.1.9.5. FACILIDAD DE MANTENIMIENTO**

Es el esfuerzo necesario para localizar y arreglar un error en un programa, es decir, la facilidad con la que una modificación puede ser realizada. La facilidad con el que se repara el software de acuerdo con los siguientes sub-atributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

### **2.1.9.6. PORTABILIDAD**

Es el esfuerzo necesario para transferir el programa de un entorno hardware/software a otro entorno diferente, es decir, la facilidad con que se lleva de un entorno a otro según los siguientes sub-atributos: adaptabilidad, facilidad de instalarse, cumplimiento y facilidad para reemplazarse.

Los factores de calidad del software ISO 9126 no necesariamente se prestan a la medición directa, sin embargo, proporcionan una base valiosa para las medidas indirectas y una lista de verificación excelente para evaluar la calidad de un sistema. La misma serán ampliadas en el capítulo evaluación del sistema.

### **2.1.10. ESTIMACIÓN DE COSTOS**

Uno de los aspectos antes de importantes para desarrollar proyectos de software es la estimación de costos, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos.

Entre los distintos métodos de estimación de costes de desarrollo de software, el modelo COCOMO (CONstructive COSt MOdel) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

COCOMO define tres modos de desarrollo o tipos de proyectos:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

## 2.2. MARCO REFERENCIAL

En este punto se hace referencia a proyectos desarrollados con anterioridad al presente proyecto, cuyas características sean de interés en cuanto a: metodología de desarrollo, herramientas tecnológicas, y sobre todo similar en el manejo de información del sistema que se desarrolla.

Los documentos organizados por trabajos similares al presente, podemos resumirlos la tabla 2.9. En sus partes más sobresalientes, donde se destaca la metodología, lenguaje de programación entre otros.

Proyecto 1: Sistema de Gestión Académica para el Instituto Normal Superior Simón Bolívar.

Proyecto 2: Sistema de Seguimiento Académico Carrera de Arquitectura Universidad Pública de El Alto.

Proyecto 3: Sistema de administración académica vía Web.

**Tabla 2.9:** Comparación de proyectos similares

ITEM	PROYECTO 1	PROYECTO 2	PROYECTO 3
Metodología	UML	OOSE/Objectory y UML	UWE y UML
Análisis de costos	Cocomo II		
Calidad	ISO 9216 y Olsina	ISO 9126	ISO y Web QEM
Tecnología	Php Mysql Apache	Php Mysql Apache	Php Mysql Apache
Organizado	En 5 capítulos y anexos	En 5 capítulos y anexos	4 capítulos y anexos.

**Fuente:** Elaboración propia

La tabla 2.2 son proyectos relativos en cuanto al manejo de información, es decir son de carácter académicas.

En cuanto a la metodología AUP, se ha consultado la propuesta de Scott W. Ambler como base de desarrollo. Para complementar, se consulto los siguientes proyectos y / tesis de grado.

- ✓ “Asesoramiento Técnico on-line para la adquisición y funcionamiento de un equipo de computación”, del postulante: Boniza Duran, consultado la parte de la metodología que se ha utilizado es UWE y UML, por sus características de complementariedad [DURAN,2007].
- ✓ “Biblioteca digital de los programas de ayuda humanitaria - save the children”, de la postulante: Cerezo Reyna, la parte de nuestro interés es en cuanto a la metodología UWE utilizado, por recomendar que para la parte de diseño de sistemas de información [CEREZO,2008].
- ✓ “Análisis, diseño e implementación de un comparador y sincronizador de bases de datos relacionales de distintos manejadores” del tesista: Calderon Giancarlo, la parte que se consulta es el uso de la metodología AUP por sus características de desarrollo de software iterativo incremental proponiendo 3 iteraciones para la fase construcción superadas por los hitos [CALDERON,2009].

## **2.3. MARCO CONCEPTUAL**

En este punto se destacan los conceptos utilizados con frecuencia en el presente proyecto, es decir parecido al glosario de términos.

### **Institución**

Las instituciones, organizaciones surgen debido a que los individuos tienen objetivos que solo puedan lograrse mediante la actividad organizada. La organización moderna se define como un proceso organizado en que los diferentes individuos interactúan para lograr objetivos comunes e influyentes en la toma de decisiones en la institución u organización.

### **Plan de estudios**

El plan de estudios es la estructura neural del sistema. En base al mismo, se realizan los procesos de gestión académica de la carrera. Asimismo, se reflejan las materias, los planes de estudio, prerequisites y carga horaria.

### **Sistema de información**

Un sistema de información es un conjunto de elementos orientados al tratamiento y administración de datos e información organizados y listos para su posterior uso, generados para cubrir una necesidad.

### **Red**

Conjunto de ordenadores conectados por medio cable, modem u otro medio de comunicación. Su finalidad es posibilitar la comunicación, el intercambio de información y compartición de información entre ellos.

## **Base de datos**

Una base de datos “es una colección de archivos a los que puedan accederse por un conjunto de programas y que contienen todos ellos datos relacionados constituyen una base de datos. Así una base de datos de una universidad, contiene archivos de estudiantes, archivos de nominas, inventarios de equipos, etc.” [JOYALES,2000].

## **Gestor de base de datos**

Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

## **Método**

El método es un proceso disciplinario que permite generar un conjunto de modelos que describen varios aspectos de un sistema de software en desarrollo, utilizando alguna notación bien definida [RUP,2010].

## **Metodología**

Colección de métodos aplicados a lo largo del ciclo de vida del desarrollo de software y unidos por una aproximación general o filosófica.

## **Modelo**

Un modelo es la representación abstracta de la realidad.

## **La web**

Sitio Web (Web-Site) es una colección de archivos creados para internet sobre un mismo tema.

## **Pagina web**

Es una fuente de información adaptada para la Word Wide Web(WWW) y accesible mediante un navegador de internet.

## **World Wide Web**

En informática, Word Wide Web o red global mundial es un sistema de documentos de hipertextos y/o hipermedias enlazadas y accesibles a través de internet.

### **2.4. MARCO LEGAL**

Actualmente Bolivia está atravesando una etapa de cambios, y en consecuencia la educación también se ve afectada por las mismas. Los habitantes de la ciudad de El Alto han tomado medidas de hecho para recuperar medidas los recursos naturales, así como lograr la formación académica y científica de su juventud. En ese contexto, la UPEA fue creada mediante la ley N° 2115 de fecha 5 de septiembre del 2000 como Universidad Publica, sin respetarse las disposiciones de la Constitución Política del Estado en sus artículos 185, 186, 187 y 189.

Sin embargo se consolida como Universidad Autónoma mediante Ley N° 2556 de fecha de 12 de noviembre de 2003 conforme mandato constitucional de la república, como efecto después de la movilización de la ciudad de El Alto [ESTATUTO,2009].

#### **2.4.1. EVALUACIÓN**

En esta parte destacaremos aspectos inherentes derechos y obligaciones de los estudiantes, así como normar las modalidades de admisión, permanencia y régimen académico y graduación.

La parte que nos interesa es la parte académica: Evaluación, ordinarias y extraordinarias (Art. 36). Ordinarias, pruebas parciales y finales. Si la suma de notas este entre 51-100 puntos entonces el estudiante aprueba la materia. Extraordinarias Segunda instancia, prueba pospuesta con 3 días de anticipación. Las pruebas de segunda instancia tendrán una nota de 51 en caso de aprobación.

## **2.4.2. ADMISIÓN ESTUDIANTIL**

### **Admisión especial**

Dispensa del cumplimiento de la prueba de suficiencia académica y el curso pre-universitario a postulantes profesionales con título en provisión nacional. Por otra parte, admite a estudiantes destacados en competencias académicas reconocidas por la UPEA y beneficiarios de convenios interinstitucionales.

### **Excelencia académica.**

Admite a estudiantes a bachilleres sobresalientes de escuelas fiscales, que hayan obtenido las tres mejores calificaciones de cuarto de secundaria. La Unidad Educativa certifica en su acta de calificación que son los tres mejores estudiantes del mencionado grado.

### **Examen de dispensación**

Admite a postulantes bachilleres que logren superar la prueba de conocimiento y un diagnóstico psicotécnico para seguir estudios superiores sin el prerrequisito del curso preuniversitario. Dicha modalidad de aprobación requiere un mínimo de puntaje 51 puntos sobre 100 puntos.

### **Preuniversitario**

El objetivo del curso preuniversitario tiene como objetivo fortalecer las capacidades cognoscitivas y desarrollar las aptitudes de los estudiantes y otorgarles instrumentos básicos para realizar estudios superiores. La nota mínima de aprobación será de 51 puntos acorde al reglamento del curso-preuniversitario.

## **Carrera paralela**

Es la inscripción simultánea de un estudiante en dos carreras dentro de la UPEA.

## **Traspaso de universidad**

Es el procedimiento académico-administrativo por el cual el estudiante universitario se moviliza voluntariamente de una universidad a otra en la misma carrera o afín.

## **Cambio de carrera**

Es el procedimiento académico-administrativo por el cual un estudiante universitario cambia de su carrera origen a otra.

### **2.4.3. MODALIDAD EXAMEN DE CLASIFICACIÓN**

Esta modalidad es habilitada para estudiantes con conocimientos previos sobre los idiomas de: Aymara, Quechua, Inglés u otros de la Carrera de Lingüística e Idiomas de la Universidad Pública de El Alto (UPEA). La misma es convocada por el Honorable Concejo de Carrera (HCC) a inicios de cada semestre [PLAN,2010], acorde a las siguientes especificaciones:

- ✓ Podrán optar esta modalidad estudiantes nuevos de la carrera y/o que demuestren no ser antiguos en la carrera de lingüística e idiomas, es decir no haber cursado la materia o congelado el semestre.
- ✓ El examen será llevado a cabo por el tribunal designado por el Honorable Consejo de Carrera.
- ✓ El examen consta de una prueba escrita sobre 40 puntos y prueba oral sobre 60 puntos.
- ✓ En cuanto a la nota obtenida, si está entre 51-75 puntos entonces clasifica al nivel II de dicha materia. Por otro lado si la nota esta entre 76-100 puntos, podrá clasificar al nivel III de dicha materia, la nota obtenida se repite para niveles I y II.

## 2.4.5. MODALIDADES DE GRADUACIÓN

Un estudiante que haya culminado con su plan de estudios y optado por las siguientes modalidades de graduación:

- ✓ Tesis de grado
- ✓ Proyecto de grado
- ✓ Examen de grado
- ✓ Internado rotatorio
- ✓ Trabajo dirigido
- ✓ Por excelencia

Una vez concluido con alguna modalidad de graduación, dicho estudiante pasa a ser estudiante no regular de la carrera.

## 2.4.6. CONVALIDACIÓN, HOMOLOGACIÓN Y COMPENSACIÓN

Son instrumentos técnicos para facilitar los cambios en los planes de estudio anteriores a los actuales, traspaso y cambios de carrera para una mejor movilidad estudiantil.

### CONVALIDACIÓN

La convalidación de estudios de un estudiante universitario, es el proceso mediante el cual y en virtud al derecho que le asiste se le permite convalidar los programas analíticos y reconocer la calificación de certificado de notas establecidos en el Historial Académico oficial de la Carrera de Lingüística e Idiomas[PLAN,2010]. Si el nombre de la asignatura, sigla y código no son iguales en una determinada asignatura, pero el contenido mínimo es igual o mayor al 60% del nuevo plan de estudios, la asignatura será reconocida como convalidada.

## **HOMOLOGACIÓN**

Si la asignatura tiene contenidos mínimos y analíticos, equivalentes a un 80 % o que sean iguales, será homologado en el Plan de Estudios de la Carrera de Lingüística e Idiomas.

## **COMPENSACIÓN**

Se aplica esta figura cuando los contenidos mínimos de las asignaturas en el Plan de Estudios de la Carrera de origen no están contempladas en el Plan de Estudios de la Carrera de Lingüística e Idiomas, pero son afines y contienen parte de la asignatura considerada, en este caso será reconocida por compensación (dos asignaturas pueden ser compensadas por una sola asignatura del Plan de Estudios de la Carrera de Lingüística e Idiomas).

### **2.5. MARCO TECNOLÓGICO**

Para el desarrollo de páginas web se tienen una variedad de lenguajes de programación como: Php, Asp, Java, NET, etc. Y la mayoría de ellas ofrecen ventajas a media que va evolucionando, incorporando nuevos componentes acorde a los requerimientos empresariales. Entre estos aspectos cabe mencionar servicios de seguridad, transacciones, soporte para servidores, Frameworks de accesos a soporte como los WebServices, Xml entre otros.

#### **2.5.1 PLATAFORMA**

Se utiliza la plataforma Windows de Microsoft instalada en la institución, en la cual la aplicación se ejecuta sin ningún problema, debido a que los usuarios están familiarizados con este entorno, que además es ampliamente difundido en los hogares y empresas.

## 2.5.2. SERVIDOR APACHE

El servidor HTTP Apache es un software libre de código abierto que se lo utiliza en plataformas Linux, Windows, Macintosh y otras. Hoy por hoy, es el servidor ampliamente utilizado en la red.

## 2.5.3. PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. Para la programación del sistema se utiliza Php5, se eligió este lenguaje de programación porque es una herramienta de desarrollo que permite crear aplicaciones web dinámicas de carácter gratuito.

## 2.5.4. MYSQL

Mysql es un gestor de base de datos que tiene las siguientes ventajas:

- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros
- MySql está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux, usando MySql, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- Permite manejar multitud de tipos para columnas.

- Permite manejar registros de longitud fija o variable.

### **2.5.5. AJAX**

Acrónimo de Asynchronous Javascript And Xml (JavaScript asíncrono y Xml) es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

El 18 de Febrero de 2005, Jesse James Garrett, se refería a AJAX como: "...una nueva forma de ver las aplicaciones web, ya no como sitios dinámicos, si no además como sitios interactivos, como la posibilidad de cargar paginas dinámicamente, sin tener que recargar toda la pagina, que es uno de los logros fundamentales de ajax" [FUENTES,2009].

Algo que debe quedar claro es que ajax no es una tecnología, si no es más bien una mezcla de tecnologías, que permiten crear aplicaciones web, increíbles y dinámicas.

### **2.5.6. JQUERY**

Para simplificar, podríamos decir que jQuery es un framework Javascript, conformada por librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Por otro lado es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General, permitiendo su uso en proyectos libres y privativos.

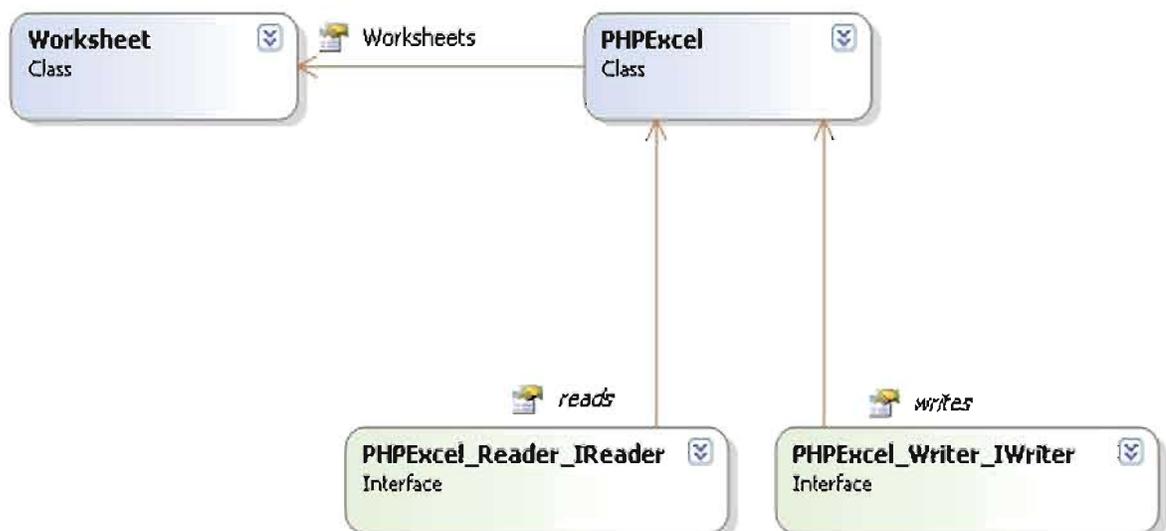
## 2.5.7. CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

## 2.5.7. GENERACIÓN DE PLANILLAS

La generación de planillas Excel, nos permitirá una mejor administración de listas y notas en forma más sencilla y conocida para los usuarios. Una de las librerías de carácter gratuito al cual podemos acceder es *PHPExcel*, la misma nos permite generar *archivos Excel con PHP*, a través de la inclusión de las clases *PHPExcel* podemos leer y manipular el contenido de una hoja de cálculo hecha con Microsoft Excel. [PHPEXCEL,2011]. Véase figura 2.21, ilustra la forma de interacción con los archivos.

Figura 2.21: Arquitectura de PhpExcel



Fuente: [PHPEXCEL,2011]

## 2.5.8. GENERACIÓN DE REPORTES CON PDF

*PHP* entre sus características primarias tiene la capacidad de generar archivos PDF, tiene funciones específicas para personalizar. Una de las clases que podemos utilizarlo en forma gratuita es: *class.ezpdf.php* y *class.pdf.php* los cuales contienen todas las propiedades y métodos necesarios para crear documentos PDF utilizando PHP.

## 2.5.9. MAGICDRAW

MagicDraw provee soporte completo para meta-modelos UML 2.0, incluyendo diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad, implantación, paquetes, componentes, estructuras compuestas y de distribución. Asimismo es posible incluir su herramienta complementaria para UWE.

MagicUWE es una herramienta CASE para el Modelado de Aplicaciones Web. La misma ha sido desarrollada para el diseño asistido por ordenador de aplicaciones Web usando UML, basada en la metodología de Ingeniería Web (UWE).

## 2.10. DISEÑO DE INTERFAZ DE USUARIO

El software presenta una interfaz amigable, que permite interactuar con los usuarios de manera fácil, donde podrá realizar operaciones sencillas. Sus características influyen en el aprendizaje del usuario, al igual que en la frecuencia de errores, cuando se introducen datos o instrucciones. Una de las formas de que un usuario se comunica con el sistema es a través de formularios "Las interfaces de formulario consisten de formularios en pantalla o formularios que se basan en la Web que despliegan campos que contienen datos o parámetros que necesitan ser comunicados al usuario"[KENDALL,2005].

## Características de una interfaz:

Las características de interfaz incluyen introducir y recibir datos, el diálogo que incita y guía a los usuarios mostrando la información.

Reglas que ayudan a desarrollar una interfaz amigable con el usuario:

1. El programa debe pedir entradas y producir salidas en forma consistente
2. Pida información con una secuencia lógica.
3. Haga obvio al usuario el tipo de error que ha cometido, y donde.
4. Distinga entre edición de campos y edición de pantalla, Algunas acciones de procesamiento:
  - ✓ Captura de datos (Descripción de cada campo y su posición)
  - ✓ Edición de datos (que datos cambiar en pantalla)
  - ✓ Recuperación de datos (Para mostrarlos)
  - ✓ Recepción de mensajes (Cuando se inicia un operación y finaliza)
  - ✓ Manejo de Pantallas (debe seguir un diseño estándar)

Uso de ventanas

- ✓ Ventana de titulo. Titulo de la pantalla.
  - ✓ Ventana de instrucciones.- como introducir datos.
  - ✓ Ventana principal de texto.- como captura de datos.
  - ✓ Área de navegación y menú.-Instruye al usuario como moverse.
  - ✓ Ventana de mensajes.-contiene mensajes de información y control.
5. Haga la edición y la revisión de errores dependientes del usuario (No es lo mismo un para un experto que para un novato).
    - ✓ Mensajes y comentarios comunicación con los usuarios.
    - ✓ Mensaje de estado la pantalla nunca debe estar en blanco.

- ✓ Mensaje de error reportan equivocaciones o eventos inesperados.
  - ✓ Mensaje de solicitud de acciones.
  - ✓ Mensaje de verificación de acciones.
6. Permita que el usuario pueda: (a) cancelar parte de la transacción (b) cancelar toda la operación. Dialogo Pregunta / Respuesta (Si / No) Confirmación o cancelación de operaciones, de eliminación, en la grabación de registros.
  7. Proporcione un mecanismo de ayuda. Pretenden auxiliar al usuario.
  8. Distinga entre aplicaciones guiados por menús y dirigidos por órdenes que facilite la navegación.
  9. Si la aplicación esta realizando un proceso largo despliegue un mensaje que se esta procesando para que no crea que se detuvo.
  10. Proporcione alternativas por omisión para las entradas estándar.
  11. Aproveche el color y el sonido. El color añade una nueva dimensión a la facilidad de uso de la pantalla, ya que atrae la atención del usuario “el color es una forma atractiva y consolidada para facilitar la entrada de datos a la computadora”[KENDALL,2005].
    - ✓ Algunas combinaciones de letreros en pantallas interactivas
    - ✓ Negro sobre amarillo
    - ✓ Verde sobre blanco
    - ✓ Azul sobre blanco
    - ✓ Blanco sobre azul
    - ✓ Amarillo sobre negro

**12.** Uso de iconos en el diseño de pantallas, simbolizan las acciones de la computadora que los usuarios puedan seleccionar.

## **CAPÍTULO III**

### **MARCO APLICATIVO**

En este capítulo se analiza en una primera instancia, la situación actual del funcionamiento del sistema en la carrera de Lingüística e idiomas con el fin de conocer los procesos académicos y sus involucrados. Las mismas son desarrolladas acorde a la metodología AUP en sus fases de: iniciación, elaboración, construcción y transición.

#### **3.1. FASE DE INICIACIÓN**

Primeramente debemos indicar la organización de la Carrera de Lingüística e Idiomas en la misma podemos ubicar la dirección, unidades, secretaria y otras. Es decir la contextualización del sistema.

##### **3.1.1. SITUACIÓN ACTUAL**

###### **3.1.1.1. ORGANIGRAMA DE LA INSTITUCIÓN**

La carrera de lingüística se encuentra organizada como en la figura 3.1. Luego describimos las funciones de los involucrados de la institución.

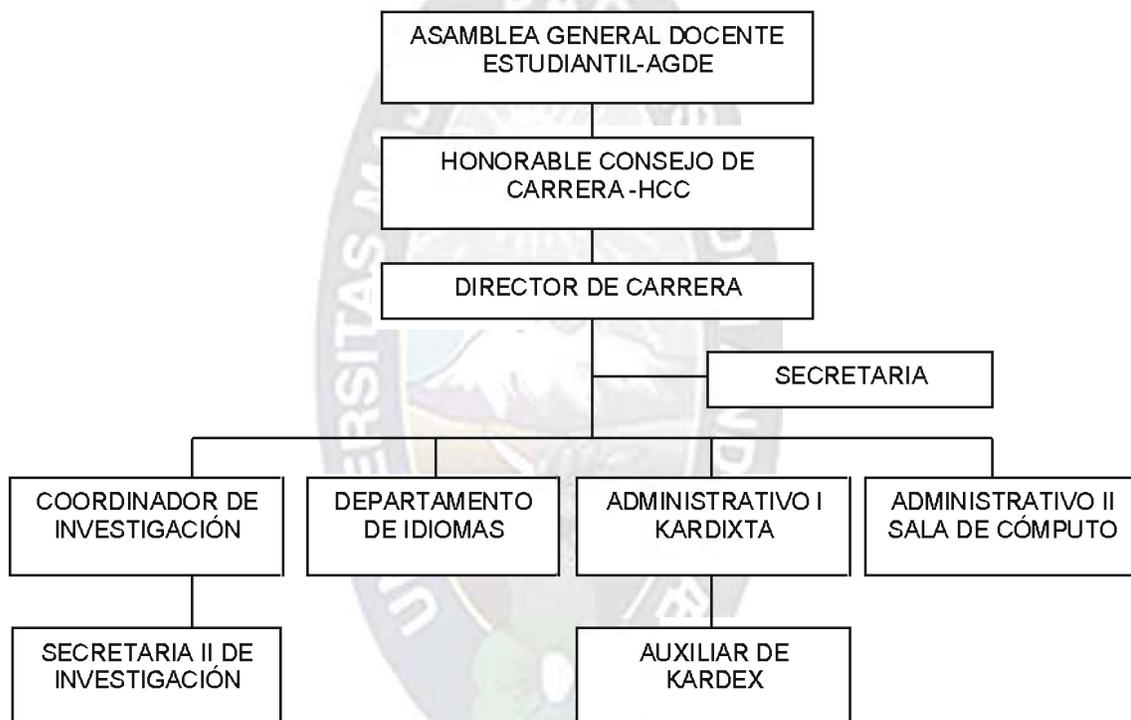
#### **ASAMBLEA DOCENTE ESTUDIANTIL**

La asamblea docente estudiantil es la instancia donde se debaten y deciden congresos de carrera, la misma es convocada por el consejo de carrera. Acorde a reglamento interno.

## HONORABLE CONSEJO DE CARRERA

El honorable consejo de carrera (HCC) es la instancia resolutoria de la carrera, se reúne a la semana una vez acorde a reglamento interno, se resuelven casos que involucren la aprobación del consejo.

**Figura 3.1:** Organigrama de la institución



Fuente: [MOF,2011]

## DIRECTOR (A) DE CARRERA

### Objetivo general

Administrar y desarrollar la carrera, coordinando y articulando la actividad académica y administrativa, relacionando con la comunidad universitaria, carreras de la UPEA, con las instituciones públicas y privadas y la sociedad en su conjunto de acuerdo a la visión y misión de la carrera.

## Funciones específicas

- ✓ Ejercer la representación oficial de la carrera ante instancias superiores de la UPEA, instituciones de educación e instituciones públicas y privadas.
- ✓ Ejercer la función de órgano ejecutivo de las decisiones tomadas en las instancias de decisión y Gobierno de carrera.
- ✓ Convocar, dirigir y presidir sesiones del Consejo de Carrera y representarla ante el Honorable Consejo Universitario y Asamblea General Docente Estudiantil.
- ✓ Cumplir y hacer cumplir el Estatuto Orgánico, los Reglamentos, manual de organización, funciones y normas de la Universidad.
- ✓ Dirigir, cumplir con las actividades académicas y administrativas de la carrera contenidas en el POA y amparadas en el presupuesto vigente de la carrera.
- ✓ Proponer políticas de investigación científica, políticas académicas y de interacción social y cultural.
- ✓ Preparar la agenda de sesiones del consejo de carrera con la documentación sustentatoria para su aprobación.
- ✓ Formular el POA consolidado en la carrera y coordinar su ejecución con las responsables de las instancias respectivas.
- ✓ Autorizar gastos y controlar la ejecución del presupuesto de la carrera.
- ✓ Coordinar las actividades con los directores de las otras carreras y demás órganos de la Universidad.
- ✓ Firmar las actas de notas, certificados y otros documentos.
- ✓ Proponer al consejo de carrera el nombramiento de las comisiones permanentes y otras que determinen en la normatividad interna.
- ✓ Aprobar las planillas de asistencia de docentes, auxiliares de docencia que ejercen una función.
- ✓ Realizar las demás funciones, afines al cargo que señala el estatuto, reglamentos y el manual de la organización y funciones de la Universidad Pública de El Alto.

## COORDINADOR DE INVESTIGACIÓN

### Objetivo general

Proponer, organizar y dirigir programas de investigación dentro de los ámbitos de competencia académica de la carrera.

### Funciones específicas

- ✓ Formular políticas de investigación para la carrera.
- ✓ Apoyar las actividades de investigación referidas a la carrera.
- ✓ Organizar actividades de investigación de acuerdo con las políticas, planes y reglamentos de la carrera.
- ✓ Fomentar y dirigir la investigación y desarrollo de investigación tecnológica en proyectos de interés regional.
- ✓ Solicitar los recursos necesarios para los proyectos de investigación de la carrera.
- ✓ Elaborar y presentar el POA de investigación.
- ✓ Establecer convenios y mantener correspondencia con instituciones, universidades y profesionales nacionales e internacionales que realicen actividades de investigación que permita la presencia de investigadores visitantes de reconocido prestigio.
- ✓ Organizar centros y grupos de investigación especializados conformados por docentes, estudiantes y egresados más competentes de la carrera y por docentes investigadores.

## ENCARGADO DEL DEPARTAMENTO DE IDIOMAS

### Objetivo general

Administrar y desarrollar el departamento de idiomas de la UPEA coordinando y articulando la actividad académica y administrativa relacionando con la comunidad universitaria y con las instituciones públicas, privadas y la sociedad en conjunto.

### Funciones específicas

- ✓ Formular políticas de enseñanza-aprendizaje
- ✓ Organizar cursos de idiomas para el público externo e interno de la universidad.
- ✓ Organizar actividades extra-académicas (seminarios, talleres, congresos y otros).
- ✓ Hacer convenios con otras instituciones públicas o privadas sobre el desarrollo de idiomas.
- ✓ Elaborar proyectos académicos para el desarrollo del departamento de idiomas.

## KARDIXTA

### Objetivo general

Ejecutar las actividades técnicas en registros académicos relacionados con el sistema de seguimiento académico.

### Funciones específicas

- ✓ Recepcionar, clasificar, registrar y archivar los documentos académicos de los estudiantes.
- ✓ Verificar y realizar la actualización de registros, fichas, acta de notas de estudiantes.

- ✓ Emitir informes preliminares a requerimientos del jefe inmediato.
- ✓ Participar en la programación de actividades académico administrativo de la carrera.
- ✓ Apoyar las acciones de información y relaciones públicas.
- ✓ Realizar las funciones afines al cargo se le asigne.
- ✓ Participación en la inscripción de estudiantes.
- ✓ Regularización de las actas de notas.
- ✓ Control del file estudiantil.
- ✓ Elaboración y verificación de record académicos.
- ✓ Elaboración de certificado de notas originales y conclusión de estudios.
- ✓ Emisión de certificados provisionales.
- ✓ Elaborar reportes académicos acorde a requerimiento de autoridades superiores.
- ✓ Elaborar estadísticas académicas.

## **SECRETARIA**

### **Objetivo general**

Llevar los archivos sistemáticos y diarios de toda la documentación académica, redacción y transcripción, recepción y despacho de documentos de la carrera. Asimismo, coadyuvar a la Dirección de Carrera en el desarrollo de actividades académicas y administrativas.

## **AUXILIAR DE KARDEX**

### **Objetivo general**

Apoyar en las actividades de mantenimiento y actualización de kardex individual del personal docente y administrativo de la universidad.

## **Funciones específicas**

- ✓ Apoyar al encargado de kardex en la elaboración de reportes y otros a requerimientos.
- ✓ Apoyar en el mantenimiento y actualización de kardex individual.
- ✓ Complementar formularios de trámites y certificaciones previa verificación de documentación.
- ✓ Ordenar sistemáticamente la información de kardex individuales.
- ✓ Realizar otros trabajos instruidos por su inmediato superior.

## **ADMINISTRATIVO BIBLIOTÉCA SALA DE CÓMPUTO**

### **Objetivo general**

Controlar y ejecutar la prestación del servicio de la Biblioteca relacionando con el análisis, evaluación de información general y material de estudio específico del control en el servicio a los usuarios.

## **SECRETARIO DE INVESTIGACIÓN**

Coordinación, realización y supervisión de actividades variadas de apoyo administrativo de investigación en redacción, transcripción, recepción y despachos de documentos de carrera.

### **3.1.1.2. SISTEMA LÓGICO**

El sistema académico que se maneja actualmente es manual, desde el registro del nuevo estudiante, inscripción, retiro / adición, seguimiento académico, emisión de documentos y otros aspectos inherentes al manejo de kardex. Para las mismas se emplean 8 planes de estudio, los cuales figuran en la tabla 3.1, en caso de estudiantes anteriores al plan 8, activan los procesos de homologación, convalidación y compensación.

**Tabla 3.1:** Planes de estudio de la carrera

Nº	PLAN	OBSERVACIÓN	NºMATERIAS
1	Plan 1(I-2001,II-2001)	Villa Esperanza	14
2	Plan 2(I-2002-A )	Villa Esperanza	12
3	Plan 3(II-2002-B)	Villa Esperanza	11
4	Plan 4(I-2003,II-2003)	Villa Esperanza	16
5	Plan 5(II-2002)	Villa Dolores	17
6	Plan 6(I-2003)	Villa Dolores	19
7	Plan 7(I-2004,I-2007)	Villa Esperanza	58
8	Plan 8(II-2007,hoy)	Villa Esperanza	62

**Fuente:** [Plan,2010]

Por otro lado también mencionamos que se maneja el plan de su unidad descentralizada Cruz Loma Coroico.

### 3.1.1.3. PROCESOS ACADÉMICOS

#### REGISTRO DE ESTUDIANTE NUEVO

Se registra un nuevo estudiante a la carrera de lingüística si cumple con una de las siguientes modalidades de ingreso que figuran en la tabla 3.2.

**Tabla 3.2:** Formas de ingreso

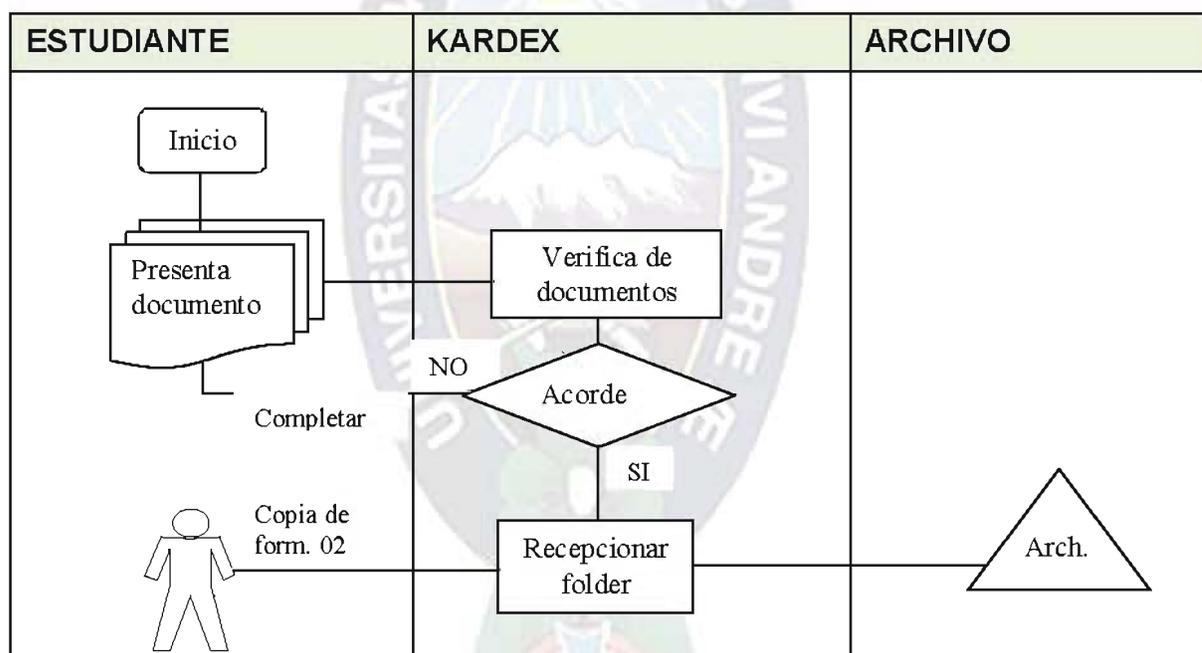
Nº	FORMAS DE INGRESO	OBSERVACIÓN
1	Admisión especial	Profesionales
2	Excelencia académica.	Uno de los tres estudiantes de cuarto de secundaria con mejor nota.
3	Examen de dispensación	Prueba aplicada a postulantes
4	Pre-Universitario	Nivela conocimientos básicos del nivel secundario.
5	Carrera paralela	Estudiantes de otras carreras de la misma universidad.
6	Traspaso de universidad	Estudiante de otra universidad estatal.
7	Cambio de carrera	Estudiantes regular de otras carreras que hayan aprobado el primer año.

**Fuente:** [PLAN,2010]

Además los estudiantes deberán adjuntar en un folder, los demás requisitos en fotocopias como son: historial académico, formulario de inscripción 02, matricula vigente, cedula de identidad, certificado de habilitación (original y fotocopia), titulo de bachiller, certificado de nacimiento y dos fotografías.

La figura 3.2 muestra el flujo de registro de nuevo estudiante. Para mayor detalle. Para el caso de pre-universitario se emite una lista de aprobados de dicho curso. Por otro lado, los estudiantes con excelencia académica son enviados con el visto bueno desde rectorado.

Figura 3.2: Registrar nuevo estudiante



Fuente: Elaboración propia

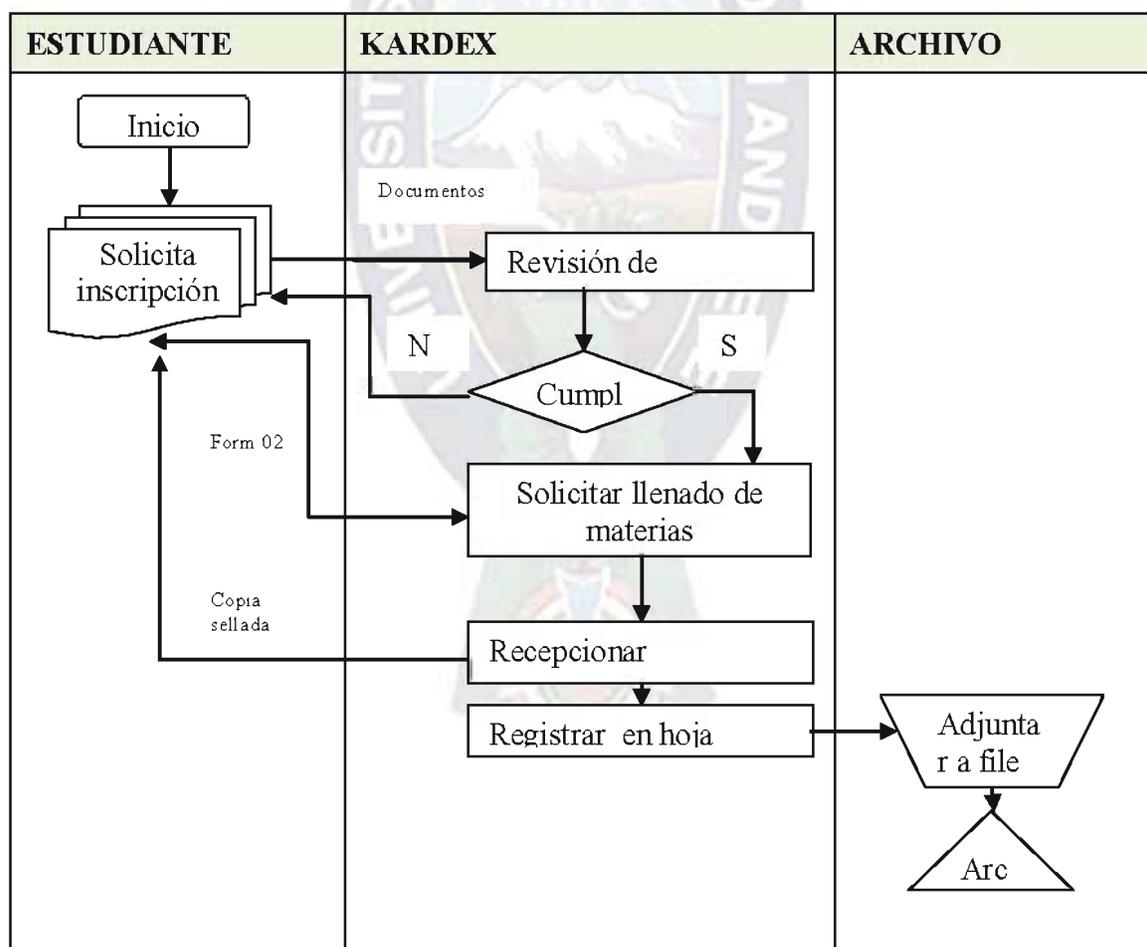
## INSCRIPCIÓN DE ESTUDIANTES

La inscripción se realiza previa presentación de fotocopia de cedula de identidad, fotocopia de matricula y llenado de formulario 02. En el formulario de inscripción se registran las materias a cursar (manualmente) por parte de los estudiantes, el cual es recepcionado por el encargado de kardex o el auxiliar de

kardex, el cual verifica que las materias y siglas sean las correctas además preguntando al estudiante que si habían aprobado el prerrequisito de la materia, sellando una copia del formulario 02 para el estudiante. La siguiente etapa es la transcripción en hojas de calculo Excel la lista de los inscritos en cada materia y gestión, que son organizadas en carpetas. Una ilustración grafica esta en figura 3.3.

Los estudiantes que no puedan cursar materias durante la gestión vigente, tienen la opción de presentar una nota que mencione su congelamiento de inscripción.

**Figura 3.3:** Proceso de inscripción



Fuente: Elaboración propia

## **INSCRIPCIÓN A EXAMEN DE CLASIFICACIÓN**

Esta modalidad es optada por los estudiantes nuevos o que hayan congelado materias de primer semestre, pueden inscribirse en la modalidad de examen de clasificación. El número de materias al cual pueden inscribirse son: una, dos y tres materias que estén en esta modalidad.

Algunos situaciones que se detectaron: que un estudiante podía inscribirse a esta modalidad ya que no se verifica que el estudiante interesado este cursando en primer semestre, es decir, un estudiante que ya haya cursado algunas materias de primer semestre pueda optar por esta modalidad, según reglamenta esta figura no debería proseguir.

## **EMISIÓN DE LISTAS OFICIALES**

Las listas oficiales son emitidas una vez concluida el proceso de inscripción, y de la recepción de listas de estudiantes que optaron por el examen de clasificación retirando las materias que ya aprobaron. El auxiliar de Kardex transcribe en hojas de calculo Excel todas los formularios de inscripción llenadas por los estudiantes, dichos formularios son apilados para el proceso de retiro y / adición de materias.

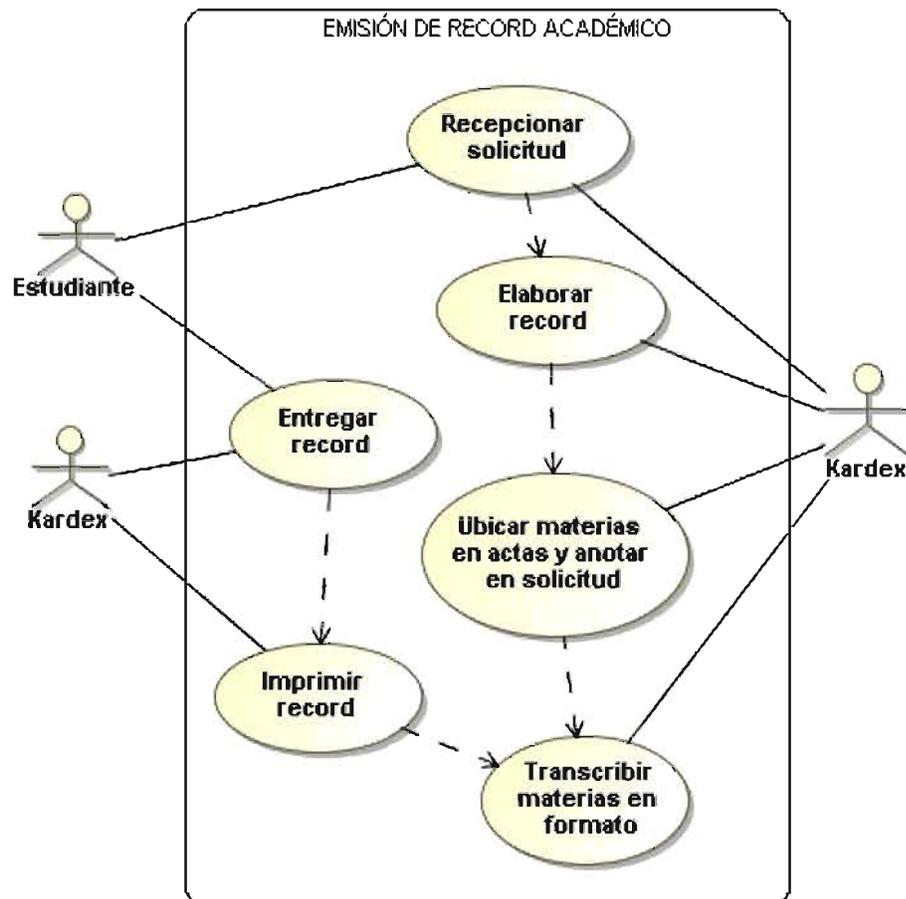
## **REGISTRO DE NOTAS**

El registro de notas se realiza al finalizar cada gestión, los docentes transcriben sus notas en hojas de calculo Excel denominados actas de calificaciones, las mismas son recepcionadas e impresas por kardex. Una vez que se haya completado con las notas de todos los docentes y materias se procede al empastado de actas. Para el caso de examen de clasificación los jurados entregan las notas de los estudiantes, para la posterior transcripción en formato de hojas Excel por parte de kardex. En el que figuran los estudiantes y sus notas obtenidas en forma numeral y literal.

## EMISIÓN DE RECORD ACADÉMICO

La emisión de record académico se la realiza después que el estudiante haya presentado una solicitud a kardex adjuntando un listado de las materias que el mencionado hubiera aprobado, en el cual figuran la materia y la gestión de aprobación. Si la solicitud no presenta observación alguna, se indica al interesado que retorne dentro de 48 horas. El auxiliar de kardex busca en acta de notas las materias que solicita el estudiante, una vez que se encuentra la nota, esta es anotada en la hoja de solicitud presentada, se repite la acción hasta completar todas las materias solicitadas. La misma podemos ilustrarlo en el caso de uso de la figura 3.4 emisión de record.

Figura 3.4: Caso de uso actual - Record académico



Fuente: Elaboración propia

Por lo expuesto, se idéntico el dominio del problema que se abordan en el proyecto, y determinar una solución viable para resolver el problema.

### 3.1.1.4. IDENTIFICACIÓN DE RIESGOS

De acuerdo a la naturaleza del proyecto se ha podido establecer los riesgos que pueden obstaculizar el desarrollo a lo largo de las distintas fases del proyecto. Las mismas son como en la tabla 3.3, donde se muestra el nombre del riesgo, la probabilidad que suceda es baja, media o alta.

**Tabla 3.3:** Riesgos del sistema

RIESGO	PROBABILIDAD DE QUE SUCEDA	IMPACTO
Las actividades definidas en el cronograma de anexos no hayan sido distribuidas de manera equitativa.	Media	Alto
Descuidar el seguimiento de las tareas y actividades definidas.	Baja	Alto
No controlar ni hacer seguimiento a los riesgos.	Media	Alto
Realizar más esfuerzo de lo estimado debido a la falta de conocimiento de algunas herramientas o tecnologías.	Alta	Medio
Los requerimientos son demasiados ambiciosos para el tiempo disponible.	Baja	Alto
Los requerimientos cambian demasiado a lo largo del proceso de desarrollo.	Medio	Alto
No poder terminar la solución acorde a cronograma o tener retrasos en las liberaciones de la aplicación.	Medio	Alto
Haber dejado de lado artefactos de software necesarios para un mejor diseño de la solución.	Baja	Bajo

**Fuente:** Elaboración propia

### Plan de riesgos

Una vez Identificado los riesgos y su impacto sobre el desarrollo del producto, se ha elaborado el siguiente plan de respuesta para mitigar, evitar o en el mejor de los casos evitarlos.

- ✓ En riegos relacionados a la planificación, es conveniente dedicar más tiempo en definir los procesos necesarios para el desarrollo del proyecto.
- ✓ En cuanto a cambio en los requerimientos, se procederá a analizarlos de acuerdo al impacto del requerimiento, se continuará con el desarrollo cambiando de manera ágil.
- ✓ Se invertirá tiempo en la investigación de las librerías necesarias que apoyen el desarrollo de la solución. Las mismas deben tener características de: facilidad de uso, documentación suficiente y que tengan las funcionalidades solicitadas. Entre las librerías más importantes están las encargadas de realizar una conexión a base de datos, Interfaz amigable, generar reportes y ejecutar sentencias SQL.
- ✓ La implementación de las pruebas se harán en paralelo con la codificación, para probar de manera rápida y ágil las funcionalidades implementadas previas a las liberaciones por cada iteración.

## **3.2. FASE DE ELABORACIÓN**

En esta fase la arquitectura debe estar desarrollada y probada, la mejor manera de hacerlo es la construcción de extremo a extremo del esqueleto de del sistema conocido.

### **3.2.1. MODELADO DEL NEGOCIO**

#### **3.2.1.1. REQUERIMIENTOS**

Aquí se detallan los requerimientos funcionales y no funcionales que se identifican durante el desarrollo del sistema.

## REQUISITOS FUNCIONALES

La tabla 3.3: Muestra los requisitos mínimos que el nuevo sistema debe satisfacer.

**Tabla 3.4:** Requisitos básicos del sistema

Ref.#	Función	Categoría
R1	El encargado de kardex debe introducir una identificación y contraseña para utilizar el sistema.	Evidente
R2	Registro de datos personales y académicos.	Evidente
R3	Inscripción de estudiantes.	Evidente
R4	Emisión de boletas de inscripción.	Evidente
R5	Retiro y / adición de materias.	Evidente
R6	Inscripción de estudiantes para examen de clasificación.	Evidente
R7	Generación de planilla de estudiantes inscritos por paralelo.	Evidente
R8	Generación de listas para examen de clasificación.	Evidente
R9	Generación de planilla de notas.	Evidente
R10	Cargado de planilla de notas.	Evidente
R11	Control interno notas.	Oculto
R12	Verificación interna de prerrequisitos.	Oculto
R13	Administración de usuarios y niveles de acceso.	Evidente
R14	Administración de pensum (materias, planes).	Evidente
R15	Emisión de record académico.	Evidente
R16	Registro de docentes.	Evidente
R17	Registro de personal administrativo.	Evidente
R18	Consultas mas frecuentes.	Evidente

Fuente: Elaboración propia

## REQUISITOS NO FUNCIONALES

Para el funcionamiento del sistema los siguientes requerimientos no funcionales se requieren:

- ✓ Microsoft Windows por parte del usuario.
- ✓ MySQL como gestor de base de datos.
- ✓ Apache como servidor.

- ✓ Php para la programación del sistema.

### 3.2.1.2. IDENTIFICACIÓN DE ACTORES

La identificación de actores nos permite conocer las personas involucradas en los distintos procesos que la carrera maneja a objeto de formar los casos de uso. Veamos las mismas en la tabla 3.4, por otro lado también se contratan personas eventuales que realizan algunas tareas delegadas por kardex.

**Tabla 3.5:** Descripción general de actores

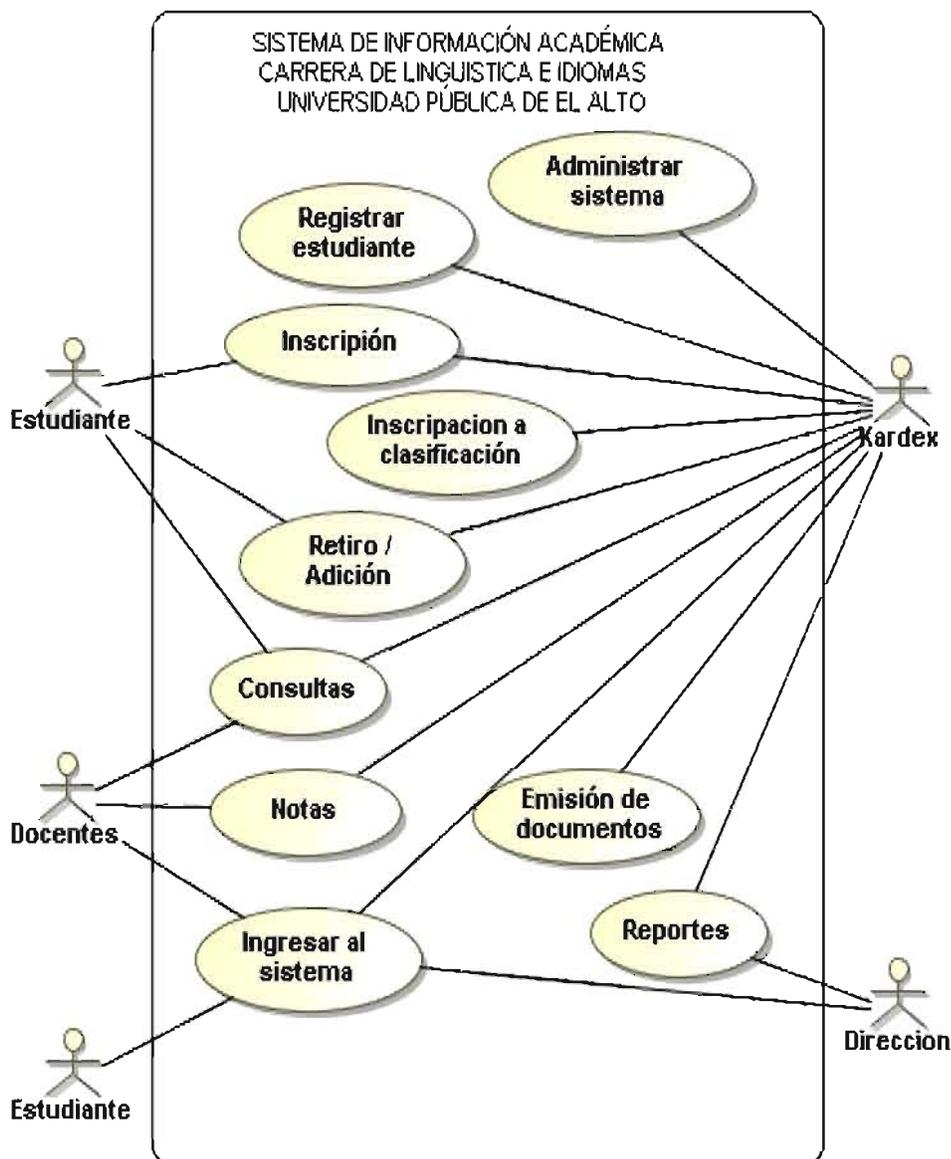
ACTOR	DESCRIPCIÓN
 Kardex	Tiene la función de registrar: estudiantes, inscripción, retiro y/o adición, emitir record académico, consultas y reportes.
 Docente	Encargado de asignar nota a estudiantes de la materia que cursa. Registra notas y consulta de las mismas.
 Estudiante	Persona principal al cual se hace seguimiento de sus datos de carácter académico.
 Director	Persona física encargada de resolver casos que involucren procesos de carácter académico.
 Auxiliar de Kardex	Persona de apoyo en actividades de académicas en Kardex.

Fuente: Elaboración propia

### 3.2.1.3. CASOS DE USO

Los casos de uso nos muestra la interacción de los actores con los procesos del sistema. El caso de uso general del sistema propuesto se ilustra en la figura 3.5, los procesos son diferentes para cada actor, pero si es común en algunos casos por ejemplo la autenticación de usuarios.

Figura 3.5: Diagrama de casos de uso del sistema



Fuente: Elaboración propia

#### 3.2.1.4. DESCRIPCIÓN DE CASOS DE USO

##### Caso de uso registrar estudiante

Las siguientes tablas muestran los casos de uso de los procesos del sistema y la interacción con los actores.

**Tabla 3.6:** Descripción caso de uso - Registrar estudiante

Caso de uso	Registrar estudiante	
Actores	Encargado de registro (personal de kardex), estudiante	
Propósito	Registrar los datos personales del estudiante.	
Referencia cruzada	R2	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex con rol de registro, carga los datos de estudiante.	2. El sistema verifica completitud de datos.
		3. El sistema almacena los datos introducidos, generando un código.
	4. Se escribe código en file de estudiante.	
Flujo alternativo	En 1. El funcionario puede buscar si el estudiante ya esta registrado en sistema. Si se quiere actualizar algún documento rezagado, se sugiere utilizar esta opción.	

**Fuente:** elaboración propia

**Tabla 3.7:** Descripción caso de uso - Inscripción

Caso de uso	Inscripción	
Actores	Estudiante, personal kardex	
Propósito	Inscribir al estudiante en una gestión, con las materias a cursar.	
Referencia cruzada	R3,R4,R11	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex con rol de inscripción, busca a estudiante.	2. El sistema despliega datos del estudiante y las materias a cursar.
	3. Selecciona paralelos para las materias e inscribe.	3. El sistema verifica datos.
		4. El sistema almacena los datos de inscripción.
	5. Selecciona la opción imprimir boleta de inscripción.	6. el sistema imprime boleta de inscripción.
	6. Entrega una copia a estudiante y el otro adjunta a su file.	
Flujo alternativo	En 1. Si no se encuentra a estudiante proceder al registro de estudiante, en caso de tener ese rol. En todo caso se comunica con administrador. En 5. Directamente puede imprimir boleta.	

**Fuente:** Elaboración propia

**Tabla 3.8:** Descripción caso de uso - Examen de clasificación

Caso de uso	Inscripción a examen de clasificación	
Actores	Estudiante, personal Kardex	
Propósito	Inscribir al estudiante para optar la modalidad de examen de clasificación.	
Referencia cruzada :	R6	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex con rol de inscripción a examen de clasificación, busca a estudiante.	2. El sistema despliega datos del estudiante y una observación indicando si puede o no optar esta modalidad.
	3. Selecciona la materia en la cual se inscribirá.	4. El sistema almacena los datos.
Flujo alternativo	En 1. Ir a consultas mostrar si esta ya registrado..	

**Fuente:** Elaboración propia

**Tabla 3.9:** Descripción caso de uso - Retiro / Adición

Caso de uso	Retiro / adición	
Actores	Estudiante, personal kardex	
Propósito	Retirar o adicionar materias de estudiantes inscritos, a objeto de solucionar algún inconveniente.	
Referencia cruzada :	R5	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex busca a estudiante para retirar materia mediante código.	2. El sistema despliega datos de inscripción de estudiante y las materias a retirar.
	3. Selecciona la materia a retirar y ejecuta la opción retirar.	4. El sistema retira las materias seleccionadas por el usuario.
	5. Personal de Kardex busca a estudiante para adicionar materia mediante código.	6. El sistema despliega datos de inscripción de estudiante y las materias a adicionar.
	7. Selecciona las materias a adicionar y ejecuta la opción adicionar	8. El sistema adiciona las materias seleccionadas por el usuario.
Flujo alternativo	En 1. Es puede ejecutar la opción 5 ya que se muestra también esta opción.	

**Fuente:** Elaboración propia

**Tabla 3.10:** Descripción caso de uso notas – Crear planilla

Caso de uso	Notas – Crear planilla	
Actores	Docente, personal kardex	
Propósito	Generar planilla de notas, en forma de pre-actas.	
Referencia cruzada	R9	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex selecciona paralelo, gestión, y materia para generar planilla.	2. El sistema genera la planilla de PRE-ACTAS en archivo Excel.
	3. Selecciona archivo pre-acta y verifica datos.	
	5. Envía archivo (s) pre-acta a medio magnético de docente.	
Flujo alternativo	En 1. Abre carpeta que contiene archivos y envía a medio magnético de docente.	

**Fuente:** Elaboración propia

**Tabla 3.11:** Descripción caso de uso notas – Cargar planilla

Caso de uso	Notas – Cargar planilla	
Actores	Docente, personal kardex	
Propósito	Cargar planilla de notas.	
Referencia cruzada	R10	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Personal de Kardex selecciona planilla con notas llenadas por docente.	2. El sistema carga el archivo Excel seleccionado, verificando integridad de datos.
		3. El sistema muestra detalles de recuperación de datos de archivo y la opción confirmar cargado.
	4. Selecciona la confirmación de cargado de datos.	5. El sistema almacena los datos recuperados.
	6. El docente revisa los datos en pantalla expresando su conformidad.	
	7. El funcionario de kardex elige la opción generar ACTA de calificación.	8. El sistema genera actas oficiales.
Flujo alternativo	En 1. En consultas verificar actas generadas.	

**Fuente:** Elaboración propia

Las demás descripciones de casos de uso se detallan en anexo F, donde se muestra los demás casos de uso.

### 3.2.1.5. GLOSARIO DE TERMINOS

Glosario de términos es la descripción textual de los elementos del modelo, para explicar términos que podrían ser ambiguos. La tabla 3.9 muestra el: término, categoría y descripción.

**Tabla 3.12:** Glosario de términos

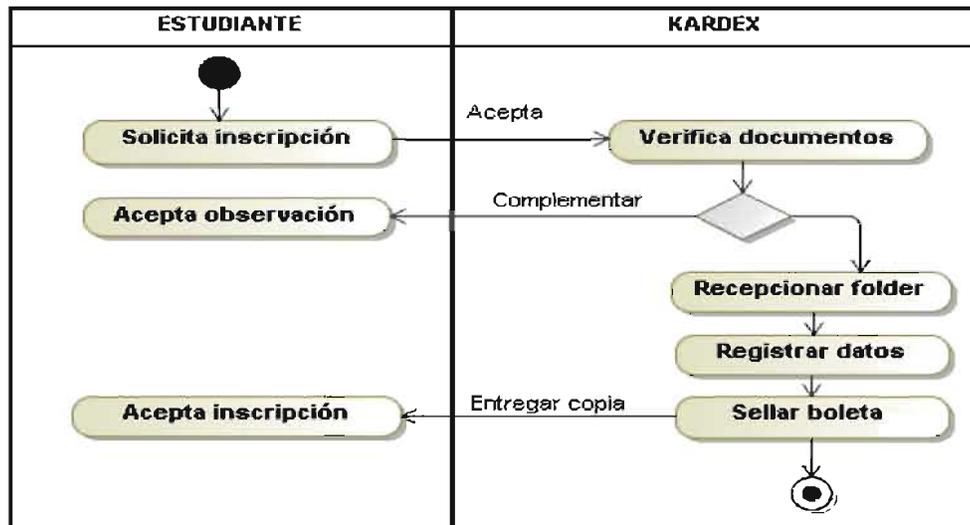
TERMINO	CATEGORIA	DESCRIPCIÓN
Pre-Actas	Tipo	Planilla donde se registran todas las notas de estudiantes, que se encuentren inscritos y que obtengan una calificación.
Kardex	Actor	Funcionario de kardex
Inscripción	Caso de uso	Proceso, en el cual un estudiante se inscribe en una o más materias.
Boleta de inscripción	Tipo	Funcionarios de kardex emite boleta de inscripción de materias inscritas.
Actas	Tipo	Reporte final de nota de estudiantes que cursaron el semestre.
Estudiante	Actor	Persona del cual se administran sus datos.
Administración de sistema	Caso de uso	Persona funcionario de kardex que tiene acceso a todos los módulos del sistema.
Retiro y/o adición	Caso de uso	Función del sistema que permite retirar o adicionar materias a estudiantes inscritos.
Consultas	Caso de uso	Función del sistema que permite realizar consultas mas frecuentes.
Inscripción a examen de clasificación	Caso de uso	Proceso que permite a los estudiantes inscribirse para liberar materias de esta modalidad.
Emisión de documentos	Caso de uso	Función del sistema que permite emitir documentos como: record académico y certificado de notas.
Notas	Caso de uso	Proceso mediante el cual el sistema genera y carga planilla de notas.
Docente	Actor	Persona física que asigna una calificación a estudiantes de la materia que dicta.

**Fuente:** Elaboración propia

### 3.2.1.6. DIAGRAMA DE ACTIVIDADES

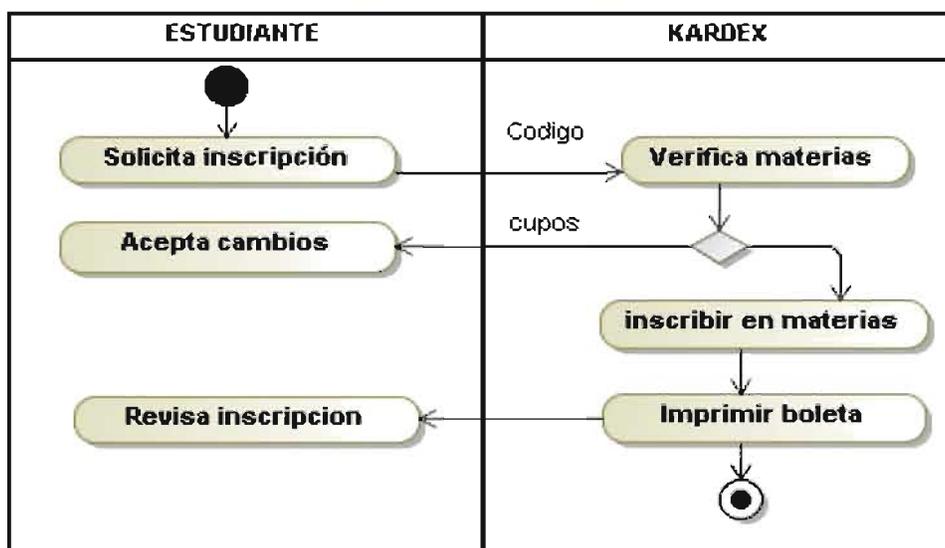
Los diagramas de actividades nos muestran el flujo de actividades entre los elementos del dominio, como podemos ver en la figura 3.8.

Figura 3.6: Diagrama de actividad - Registro de estudiante



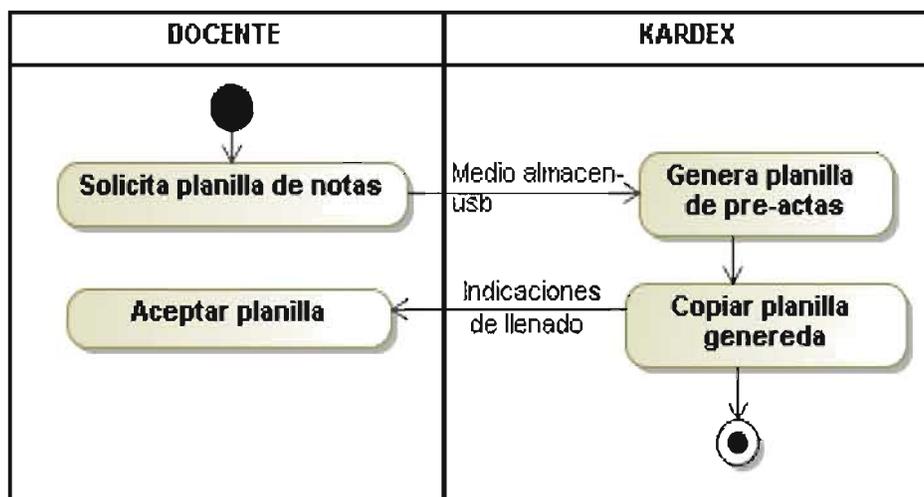
Fuente: Elaboración propia

Figura 3.7: Diagrama de actividad - Inscripción



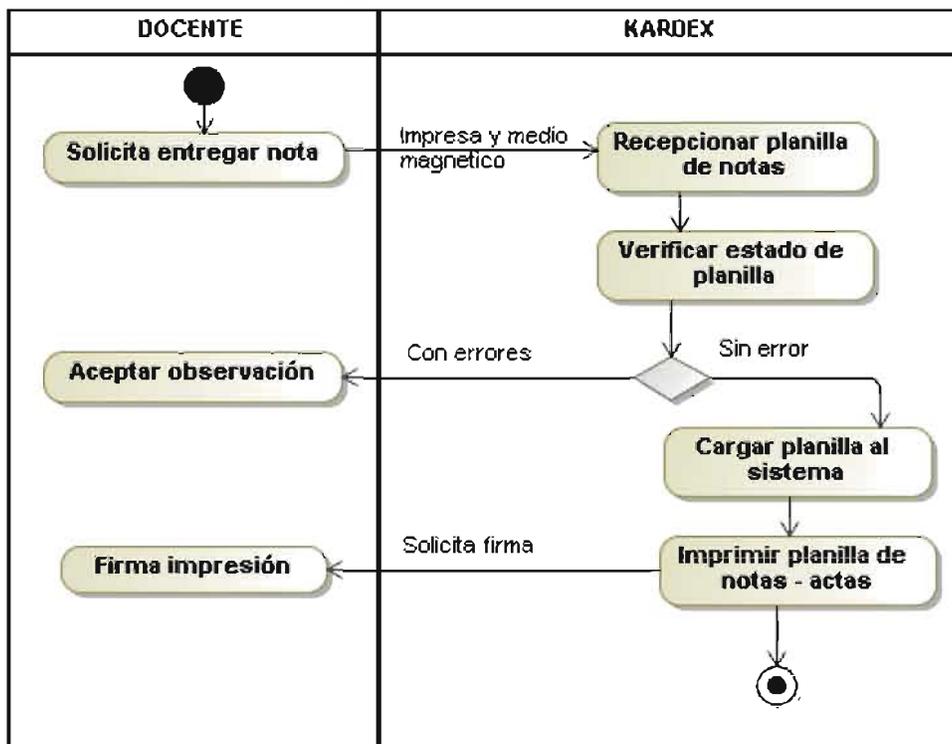
Fuente: Elaboración propia

Figura 3.8: Diagrama de actividad - Generación de planillas.



Fuente: Elaboración propia

Figura 3.9: Diagrama de actividad – Cargar planilla con notas.



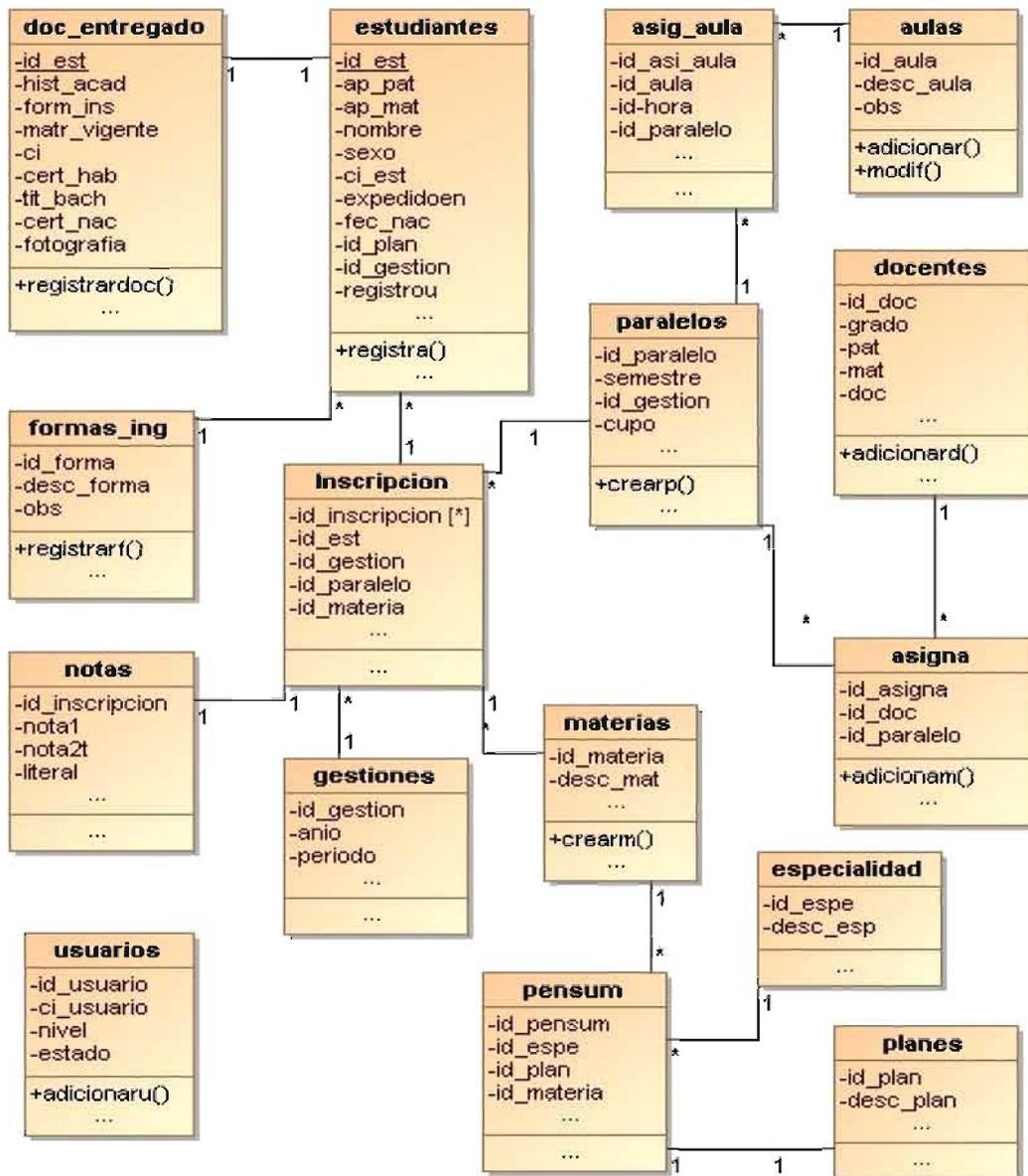
Fuente: Elaboración propia

Los demás diagramas de actividades se muestran en el anexo F.

### 3.2.2. MODELO CONCEPTUAL

El modelo conceptual muestra las principales entidades del sistema, la misma podemos representarlo por un diagrama de clases. La base sobre la cual se identifica estos conceptos es a partir de los requisitos y el contexto en la cual se basan. El diseño conceptual preliminar del sistema es como en la figura 3.6

Figura 3.10: Modelo conceptual

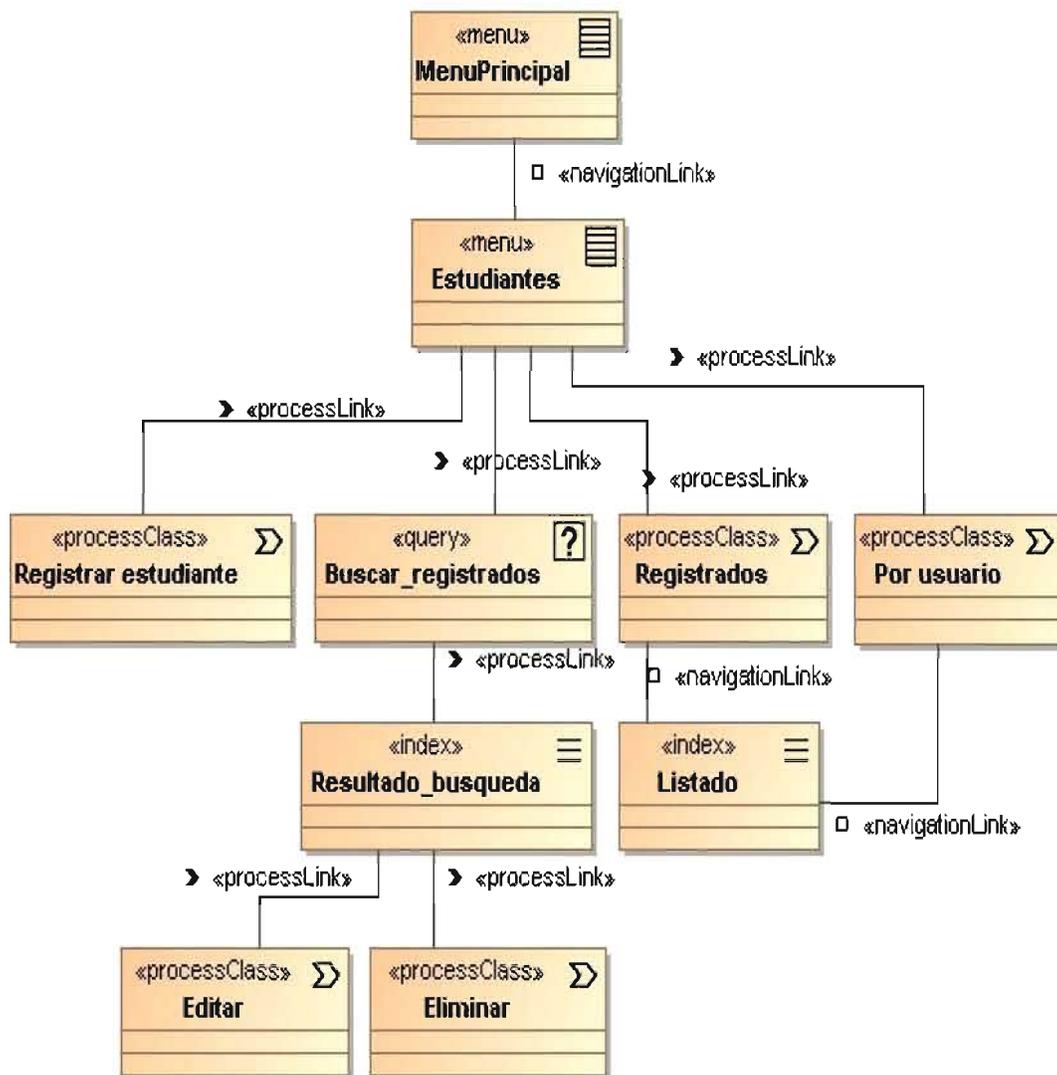


Fuente: Elaboración propia

### 3.2.3. MODELO NAVEGACIONAL

En esta parte especificaremos que objetos pueden ser visitados mediante la navegación a través de la aplicación web. Las mismas son presentadas por diagrama de clases estereotipadas. Empecemos mostrando el modulo de registro de estudiantes en la figura 3.11, esta disponible para usuario administrador, también para usuarios con rol registro.

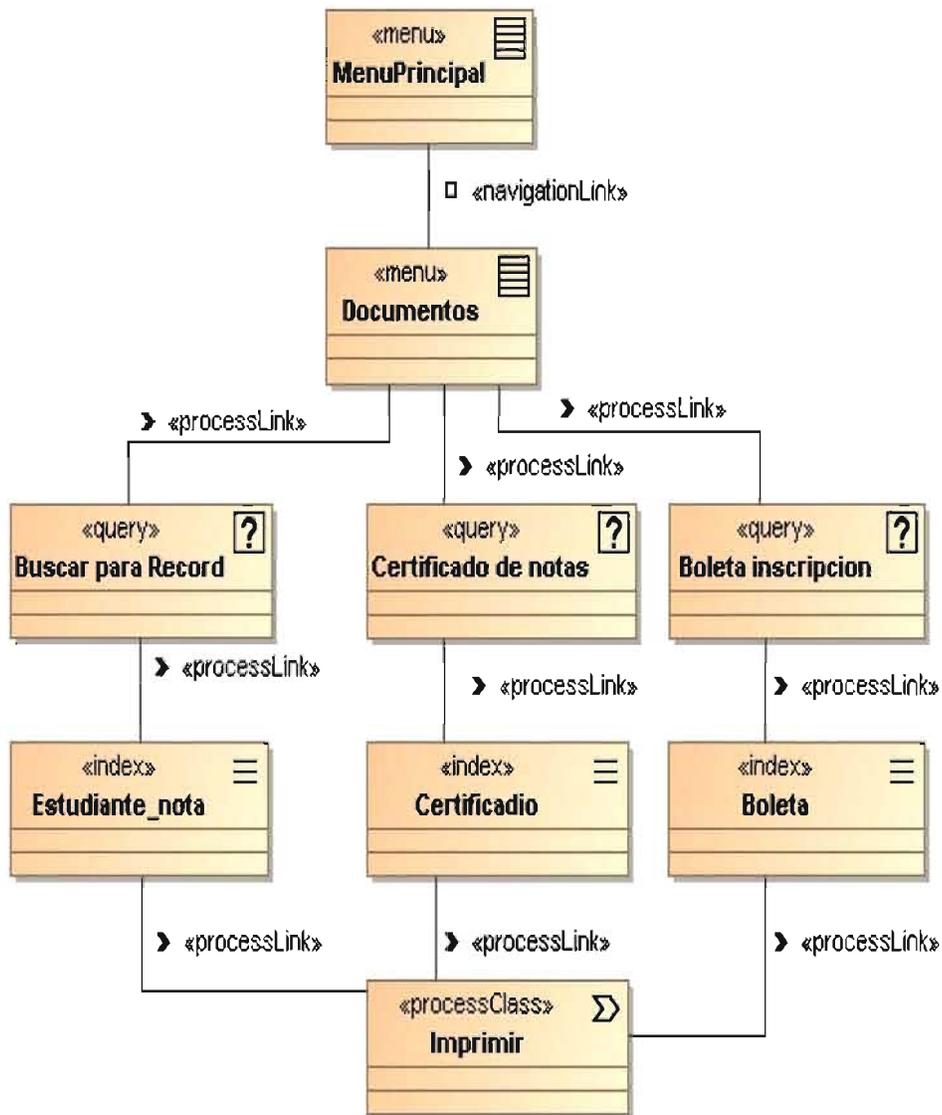
Figura 3.11: Diagrama de navegación - Estudiantes



Fuente: Elaboración propia

El módulo de documentos nos permite acceder a los operaciones de emisión de documentos como ser: record académico, certificado de notas, alumno regular, conclusión de estudios y boleta de inscripción. La figura 3.12 ilustra la navegación por documentos, esta opción estará disponible para usuarios de kardex que tengan este rol de manejo de documentos.

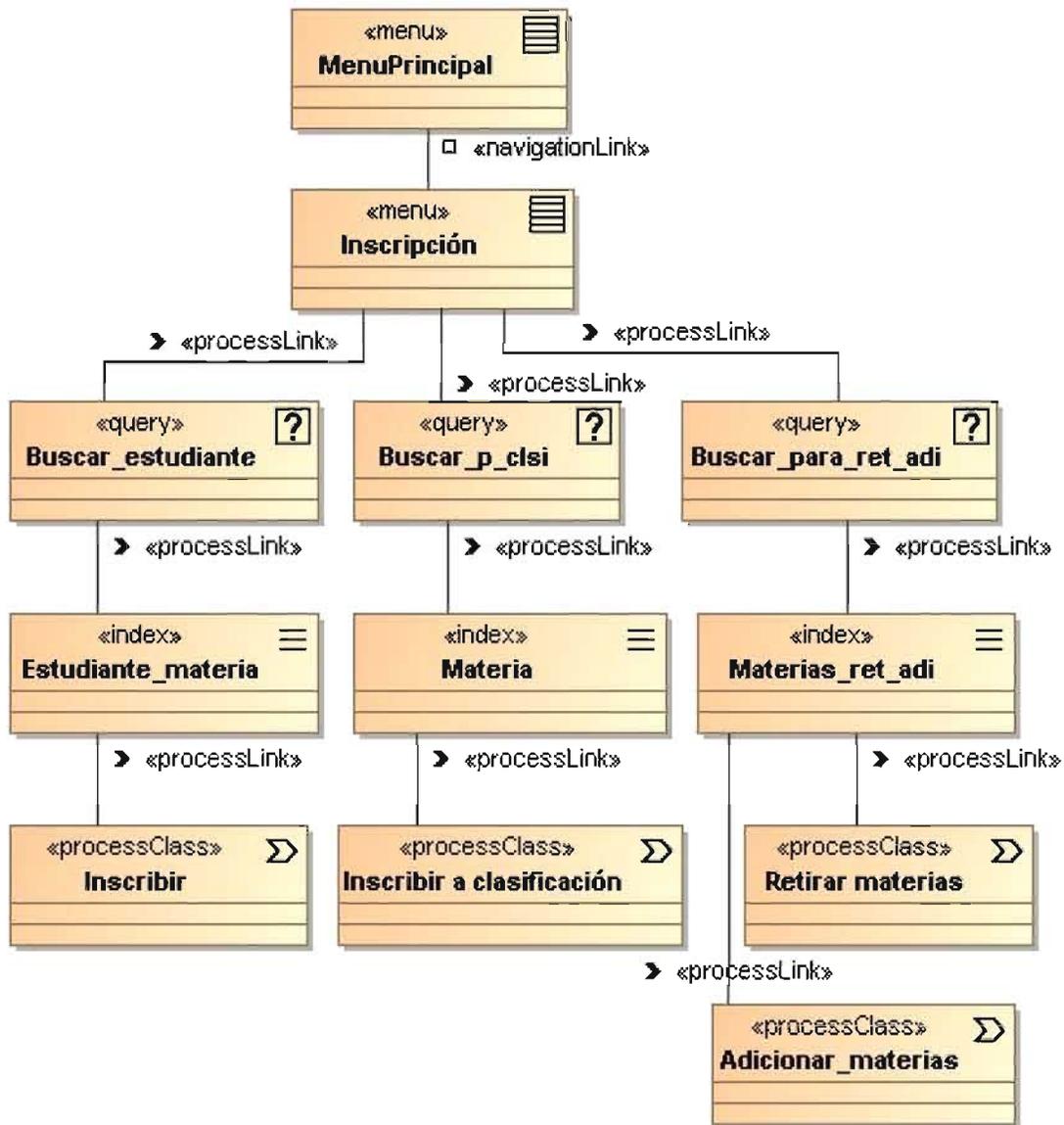
**Figura 3.12:** Diagrama de navegación - Documentos



Fuente: Elaboración propia

El módulo de inscripción nos permite navegar a través de esta opción accediendo a las opciones como ser: buscar estudiante para inscribir en una gestión, en examen de clasificación, retirar y/o adicionar materias. La figura 3.13, ilustra esta opción.

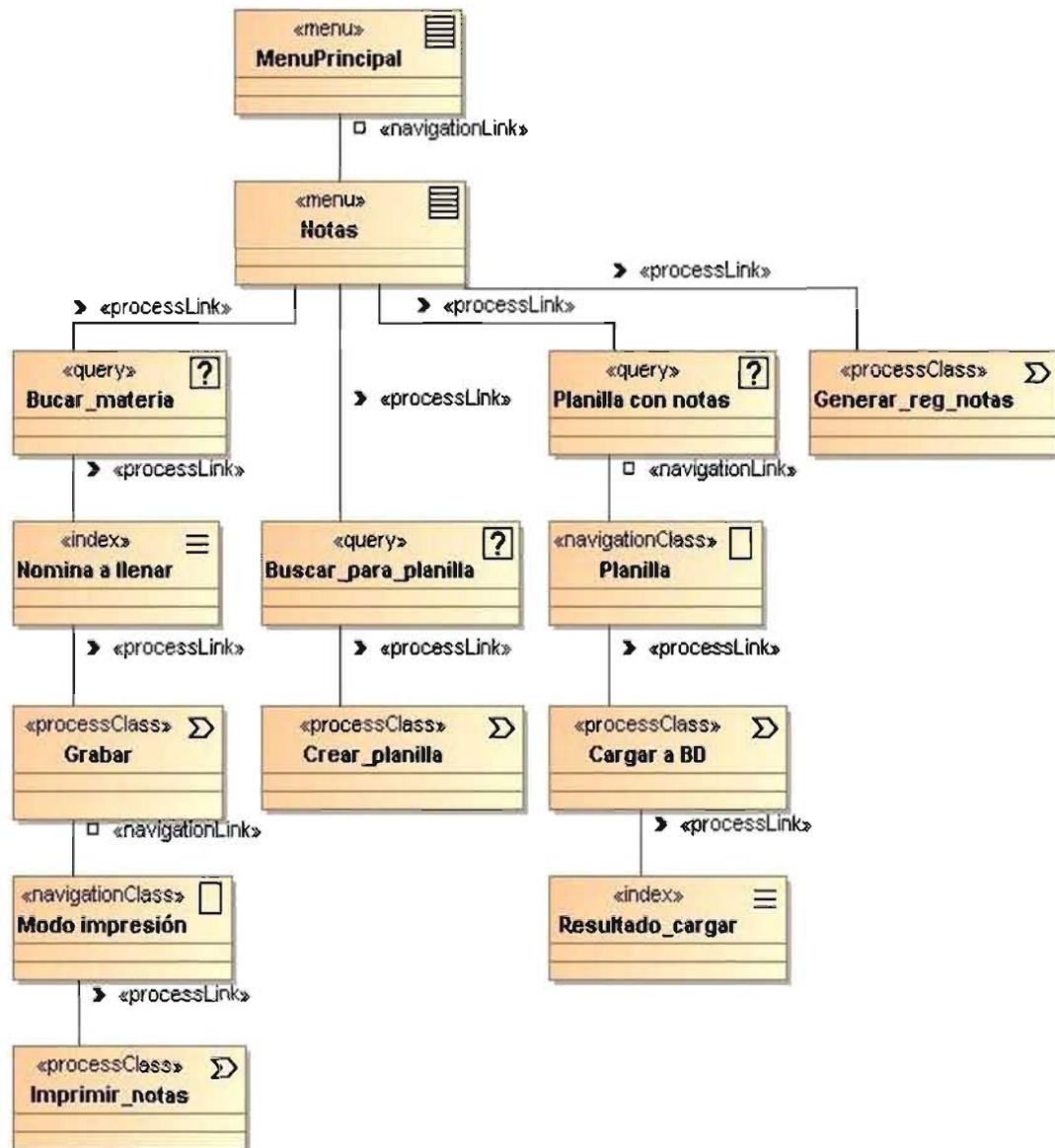
Figura 3.13: Diagrama de navegación - Inscripción



Fuente: Elaboración propia

El módulo de notas nos permite navegar a través de estas opciones a los demás enlaces como ser: llenar notas transcribiendo en sistema, generar planilla de notas, cargar planilla con notas y generar registro de notas a partir de la inscripción. La figura 3.14, muestra esta opción.

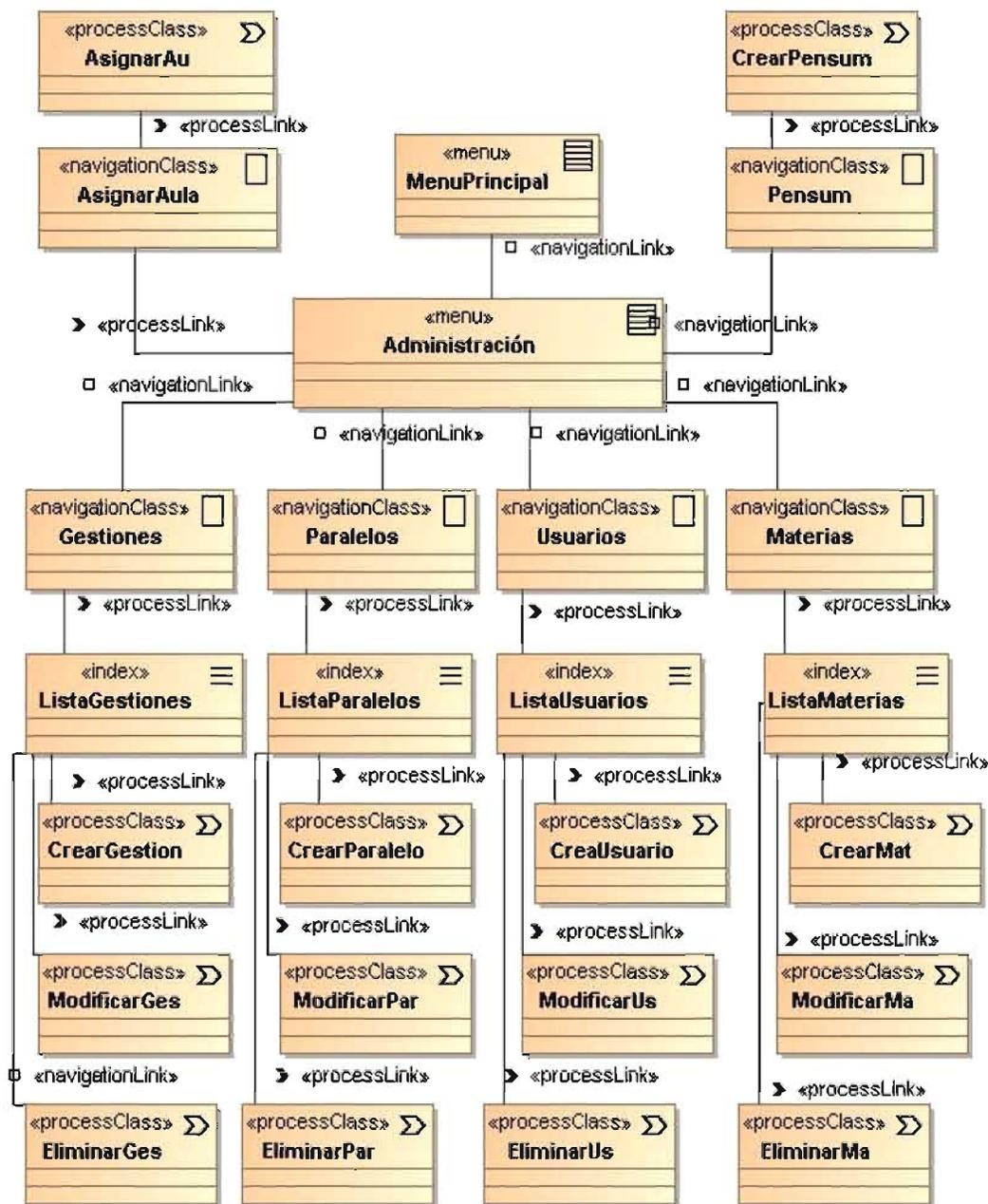
Figura 3.14: Diagrama de navegación - Notas



Fuente: Elaboración propia

El módulo de administración nos permite configurar el sistema con opciones como ser: gestiones, pensum, paralelos, usuarios, materias y aulas. Véase La figura 3.15.

Figura 3.15: Diagrama de navegación - Administración



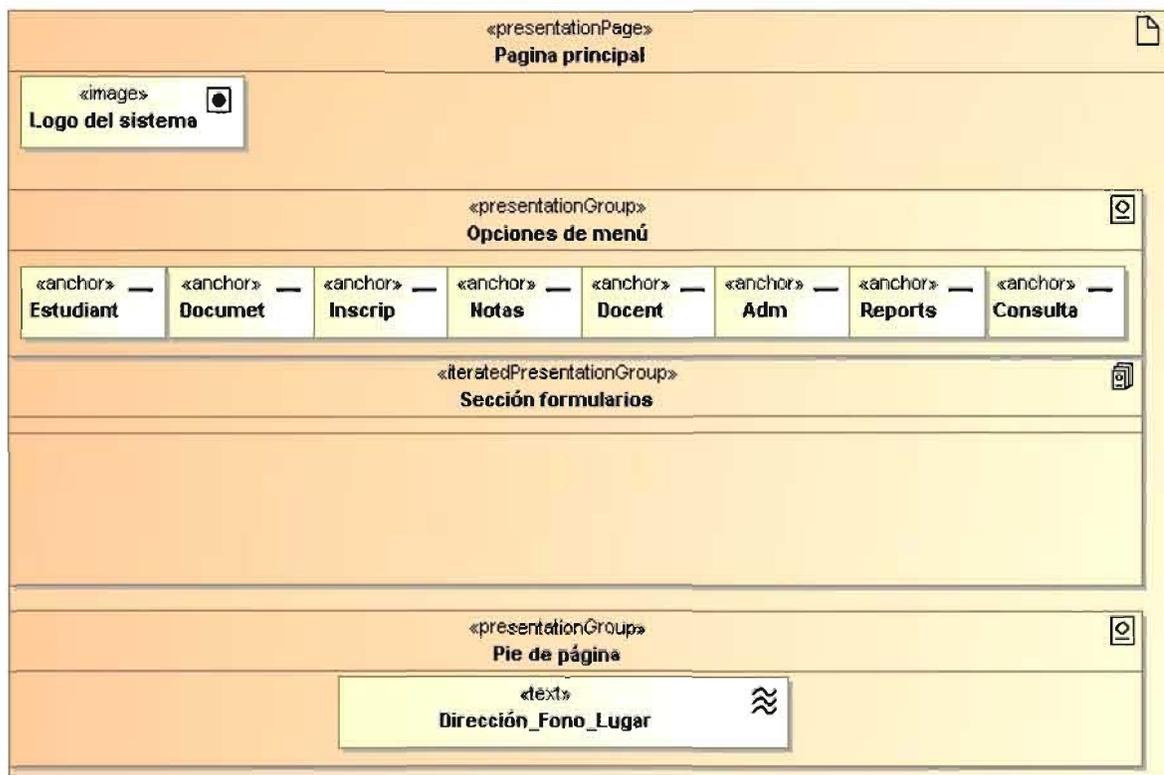
Fuente: Elaboración propia

Los demás diagramas están en anexo H.

### 3.2.4. MODELO DE PRESENTACIÓN

El modelo de presentación nos permite mostrar en forma abstracta de interfaz de usuario (UI) de la aplicación, mostrando el contenido y estructura de nodos simples. Primeramente vemos en la figura 3.16 el modelo de presentación de la página principal del sistema.

**Figura 3.16:** Diagrama de presentación - Pagina principal



Fuente: Elaboración propia

La figura 3.16 muestra el nodo principal del sistema, dicho diagrama se encuentra detallada con estereotipos de texto, grupo, grupo de alternativas, página y la sección de formulario. En la sección formularios se mostrarán los demás formularios que integra el sistema, en forma interactiva. El segundo formulario que se muestra es el diagrama de presentación registro de estudiantes, como en la figura 3.17.

Figura 3.17: Diagrama de presentación - Registro de estudiantes

«inputForm»  
**Registrar\_estudiante**

«text»  
**Título del módulo**

---

<span>«textInput»</span> <b>Ap_pat</b>	<span>«textInput»</span> <b>Ap_mat</b>	<span>«textInput»</span> <b>Nombre</b>	<span>«selection»</span> <b>SexoMF</b>		
<span>«textInput»</span> <b>Cedula</b>	<span>«selection»</span> <b>Expedido</b>	<span>«text»</span> <b>Fec_nac</b>	<span>«textInput»</span> <b>Día</b>	<span>«selection»</span> <b>Mes</b>	<span>«selection»</span> <b>Año</b>
<span>«selection»</span> <b>Pais</b>	<span>«selection»</span> <b>Lugar</b>	<span>«textInput»</span> <b>Provincia</b>			

«text»  
**Datos de referencia**

---

<span>«text»</span> <b>Dirección</b>	<span>«textInput»</span> <b>Zona</b>	<span>«textInput»</span> <b>Calle_Av.</b>	<span>«textInput»</span> <b>Número</b>
<span>«textInput»</span> <b>Fono</b>	<span>«textInput»</span> <b>Correo</b>	<span>«selection»</span> <b>Forma de ingreso</b>	

«text»  
**Documentos presentados**

---

<span>«selection»</span> <b>Historial</b>	<span>«selection»</span> <b>Matricula</b>	<span>«selection»</span> <b>Certificado hab.</b>	<span>«selection»</span> <b>FormInscripcion</b>	<span>«selection»</span> <b>FotocopiaCI</b>
<span>«selection»</span> <b>TítuloBachiller</b>	<span>«textInput»</span> <b>NúmeroTit</b>	<span>«textInput»</span> <b>institucionTítulo</b>	<span>«textInput»</span> <b>AñoTítulo</b>	

«text»  
**Datos academicos**

---

<span>«textInput»</span> <b>Codigo</b>	<span>«textInput»</span> <b>Contraseña</b>	<span>«textInput»</span> <b>Registro_U</b>	<span>«textInput»</span> <b>MatriculaU</b>	<span>«textInput»</span> <b>Nro Folder</b>
<span>«selection»</span> <b>PeriodoIngreso</b>	<span>«selection»</span> <b>Especialidad</b>	<span>«selection»</span> <b>Plan</b>	<span>«fileUpload»</span> <b>Fotografia</b>	

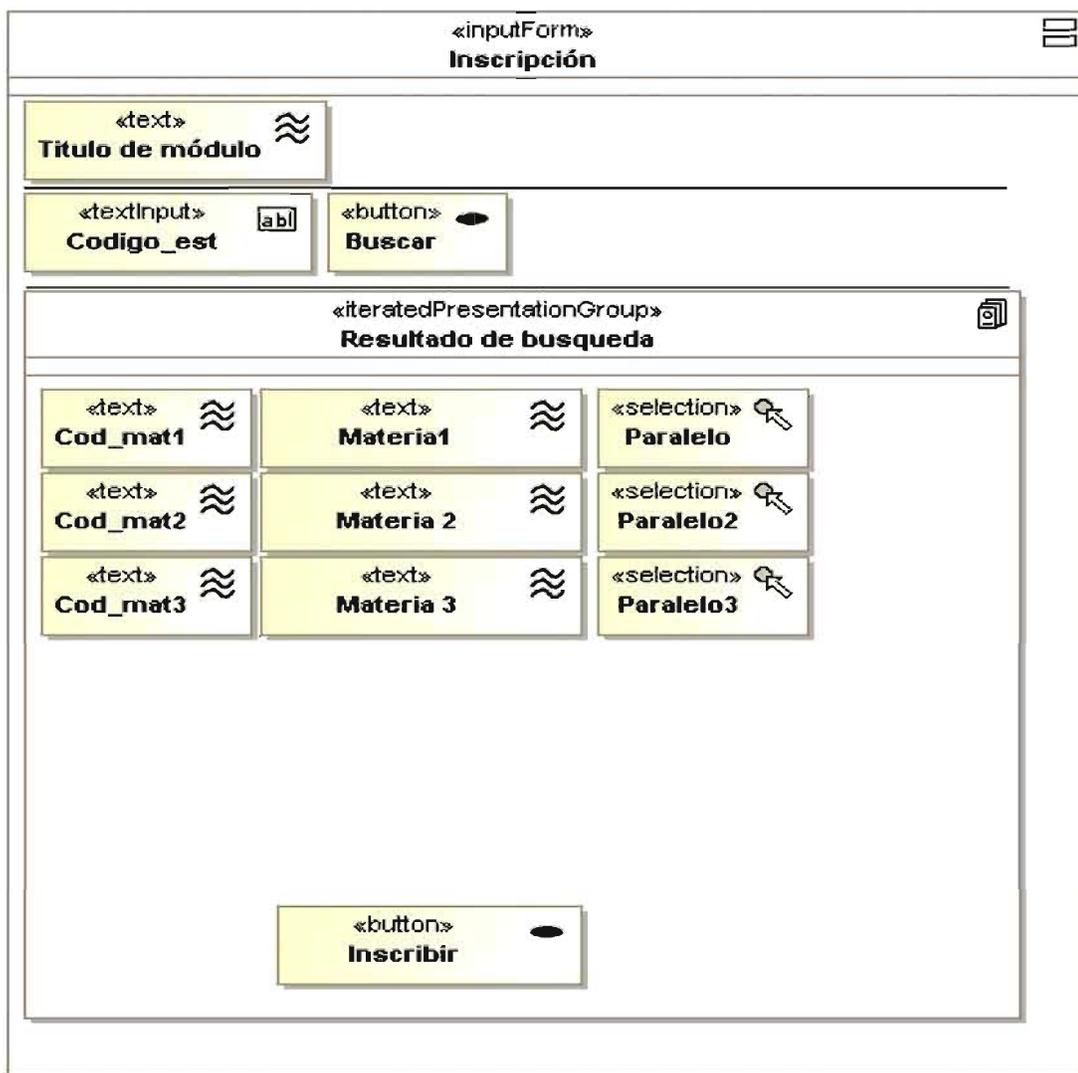
«textInput»  
**Observaciones**

«button»  
**Guardar**

Fuente: Elaboración propia

El formulario que se despliega al ingresar en la opción inscripción, es representada en el diagrama de presentación inscripción de la figura 3.18. El formulario solicita al usuario introducir el código de estudiante valido y proceder a la respectiva búsqueda.

Figura 3.18: Diagrama de presentación - Inscripción



Fuente: Elaboración propia

Si el resultado es exitoso, se muestra las materias a los cuales el estudiante pueda inscribirse.

Para generar planillas utilizamos el diagrama de presentación generar planilla de la figura 3.19 y para el cargado de notas el diagrama de presentación cargar planilla de notas de la figura 3.20.

Figura 3.19: Diagrama de presentación - Generar planilla

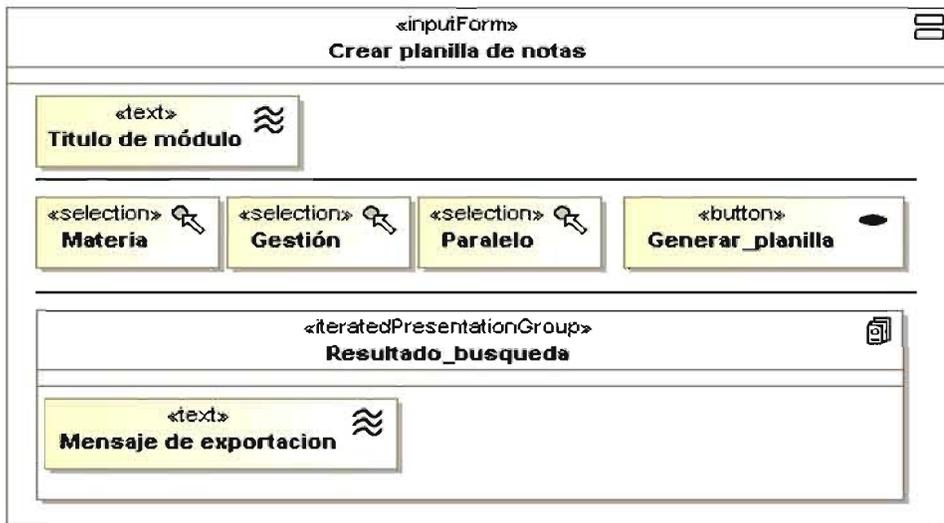
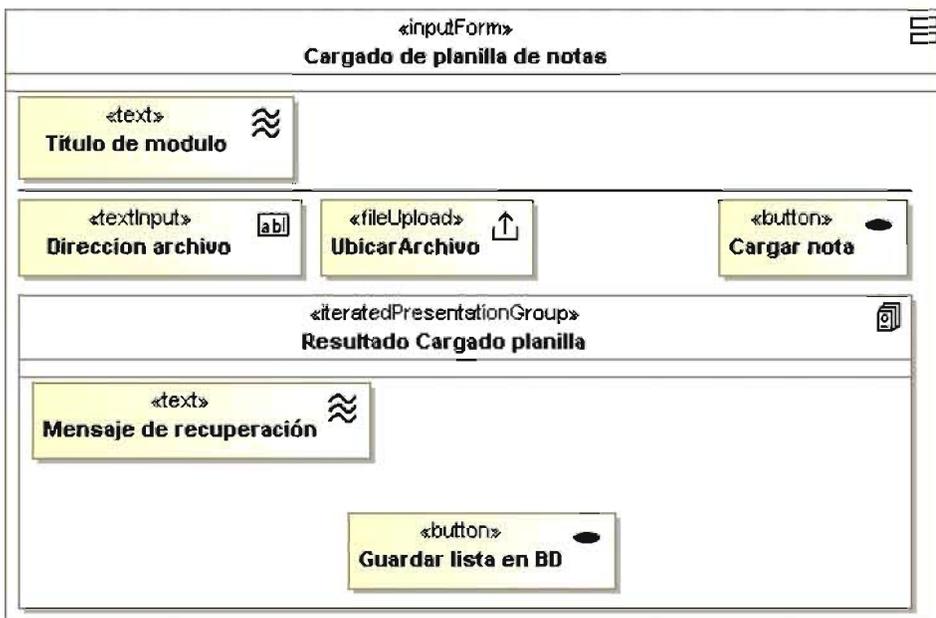


Figura 3.20: Diagrama de presentación - Cargar planilla

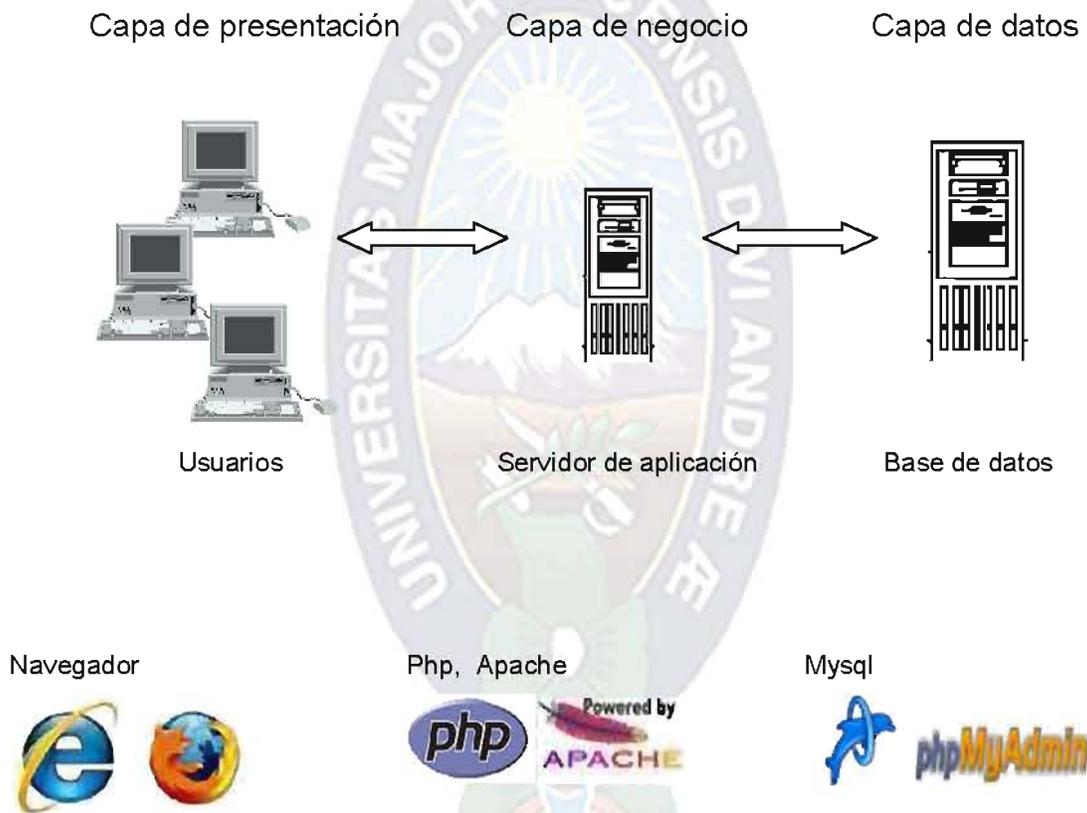


Los restantes diagramas de presentación se muestran en anexo I.

### 3.2.5. ARQUITECTURA DEL SISTEMA

En este punto se realiza una descripción de la arquitectura 3-capas: capa de interfaz o presentación, capa lógica de negocios y una capa de base de datos. La misma se ilustra en la figura 3.21.

Figura 3.21. Arquitectura del sistema



Fuente: Elaboración propia

Para la interfaz de usuario se utiliza un navegador de internet como Internet Explorer, Firefox u otros. Para la lógica interna se utiliza Php y Apache que nos permiten interactuar usuario y base de datos. Por otro lado MySQL, con su interfaz de manejador PhpMyadmin u otro manejador, nos permiten almacenar la estructura física de los datos.

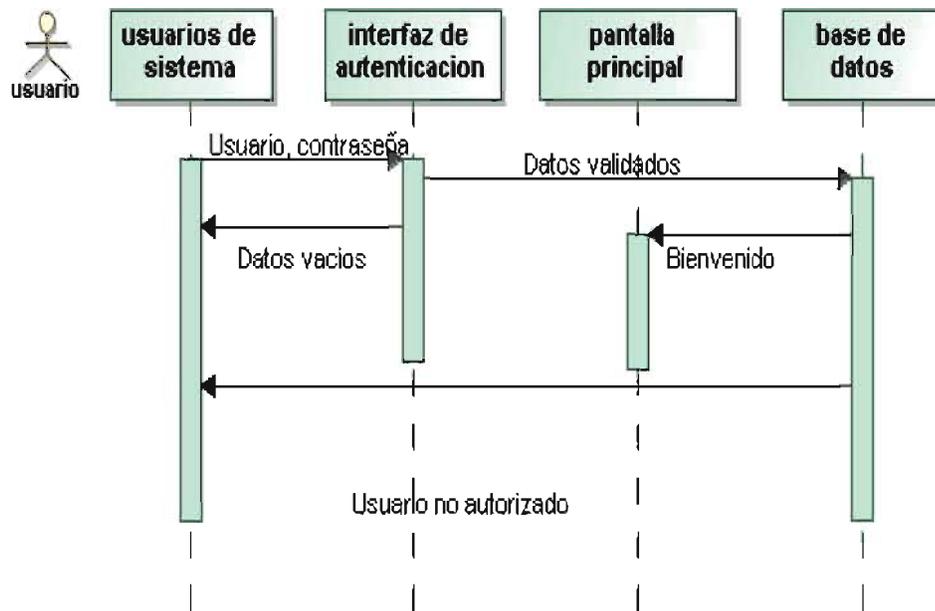
### 3.3. FASE DE CONSTRUCCIÓN

Como se vio en el capítulo anterior el objetivo de esta disciplina es transformar el modelo en código fuente, en este caso script y llevar las pruebas más básicas.

#### 3.3.1. DIAGRAMA DE SECUENCIA

Los diagramas de secuencia muestran de manera grafica los eventos u operaciones que los actores ejecutan al interactuar con el sistema. La primera interacción que todo actor realiza es el acceso al sistema, la misma es ilustrada en la figura 3.22.

Figura 3.22: Diagrama de secuencia - Acceso al sistema

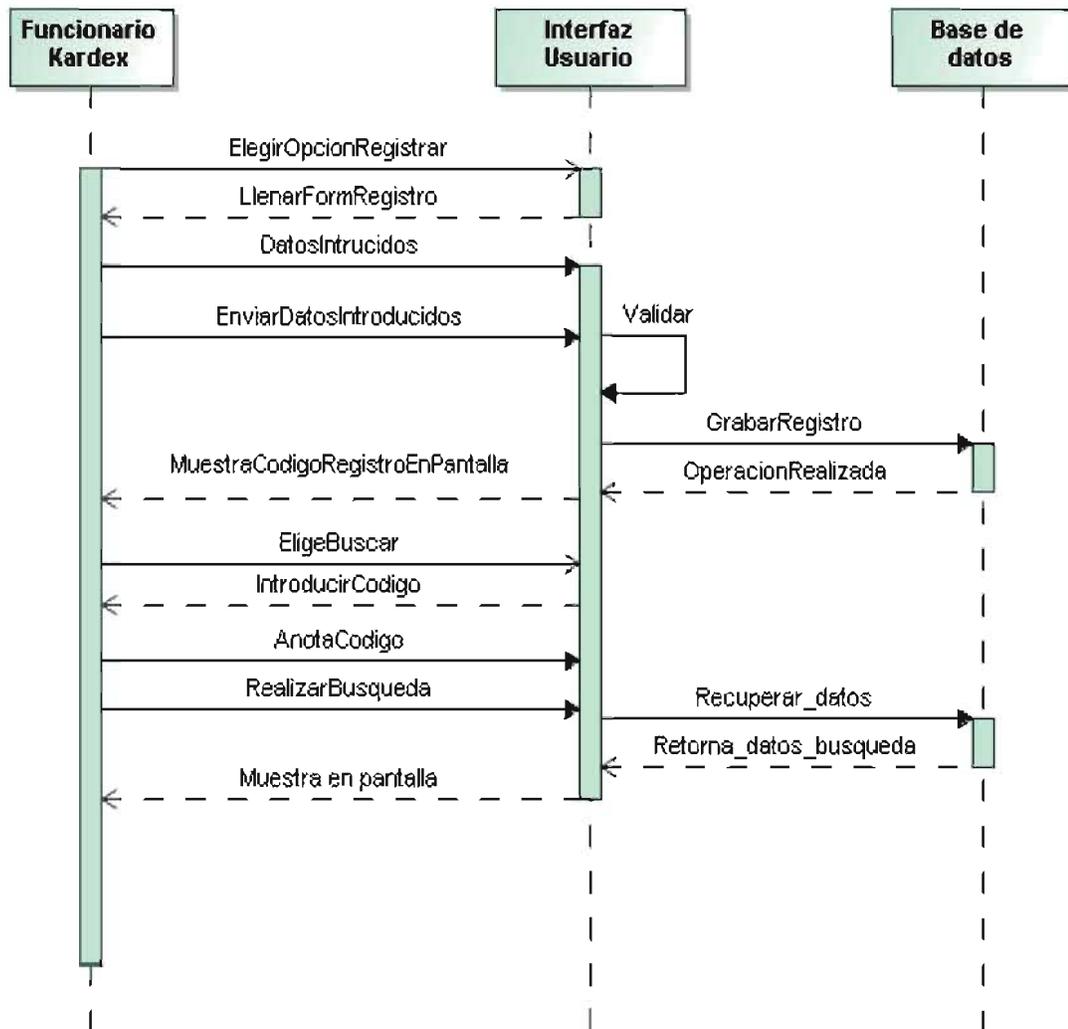


Fuente: Elaboración propia

El sistema solicita al usuario se identifique a través de la introducción de un usuario y una contraseña. En caso de que las mismas sean las correctas y estén almacenadas en la base de datos, se le habilitara la pantalla principal con las opciones de acuerdo al privilegio otorgado.

Una vez autenticado el usuario, este ya puede trabajar a partir del menú principal del sistema. Una de las opciones disponibles del sistema es el de registro de estudiantes, la misma es como en la figura 3.23.

**Figura 3.23:** Diagrama de secuencia – Registro de estudiantes

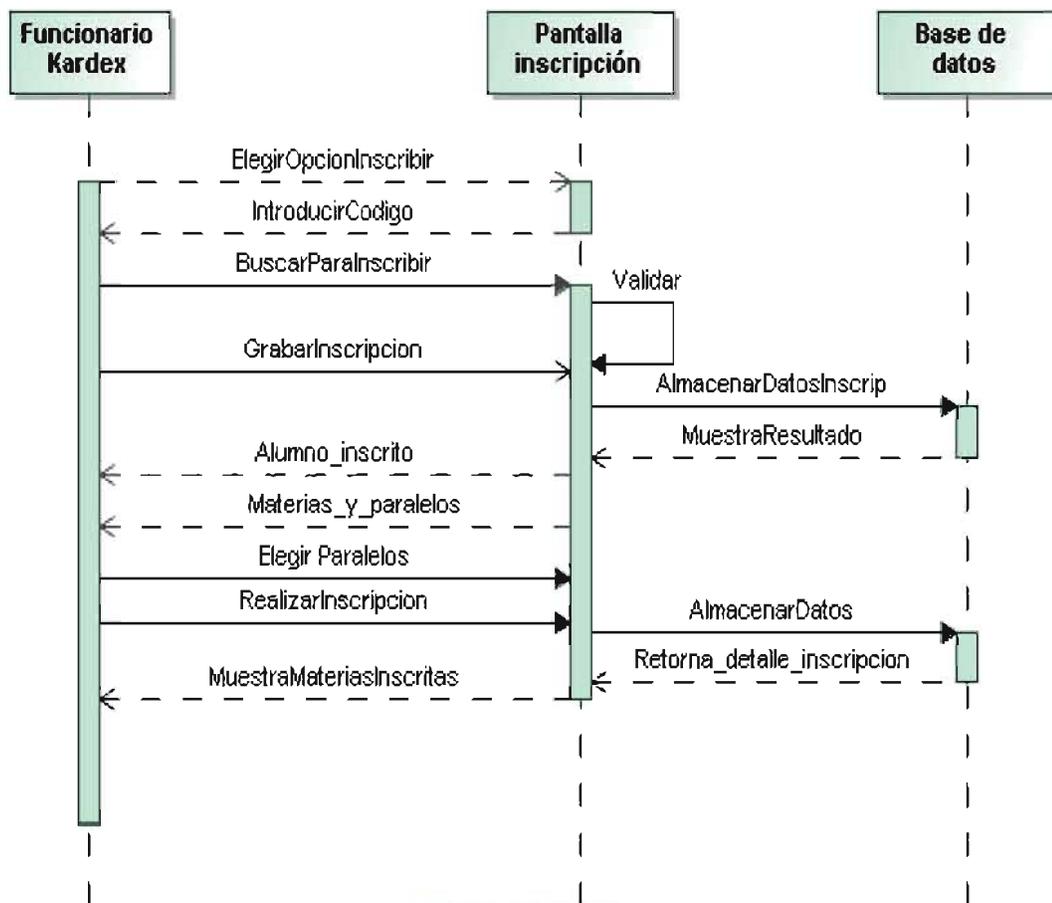


Fuente: Elaboración propia

Previo al proceso de registro de estudiantes, el funcionario de Kardex verifica si todos los documentos están en orden. Se registrara únicamente a los estudiantes a registrar no este en sistema, caso contrario el sistema desplegara el estudiante que intenta registrar ya existe.

Una vez concluida con el proceso de registro, los datos ya están disponibles para su posterior uso. Es así, que para la inscripción en una nueva gestión se utiliza estos datos. La misma se ilustra en la figura 3.24, en caso de ya haber realizado su inscripción el sistema rechaza esa petición. Por otro lado, si cumple con los requisitos se procede a la inscripción.

**Figura 3.24:** Diagrama de secuencia – Inscripción



Fuente: Elaboración propia

Una vez que pasa el periodo de inscripción, se procede a la generación de listas respectivas por paralelo. Por otro lado también se generan las planillas notas en pre-actas para el llenado de notas por parte de los docentes. La misma podemos verlos en la figura 3.25, ya su respectivo cargado al sistema una vez que haya sido llenado como en la figura 3.26.

Figura 3.25: Diagrama de secuencia – Crear Planilla

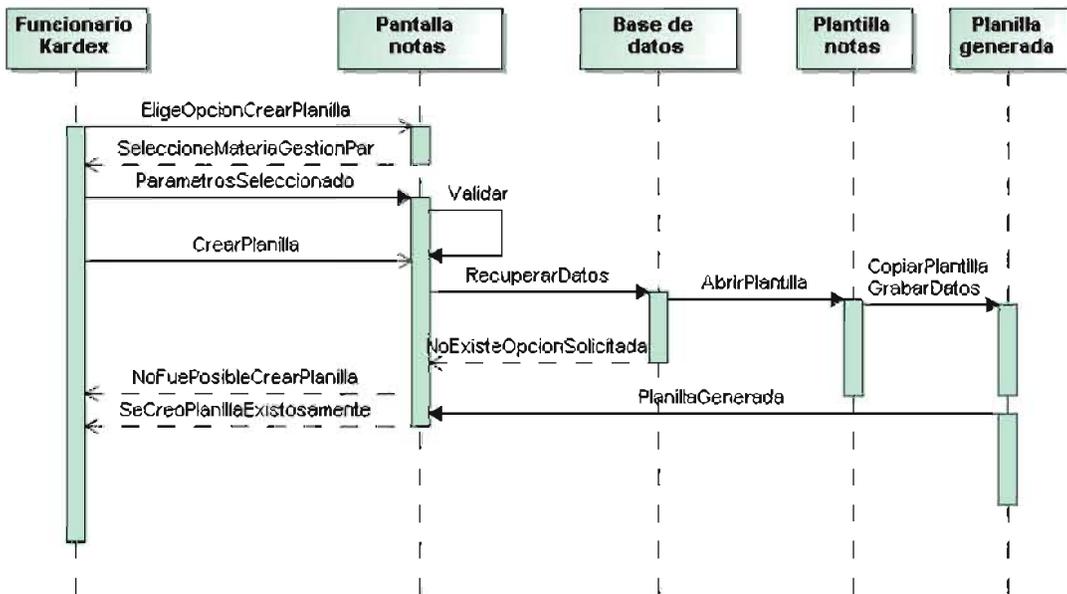
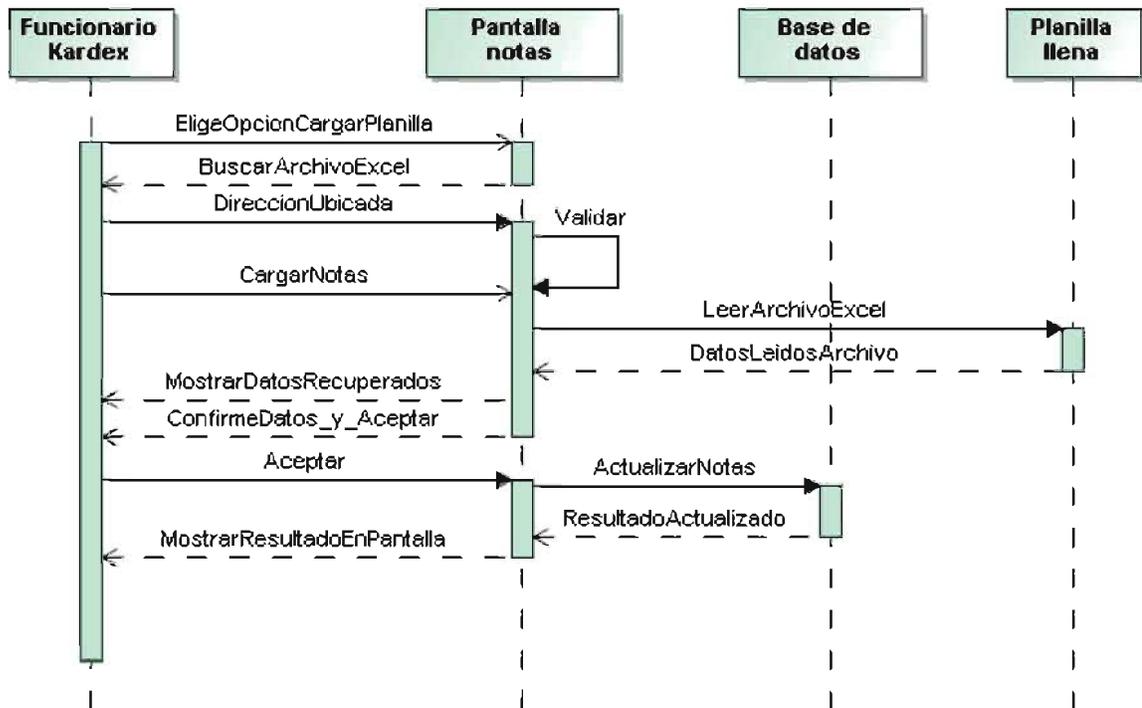


Figura 3.26: Diagrama de secuencia – Cargar Planilla



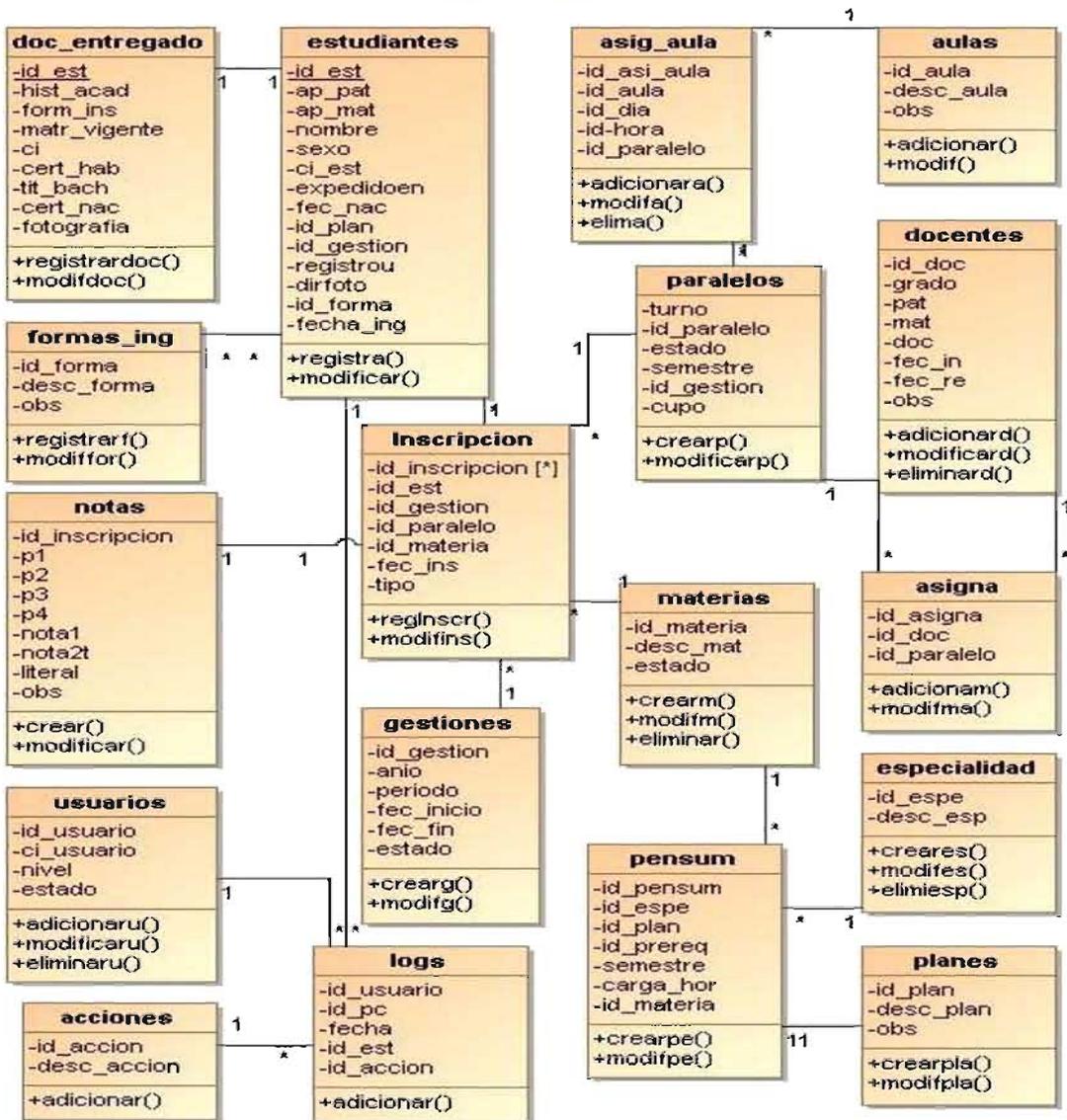
Fuente: Elaboración Propia

Los restantes diagramas de secuencia se encuentran en anexo J.

### 3.3.2. DIAGRAMA DE CLASES

El diagrama de clase a diferencia del modelo conceptual que es solo una abstracción del dominio, permite identificar las clases del software que interactúan para llevar a cabo los procesos del sistema. A partir de este diagrama clase es posible generar un modelo relacional de datos, donde una clase representa una tabla y un atributo un campo. Véase la figura 3.27.

Figura 3.27: Diagrama de clases del sistema

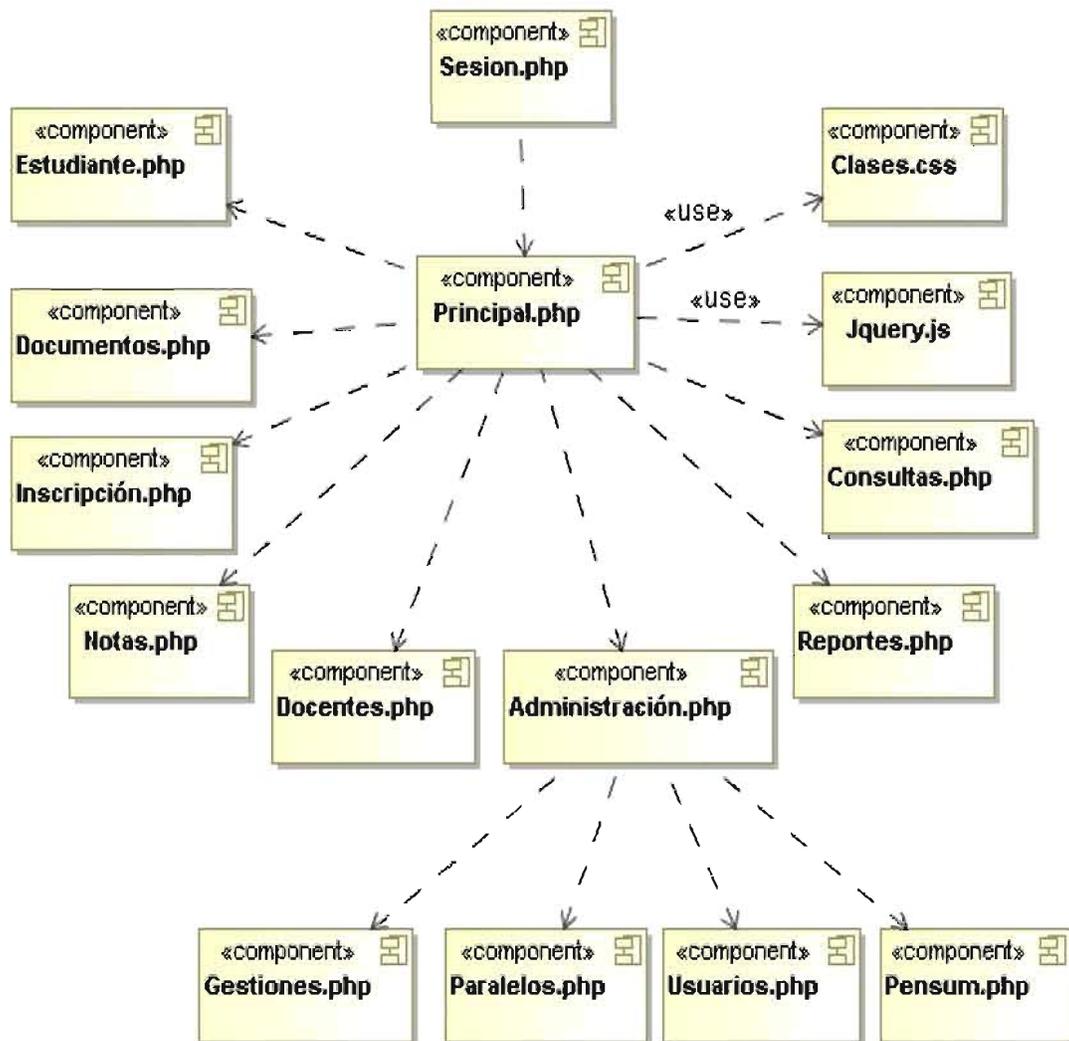


Fuente: Elaboración propia

### 3.3.3. DIAGRAMA DE COMPONENTES

Un diagrama de componente nos muestra la organización y las dependencias entre un conjunto de componentes del sistema. Se utilizan para agrupar clases en componentes o módulos. Véase figura 3.28, adicionalmente a los desarrollados se utiliza JQuery y las clases donde se almacena la configuración de interfaz de usuario.

Figura 3.28: Diagrama de componentes del sistema

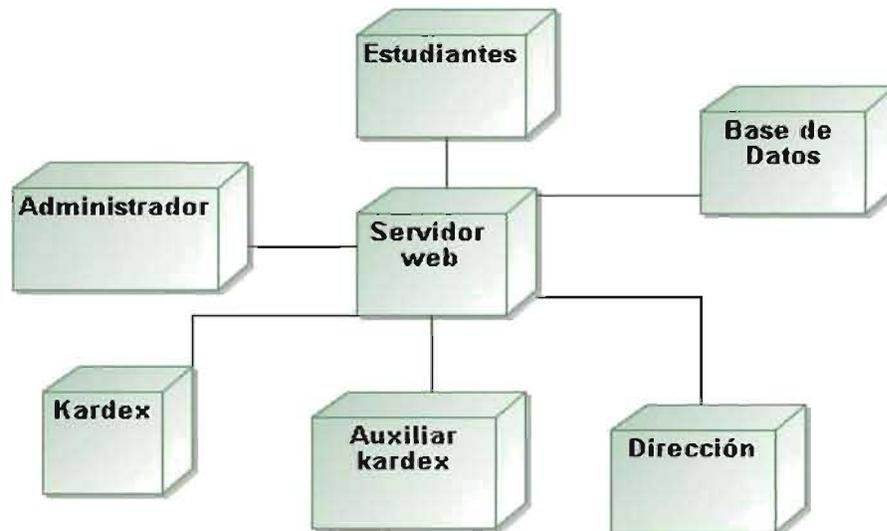


Fuente: Elaboración propia

### 3.3.4. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue nos muestra como y donde se instalara el sistema en forma física. La figura 3.29 ilustra una configuración básica.

Figura 3.29: Diagrama de despliegue del sistema



Fuente: Elaboración propia

### 3.3.5. ITERACIONES DEL SISTEMA

El desarrollo del sistema propiamente dicho se la realiza a través de tres iteraciones incrementales. En esta parte se prioriza los requerimientos mas críticos del sistema. Los desplegables se organizan como en la tabla 3.k.

Tabla 3.13: Cronograma de entregables

Nº	ENTREGABLE	ITER	FECHA
1	Modulo registro de estudiantes	1	10-10-2011
2	Módulos de inscripción, retiro/adición, clasificación	2	25-10-2011
3	Modulo de notas (semestrales, clasificación)	2	30-10-2011
4	Modulo de documentos (record, certificado notas)	2	05-11-2011
5	Modulo de consultas básicas	3	10-11-2011
6	Modulo de Reportes	3	20-11-2011

Fuente: Elaboración propia

### 3.3.5.1. PRIMERA ITERACIÓN

El módulo más importante es el registro de estudiantes, para el mismo es necesario tener previamente generado el pensum, materias, formas de ingreso, gestiones y planes con la cual se registran estudiantes al sistema. También se realiza el registro de usuarios para el manejo de sistemas, la misma se inicia con la pantalla de inicio de sesión como en la figura 3.30, esta pantalla es necesario ya para el despliegue es necesario el manejo de usuario.

**Figura 3.30:** Pantalla de inicio de sesión



Logo:  **Sistema de Información Académica**

**IDENTIFICACIÓN**

Usuario:

Contraseña:

Ingresar Cerrar

**Fuente:** Elaboración propia

Si el usuario y contraseña introducida por el funcionario de Kardex es correcta, el sistema despliega el menú de la figura 3.31.

**Figura 3.31:** Menú principal del sistema



**Fuente:** Elaboración propia

Recordemos que aquí se materializa el modelo de presentación Menú Principal diseñado en la fase de elaboración.

Para cumplir con esta iteración se desarrollo el modulo de registro de estudiantes, dicha pantalla se ilustra en la figura 3.32, la misma se muestra luego de ingresar en la opción estudiante del menú principal.

**Figura 3.32:** Módulo estudiantes – Registrar estudiante

## ESTUDIANTES

Registro y habilitación de estudiantes

Registrar
Registrados
Habilitar

**DATOS PERSONALES**

Sexo    
 (\*)Ap. Paterno (\*)Ap. Materno (\*)Nombre(s) (\*) M F

--  Fecha nacimiento  -- Mes -- --Año--  
 (\*)Cedula Exp. Día Mes Año

BOLIVIA LA PAZ   
 País Lugar de nacimiento Provincia

---

**DATOS DE REFERENCIA**

Dirección     
 (\*)Zona Calle/Av. Número

Fono  Correo  Forma Ingreso Elija esp

---

**DOCUMENTOS EN FILE**

SI	NO	TIPO DE DOCUMENTO	Marcar documentos recepcionados
<input type="radio"/>	<input type="radio"/>	Historial académico	
<input type="radio"/>	<input type="radio"/>	Formulario de inscripción 02	
<input type="radio"/>	<input type="radio"/>	Matrícula vigente	
<input type="radio"/>	<input type="radio"/>	Fotocopia CI	
<input type="radio"/>	<input type="radio"/>	Certificado de habilitación	
<input type="radio"/>	<input type="radio"/>	Título de bachiller	<input type="text"/> <input type="text"/> <input type="text"/> Número Institución Año
<input type="radio"/>	<input type="radio"/>	Fotocopia de certificado nacimiento	
<input type="radio"/>	<input type="radio"/>	Fotografía	

Observación sobre documentación

---

**DATOS ACADÉMICOS**

Código Contraseña Reg. Univ. Matrícula NºFolder

- Elija - - Elija - - Elija -  Cargar Foto...  
 Perido Ingreso Especialidad Plan Fotografía

Observaciones

Guardar

Fuente: Elaboración propia

### 3.3.5.2. SEGUNDA ITERACIÓN

El módulo que se prioriza en esta iteración es la de inscripción, retiro y/o adición, inscripción a examen de clasificación y emisión de boleta de inscripción para efectivizar este módulo se requiere el registro de paralelos. Véase inscripción en la figura 3.33.

**Figura 3.33:** Módulo Inscripción

**INSCRIPCION**  
Nuevos y antiguos

**Inscribir** Boleta Inscritos Para clasificacion Ret/Adi Borrar Ins

COD:  II-2011

**Fuente:** Elaboración propia

El funcionario de kardex introduce código de estudiante y luego presiona buscar. En caso, que el estudiante no haya realizado su inscripción despliega la siguiente pantalla.

**Figura 3.34:** Módulo Inscripción – Lista materias

1. Se encontro a:ACARAPI LAURACIA MARCELA, COD:ALM-20027:

SIGLA	MATERIA(S)	PAR
LIN-131	LINGUISTICA GENERAL	---
LIN-121	TALLER DE LENGUAJE I	---
LIN-132	FONETICA Y FONOLOGIA	A B C
HIS-181	HISTORIA Y CULTURA ANDINA	---
LIN-101	AYMARA I	---
LIN-201	IDIOMA I- INGLES I	---
LIN-301	IDIOMA I - QUECHUA I	---
INV-231	TEORIAS Y METODOLOGIAS DE LA INVESTIGACION	---
PED-211	PEDAGOGIA Y LENGUA	---

**Inscribir**

El funcionario de kardex selecciona los paralelos en la cual se inscribirá al estudiante y presiona inscribir.

Si el estudiante ya realizo su inscripción sale el siguiente mensaje de interacción con el usuario.



Otra modalidad de inscripción que se realiza es la inscripción para examen de clasificación explicada anteriormente, la figura 3.35 muestra la pantalla para este fin.

**Figura 3.35:** Módulo inscripción - Inscribir a examen de clasificación

## INSCRIPCION

Nuevos y anitquos

Inscribir	Boleta	Inscritos	<b>Para clasificacion</b>	Ret/Adi	Borrar Ins
COD:	CSG-150591	I-2011	LIN-101 AYMARA I	Buscar	

1. Se encontro a : CARVAJAL SUMI GLADYS MARISOL, COD : CSG-150591

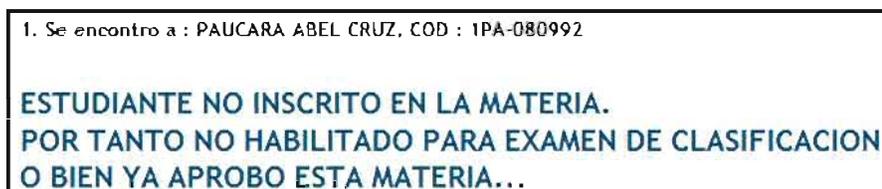
## ESTUDIANTE INSCRITO EN LA MATERIA.

Habilitar para examen de clasificacion

En caso de reprobacion, tendra que cursar la materia...

**Fuente:** Elaboración propia

Cuando se presenta el caso de que el estudiante no cumpla requisitos el sistema muestra el mensaje de interacción



En esta iteración también se ha desarrollado el módulo de retiro y/o adición de materias, véase la figura 3.36. Kardex introduce código de usuario y procede a la búsqueda.

**Figura 3.36:** Módulo inscripción – Retiro y/o Adición

**INSCRIPCION**  
Nuevos y antiguos

Inscribir    Boleta    Inscritos    Para clasificacion    **Ret/Adi**

COD: AQE-210992    II-2011    **Mostrar**

La búsqueda despliega la siguiente pantalla, donde se puede realizar las operaciones de retiro o adición de materias.

**Figura 3.37:** Módulo inscripción – Boleta de inscripción

ESTUDIANTE:	ARUQUIPA QUISPE ERICKA, MARIBEL	COD:	AQE-210992
ESPECIALIDAD:	LINGUISTICA E IDIOMAS	PLAN:	PLAN NIRO 8 (II-2007) V, ESPERANZA

MATERIAS INSCRITAS - GESTION:II-2011

SEM	SIGLA	MATERIA	PAR	RET
1	LIN-131	LINGUISTICA GENERAL	A	<input type="checkbox"/>
1	LIN-121	TALLER DE LENGUAJE I	A	<input type="checkbox"/>
1	LIN-132	FONETICA Y FONOLOGIA	A	<input type="checkbox"/>
1	HIS-181	HISTORIA Y CULTURA ANDINA	A	<input type="checkbox"/>
1	LIN-201	IDIOMA I - INGLES I	A	<input type="checkbox"/>

**Retirar**

MATERIAS A ADICIONAR - GESTION:II-2011

SIGLA	MATERIA(S)	PAR
INV-231	TEORIAS Y METODOLOGIAS DE LA INVESTIGACION	--- <input type="checkbox"/>
LIN-103	AYMARA III	--- <input type="checkbox"/>
LIN-301	IDIOMA I - QUECHUA I	--- <input type="checkbox"/>
PED-211	PEDAGOGIA Y LENGUA	--- <input type="checkbox"/>

**Adicionar**

Fuente: Elaboración Propia

Durante esta iteración también se ha desarrollado el modulo de manejo de notas, la misma es como en la figura 3.38. Despliega opciones como transcribir nota, crear planilla de notas, cargar planilla de notas y similar operación para examen de clasificación.

**Figura 3.38:** Módulo Notas

## NOTAS

Registro y actualización

The screenshot shows a web interface for the 'NOTAS' module. At the top, there are five tabs: 'Transcribir' (highlighted in blue), 'Crear planilla', 'Cargar planilla', 'Cargar Clasificación', and 'Generar reg. notas'. Below the tabs is a search form with the following fields: 'MATERIA' (dropdown menu with 'LIN-131 LINGUISTICA GENERAL' selected), 'SEMESTRE' (dropdown menu with '-Elija-' selected), 'GESTION' (dropdown menu with '-Elija-' selected), 'PAR' (dropdown menu with '---' selected), and 'ACCION' (dropdown menu with 'Buscar' selected). There is also a 'Listar' button. A dropdown menu for 'PAR' is open, showing options 'A', 'B', 'C', and 'D'.

Fuente: Elaboración propia

En este caso utilizemos transcripción y elijamos materia, gestión y paralelo desplegara la pantalla como en la figura 3.39.

**Figura 3.39:** Módulo de notas – Transcribir notas

The screenshot shows the same web interface as Figure 3.38, but with the 'Transcribir' tab selected. The search form now has 'MATERIA' set to 'LIN-121 TALLER DE LENGUAJE I', 'SEMESTRE' set to 'Primero', 'GESTION' set to 'II-2011', and 'PAR' set to 'A'. The 'Buscar' button is highlighted with a dashed border.

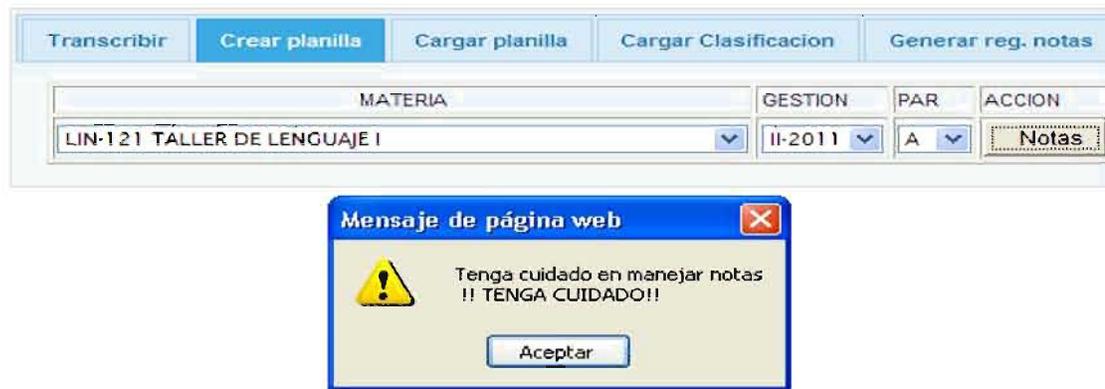
MATERIA: TALLER DE LENGUAJE I  
SEMESTRE: 1  
PARALELO: A  
TURNO: M

CODIGO	CI	PATERNO	MATERNO	NOMBRE	1P 25%	2P 25%	3P 25%	NF 25%	N 1T	N 2T	N.F.
1MN-080991	8283878		MITA	NANCY DELIA	0	0	0	0	0	0	0
1PA-080992	7069716		PAUCARA	ABEL CRUZ	0	0	0	0	0	0	0
AAM-060690	9250957	APAZA	ALCON	MARTHA REVECA	0	0	0	0	0	0	0

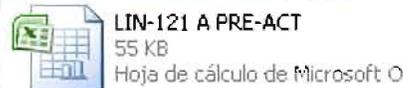
Fuente: Elaboración propia

La segunda opción del manejo de notas es la generación de planilla de notas semestrales. Véase figura 3.40, donde se inicia con la creación de planilla de notas.

Figura 3.40: Módulo notas – Crear planilla de notas



La planilla de notas que se logra generar con la opción anterior esta en formato Excel para el llenado de notas por parte de los docentes, la misma es distribuida al finalizar cada gestión.



Donde LIN-121 es el nombre de la materia, A es el paralelo y PRE-ACT es el nombre con el que se agrega como pre-actas, al abrir el archivo se abre como la figura 3.41.

Figura 3.41: Módulo notas – Vista de planilla



*Universidad Pública de El Alto*  
Creada por ley: 2115 del 5 de septiembre de 2000 y autónoma por ley 2556 del 12 de noviembre

**PRE-ACTA DE CALIFICACIONES**

Área Académica: Ciencias Sociales

Carrera	LINGÜÍSTICA E IDIOMAS	Paralelo	1-A
Materia	TALLER DE LENGUAJE I	Sigla	LIN-121
Docente:		Gestión	II-2011

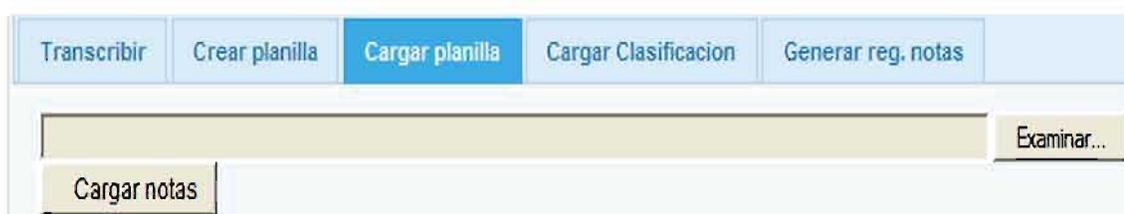
  

N°	NOMINA DE ESTUDIANTES	N° Registro Universitario	1° P.	2° P.	3° P.	E.F.	1° T	2° T	N.F.	NOTA LITERAL	OBS.
			25%	25%	25%	25%	100%				
1	HUCHANI EDWIN	6660									NSP
2	MITA NANCY DELIA	10039611									NSP
3	PAUCARA ABEL CRUZ	11005530									NSP
4	APAZA ALCON MARTHA REVECA	11005631									NSP
5	APAZA MAMANI JULIA	11005524									NSP
6	ARUQUIPA QUIISPE ERICKA MARIBEL	11005608									NSP
7	AVILA CHOQUEHUANCA GRACIELA	10039658									NSP
8	CACERES CAJAS JUAN	0									NSP
9	CALLE QUIISPE GREGORIO GERMAN	8002983									NSP
10											NSP

Fuente: Elaboración Propia

Una vez llenado la planilla con notas, se procede a cargar las mismas al sistema acorde con su materia, a través de la opción cargar planilla. Ver figura 3.42.

**Figura 3.42:** Módulo notas - Cargar planilla de notas



**Fuente:** Elaboración propia

El proceso es similar para el caso de notas de examen de clasificación.

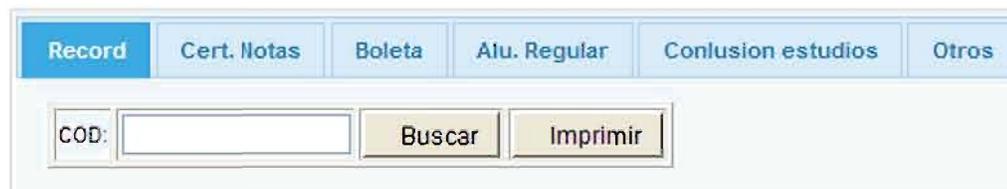
### 3.3.5.3. TERCERA ITERACIÓN

En esta etapa se prioriza el módulo de emisión de documentos básicos como salida del sistema. La figura 3.43 ilustra parte de la pantalla con la cual el usuario interactúa.

**Figura 3.43:** Pantalla – Emisión de documentos

#### EMISIÓN DE DOCUMENTOS

Salidas del sistema



**Fuente:** Elaboración propia

Al introducir el código de estudiante y haciendo Click en Buscar, se muestra el documento como en la figura 3.44 cuando la búsqueda es exitosa, en otro caso muestra no existe record, sino despliega una de las razón por las cuales no fue posible elaborar la documentación solicitada.

**Figura 3.44:** Módulo documentos – Record académico

UNIVERSIDAD PUBLICA DE EL ALTO  
CARRERA DE LINGUISTICA E IDIOMAS

RECORD ACADEMICO (Segun Pensum)

<b>ESTUDIANTE:</b> MACHACA LAYME EMMA	<b>CI:</b> MLE-290385
<b>ESPECIALIDAD:</b> LINGUISTICA E IDIOMAS	<b>PLAN:</b> PLAN NRO 8 (II-2007) V.ESPERANZA

SIGLA	MATERIA	NOTA	GEST
LIN-101	AYMARA I	80	II-2011
LIN-102	AYMARA II	80	II-2011

**RESUMEN**

PROMEDIO:	80
TOTAL APROBADOS:	2

**Fuente:** Elaboración propia

Por otro lado también se genera certificado de notas, la misma podemos verlo en la figura 3.45.

**Figura 3.45:** Pantalla – Certificado de notas

				420
MAMANI	BAUSTISTA	ROXANA		
9236478	11005621	LINGUISTICA E IDIOMAS		
	II-2011	2011		
LIN-101 AYMARA I	5	76	SETENTA Y SEIS	4 APROBADO
LIN-102 AYMARA II	5	76	SETENTA Y SEIS	4 APROBADO

**Fuente:** Elaboración propia

Las demás pantallas se muestran en anexo K.

Con la última iteración el sistema prácticamente ya se está concluyendo, los módulos de las iteración ya están en producción.

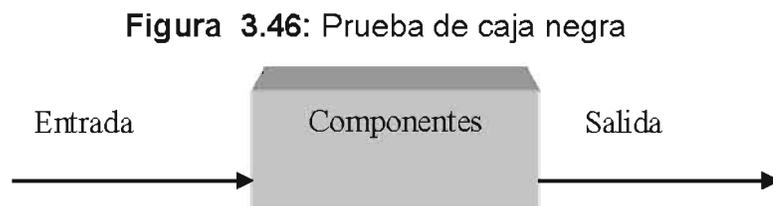
### 3.4. FASE DE TRANSICIÓN

En esta fase se valida y despliega el sistema en el entorno de producción. Para el cual se aplica pruebas extensivas a lo largo de esta fase, una buena afinación del proyecto tiene lugar aquí incluyendo las modificaciones a los defectos significativos. Destacamos las pruebas de caja blanca y caja negra con el objetivo descubrir defectos del software. "La prueba no puede asegurar la ausencia de defectos, solo puede demostrar que existen defectos en el software" [PRESSMAN, 2005].

#### 3.4.1. PRUEBA DEL SISTEMA

El motivo de la selección de este método es la utilidad que tiene para probar software. La prueba de caja negra intenta descubrir errores en las siguientes categorías (1) funciones incorrectas o ausentes, (2) errores de interfaz, (3) errores en la estructura de datos o acceso a base de datos externas, (4) errores de rendimiento y (5) errores en la inicialización y terminación [PRESSMAN, 2005].

El mismo consiste en catalogar las entradas y salidas primero en forma global y después en cada uno de sus componentes en forma independiente y la manera en que estas son transformadas para poder comprender el objetivo de cada componente. Figura 3.46



**Fuente:** Elaboración Propia

- ✓ Se realizo pruebas unitarias, es decir cada modulo en el momento de la construcción previo a puesta en producción.
- ✓ Pruebas de integración, esta prueba se realiza cuando todos los módulos están diseñados y desarrollados para luego integrarlos y posteriormente realizar prueba de todo el sistema.
- ✓ Pruebas del software, una vez integrado todos los módulos, procedemos a la prueba del software con datos reales.
- ✓ Pruebas de implantación, esta prueba se realizo en forma incremental es decir que, ya parte del sistema esta en plena producción acorde a la metodología AUP.
- ✓ Pruebas de aceptación, pasado el periodo de pruebas se espera que el sistema sea aceptado por los usuarios finales.

### **3.4.2. POLÍTICAS DE SEGURIDAD**

La seguridad es responsabilidad de todos aquellos que están en contacto con el sistema. La misma lo enfocamos desde tres aspectos interrelacionados: físico, lógico y conductual.

#### **SEGURIDAD FÍSICA**

- ✓ A la oficina de sistemas se debería restringir el acceso físico de otras personas, que no sean propios de esta oficina, para mayor seguridad.
- ✓ El servidor debe ser manejado por el personal autorizado para la administración de base de datos. En caso de mantenimiento técnico este deberá consultar al administrador del sistema.

- ✓ El personal de limpieza debe estar consiente de las restricciones de seguridad.

## **SEGURIDAD LÓGICA**

Se refiere, al acceso de los usuarios a los recursos y a la información que se dispone en un ordenador, y al uso correcto de los mismos. Cuando se usan, permiten al usuario entrar al sistema o a una parte particular de una base de datos con una contraseña correcta. Entra las cuales destacamos:

- ✓ Seguridad del motor Sql Manger for MySql, contrseña de accesos a base de datos.
- ✓ Seguridad del sistema, mediante el manejo de usuarios de la aplicación y contraseñas.
- ✓ Sistema de encriptación de contraseñas para que el usuario confié en el sistema.
- ✓ Se puede realizar Back Up de la base de datos, solo por usuarios con este rol de manejo de sistemas.
- ✓ Se puede realizar restauración del sistema en caso de eventualidades, es decir recuperar una copia anterior.

## **SEGURIDAD CONDUCTUAL**

- ✓ Crear una conciencia en el manejo de información de usuarios a través capacitaciones planificadas, para que los mismos reconozcan el valor que tiene la información.
- ✓ Identificación de empleados que eventualmente tendrán acceso a las computadoras, datos e información, para asegurar que sus intereses son

consistentes con los intereses de la organización y que entienden por completo la importancia de llevar a cabo los procedimientos de seguridad.

- ✓ Se creara un manual de funciones que describa, las políticas con respecto a la seguridad para que los empleados estén totalmente conscientes de las expectativas y responsabilidades.

### **3.4.3. RESTRICCIONES DEL SISTEMA**

Esta parte es muy importante para que no pueda haber otras expectativas, las mismas se mencionan a continuación:

- ✓ Es Sistema funciona dentro del entorno Windows posteriores a Windows SP2, y es posible instalar en Linux utilizando LAMP.
- ✓ El sistema esta funcionando en un entorno de red Intranet, pero se tiene previsto para que pueda ser utilizada desde internet, una vez adquirida su dominio, acorde a las políticas que la institución adopte.

### **3.4.4. CAPACITACION**

La capacitación del personal que maneje el sistema de información, se realizara de los siguientes puntos y tematicas.

#### **Encargados de sistemas:**

- ✓ Instalación del sistema
- ✓ Back Up y Restauración de la Base de datos sistema.
- ✓ Alta de usuarios, gestiones, paralelos, aulas, horarios y pensum.
- ✓ Ubicación física de equipos.
- ✓ Política de manejo de usuarios.

**Kardex y/o auxiliares de kardex y funcionarios eventuales de transcripción:**

- **Ingreso al sistema**, la importancia del manejo de los procesos.
- **Registro de estudiantes**, el cual consta de captura de datos y las actividades que debe realizar por ejemplo verificar documentos.
- **Inscripción**, materias, paralelos, errores, impresión de boleta y otros casos que podrían presentarse.
- **Inscripción a examen de clasificación**, quienes pueden inscribirse con esta modalidad, procedimiento y cuidados.
- **Retiro y/o Adición**, solo para estudiantes inscritos.
- **Manejo de notas**, creación de planillas, cargado de notas tanto semestrales como de examen de clasificación.
- **Emisión de documentos**, quienes pueden emitir y que documentos están disponibles para los usuarios.
- **Consultas del sistema**, en que casos acudir a una consulta y que tipos de consultas se tiene disponible.

## **CAPÍTULO IV**

### **EVALUACIÓN DEL SISTEMA**

En este capítulo se evalúa al sistema de información académica para la carrera de lingüística e idiomas, como cualquier producto es sometido a criterios de calidad, que nos permitan medir, analizar y comprender el grado de cumplimiento de una serie de características de un sistema de información, es decir "...debe hacer lo que dice que hace" [PIATTINI,2003].

Los factores de calidad no necesariamente se prestan a la medición directa, sin embargo, proporcionan una base valiosa para las medidas indirectas y una lista de verificación excelente para evaluar la calidad de un sistema.

La estándar ISO-9126 que identifica seis factores clave de calidad de software, los mismos coinciden con los definidos para la evaluación de calidad de aplicaciones web, como son factores de: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad.

#### **4.1. FACTORES DE CALIDAD**

##### **4.1.1. FUNCIONALIDAD**

Este atributo valora el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global. La funcionalidad es el grado en que el software satisface las necesidades que indican los siguientes sub-atributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

## **PUNTO FUNCIÓN**

Para el cálculo de punto función se toma en cuenta cinco características de dominio de información como son número de: entradas, salidas, peticiones, archivos y interfaz externa. Luego se realiza el cálculo de punto función hallando la suma de estas características, parámetros de medición y el factor ponderación que es el punto medio de ponderación.

### **Número de entrada de usuario**

Se cuenta cada entrada de usuario que proporciona datos al sistema

### **Número de salidas de usuario**

Se refiere a cada salida que proporciona el sistema al usuario, entre estos pueden estar: informes, pantallas, mensajes de error, etc.

### **Número de peticiones de usuario**

Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salida interactiva.

### **Número de archivos**

Se cuenta cada archivo maestro lógico, estos puede ser: grupo lógico de datos, o un archivo independiente.

### **Número de interfaces externas**

Se cuenta todos las interfaces legibles por la maquina que se utilizan para transmitir información a otro sistema.

**Tabla 4.1:** Cálculo de puntos de función no ajustados

Parámetro de medición	Cuenta	Factor de ponderación			Total
		Simple	Medio	Complejo	
Número de entradas de usuario	16	3	4	6	48
Número de salidas de usuario	9	4	5	7	36
Número de peticiones de usuario	16	3	4	6	48
Número de archivos	22	7	10	15	154
Número de interfaces externas	3	5	7	10	15
<b>Cuenta_Total</b>					<b>301</b>

Fuente: [PRESSMAN,2005]

Calculo de puntos funciones ajustadas

**Tabla 4.2:** Valores de ajuste de complejidad

Nº	FACTOR	VALORACION DE FACTOR						
		No influye	Incidental	Moderado	Medio	Significativo	Esencial	VALOR OBTENIDO
		0	1	2	3	4	5	
1	¿Requiere el sistema de copias de seguridad?					x	5	
2	¿Requiere comunicación de datos?					x	5	
3	¿Existe funciones de procesamiento distribuido?				x		3	
4	¿Es crítico el rendimiento?				x		3	
5	¿Se ejecutan en un entorno operativo y utilizado?					x	4	
6	¿Requiere el sistema entrada de datos en línea?					x	4	
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?					x	4	
8	¿Se actualiza los archivos maestros en forma interactiva?					x	4	
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?			x			2	
10	¿Es complejo el procesamiento interno?					x	4	
11	¿Se ha diseñado el código para ser reutilizable?			x			2	
12	¿Están incluidas la conversión y la instalación?			x			2	
13	¿Se ha diseñado el sistema para múltiples instalaciones en diferentes organizaciones?				x		3	
14	¿Es flexible en cuanto a cambios?				x		3	
<b>CUENTA TOTAL (ΣFi)</b>							<b>48</b>	

Fuente: [PRESSMAN,2005]

Para el ajuste se utiliza la ecuación

$$PF = \text{cuenta-total} \times [0.65 + 0.01 \times (\sum Fi)] \quad (1)$$

Remplazando valores obtenidos, se tiene:

$$PF_{\text{obtenido}} = (301) \times [0.65 + 0.01 \times (48)]$$

$$PF_{\text{obtenido}} = 340,13. \quad (2)$$

Ahora suponemos para un nivel de 100% de completitud de los factores ajustables que sería 70.

$$PF_{\text{ideal}} = (301) \times [0.65 + 0.01 \times (70)]$$

$$PF_{\text{ideal}} = 406,35. \quad (3)$$

Luego calculamos el % de funcionalidad real ( $PF_{\text{real}}$ ) dividiendo (2) / (3), se tiene:

$$PF_{\text{real}} = PF_{\text{obtenido}} / PF_{\text{ideal}}$$

$$PF_{\text{real}} = 340,13 / 406,35$$

$$PF_{\text{real}} = 0,837$$

Interpretando, el sistema tiene un 83.7% de funcionalidad o utilidad para la institución.

#### 4.1.2. CONFIABILIDAD

Recordemos la confiabilidad representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la probabilidad de que un sistema presente fallos según la función:

$$F(t) = (PF_{\text{real}}) * (e^{-\lambda t}) \quad (4)$$

Sean los datos:

$$PF_{\text{real}} = 83.7\%,$$

$$\lambda = 0,1 \text{ (es decir 1 error en cada 10 ejecuciones)}$$

$$t = 12 \text{ meses}$$

$$F(12) = (0,837) * (e^{-(0.1*12)})$$

Remplazando datos en (4), se tiene:

$$F(12) = 0.25$$

Por tanto la probabilidad de que el sistema no presente fallos en una gestión es hallando el complemento, se tiene:

$$G(T)=1-F(T)$$

$$G(12)=1-0.25$$

$$G(T) = 0.75.$$

Por tanto: la confiabilidad del sistema es 75%, es decir en un 75% de ejecuciones del sistema, este no pueda presentar errores.

#### 4.1.3. USABILIDAD

Recordemos que la usabilidad es el esfuerzo necesario para aprender a operar con el sistema, preparando los datos de entrada e interpretar los datos de salida. Para valorar la usabilidad se ha empleado el instrumento como en la tabla 22.2 tomando en cuenta preguntas cerradas a objeto de tabular rápidamente los datos, participaron de la aplicación del instrumento: director de carrera, la kardixa, auxiliar de Kardex, persona contratada para la tabulación de datos históricos, y un representante del centro de estudiantes de la carrera de lingüística.

**Tabla 4.3:** Test de evaluación de usabilidad

N°	PREGUNTA	RESPUESTAS		Interpretación
		SI	NO	
1	¿El sistema es comprensible?	5	1	0.833
2	¿Es atractivo a la vista?	4	2	0.663
3	¿El sistema hace lo que dice que hace?	6	0	1.000
4	¿Las respuestas del sistema son satisfactorias?	5	2	0.833
5	¿Parece complicado las tareas que realiza?	4	2	0.663
6	¿Puede utilizarlo?	5	1	0.833

**Fuente:** Elaboración propia

De acuerdo a los datos obtenidos en la tabla 22.2, se concluye que el sistema tiene una usabilidad del 0.804, es decir del 80.4 % de usabilidad.

#### 4.1.4. EFICIENCIA

Es el grado en el que el software emplea de manera óptima los recursos del sistema, como lo indica los siguientes sub-atributos: comportamiento en el tiempo y comportamiento de los recursos. Es decir no presenta mayor lentitud y los recursos de hardware pueden ser compartidos con otras aplicaciones.

#### 4.1.5. FACILIDAD DE MANTENIMIENTO

Es el esfuerzo necesario para localizar y arreglar un error en un programa, es decir, la facilidad con la que una modificación puede ser realizada. Para este fin Pressman nos sugiere el Índice de Madurez del Software (IMS) para determinar la estabilidad de un producto software. Dicha IMS es calculada según la ecuación (5).

$$IMS = [M_T - (F_a + F_c + F_d)] / M_T \quad (5)$$

Donde:

$M_T$  = Número de módulos de la versión actual

$F_c$  = Número de módulos en la versión actual que se han cambiado

$F_a$  = Número de módulos en la versión actual que se han añadido

$F_d$  = Número de módulos de la versión anterior que se han borrado en la versión actual.

Los parámetros que se obtiene con el sistema es:

$$M_T = 8,$$

$$F_c = 0,$$

$$F_a = 1,$$

$$F_d = 0.$$

Remplazando en (6)

$$IMS = [8 - (0 + 1 + 0)] / 8 = 0,875$$

Interpretando se tiene una madurez de 87% lo cual indica que el sistema empieza a estabilizarse.

#### 4.1.6. PORTABILIDAD

Es el esfuerzo necesario para transferir el programa de un entorno (hardware / software) a otro entorno diferente, es decir, la facilidad con que se lleva de un entorno a otro.

#### SERVIDOR

Es portable en entornos Windows XP, server 2000 o posteriores que soporten:

- ✓ Apache 2.2 en adelante
- ✓ MySql 3.3 en adelante
- ✓ Php 5.0 En adelante

Por otro lado también es posible instalar en un entorno Linux utilizando LAMPP que es similar a XAMPP pero para Linux.

En cuanto a hardware

- ✓ Pentium IV mínimo
- ✓ RAM de 1GB
- ✓ Procesador de 2GB
- ✓ Disco duro de 40 GB

#### CLIENTE

Por otro lado, para la ejecución en los equipos cliente se requiere de un navegador de internet; superior a internet Explorer 6.0, Mozilla Firefox (Probados), no se probó en otros casos, pero suponemos que funciona ya que se desarrollo en un lenguaje de programación en un lenguaje que no requiere ser instalado en un equipo cliente.

Para la parte de hardware es deseable que tengan las siguientes características como mínimo:

- ✓ Pentium IV
- ✓ RAM de 256 MB

- ✓ Procesador de 1,7 GB
- ✓ Disco duro de 10 GB

## 4.2. COSTO / BENEFICIO

El análisis de costo y beneficio del sistema que el cliente valora cuando los beneficios exceden a los costos de la implementación del sistema. Por lo tanto analicemos costo y beneficios.

### 4.2.1. ANALISIS DE COSTOS

En la cuantificación del costo del sistema se tomara en cuenta los siguientes aspectos:

- ✓ Costo de desarrollo del software
- ✓ Costo de implementación del sistema
- ✓ Costo de elaboración del proyecto

#### 4.2.1.1. COSTO DE DESARROLLO DEL SOFTWARE

Para hallar el costo de construcción software utilizaremos el valor obtenido como  $P_{fobtenido}$  con la formula de la ecuación (1), que es de 340,13. Y además utilizamos el factor de conversión a KLDC de la tabla 4.3.

**Tabla 4.4:** Conversión de puntos de función a KLDC

LENGUAJE	NIVEL	FACTOR LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL / I	4	80
Ansi / Cobol 74	3	107
Visual Basic	7.00	46
Asp	9.00	36
PHP	11.00	29
Visual C++	9.50	34

Fuente: [CHAMBI,2007]

$$LDC = PF_{\text{Obtenido}} * \text{Factor LDC} / PF \quad (6)$$

$$LDC = 340.13 * 29$$

$$LDC = 9863,77. \rightarrow KLDC = (9863,77 / 1000)$$

$$\text{Entonces } KLDC = 9.86$$

Ahora bien, aplicando las formulas de esfuerzo, tiempo calendario y personal requerido.

$$E = a_b (KLDC)^{b_b} \quad (7)$$

$$D = c_b (KLDC)^{d_b} \quad (8)$$

Donde:

E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en meses cronológico

KLDC: Número de líneas de código distribuido en miles

**Tabla 4.5:** Constantes de COCOMO

Proyecto de software	$a_b$	$c_b$	$b_b$	$d_b$
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: [PRESSMAN,2005]

En este caso el sistema desarrollado estaría entre los semi-acoplados, porque es de tamaño y complejidad intermedia, con variados niveles de experiencia en el que el equipo de satisfacer requisitos poco o medio rígidos.

$$E = 3.0 * (9.86)^{1.12}$$

$$E = 38.9$$

$$D = 2.5 * (38.9)^{0.35}$$

$$D = 9.0$$

Por tanto el personal requerido es:

$$\text{Número de programadores} = E/D = 38.9/9$$

$$\text{Número de programadores} = 4.3$$

Conclusión se requieren 5 programadores.

Si el salario que percibe un programador es de 1000, entonces se tiene:

Costo de desarrollo = numero de programadores \* por salario de programador.

Costo de desarrollo = 5\*1000= 5000 Bs / mes (Los 5 programadores ganan en un mes)

Ahora por los 9 meses se tiene:

Costo total del proyecto=5000\*9

Por tanto el costo del sistema desarrollado es 45000 Bs, o su equivalente en dólares 5625 \$us.

#### 4.2.1.2. COSTO DE IMPLEMENTACIÓN DEL SISTEMA

Tabla 4.6: Costo de licencias

SISTEMA OPERATIVO	LICENCIA(\$us)
Windows 2008 server	100
Windows XP SP3	100
Total	200

Por otro lado se requiere un costo estimado del hardware a comprar, aunque la institución ya cuenta con estos equipos.

Tabla 4.7: Costo de equipos

EQUIPO	CANTIDAD	COSTO UNITARIO (\$US)	COSTO TOTAL (\$US)
Servidor Pentium 2i, 2 GHz, RAM 2GB , HD 500GB	1	1000	1000
Terminal Core 2Ghz, 1GB de RAM, HD 80GB	3	600	1800
Switch de 8 puertos	1	40	40
UPS	1	200	200
Total			3040

Fuente: Cotización de proformas

#### 4.2.1.3. COSTO DE ELABORACIÓN DEL PROYECTO

Durante la construcción del sistema se computan los siguientes aspectos que implican costo previo a la programación del sistema. Las mismas son como en la tabla 4.8.

**Tabla 4.8:** Costo de elaborar proyecto

ACTIVIDAD	COSTO(Bs)
Análisis y diseño	1600
Bibliografía	500
Material de escritorio	200
Internet	400
Otros	180
TOTAL	2880

**Fuente:** acorde a [CHAMBI,2007]

Finalmente el costo del sistema será:

Costo de programación	5625 \$us
Costo de implementación	(200 + 3040) \$us
Costo de elaborar proyecto	360 \$us.
<b>Total</b>	<b>9225 \$us.</b>

Nótese que el sistema se implementa con equipos ya existentes, esto no implica que no haya que adquirirlos para este propósito.

#### 4.2.2. ANALISIS DE BENEFICIOS

Los beneficios del sistema son más de carácter intangible. Para el cual se utilizan los indicadores de la tabla 4.9, que nos permite comparar indicadores para con el sistema anterior con el que se implementa, entre los aspectos que se destacan esta la velocidad de procesamiento, control de procesos, integridad de información y los datos de salida para la toma de decisiones.

**Tabla 4.9:** Análisis de beneficios

INDICADOR	BENEFICIO	SISTEMA ANTERIOR	SISTEMA NUEVO
Incremento en velocidad	<ul style="list-style-type: none"> <li>✓ Búsqueda de estudiantes</li> <li>✓ Consulta de estudiantes</li> <li>✓ Consulta docentes</li> <li>✓ Consulta para toma de decisión</li> </ul>	Manual 10-35 minutos 5-10 minutos Información incompleta	Automático 1-2 minutos 0.5 – 1 min Información confiable.
Volumen de información	<ul style="list-style-type: none"> <li>✓ 7 consultas de docentes al día (Promedio)</li> <li>✓ 30 consultas estudiantes de al día (Promedio)</li> </ul>	6 hojas solicitudes y consultas. 50 hojas entre solicitudes, registro y consulta	La información es almacenada en medio magnética.
Control a procesos	<ul style="list-style-type: none"> <li>✓ Registro de información</li> <li>✓ Cuadros estadísticos</li> </ul>	En 20% existen errores. Los reportes tienen un margen de error de 15%	Se reduce el % de error por su entrada de datos. Los cálculos estadísticos no presentan errores.
Información integra	Coordinación en los procesos de manejo académico	Manejan distintas personas, que podrían cometer algún error.	La información es centralizada y accedida por usuario acorde a un rol.
Información para toma de decisiones	<ul style="list-style-type: none"> <li>✓ Para fuentes externas</li> <li>✓ Para directivos</li> </ul>	El tratamiento de la información tarda mucho, lo cual posterga la toma de decisión oportuna.	Facilita información básica para la toma de decisión oportuna.

**Fuente:** resultados del sistema

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

Este capítulo describe los resultados obtenidos de los objetivos planteados para el desarrollo del proyecto, las recomendaciones y los futuros trabajos.

#### 5.1. CONCLUSIONES

Se ha alcanzado los siguientes objetivos:

- ✓ Se desarrolló el módulo de registro de estudiantes nuevos, la misma permite a la institución capturar información sin redundancia y con menos errores.
- ✓ Se desarrolló un módulo de inscripción de estudiantes nuevos y antiguos, que permite verificar prerequisites y las materias a cursar.
- ✓ Se implementó el módulo de retiro y/o adición de materias según pensusum de la carrera.
- ✓ Se desarrolló el módulo de manejo de notas semestrales y notas de examen de clasificación generando planillas de notas y cargado de los mismos.
- ✓ Se desarrollo el módulo de consultas más frecuentes.
- ✓ Se implementa el módulo para generar reportes y estadísticas para la toma de decisiones por parte de las autoridades.

- ✓ Los criterios de calidad que se utilizaron para la valoración del sistema son: Funcionalidad, Portabilidad, Usabilidad, Mantenimiento y fiabilidad.

Por lo tanto podemos decir que: se ha desarrollado e implementado el sistema de información académica para la carrera de lingüística e idiomas de la Universidad Pública de El Alto, que apoya los procesos de centralización y organización de la información, para brindar resultados oportunos y eficientes.

## 5.2. RECOMENDACIONES

Para la complementación del sistema y analizando los beneficios obtenidos, recomendamos los siguientes aspectos:

- ✓ Desarrollar un portal Web que permita interactuar con la comunidad universitaria, a través de foros y publicaciones.
- ✓ Implementar un sistema de información que administre la biblioteca, coadyuvando en la búsqueda de material bibliográfico de manera eficiente.
- ✓ Integrar el curso pre-universitario al sistema de información académica para automatizar sus procesos.
- ✓ Realizar evaluaciones periódicas de la información producida por el sistema, con el fin de determinar las nuevas necesidades, es decir mantenimiento.

## BIBLIOGRAFÍA

- [ALARCON,2000] ALARCON R. 2000, DISEÑO ORIENTADO A OBJETOS CON UML, Madrid, España.
- [DATE,2001] DATE C.J., 2001, Introducción a los SISTEMAS DE BASE DE DATOS, Séptima edición, Prentice Hall, PEARSON EDUCACIÓN, México.
- [ESTATUTO,2009] Estatuto Orgánico, Reglamentos Generales, Resoluciones del II Congreso Ordinario, 2009, Universidad Pública de El Alto.
- [HERNÁNDEZ,2006] Hernández Sampieri R., Fernández C. Batista P., Baptista P., METODOLOGÍA DE LA INVESTIGACIÓN, 2006, Cuarta edición, McGraw-Hill, México D.F.
- [KENDALL,2005] Kendall K. Kendall J., 2005, ANALISIS Y DISEÑO DE SISTEMAS, Sexta edición, PEARSON EDUCATION, México.
- [KRUCHTEN,2000] The Rational Unified Process: An Introduction, 2000 Addison Wesley.
- [LARMAN,1999] LARMAN C. 1999, UML Y PATRONES, Primera Edición, Prentice Hall Hispanoamérica, México.
- [LEFFINGWELL,2011] Linffingwell D., 2011, Agile Software Requirements, First Printing, Addison-Wesley, PEARSON EDUCATION, Massachusetts- United States.
- [MOF,2011] UNIVERSIDAD PÚBLICA DE EL ALTO, 2011, MANUAL DE ORGANIZACIÓN Y FUNCIONES DE LA CARRERA DE LINGÜÍSTICA E IDIOMAS, El Alto, La Paz-Bolivia.
- [PLAN,2010] PLAN CURRICULAR ACTUALIZADO, 2010, Carrera de Lingüística e Idiomas, Universidad Pública de El Alto.
- [PIATTINI,2003] Piattini M., García F. 2003, Calidad en el desarrollo y mantenimiento del software, MEXICO D.F: ALFAOMEGA GRUPO EDITORES, S.A. de C.V.
- [PRESSMAN,2005] Pressman R. 2005, INGINIERÍA DE SOFTWARE Un enfoque práctico, Sexta edición, McGraw-Hill/Interamericana de España, México D.F.

[RJB,2000] RUMBAUGH J., JACOBSON I., BOOCH G., 2000, EL LENGUAJE UNIFICADO MANUAL DE REFERENCIA, Primera Edición, ADDISON WESLEY, Madrid, España.

[SCHMULLER,2001] Schmuller J. 2001, UML en 24 horas, Primera Edición, Prentice Hall, México.

## TESIS DE GRADO

[CALCINA,2007] Calcina V.,2007, *Sistema de seguimiento académico carrera de arquitectura universidad pública de el alto*, Universidad Mayor de San Andrés, Carrera de Informática, La Paz-Bolivia.

[CHAMBI,2007] Chambi D., 2007, *Sistema de Gestión Académica para el Instituto Normal Superior Simón Bolívar*, Universidad Mayor de San Andrés, Carrera de Informática, La Paz-Bolivia.

[CARPIO,2008] Carpio F., 2008, *Sistema de Administración Académica vía Web Caso: Escuela de Idiomas*, Universidad Mayor de San Andrés, Carrera de Informática, La Paz-Bolivia.

[AMARO,2007] Amaro S., Valverde R., 2007, *Metodologías Agiles*, Universidad Nacional de Trujillo - Facultad de Ciencias Físicas y Matemáticas, Trujillo Perú.

[ALVARADO,2009] Alvarado R, 2007, "*Sistema Integral de Gestión de Biblioteca*", UNIVERSIDAD TECNOLÓGICA SAN ANTONIO DE MACHALA, Escuela de Informática, Machala-EI-Oro Ecuador.

[DURAN,2007] Duran B, 2007, "*Asesoramiento técnico on-line para la adquisición y funcionamiento de un equipo de computación*", Universidad Mayor de San Andrés, Carrera de Informática, La Paz-Bolivia.

[CEREZO,2007] Cerezo R, 2008, "*Biblioteca digital de los programas de ayuda humanitaria - save the children*", Universidad Mayor de San Andrés, Carrera de Informática, La Paz-Bolivia.

[CALDERON,2009] Calderón G, 2009, "*Análisis, diseño e implementación de un comparador y sincronizador de bases de datos relacionales de distintos manejadores*", Universidad Católica, Facultad de Ciencias e Ingeniería, Perú.

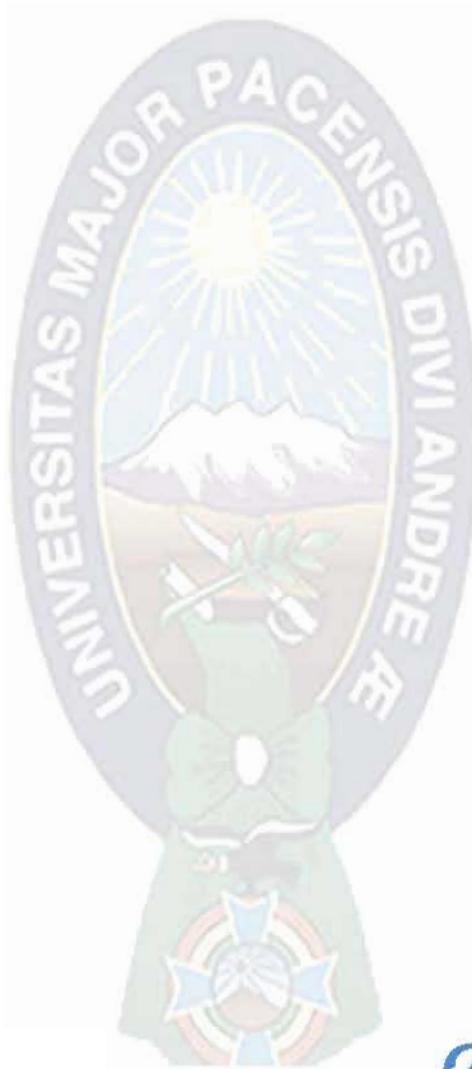
- [KOCH,2001] Koch N., Kraus A., Hennicker R., 2001, The Authoring Process of the UML-based Web Engineering Approach, Institute of Computer Science, Ludwig-Maximilians University of Munich, München, Germany.
- [OLSINA,1999] Olsina L., 1999, Ingeniería de Software en la Web, “*Metodología Cuantitativa para Evaluación y Comparación de la Calidad de Sitios Web*”, Tesis Doctoral, Universidad Nacional la Plata-Argentina.

## REFERENCIAS EN INTERNET

- [AMBLER,2005] Scott W. Ambler, 2005, *Proceso Unificado Ágil v1.1*, UNIVERSIDAD NACIONAL DE COSTA RICA, <http://cgi.una.ac.cr/AUP/index.html>, traducido por: Diego Alpízar Naranjo e Iván Argüello Oviedo 2006  
Fecha de acceso: 20-08-2011
- [CURSO,2009] *CURSO DE DESARROLLO ÁGIL*, Instituto Nacional de tecnología de la comunicación, junio 2009.  
<http://www.inteco.es/accesibilidad/formacion/manualesyguia/>  
Fecha de acceso: 22-08-2011
- [FUENTES,2009] MANUAL DE AJAX, las entrañas de AJAX, Segunda edición, 2009.
- [JQUERY,2010] Alvarez M., 2010, *Manual de JQuery*, <http://www.desarrolloweb.com/manuales/manual-jquery.html>  
Fecha de acceso: 12-08-2011.
- [PHPEXCEL,2011] Balleau M., *PHPEXcel Developer Documentation*, Versión 1.7.6. <http://www.doplex.com/PhpExcel/>  
Fecha de acceso: 10-08-2011.
- [UWE,2010] *Estudio UWE(UML-based Web Engineering)*, UNIVERSIDAD CARLOS III, Madrid, España.  
<http://uscii.es/publicaciones/uwe.pdf>  
Fecha de acceso: 15-08-2011.
- [UWE1,2008] Kroib C., Koch N., *UWE Metamodel and Profile User Guide and Reference*, 2008, Institute for Informatics, Ludwig-Maximilians-Universität München, Germany  
<http://www.pst.ifi.lmu.de/projekte/uwe>
- [WIKI,2011] *Agile Unified Process de Scoth Ambler*  
<http://es.wikipedia.org/wiki/aup>  
Fecha de acceso: 15-08-2011.



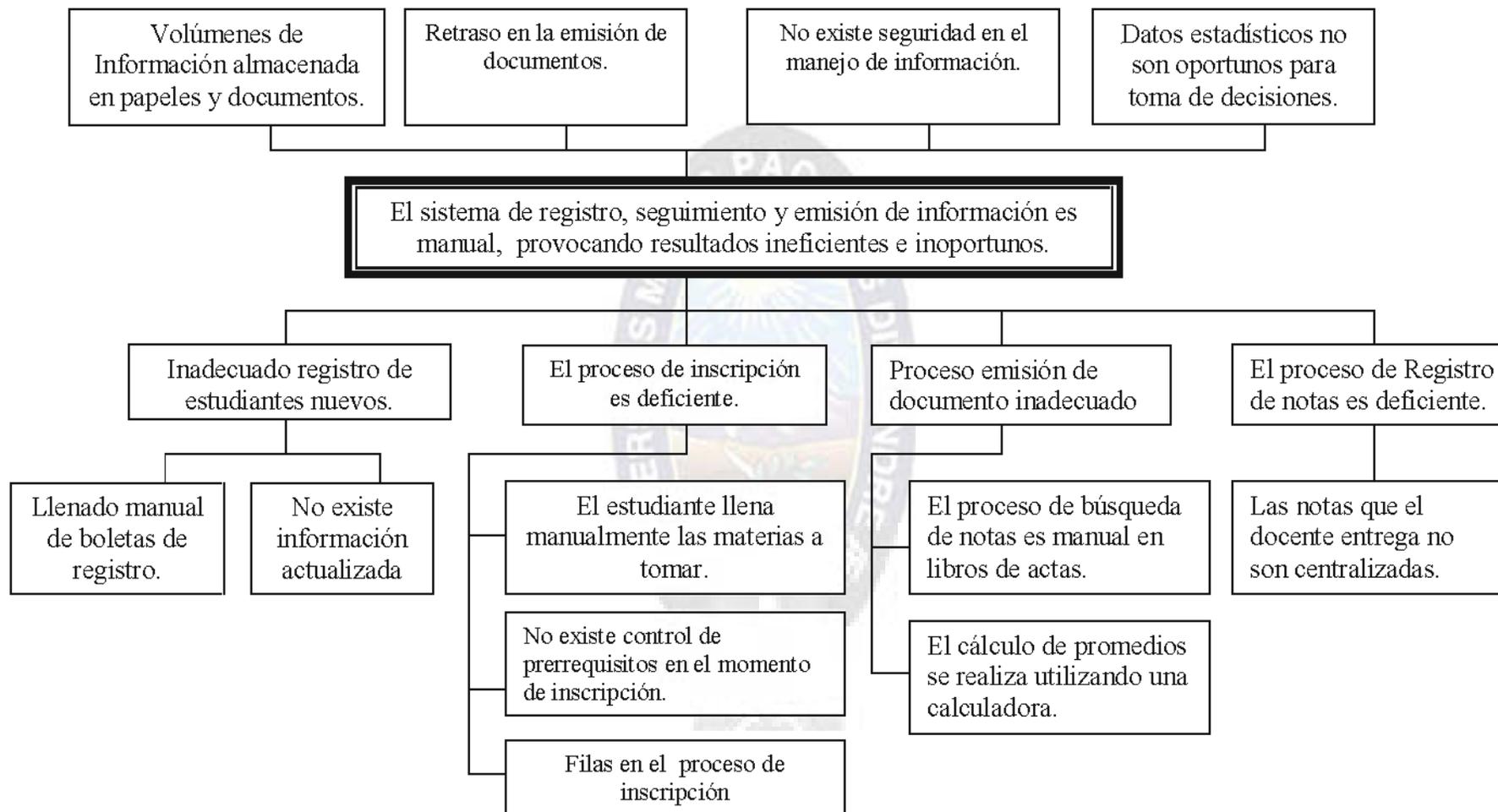
# *Anexos*



## *Anexo A*

Árbol de problemas

Figura A.1: Árbol de problemas

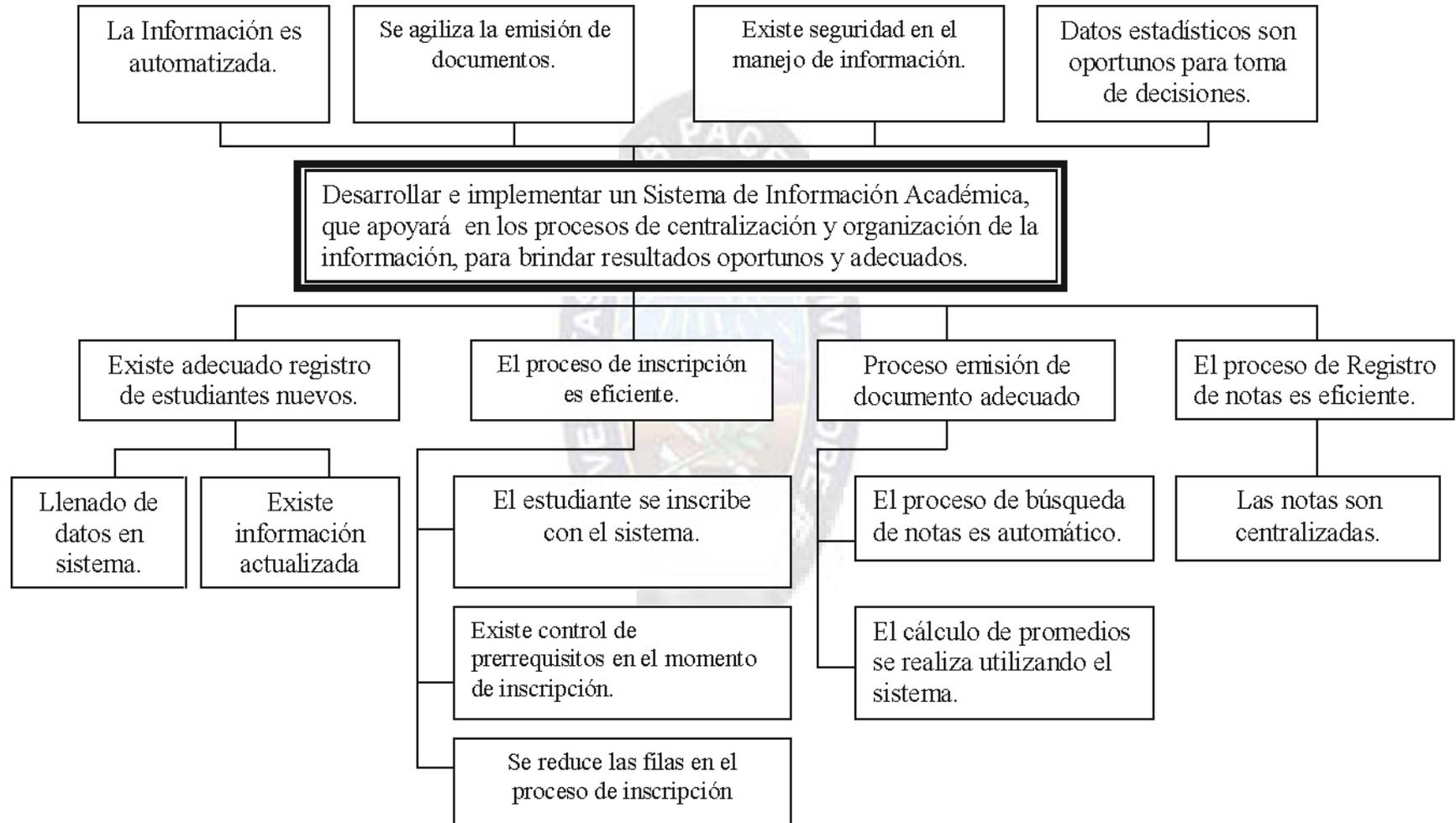




## *Anexo B*

Árbol de objetivos

Figura B.1: Árbol de objetivos





## *Anexo C*

*Marco lógico*

*Tabla C.1: Marco lógico*

MATRIZ DE PLANIFICACIÓN DEL PROYECTO  
SISTEMA DE GESTIÓN ACADÉMICA CARRERA DE LINGÜÍSTICA E IDIOMAS

Fecha aproximada de conclusión del proyecto 10/10/2011

RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACIÓN	SUPUESTOS IMPORTANTES
<i>FIN</i> Apoyo al mejor manejo de la información académica de los estudiantes.	Se reduce el tiempo en emisión de historias académicas, documentos y seguimiento.	Realizar las pruebas necesarias en la institución acorde a cronograma.	Mantener en marcha el sistema de gestión académica.
<b>PROPÓSITO</b> Desarrollo e Implementación de un sistema de información, para una gestión académica, que centralice y organice la información de los estudiantes de lingüística e Idiomas.	La emisión de record académico se reduce de un promedio de 30 minutos a 1 minuto, a partir de la gestión II-2011, es decir los nuevos que ingresan.  Los estudiantes son atendidos en menor tiempo en cuanto a consultas, partir de la gestión II-2011.  Se elimina algunos procesos manuales en el control y manejo de retiro y/o adición de materias a partir de la gestión II-2011.	Aval certificado de la Dirección de carrera de Lingüística e Idiomas.  Aval certificado por el revisor  Encuesta realizada a los usuarios  Proyecto de grado concluido	Respaldo de la Dirección de Carrera de Lingüística e Idiomas en la coordinación de actividades.  Disponibilidad de datos. Cumplimiento de cronogramas  Disponibilidad de documentación  El personal adopta la nueva forma de trabajo

<p><b>PRODUCTO</b></p> <p>1.-Modulo de Registro adecuado de datos de los estudiantes.</p> <p>2. Modulo de inscripción de estudiantes acorde a prerrequisitos.</p> <p>3. Modulo de asignación de docentes.</p> <p>4.- Modulo para el registro de notas.</p> <p>5.-Modulo para Generar historias académicas para cada estudiante.</p> <p>6.-Modulo de consultas mas frecuentes.</p>	<p>Se entrega hasta 15 de septiembre de 2011.</p> <p>Se entrega hasta el 20 de septiembre de 2011.</p> <p>Se entrega hasta el 30 de septiembre de 2011.</p> <p>Se entrega hasta 20 de octubre de 2011.</p> <p>Revisado y probado hasta 30 de octubre de 2011.</p> <p>Revisado y probado hasta 15 de noviembre de 2011.</p>	<p>Visto bueno por parte de Kardex.</p> <p>Visto bueno por parte de Kardex</p>	<p>Se dispone de recursos informáticos y herramientas para la implementación del sistema</p> <p>Continuidad de la institución con esta política</p> <p>Se tiene un sistema que cumple con los requerimientos.</p>
<p><b>ACTIVIDADES</b></p> <p>1. Fase de inicio</p> <p>2. Fase de elaboración</p> <p>3. Fase de construcción</p> <p>4. Fase de transición</p> <p>5. Planificación y Capacitación de personal.</p>	<p>1. 4 semanas/hombre 1000 Bs.</p> <p>2. 4 semanas/hombre 1000 Bs.</p> <p>3. 6 semanas/hombre 7000 Bs.</p> <p>4. 3 semanas/hombre 1500 Bs.</p> <p>5. 1 semanas/hombre 500 Bs.</p> <p>El tiempo estimado para la conclusión del proyecto es de 18 semanas a un costo Total 12000 Bs.</p>	<p>Registro y seguimiento por el Tutor y Revisor del proyecto de grado.</p> <p>Documentación formal del proyecto.</p> <p>Manuales de usuario y de sistema</p> <p>Cronograma e informes de capacitación.</p>	<p>Tener aceptación de la institución para la realización del proyecto, la misma se compromete a prestar cooperación con el desarrollo del mismo.</p> <p>La documentación tenga bibliografía y metodologías.</p>



## *Anexo D*

Instrumentos de diagnóstico



Distinguido Director de Carrera:

Con el propósito de conocer un poco mejor su institución, le pedimos muy respetuosamente nos apoye con el llenado sincero de la presente encuesta que responderá en nuestras expectativas; gracias por su aporte.

(En todos los casos que vea pertinente marque con X la respuesta).

1. Sexo V  M

2. Años de servicio en la institución .....

3. Puesto que desempeña.....

4. Como califica su gestión.....

5. ¿Qué debería mejorar su dirección respecto a Kardex?

R.....

Datos técnicos

Criterio	SI	NO
Su carrera cuenta con equipos de computación		
Kardex cuenta con manuales de funciones y procedimientos		
Laboratorio de computación		
Internet		
Red		
Servidores		
Programas que mas utiliza		
Cuenta con unidad de sistemas		

12. ¿Qué opina sobre las TIC'S?

.....

13. ¿Podría mencionarnos algunos ejemplos?

.....

14. ¿Como emplearía para mejorar su institución ?.....

.....

15. ¿En Kardex?

.....

.....



Distinguido Encargado(a) de kardex:

Con el propósito de conocer un poco mejor el funcionamiento de los procesos que atiende, le pedimos muy respetuosamente nos apoye con el llenado sincero de la presente encuesta que responderá en nuestras expectativas; gracias por su aporte.

(En todos los casos que vea pertinente marque con X la respuesta).

1. Sexo V  M

2. Años de servicio en la institución .....

3. Puesto que desempeña.....

4. Como califica su gestión.....

5. ¿Qué debería mejorar en las tareas que Ud. Atiende en kardex?

R.....

6. Procesos que atiende en sus oficinas:

Nº	PROCESO	Manual		Duración (Minutos)	Procedimiento	Involucrado
		SI	NO			
1						
2						
3						
4						
5						
6						
7						
8						
9						

Preliminar



7. ¿Alguna vez se intento implementar un sistema que administre kardex?

.....  
 .....  
 .....  
 .....

8. SI  NO

9. En caso de que haya marcado si. ¿Por qué no funcionó?

.....  
 .....

10. Lista de programas que maneja:

Nº	PROGRAMA	Instalado		Lo utilizo para:
		SI	NO	
1				
2				
3				
4				
5				
6				

11. ¿Cuenta con equipos de computación al cual tiene acceso?

.....  
 .....

12. ¿Ud. Apoyaría un proyecto que automaticé las tareas mas morosas?

.....  
 .....



## *Anexo E*

*Cronograma de actividades*

Figura E.1: Cronograma de actividades



Fuente: Elaboración Propia



# *Anexo F*

Casos de uso

**Tabla F.1:** Descripción caso de uso – Emisión documentos

Caso de uso	Emisión de documentos	
Actores	Personal kardex	
Propósito	Elaborar documentos como record académico.	
Referencia cruzada	R15	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Kardex introduce código de estudiante y elige opción buscar.	2. El sistema elabora record académico del estudiante buscado, mostrando en pantalla los resultados.
	3. Kardex verifica datos desplegados.	
	4. Kardex selecciona la opción imprimir.	5. El sistema imprime record académico.
Flujo alternativo	En 3. Se puede dar orden de imprimir directamente.	

**Fuente:** Elaboración propia

**Tabla F.2:** Descripción caso de uso - Consultas

Caso de uso	Consultas	
Actores	Personal kardex	
Propósito	Realizar búsquedas de información más frecuentes, listas por paralelo.	
Referencia cruzada	R18	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Kardex selecciona parámetros y/o datos de entrada y elige la opción buscar.	2. El sistema realiza la búsqueda, mostrando resultado en pantalla.
	3. Kardex verifica los datos desplegados de consulta.	
	4. Kardex elige la opción imprimir.	5. el sistema imprime consulta.
Flujo alternativo	En 4. Kardex solicita exportar la consulta a archivo tipo Excel o Pdf.	

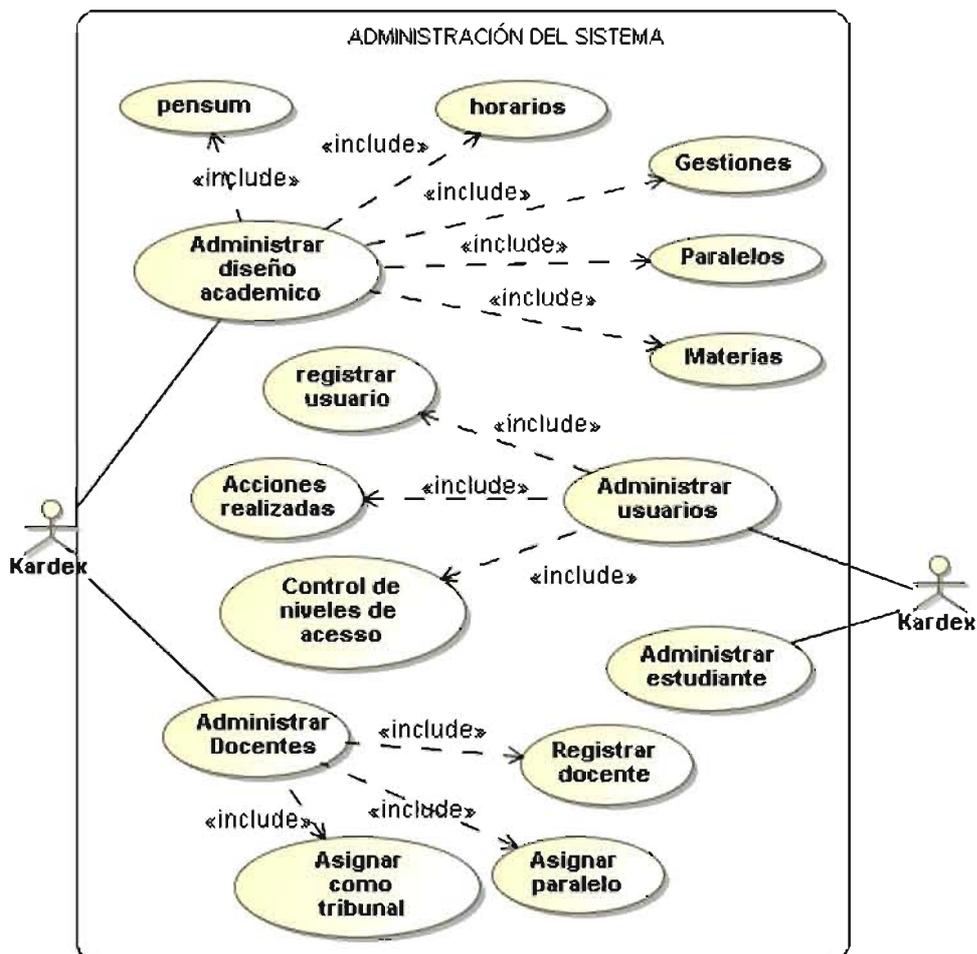
**Fuente:** Elaboración propia

**Tabla F.3:** Descripción caso de uso – Ingresar al sistema

Caso de uso	: Ingresar al sistema	
Actores	: Personal Kardex, Docentes, estudiantes	
Propósito	: Permitir el manejo del sistema a usuarios registrados previamente, bajo un nivel de acceso.	
Referencia cruzada	: R1	
Flujo normal de eventos	<b>Actores</b>	<b>Sistema</b>
	1. Usuarios autorizados ingresan su USUARIO Y CONTRASEÑA. Haciendo click en ingresar.	2. El sistema verifica si los datos ingresados corresponden a un usuario activo. 3. El sistema muestra el menú principal acorde al privilegio del usuario.
Flujo alternativo	En 2. El sistema muestra mensaje de usuario no autorizado comuníquese con su proveedor.	

Fuente: Elaboración propia

**Figura F.1:** Diagrama de caso de uso - Administración



Fuente: Elaboración propia

**Tabla F.4:** Descripción de caso de uso – Registrar usuario

Caso de uso	:	Registrar usuario	
Actores	:	Encargado de kardex que administre el sistema	
Propósito	:	Registrar funcionarios que utilizarán el sistema	
Referencia cruzada	:	R13	
Flujo normal de eventos		<b>Actores</b>	
		<b>Sistema</b>	
		1. Personal de Kardex con rol de administrador, carga datos de usuario.	
		2. El personal de kardex pide a usuario introducir su contraseña.	3. El sistema verifica completitud de datos.
		3. El funcionario de kardex ordena grabar	4. Almacena datos introducidos.
	5. Recomienda a usuario normas de seguridad.		
Flujo alternativo		Verificación previa si ya esta registrado.	

Fuente: Elaboración propia

**Tabla F.5:** Descripción de caso de uso – Registrar docente

Caso de uso	:	Registrar docente	
Actores	:	Encargado de kardex que administre el sistema	
Propósito	:	Registrar docentes que sus datos figuren en acta de notas y utilicen el sistema.	
Referencia cruzada	:	R16	
Flujo normal de eventos		<b>Actores</b>	
		<b>Sistema</b>	
		1. Kardex con rol de administrador, introduce los datos del docente y elige la opción guardar.	2. El sistema verifica completitud de datos y almacena registro.
			3. El sistema muestra mensaje de registro exitoso.
		4. Kardex busca mediante C.I. en docentes registrados	5. El sistema despliega resultado de búsqueda.
	5. Kardex esta conforme con registro.		
Flujo alternativo		En 1. Kardex puede saltar al paso 4 para verificar si docente ya fue registrado.	

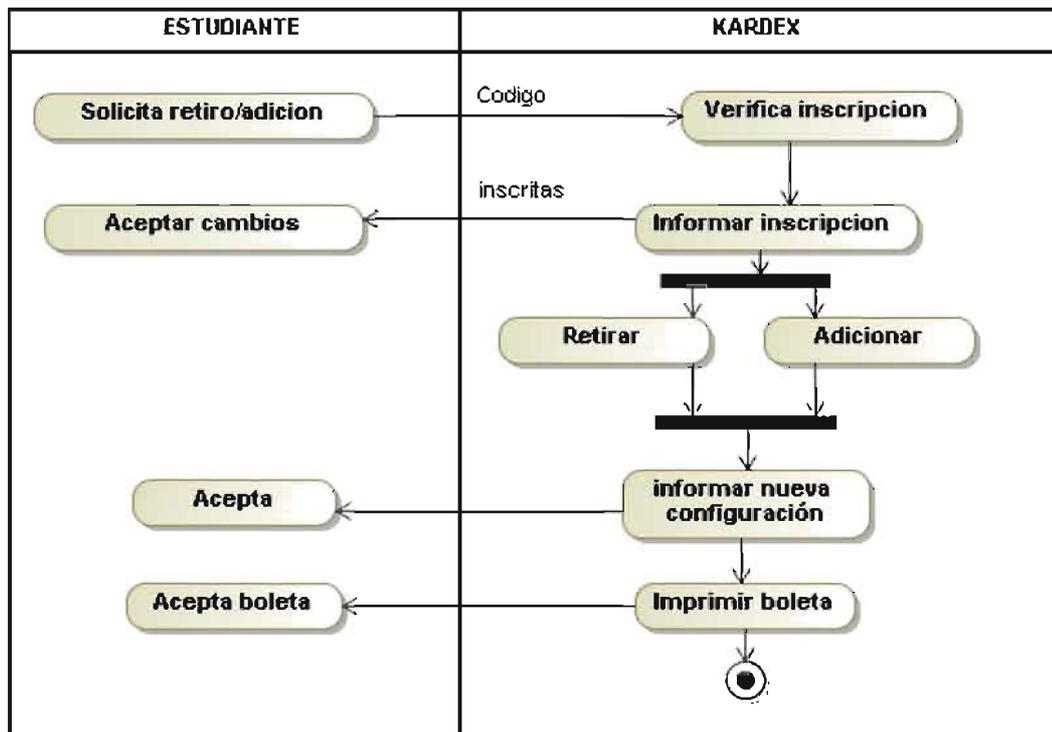
Fuente: Elaboración propia



## *Anexo G*

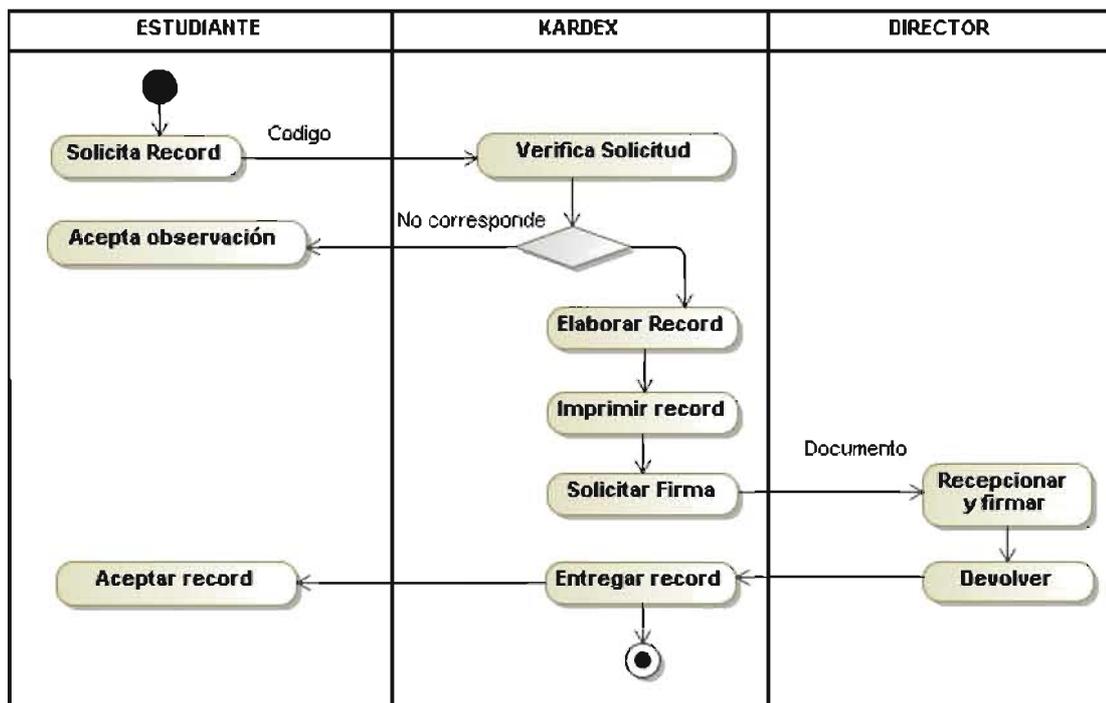
Diagramas de actividad

Figura G.1: Diagrama de actividad – Retiro y/o Adición



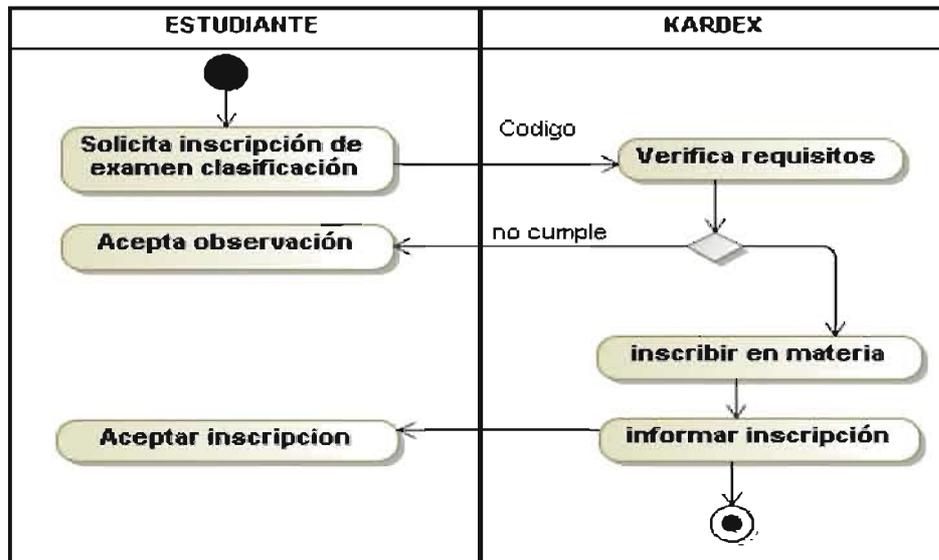
Fuente: Elaboración Propia

Figura G.2: Diagrama de actividad – Record Académico



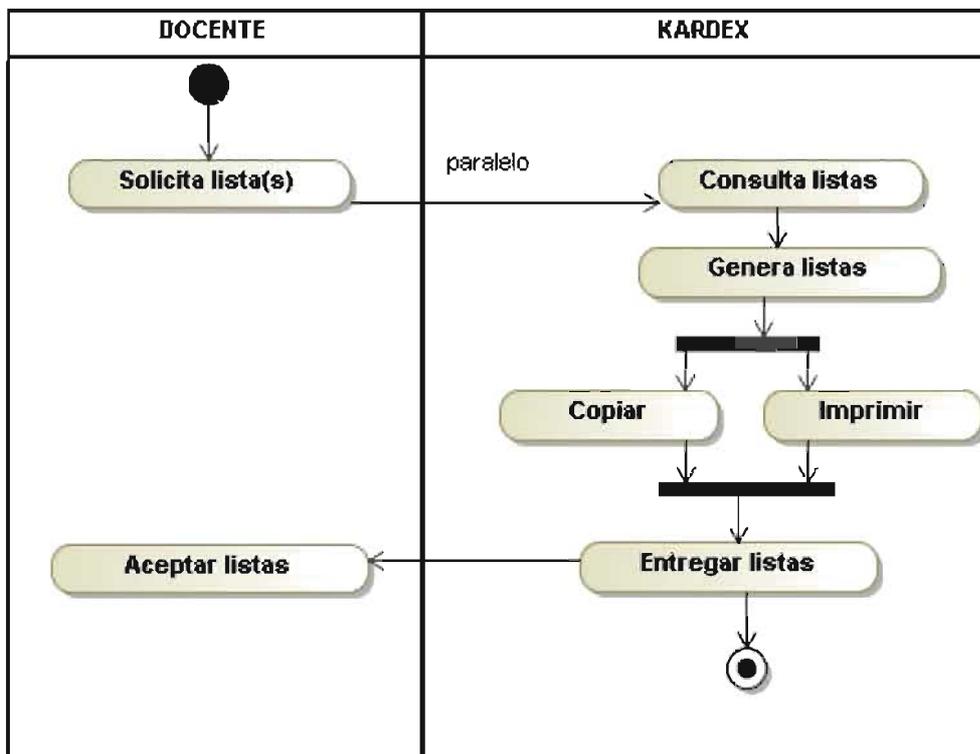
Fuente: Elaboración Propia

Figura G.3: Diagrama de actividad – Examen de clasificación



Fuente: Elaboración propia

Figura G.4: Diagrama de actividad – Generación de listas



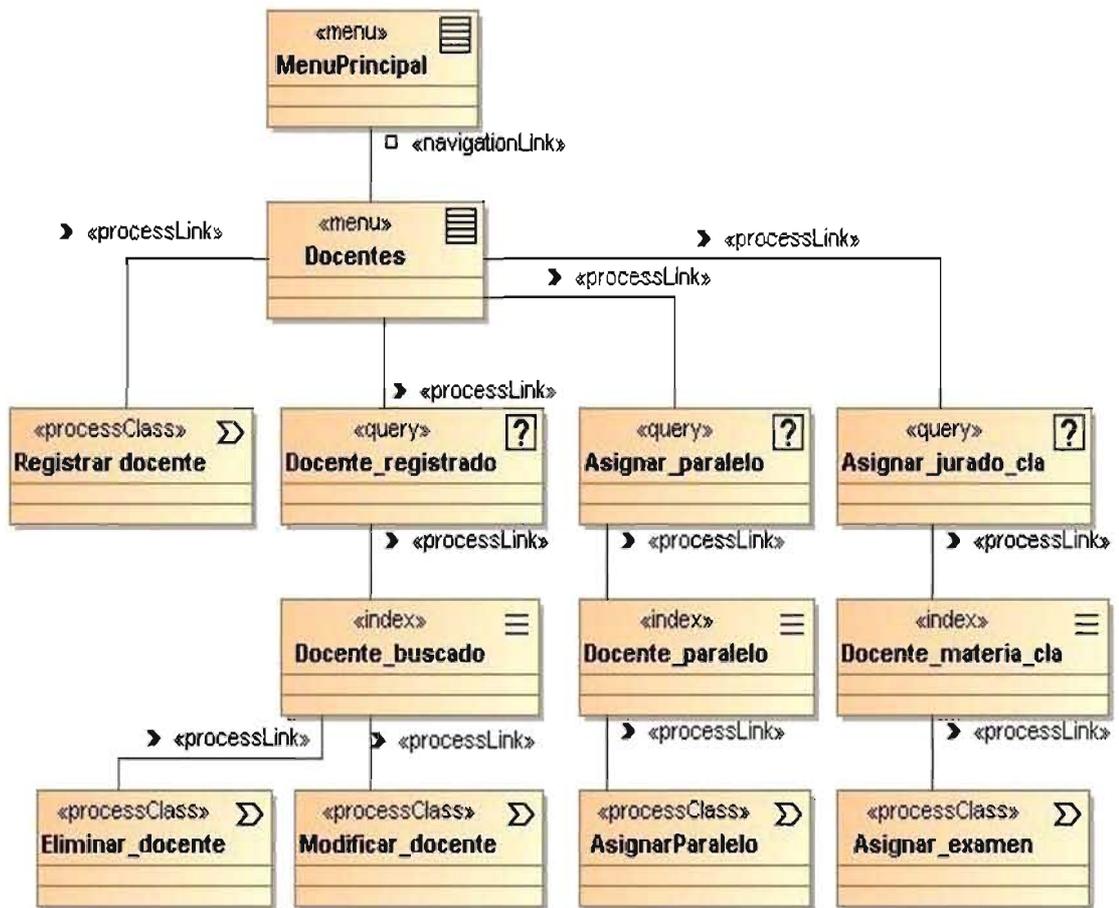
Fuente: Elaboración propia



## *Anexo H*

*Diagramas de navegación*

Figura H.1: Diagrama de navegación - Docentes



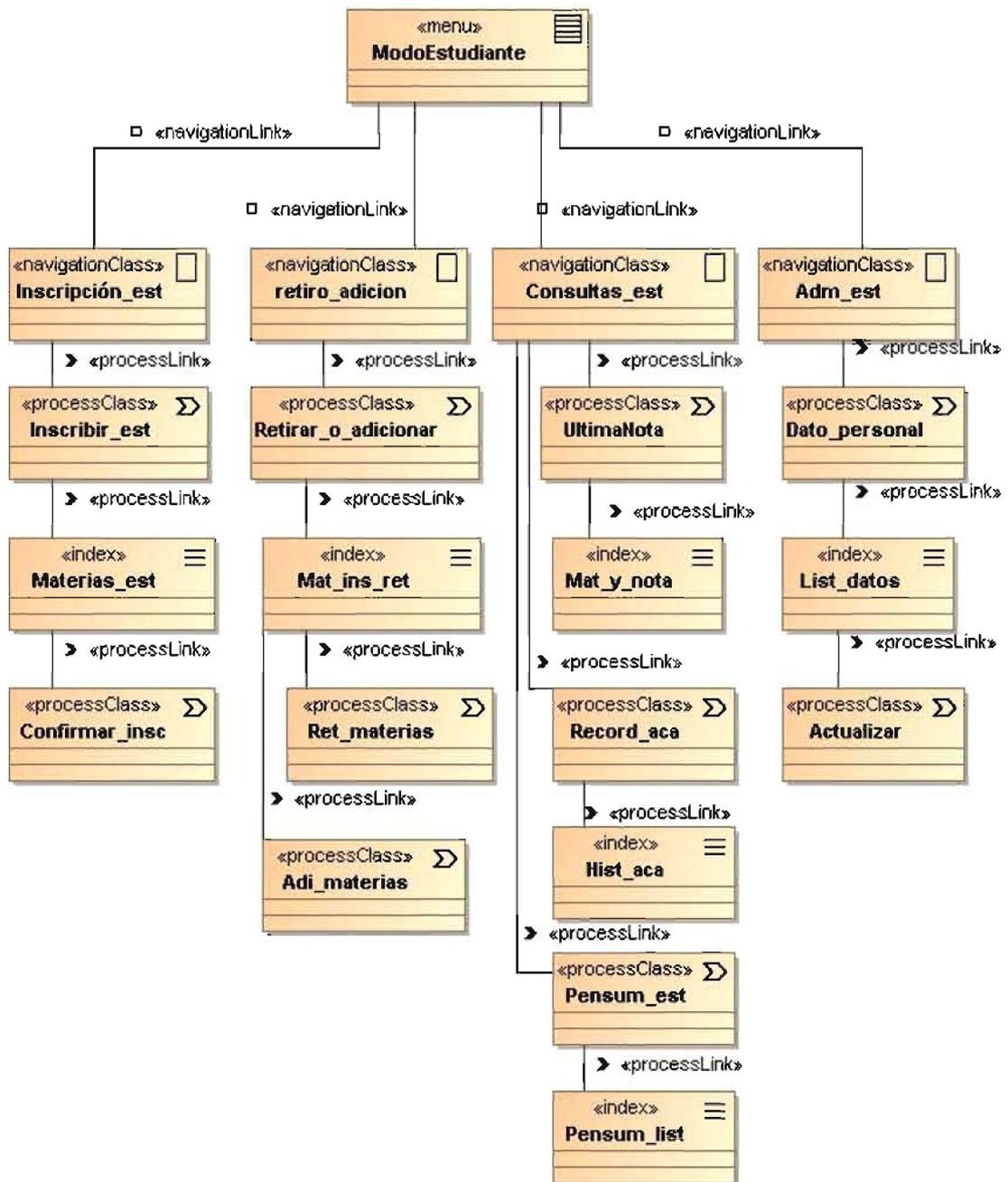
Fuente: Elaboración propia

Figura H.2: Diagrama de navegación - Reportes



Fuente: Elaboración propia

Figura H.3: Diagrama de navegación – Modo estudiante



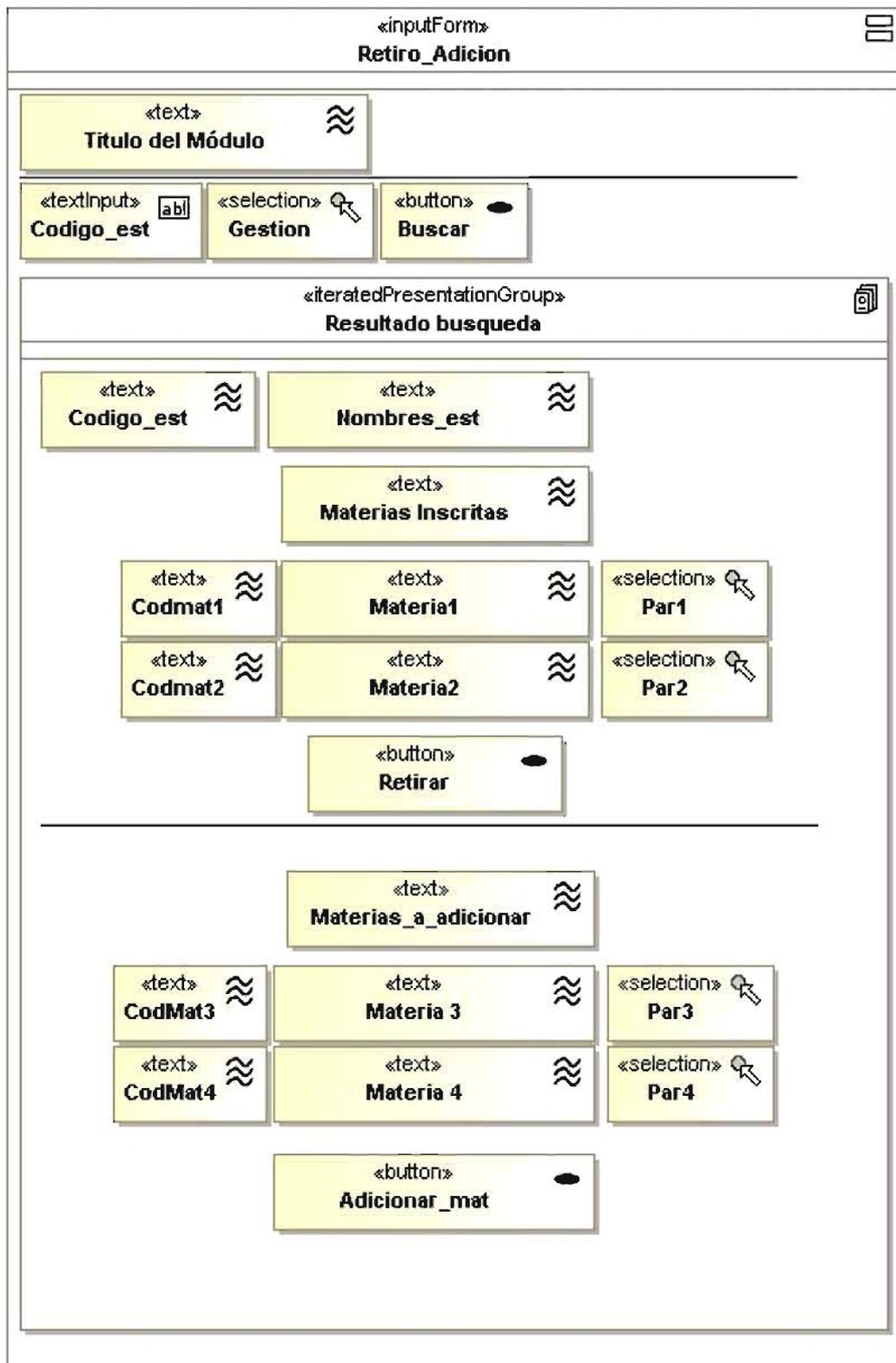
Fuente: Elaboración propia



## *Anexo I*

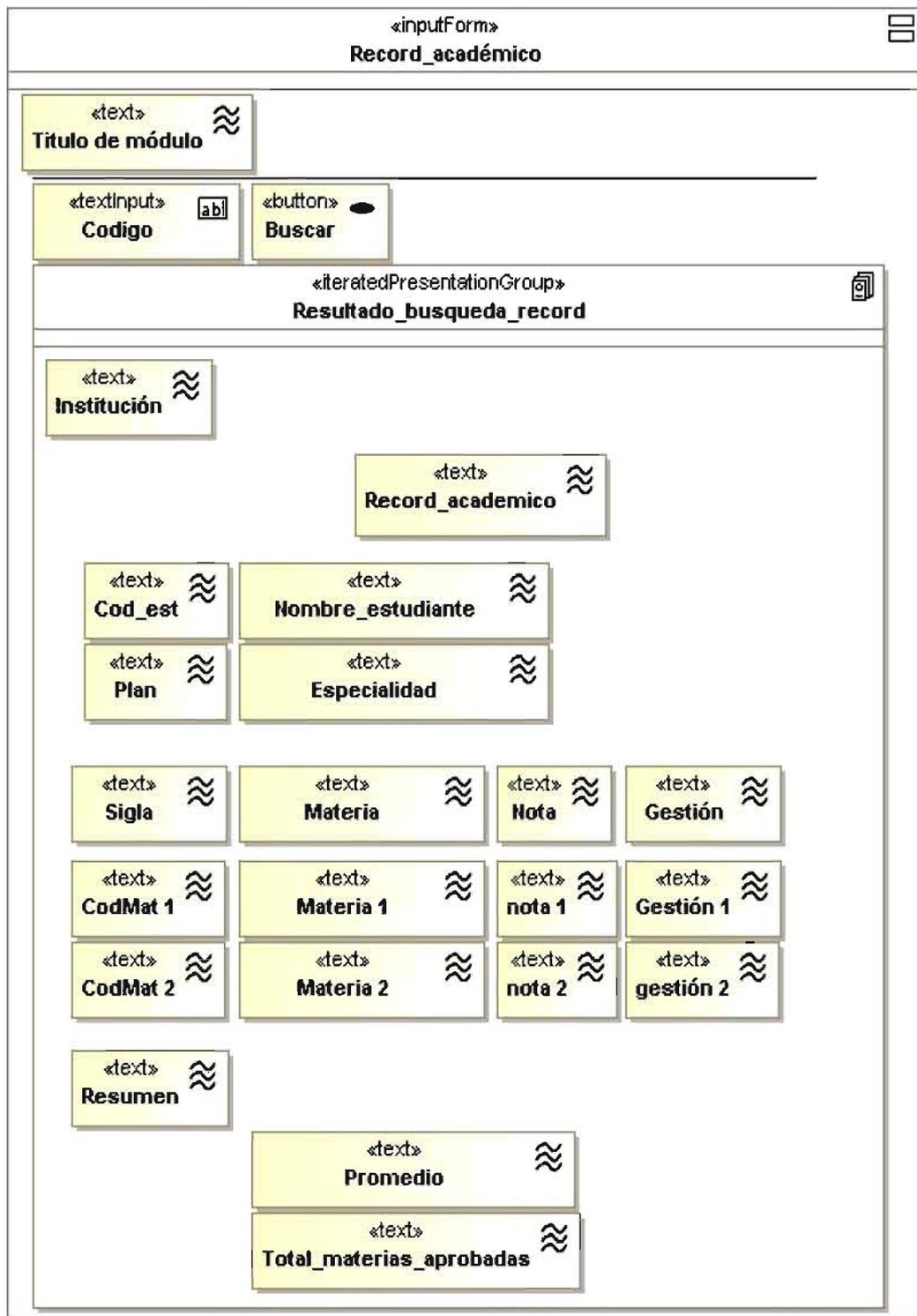
Diagramas de presentación

Figura I.1: Diagrama de presentación – Retiro y/o Adición



Fuente: Elaboración propia

Figura I.2: Diagrama de presentación – Record Académico



Fuente: Elaboración propia



## *Anexo J*

Diagramas de secuencia

Figura J.1: Diagrama de secuencia – Examen de clasificación

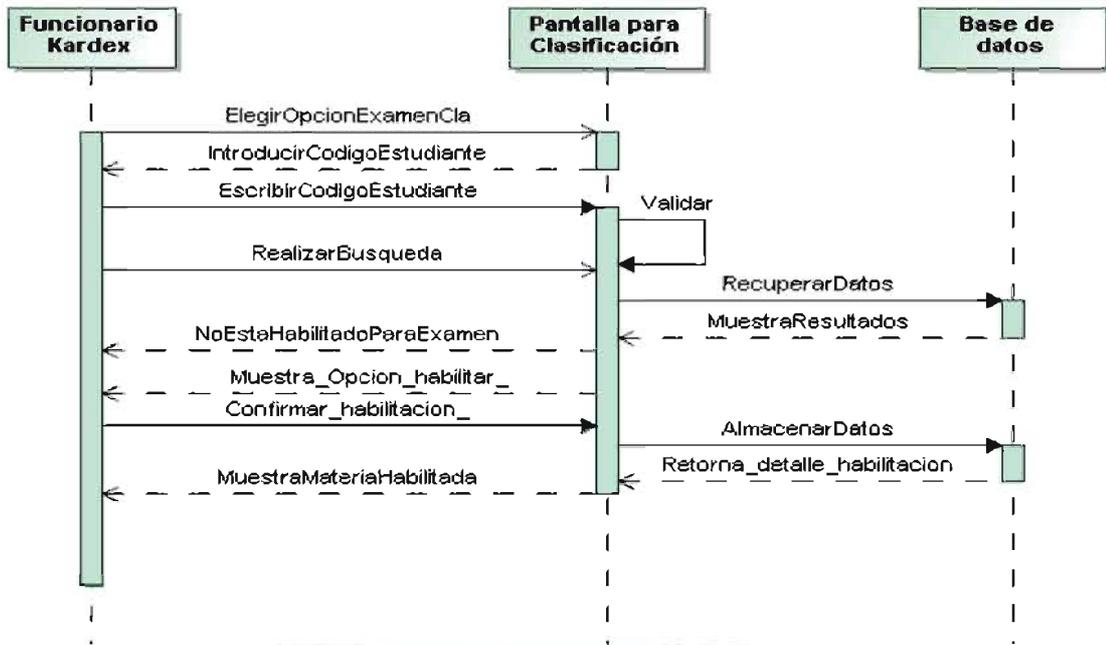
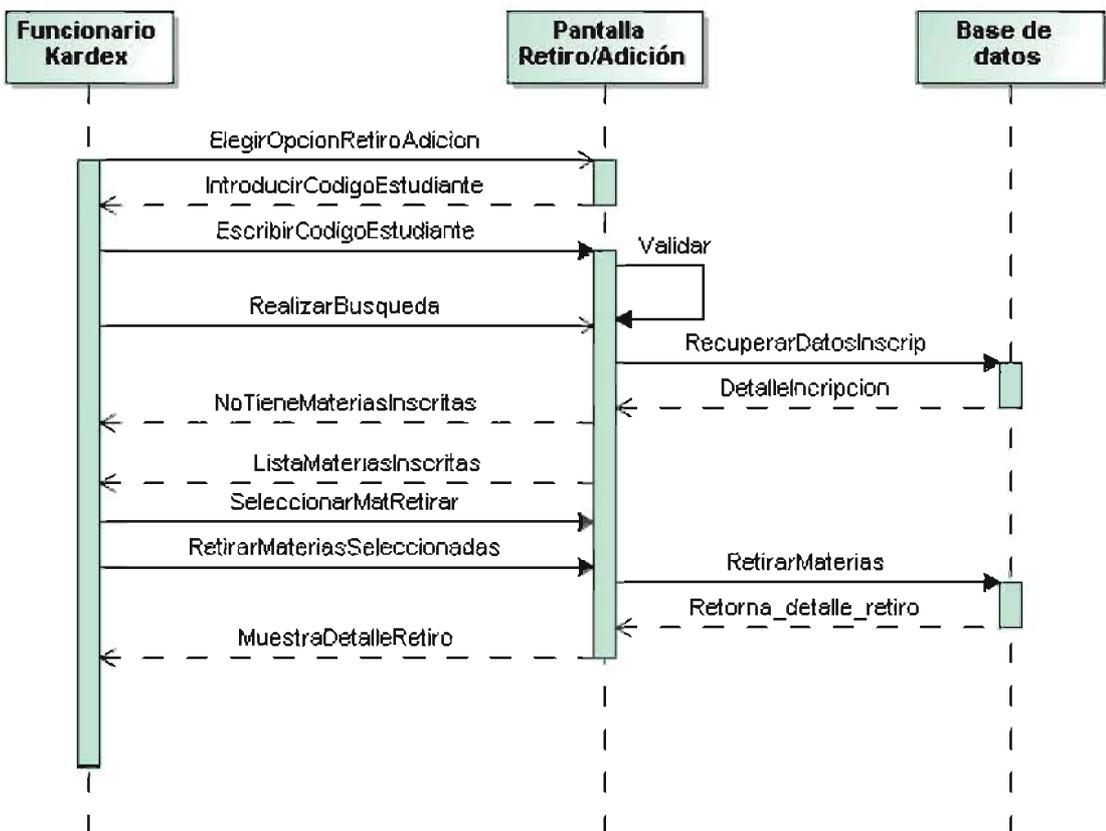


Figura J.2: Diagrama de secuencia – Retirar materia(s)



Fuente: Elaboración propia

Figura J.3: Diagrama de secuencia – Adicionar materia(s)

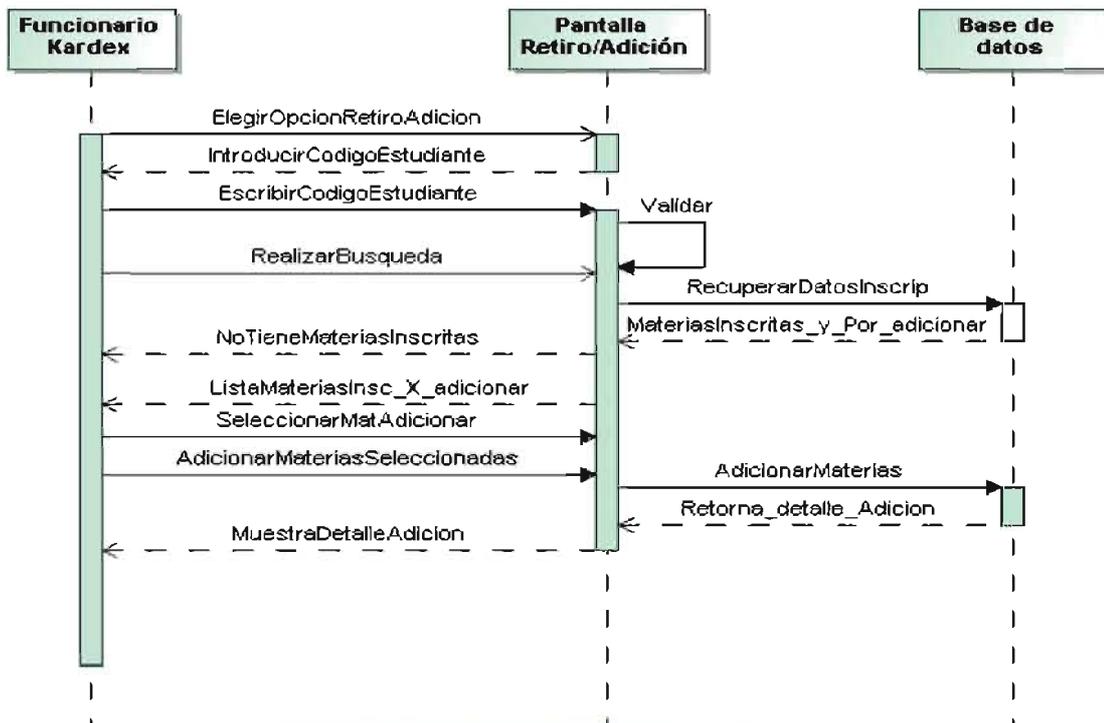
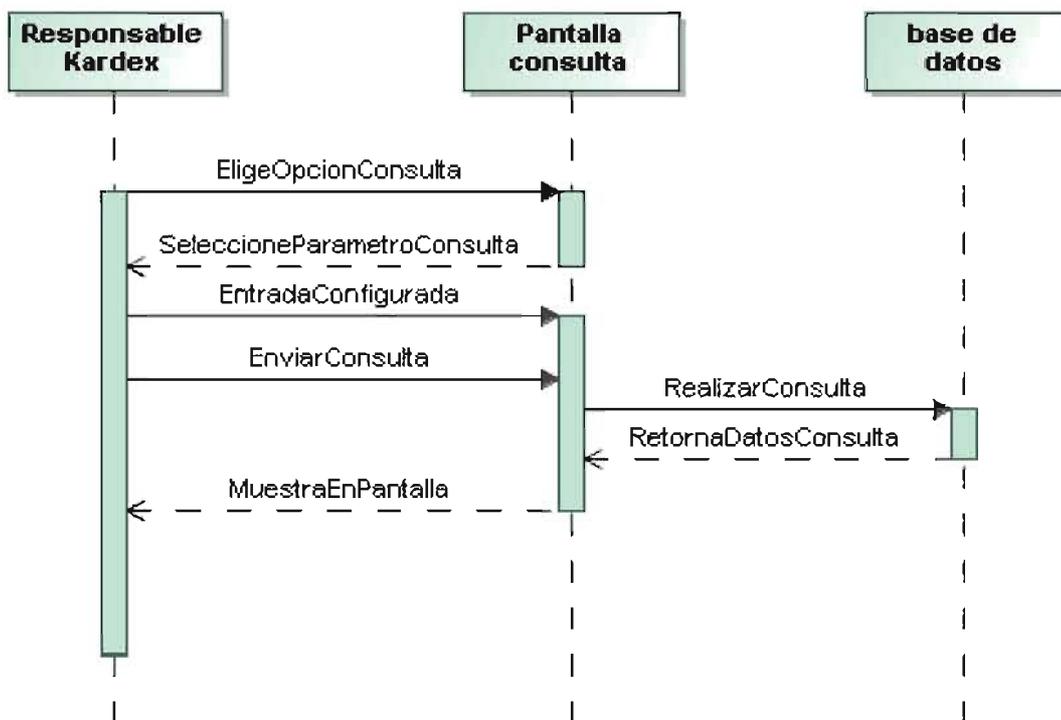


Figura J.4: Diagrama de secuencia – Adicionar materia(s)



Fuente: Elaboración propia



## *Anexo K*

Pantallas del sistema

Uno de los modulo que se desarrollo es el de registro de docentes para acta de notas, horarios y asignación como tribunal.

**Figura K.1: Módulo docente – Registrar docente**

The screenshot shows a web application interface for registering a teacher. The top navigation bar includes tabs: Registrar, Registrados, Asignar materia, Lista, Tribunal Clasi, and Cons5. The main form is titled 'DATOS PERSONALES' and contains several input fields: Cedula, Exp., Grado, Ap. Paterno, Ap. Materno, Nombre(s), Sexo (radio buttons for M and F), and País. Below this is the 'DATOS DE REFERENCIA' section with fields for Dirección, Zona, Calle/Av., Número, Teléfono, and Correo. At the bottom, there are fields for Fecha ingreso, Fecha retiro, and Observacion, followed by a 'Guardar' button.

Fuente: Elaboración propia

Ahora veamos una de sus pestañas que asignar materia:

**Figura K.2: Módulo docente – Asignar materia**

The screenshot shows the 'Asignar materia' form. The navigation bar is the same as in Figure K.1. The form contains three dropdown menus: 'DOCENTE' with the placeholder '---Elija docente--', 'MATERIA' with '--Elija materia--', and 'PARALELO' with '--Elija paralelo--'. Below these is an 'Asignar' button.

Al desplegar la opción docente y materia se tiene.

- 
- The screenshot shows two expanded dropdown menus. The left menu, labeled '---Elija docente--', lists several names: ALAVI ALAVI ZCARIAS, APAZA APAZA IGNACIO, ANDIA ANDIA DOUGLAS EDUARDO, APAZA APAZA MARCELO, ADUVIRI ADUVIRI JOSE DAVID, ARUQUIPA ARUQUIPA JAVIER DOMINGO, AYMA AYMA DONATO, and BALISTITA BALISTITA BECINA. The right menu, labeled '--Elija materia--', lists course codes and titles: LIN-121 TALLER DE LENGUAJE I, LIN-131 LINGUISTICA GENERAL, LIN-132 FONETICA Y FONOLOGIA, HIS-181 HISTORIA Y CULTURA ANDINA, LIN-101 AYMARA I, and LIN-201 IDIOMA INGLÉS I.

Del mismo modo los paralelos que se crean cada gestión, y al hacer click en el botón asignar este queda asignado.

**Figura K.3: Módulo documentos - Boleta de inscripción**

UNIVERSIDAD PUBLICA DE EL ALTO  
CARRERA DE LINGUISTICA E IDIOMAS

**BOLETA DE INSCRIPCION**  
GESTION: II-2011

<b>ESTUDIANTE:</b>	CARVAJAL SUMI GLADYS MARISOL	<b>COD:</b>	CSG-150591
<b>ESPECIALIDAD:</b>	LINGUISTICA E IDIOMAS	<b>PLAN:</b>	PLAN NRO 8 (II-2007) V.ESPERANZA

SEMESTRE	SIGLA	MATERIA	PARALELO
1	LIN-131	LINGUISTICA GENERAL	B
1	LIN-121	TALLER DE LENGUAJE I	B
1	LIN-132	FONETICA Y FONOLOGIA	B
1	HIS-181	HISTORIA Y CULTURA ANDINA	B
1	LIN-101	AYMARA I	B
1	LIN-201	IDIOMA I- INGLES I	B

6 materias-inscritas

NO tiene materias RETIRADAS

NO tiene materias ADICIONADAS

Fuente: Elaboración Propia

Una de las opciones del modulo administrador es la creación de paralelos, como se ve en la figura k.4.

**Figura k.4: Módulo administración – Crear paralelo**

**ADMINISTRACION**  
Tareas específicas

Gestiones	Nuevo Par	Lista Par	User	Materia	Pensum	Aulas
-- Elija --	--Elija--	-Elija-	40	--Elija--	--Elija--	
GESTION	SEMESTRE	CUPO	ESTADO	TURNO		
Crear						

--Elija--  
 A  
 B  
 C  
 D  
 E  
 F

Fuente: Elaboración propia



# *Documentos*