

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



PROYECTO DE GRADO

**SISTEMA DE CONTROL DE TRÁMITES UTILIZANDO
MAQUINAS DE TURING**

**CASO: DIVISIÓN DE GESTIONES ADMISIONES Y
REGISTROS U.M.S.A.**

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

POSTULANTE: Patricia Justina Nacho

TUTOR: Lic. Nancy Orihuela Sequeiros

REVISOR: Lic. Brígida Carvajal Blanco

ASESOR: Lic. Ivan Vallejos Plaza

LA PAZ – BOLIVIA

2007

ÍNDICE

CAPÍTULO I MARCO CONTEXTUAL

1.1 Introducción.....	1
1.2 Antecedentes.....	2
1.3 Planteamiento del problema.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo Principal.....	5
1.4.2 Objetivos Secundarios.....	5
1.5 Justificación.....	6
1.5.1 Justificación Económica.....	6
1.5.2 Justificación Social.....	6
1.5.3 Justificación Tecnológica.....	6
1.5.4 Justificación Académica.....	6
1.6 Límites y alcances.....	7
1.6.1 Límites.....	7
1.6.2 Alcances.....	7
1.6.3 Herramientas.....	8
1.7 Aporte teórico.....	8
1.8 Aporte práctico.....	9
1.9 Metodología.....	9

CAPÍTULO II MARCO TEÓRICO

2.1 Introducción.....	11
2.2 División de Gestiones Admisiones y Registros y sus funciones.....	11
2.2.1 Proceso de Carreras Paralelas.....	13
2.2.2 Proceso de Devolución de documentos.....	14
2.2.3 Proceso de Readmisiones.....	15
2.2.4 Proceso de Modalidades de Graduación.....	16
2.2.5 Proceso de Admisiones Especiales.....	17
2.3 Metodología RUP.....	19
2.3.1 Características de la metodología RUP.....	20
2.3.1.1 Dirigido por casos de uso.....	20
2.3.1.2 Centrado en la arquitectura.....	20

2.3.1.3 Iterativo e incremental.....	21
2.3.2 Fases de la metodología RUP.....	21
2.3.2.1 Fase de inicio.....	21
2.3.2.2 Fase de elaboración.....	21
2.3.2.3 Fase de construcción.....	22
2.3.2.4 Fase de transición.....	22
2.4 Arquitectura Spring.....	22
2.5 Estándar de IEEE 830.....	23
2.6 Máquinas de Turing.....	26
2.6.1 Definición.....	26
2.6.2. Elementos de una Máquina de Turing.....	27
2.6.3 La Máquina de Turing como aplicación.....	28
2.6.4 Forma de implementación de la Máquina de Turing en el proyecto.....	29
2.7 Workflows.....	30
2.8 Base de datos.....	31
2.8.1 Bases de datos orientadas a objetos.....	31
2.8.2 Bases de datos relacionales.....	32
2.8.2.1 Documentos XML.....	33
2.8.2.2 Motor de Persistencia ORM.....	34
2.8.3 Bases de datos de conocimiento.....	34

CAPÍTULO III MARCO APLICATIVO

3.1 Introducción.....	36
3.2 Máquinas de Turing de los procesos.....	37
3.3 Etapas de la metodología RUP.....	43
3.3.1 Modelado del negocio.....	43
3.3.2 Modelado de requisitos.....	46
3.3.2.1 Diagramas de casos de uso.....	47
3.3.3 Modelado del análisis.....	50
3.3.3.1 Diagramas de clases de análisis.....	51
3.3.3.2 Diagramas de secuencias.....	54
3.3.3.3 Diagramas de actividades.....	59
3.3.4 Modelado del diseño.....	64
3.3.4.1 Diagrama de clases.....	64

3.3.4.2 Arquitectura del sistema.....	65
3.3.4.3 Modelado de datos.....	66
3.4 Implementación.....	69
3.4.1 Pantallas del sistema.....	70
3.4.2 Prueba de la caja blanca.....	74
3.4.2 Prueba de la caja negra.....	76
3.5 Despliegue.....	78
3.5.1 Modelo de Despliegue.....	78
3.6 Seguridad del sistema.....	79
3.7 Capacitación del personal.....	80

CAPÍTULO IV CONCRECIÓN DEL MODELO

4 Métricas de calidad.....	82
4.1 Métricas orientadas a la función.....	82
4.2 Métricas de mantenimiento.....	85
4.3 Portabilidad.....	86
4.4 Confiabilidad.....	86
4.5 Conclusiones.....	88
4.6 Recomendaciones.....	89

REFERENCIAS BIBLIOGRÁFICAS

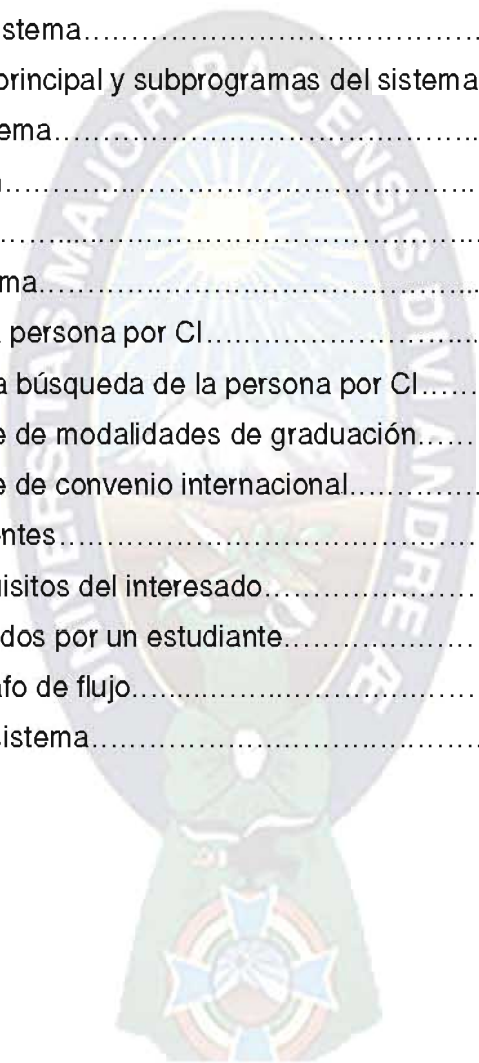
ANEXOS

- A. Marco Lógico
- B. Análisis de involucrados
- C. Árbol de problemas
- D. Árbol de Objetivos
- E. Análisis de alternativas
- F. Diagrama de flujo de datos de los procesos
- G. Encuestas y resultados
- H. Modelo Vista Controlador
- I. Documento de Visión
- J. Especificación de requerimientos software según el Estándar IEEE 830
- K. Especificación de casos de uso
- L. Reportes emitidos por el sistema
- M. Diagrama de clases

LISTA DE FIGURAS

Figura	Pag.
1 Organigrama DGAR.....	12
2 Metodología RUP.....	20
3 Módulos de Spring.....	22
4 Esquema de una Máquina de Turing representada por un grafo.....	26
5 Esquema de funcionamiento de la Máquina de Turing en el proyecto.....	29
6 Etapas de la metodología RUP.....	37
7 Máquina de Turing para Modalidades de Graduación.....	38
8 Máquina de Turing para Readmisiones.....	39
9 Máquina de Turing para Devolución de documentos,.....	39
10 Máquina de Turing para Admisiones especiales.....	41
11 Máquina de Turing para Carreras Paralelas.....	41
12 Modelado del negocio.....	43
13 Diagrama de casos de uso del negocio.....	44
14 Modelo de Objetos de Control de Modalidades de graduación.....	45
15 Modelo de Objetos de Control Carrera Paralelas.....	45
16 Modelo de Objetos de Control de Readmisiones.....	46
17 Modelo de Objetos de Control Devolución de documentos.....	46
18 Modelo de Objetos de Control de Admisiones Especiales.....	46
19 Diagrama de casos de uso Control Modalidades de Graduación.....	47
20 Diagrama de casos de uso Control Carreras Paralelas.....	48
21 Diagrama de casos de uso Control de Readmisiones.....	48
22 Diagrama de casos de uso Control Devolución de documentos.....	49
23 Diagrama de casos de uso Control de Admisiones Especiales.....	49
24 Esquema de clases de diseño.....	51
25 Diagrama de clases del proceso: Control de Modalidades de Graduación.....	52
26 Diagrama de clases del proceso: Control de Carreras Paralelas.....	52
27 Diagrama de clases del proceso: Control de Readmisiones.....	53
28 Diagrama de clases del proceso: Control de Devolución de documentos.....	53
29 Diagrama de clases del proceso: Control de Admisiones Especiales.....	54
30 Diagrama de secuencias: Control de Modalidades de Graduación.....	55
31 Diagrama de secuencias: Control de Readmisiones.....	56
32 Diagrama de secuencias: Control de Devolución de documentos.....	57

33 Diagrama de secuencias: Control de Admisiones Especiales.....	58
34 Diagrama de secuencias: Control de Carreras Paralelas.....	59
35 Diagrama de actividades: Control de Readmisiones.....	60
36 Diagrama de actividades: Control de Modalidades de Graduación.....	61
37 Diagrama de actividades: Control de Devolución de documentos.....	62
38 Diagrama de actividades: Control de Admisiones Especiales.....	63
39 Diagrama de actividades: Control de Carreras Paralelas.....	64
40 Arquitectura de datos del sistema.....	65
41 Arquitectura del programa principal y subprogramas del sistema.....	66
42 Modelado de datos del sistema.....	67
43 Diagrama Entidad Relación.....	68
44 Diagrama de componentes.....	69
45 Pantalla de ingreso al sistema.....	70
46 Pantalla de búsqueda de la persona por CI.....	70
47 Pantalla de resultados de la búsqueda de la persona por CI.....	71
48 Pantalla de inicio de trámite de modalidades de graduación.....	71
49 Pantalla de inicio de trámite de convenio internacional.....	72
50 Pantalla de trámites pendientes.....	72
51 Pantalla de llenado de requisitos del interesado.....	73
52 Pantalla de trámites realizados por un estudiante.....	73
53 Diseño procedimental y grafo de flujo.....	75
54 Modelo de despliegue del sistema.....	79



LISTA DE TABLAS

Tabla	Pag.
1 Tabla de acción de la Máquina de Turing.....	28
2 Tabla de estados para Inscripción de Profesionales.....	42
3 Cronograma de capacitación del personal de la División de Gestiones.....	80
4 Calculo de cuenta total de la métrica de Punto Función.....	83
5 Calculo de Ajuste de complejidad.....	84
6 Escala de valores para Punto Función.....	85
7 Tabla comparativa de resultados.....	88





Dedicatoria

A mi mamá Teófila por haber sido la forjadora de mi desarrollo como persona y apoyarme en las buenas y malas de mi vida. A mi papá Hilarión por ser ejemplo de trabajo y enseñarme lo más importante de la vida que es luchar y superarse en cada momento.

A Geybert por cuidarme en todo momento y darme su amor y ternura cada día.

Patricia Justina Nacho

AGRADECIMIENTOS

A la Lic. Nancy Orihuela Sequeiros por guiarme de forma paciente, dándome consejos para concluir el proyecto y por la motivación constante que siempre me ha brindado.

A la Lic. Brígida Carvajal Blanco por haber sido parte de mi proyecto con las revisiones constantes del documento, por sus sugerencias que dieron forma al documento.

Al Lic. Ivan Vallejos Plaza, por el asesoramiento constante en el desarrollo del proyecto, por sus consejos y recomendaciones que me sirvieron de mucho para el desarrollo y conclusión del sistema.

Al Lic. Edgar Clavijo, por las recomendaciones, sugerencias y colaboración brindada al inicio del proyecto, que fueron realmente útiles para el desarrollo del mismo.

A la Lic. María del Carmen Calvetty Amboni por haberme dado la oportunidad de desarrollar el presente proyecto en la División de Gestiones, por confiar en mí y apoyarme en todos los aspectos necesarios para la conclusión del mismo.

Al todo el personal de la División de Gestiones Admisiones y Registros de la UMSA por ayudarme a desarrollar el proyecto con sus observaciones y sugerencias, por brindarme su amistad desinteresada el cual es muy bien correspondido.

A mi mamá Teófila Nacho por ser mi amiga y confidente, por darme fortaleza en todo momento y estar siempre presente en mi vida dándome todo su amor y cariño.

A mi papá Hilarión Nacho por su apoyo constante, a mi mamá Marcelina por su cariño, a Geybert por ser mi ángel guardián, a toda mi familia, por motivarme cada día a seguir adelante, por confiar en mí y ser parte importante de mi vida.

A Eddy Valero por apoyarme en cada momento, por estar siempre cuando más lo necesito y sobre todo por estar conmigo en las buenas, malas y peores.

A todos mis amigos y amigas por incentivar me en cada momento para terminar este trabajo.

Gracias.

RESUMEN

La División de Gestiones Admisiones y Registros perteneciente a la Universidad Mayor de San Andrés es la unidad encargada del ingreso, movilidad estudiantil, modalidades de graduación y otros procesos complementarios que en algún momento un estudiante realiza en su vida universitaria.

Cada proceso es realizado mediante un trámite que es iniciado a solicitud de un estudiante, cada trámite necesita ser registrado, controlado en cada paso del mismo y por último es necesario obtener reportes en cualquier momento.

El presente sistema realiza el seguimiento y control de trámites desde el registro inicial hasta la conclusión del mismo. El control del trámite se realiza de forma dinámica gracias a que utiliza una máquina de Turing que es un autómata que posee una tabla de estados y permite que el sistema tenga un buen control en el seguimiento de cada trámite; además el hecho de tener una máquina de Turing inmerso en la base de datos hace que se constituya una base de conocimiento ya que la misma controla el funcionamiento del sistema, por lo tanto el sistema trabaja con una base de datos híbrida cuyo gestor es Postgres.

Para el desarrollo del sistema se utilizó la metodología RUP que tiene como herramienta al UML, esta metodología posee las siguientes características: es dirigido por casos de uso, es centrado en la arquitectura y por último es iterativo incremental lo cual lo hace una metodología ideal para el desarrollo de sistemas grandes. La programación se realizó con el lenguaje de programación Java bajo la arquitectura Spring, se utilizó como servidor web el Tomcat-Jakarta y sistema operativo Linux.

El sistema permite realizar el seguimiento y control de trámites de manera óptima, centralizando la información generada en un servidor que tiene suficientes medidas de seguridad para evitar que la información se pierda o sea alterada en cualquier momento.

CAPÍTULO I

MARCO CONTEXTUAL

Este capítulo presenta una descripción general de la División de Gestiones Admisiones y Registros, las justificaciones, objetivos, límites y alcances del presente proyecto de grado.

1.1 INTRODUCCIÓN

La tecnología actual brinda herramientas que ayudan a automatizar los datos, dejando de esta manera al olvido las formas tradicionales de guardar la información. El ordenador es un medio por el cual podemos utilizar esas herramientas para que así se pueda disponer de la información en el instante preciso y de una forma segura.

La División de Gestiones, Admisiones y Registros (DGAR) de la Universidad Mayor de San Andrés (UMSA), es una de las entidades más importantes de la UMSA ya que es la encargada de los procesos de: admisión, movilidad estudiantil, modalidades de graduación y otros complementarios como ser la legalización de documentos y otros. Cada uno de los procesos anteriormente mencionados, son realizados por medio de trámites que son iniciados a solicitud de la comunidad estudiantil y previo cronograma de actividades preparado por la División.

Cada uno de los procesos que maneja la DGAR, deben de ser realizados bajo ciertas normas de seguridad ya que se trata de información de vital importancia no solamente para la División sino también para otras Divisiones pertenecientes a la UMSA y unidades académicas como ser las distintas carreras de la Universidad, dicha información será base fundamental para la realización y verificación de otros procesos que son responsabilidad de las entidades anteriormente mencionadas.

La conectividad de sistemas hoy en día es un factor muy importante e indispensable para la transmisión de información de un punto a otro y también para reducir el tiempo de servicio,

pero la conectividad a la vez esta ligada al concepto de seguridad que garantiza la integridad y confidencialidad de la información en todo lugar y todo momento.

Por esta razón es necesario realizar sistemas que sean acordes a los nuevos avances tecnológicos y que en su desarrollo tengan en claro las importancia que tiene la conectividad con otros sistemas y la seguridad que implica la confidencialidad e integridad de la información manejada, un producto software ante todo debe de ser seguro en todo momento de su ciclo de vida.

En el presente capítulo se menciona la situación problemática de la División de Gestiones, se plantea los objetivos y justificaciones del presente proyecto como también las limitaciones y alcances que presenta.

En el capítulo dos se describe la metodología utilizada, se hace mención de conceptos importantes que serán aplicados en el desarrollo del presente trabajo, además de las herramientas utilizadas en el sistema.

En el capítulo tres se desarrolla el análisis y diseño del sistema como la aplicación de los conceptos y herramientas mencionadas en el capítulo dos.

En el capítulo cuatro se desarrollan conceptos sobre la calidad del sistema y la aplicación de cada una de las métricas de calidad.

1.2 ANTECEDENTES

La DGAR es una División dependiente de Secretaria General de la UMSA, atiende diferentes trámites que son realizados por todo estudiante universitario en algún momento de su vida académica dentro de la Universidad.

Esta División tiene dos tipos de usuarios¹ los usuarios externos e internos. Se consideran como usuarios externos a: Embajadas de países de convenio con la Universidad, Universidades del Sistema Universitario, Ministerio de Relaciones Exteriores y Culto, y por

2

¹ Se considera usuario a toda persona, unidad, sistema o entidad que utiliza lo servicios o los procesos que realiza la DGAR

último a los Colegios Secundarios. Por otra parte los usuarios internos están constituidos por todas las unidades Académicas y Administrativas de la UMSA.

Los procesos que realiza la DGAR son:

- Modalidad de Graduación (Defensa de Tesis, Defensa de Tesis de Curso de Postgrado, Defensa de trabajo dirigido, Examen de grado, Proyecto de grado, Tesina, Internado rotatorio, Graduación por excelencia).
- Legalización de documentos académicos.
- Admisión de alumnos extranjeros (Nuevos y con traspaso).
- Convenios internacionales
- Readmisión
- Devolución de Documentos.
- Carrera Paralela.
- Traspasos internos (Cambio de carrera)
- Traspasos de la UMSA a Universidades del Sistema.
- Traspasos de Universidades del Sistema a la UMSA.
- Anulación de Traspasos.
- Traspasos del exterior.
- Solicitud de reactualización y desistimiento del trámite.
- Admisiones especiales (Profesionales Universitarios, Militares, Oficiales de Policía, maestro Normalistas y Egresados del Instituto Técnico Industrial Pedro Domingo Murillo).
- Certificación de Alumno Regular.
- Convalidación a Vestibulares.
- Matriculación universitaria.
- Legalización de matrículas.
- Tramitación y viabilización de Matrículas Retroactivas.
- Reconocimiento de curso Pre-Facultativo.
- Prorrogas para la presentación de Diplomas de Bachiller.

Los anteriores procedimientos en su mayoría son realizados de forma manual a excepción del proceso de matriculación, que ya cuenta con un Sistema integrado con todas las unidades académicas de la UMSA.

Cada proceso realizado por la DGAR esta ligado a lo que es un trámite para lo cual se tienen que cumplir ciertos requisitos indispensables y se sigue de forma general los siguientes pasos:

- Registro de datos personales del interesado y la fecha de inicio del trámite solicitado, este registro es realizado en un libro de actas.
- Revisión de los requisitos presentados, si es que faltara alguno de ellos el trámite es suspendido hasta la presentación del requisito faltante.
- Control de forma manual en cada fase del trámite.
- La finalización del trámite casi siempre es con la entrega de un documento al interesado el cual es elaborado de forma manual.

Para poder conocer de mejor manera el tratamiento de los trámites se presenta en la parte de anexos en la sección F diagramas de flujo de datos² que representan de mejor manera lo anteriormente expresado, estos diagramas muestran los procesos que serán automatizados con el Sistema de Información de Gestiones Admisiones y Registros.

En la gestión 1998 se realizó un proyecto de grado titulado *Sistema de Información para Registro y Control de Gestiones* realizado por Germán Lipan Llanos [Lipan,1998], el cual no fue implementado en la DGAR por razones desconocidas, dicho proyecto utiliza una metodología estructurada.

1.3 PLANTEAMIENTO DEL PROBLEMA

Actualmente la DGAR para realizar los trámites que están a su cargo se basa en el Manual de Procedimientos donde se reglamenta cada uno de los requisitos necesarios para cada proceso, además de la forma como se debe de proceder en cada uno de los mismos.

Para cada proceso existe una persona responsable de la realización del mismo se lo denomina con el cargo de Técnico I, sus funciones en forma resumida son:

- Registrar un trámite en un libro de actas.
- Asignarle un número de trámite a la persona interesada con una ficha.
- Revisar que el interesado presente todos los requisitos necesarios.

4

² Un diagrama de flujo de datos representa gráficamente los procesos y flujos de datos en un sistema de negocios.

- Controlar el seguimiento del trámite, esto con la ayuda del número de ficha anteriormente asignado.
- Finalizar el trámite con la elaboración del documento final solicitado por el interesado ó con el que corresponda al proceso elegido.

La mayoría de los procesos de la DGAR son realizados de forma manual por lo que se pierde mucho tiempo en la realización de los mismos, además este factor implica la elaboración de reportes ineficientes y tardíos por no contar con un número adecuado de computadoras, la DGAR solamente cuenta con tres ordenadores que son insuficientes para los ocho Técnicos I que trabajan en la Unidad.

Después de haber realizado la recopilación de datos en la DGAR³ y un estudio de los requerimientos de cada unos de los funcionarios de esta División se ha podido llegar al siguiente planteamiento del problema:

El manejo de los procesos de la DGAR de forma manual genera una enorme pérdida de tiempo, inadecuado seguimiento y control de trámites además de la elaboración de reportes morosos y tardíos.

1.4 OBJETIVOS

1.4.1 Objetivo Principal

Optimizar el seguimiento y control de los trámites que involucran los procesos realizados por la DGAR, para mejorar la calidad del servicio existente hacia los usuarios y controlar de manera automática el seguimiento de los mismos.

1.4.2 Objetivos secundarios

- Mejorar el seguimiento y control de trámites.
- Diseñar una Máquina de Turing para el control de trámites.
- Diseñar una Base de Datos híbrida.
- Elaborar e imprimir reportes de acuerdo a requerimientos en el momento preciso.

5

³ Para esta recopilación de datos se tomo como referencia el levantamiento de campo de la DGAR elaborada anteriormente, las encuestas realizadas se encuentran a detalle en la sección de Anexos G

1.5 JUSTIFICACIÓN

1.5.1 Justificación Económica

- La automatización de los procesos permitirá reducir costos pertinentes a la actualización y mantenimiento dentro del seguimiento y control de trámites.
- La reducción de tiempo permitirá poder dar mayor cabida a usuarios que soliciten la realización de los diferentes trámites, lo que implica mayores ingresos al Tesoro Universitario.

1.5.2 Justificación Social

Implementar un Sistema de Información, se fundamentará en el hecho de poder brindar un instrumento de servicio a la población universitaria de la UMSA.

1.5.3 Justificación Tecnológica

El presente proyecto esta desarrollado bajo la arquitectura Modelo Vista Controlador (MVC)⁴ que es una de las últimas arquitecturas de programación que brinda seguridad ante Hackers u otro tipo de intrusos que puedan ser una amenaza para el manejo de la información.

La tecnología contiene software para el soporte de Base de Datos cuyo gestor es Postgres, bajo la plataforma Linux, teniendo como interfaz páginas web realizadas en JSP bajo la arquitectura Spring.

1.5.4 Justificación Académica

Se lleva a la práctica la teoría de lo que es una Máquina de Turing y muestra la aplicabilidad del mismo en un sistema donde es necesario la toma de decisiones.

6

⁴ El MVC es una arquitectura basada en vistas, modelo y controladores para conocer mejor esta nueva arquitectura vea la sección Anexos H

1.6 LÍMITES Y ALCANCES

1.6.1 Límites

No se desarrolla un módulo para la elaboración de formularios que son necesarios en realización de algunos trámites, por lo que se seguirán utilizando los ya existentes y prediseñados.

1.6.2 Alcances

Los alcances del Sistema de Información de Gestiones Admisiones y Registros son:

- Registro y control automatizado de las diferentes Modalidades de graduación.
- Registro y control automatizado de Devolución de documentos.
- Registro y control automatizado de Readmisiones.
- Registro y control automatizado de Carreras Paralelas.
- Registro y control automatizado de Inscripción de profesionales.
- Registro y control automatizado de Inscripción bajo convenios internacionales.
- Registro y control automatizado de Inscripción bajo convenios nacionales.
- Registro y control automatizado de Inscripción de estudiantes de competencias académicas
- Registro y control de asignación de plazas para casos especiales.
- Registro y control de asignación de tareas a cada usuario para la atención de trámites.
- Reporte de trámites atendidos por cada funcionario de la División.
- Reporte del seguimiento de cada trámite.
- Reporte de trámites finalizados por facultades y carreras de acuerdo a intervalos de tiempo.
- Reporte de orden de canje de matrícula universitaria.
- Reporte de resoluciones para modalidades de graduación
- Reporte de orden para defensa de modalidad de graduación.
- Reporte de historial de trámites realizados por estudiante universitario.

1.6.3 HERRAMIENTAS

Las herramientas que se utilizarán para el desarrollo del sistema son:

HARDWARE:

- Un computador que tenga las siguientes características:
 - Microprocesador mayor a 1Ghz.
 - Memoria Ram de 256Mb
 - Disco duro mayor o igual a 40 Gbyte
 - Tarjeta de video mayor o igual a 64Mb
- Una impresora Laser de última generación.

SOFTWARE

- Visio 2003
- Rational Rouse (para realizar diagramas en el análisis y diseño del sistema)
- Manejador de Base de Datos Postgres.
- Lenguaje de programación JE2DK (Java empresarial)
- Arquitectura Spring.
- Servidor de paginas web Apache Tomcat

MATERIAL DE OFICINA

- 200 hojas bonn tamaño carta.
- 20 unidades de diskets.
- 20 unidades de CD's.
- Una docena de lápices.
- Una docena de bolígrafos.
- Una docena de borradores.

1.7 APORTE TEÓRICO

Se aplicará la teoría de Máquinas de Turing para el desarrollo del sistema ya que el mismo ayudará en la toma de decisiones tanto en el seguimiento y control de cada trámite

En el marco teórico se ve más a detalle la teoría de la Máquina de Turing y la aplicación del mismo en el desarrollo del sistema.

1.8 APORTE PRÁCTICO

El aporte práctico del proyecto es la implementación de una máquina de Turing dentro de la Base de Datos, las transiciones de estado e_0 a un estado e_1 de la máquina estarán detalladas en una tabla de transiciones, donde el alfabeto de la máquina de Turing está formado por todas las etapas existentes dentro de un trámite.

1.9 METODOLOGÍA

Para poder definir los problemas existentes y plantear los objetivos del presente proyecto de grado se ha utilizado el análisis de Marco lógico que involucra los siguientes pasos:

- Análisis de involucrados.
- Análisis de problemas
- Análisis de objetivos
- Análisis de alternativas

Este análisis se encuentra en la parte de Anexos A,B,C Y D del presente documento.

Para el análisis y desarrollo del sistema la metodología que se utilizará es el desarrollo del sistema será el Proceso Unificado de la Racional (RUP)⁵, por que brinda las siguientes características:

- Es iterativo e incremental.
- Es dirigido por los casos de uso
- Centrado en la arquitectura

El ser iterativo e incremental da beneficio de: Obtener un sistema robusto, reducir el riesgo de tener un mal producto, reducir el riesgo de no obtener el producto en el tiempo previsto y permite atacar problemas con requisitos incompletos.

9

⁵ RUP es el Proceso Unificado de la Racional que utiliza herramientas de UML para el análisis y diseño de sistemas

Dirigido por los casos de uso implica que:

- Casos de uso que proporcionan los requisitos funcionales del sistema
- Modelo de casos de uso que describen toda la funcionalidad del sistema
- Filosofía de desarrollo que involucra los modelos de: casos de uso, análisis, diseño e implementación.
- Trazabilidad del sistema que implica cambios en requisitos de un caso de uso fácil de detectar las clases y componentes que afectan.

Centrado en la arquitectura involucra que con el uso de los casos de uso se describe la funcionalidad del sistema, la arquitectura define la forma del sistema y se describe mediante vistas que incorporan el 10% de los casos de uso más relevantes.



CAPÍTULO II

MARCO TEÓRICO

Este capítulo presenta la descripción a detalle de la División de Gestiones Admisiones y Registros, la metodología que se utilizará para el desarrollo del proyecto además de las diferentes herramientas a utilizar.

2.1 INTRODUCCIÓN

Como se mencionó en el capítulo anterior, la División de Gestiones, Admisiones y Registros se encarga de atender trámites relacionados con la admisión, movilidad estudiantil, modalidades de graduación y otros, la atención de los trámites están a cargo de doce personas incluida la Jefe de la División, cada uno de los cuales desempeña una función muy importante.

El principal objetivo de la DGAR es el de brindar un servicio rápido y eficiente a sus usuarios y para ello se desea contar con un sistema informático que pueda cubrir las necesidades de los usuarios en general.

Para realizar tal objetivo se tomara como modelo de desarrollo la metodología RUP cuyas características se detallan en el presente capítulo.

Además se muestra la arquitectura del sistema mediante diagramas que representan aspectos importantes para el desarrollo y conclusión del sistema informático.

2.2 DIVISIÓN DE GESTIONES ADMISIONES Y REGISTROS Y SUS FUNCIONES

Orgánicamente la División de Gestiones Admisiones y Registros depende de Vicerrectorado, la función principal que tiene es el de planificar, organizar, coordinar y efectivizar los procesos administrativos de cada una de las unidades comprendidas dentro la misma de acuerdo a cronogramas y calendarios establecidos por instancias superiores y los

establecidos de acuerdo a la pertinencia de los trámites que se realizan en las diferentes épocas del año académico.

A continuación se muestra el esquema organizacional de la División de Gestiones Admisiones y Registros.

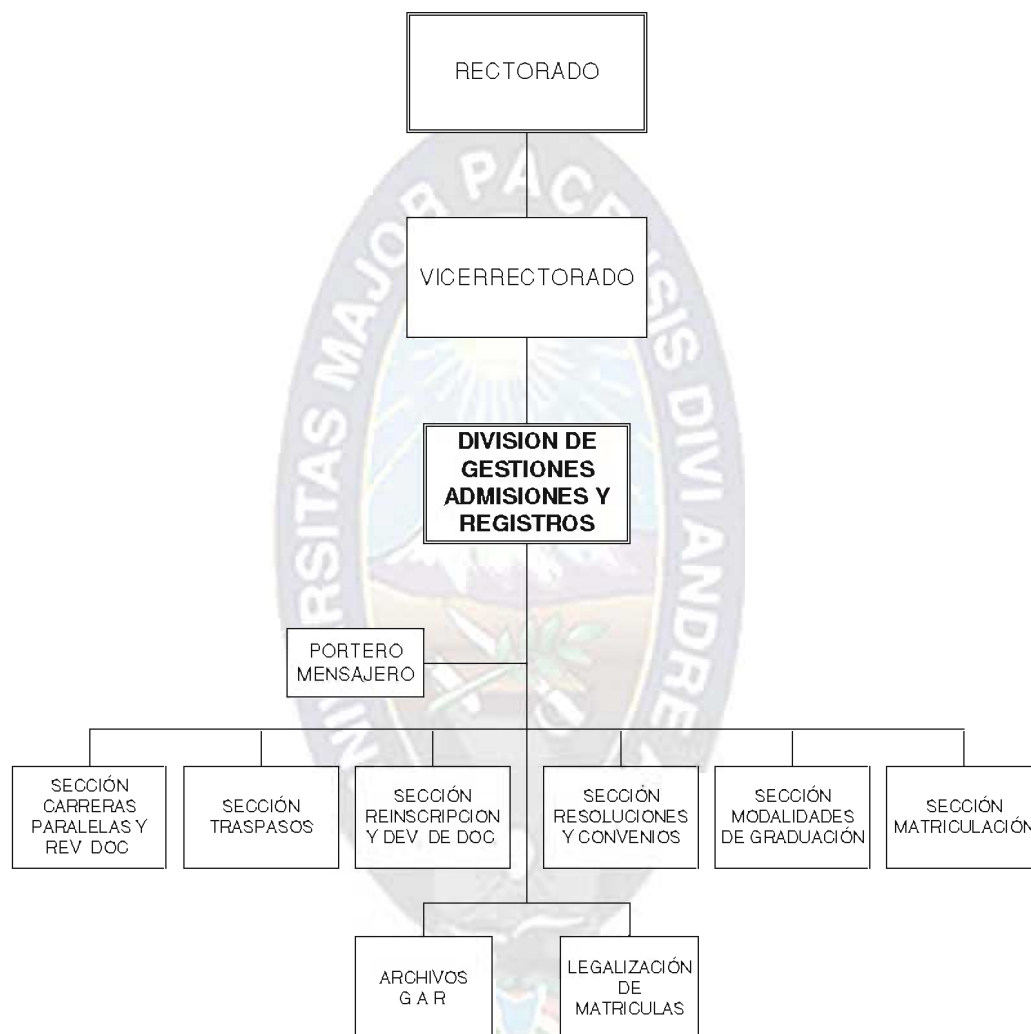


Fig. 1 Organigrama DGAR

Fuente: [Levantamiento de campo DGAR]

Las funciones específicas que tiene la División son:

- Planificar, coordinar y efectivizar el proceso de matriculación de cada gestión académica, de acuerdo a calendario establecido para alumnos nuevos, antiguos observados, egresados y admisiones especiales.

- Recepción y verificación de los documentos en los diferentes trámites como ser, defensa de tesis y proyectos de grado, trabajos dirigidos, exámenes de grado, inscripción de profesionales, reinscripciones, legalizaciones, traspasos, devolución de documentos y otros, en el marco de disposiciones emanadas dentro la UMSA.
- Elaborar resoluciones, legalizaciones de certificados de notas, programas de estudio, cartas, elaboración de informes, extensión de orden de compra de matrícula y legalización de matrícula.

De acuerdo a los alcances del proyecto planteado en el capítulo I a continuación se detallan los procesos que se automatizaran con el sistema.

2.2.1 Proceso de Carreras Paralelas

Para este proceso previamente debe de haber un número de plazas asignadas por cada carrera de la UMSA, ya que la asignación de cupos dependerá de dicho dato.

Una vez que la carrera establece el número de plazas para viabilizar el trámite de carrera paralela, la DGAR publica el calendario para la recepción de solicitudes, publicada la relación de solicitudes admitidas, el interesado deberá cumplir los siguientes requisitos y procedimiento:

Requisitos:

- a) Presentar una carta al Jefe de la División de Gestiones Admisiones y Registros, solicitando carrera paralela mencionando la carrera de origen y la carrera de destino.
- b) Adjuntar fotocopia de título de bachiller, certificado de notas (originales o fotocopias) o seguimiento académico firmado por el director de carrera, plan de estudios con firma y sello de la carrera de origen, fotocopia de la última matrícula universitaria de la carrera de origen.
- c) Los estudiantes egresados presentarán certificado de egreso original y fotocopia.
- d) Haber aprobado por lo menos el 65% del plan de estudio de su carrera origen.
- e) Alcanzar un promedio mínimo general del 60% en las notas aprobadas.
- f) El alumno que hubiera aprobado el 80% o más de las materias del plan de estudios de la carrera de origen queda liberado del requisito del inciso e).

Procedimiento:

El Técnico encargado de carreras paralelas debe de verificar el cumplimiento de los requisitos, si todo esta correcto entonces proporcionará la autorización para compra de timbres universitarios y formulario de carrera paralela al interesado.

Los estudiantes admitidos deberán realizar los siguientes pasos:

- Llenar el formulario hasta la firma del Interesado.
- Llevar los documentos a la carrera origen para el informe correspondiente y recogerá la documentación con la firma del Jefe de Carrera origen.
- Cumplido el paso anterior, el estudiante llevará los documentos a la carrera destino para la firma correspondiente del Jefe de Carrera destino.
- Devolverá la documentación a la DGAR y recogerá la Orden de compra de matrícula, que le habilitará a matricularse en las fechas establecidas para el efecto.
- Una vez concluido el trámite, una copia del formulario quedará con el interesado, otra copia será enviada a la carrera destino y la otra copia junto con los documentos quedará en el archivo de Gestiones.

2.2.2 Proceso de Devolución de documentos

Podrán solicitar la devolución de documentos los estudiantes que fueron matriculados y hubieran aprobado sólo una materia o ninguna. También los estudiantes que aprobaron el examen de dispensación o curso pre-facultativo en una nueva carrera, independientemente del número de materias vencidas.

Requisitos:

- a) Una carta dirigida al Jefe de la División de Gestiones, Admisiones y Registros.
- b) Última matrícula.
- c) Timbres universitarios

Procedimiento:

El trámite será iniciado por el interesado en la DGAR, donde presentará una carta solicitando la Devolución de Documentos con la información oficial y académica.

El personal encargado remitirá la solicitud a la Carrera para su correspondiente informe, si el estudiante hubiera vencido una materia o ninguna, se remitirá a la DGAR el informe y el fólder académico del solicitante, si el estudiante hubiera vencido dos o más materias y no cuente con la aprobación de examen de dispensación o curso pre-facultativo, la carrera emitirá el informe a la DGAR indicando la no procedencia del trámite.

La DGAR, entregará al interesado los documentos personales y un certificado por el cual consta que ha dejado de ser estudiante de la carrera mencionada. Para el efecto, el interesado deberá adjuntar timbres universitarios por el valor establecido en la gestión y devolver la matrícula universitaria que hubiera adquirido en su vida estudiantil.

2.2.3 Proceso de Readmisiones

Un estudiante de la UMSA puede realizar este trámite si hubiera abandonado sus estudios por más de tres años y mínimo tuviera dos materias aprobadas en su carrera y deberá presentar los siguientes requisitos:

Requisitos:

- a) Formulario de reinscripción, previa autorización de la oficina de Gestiones.
- b) Original de la última matrícula o fotocopia legalizada.
- c) Fotocopia del Título de Bachiller.
- d) Certificado de notas de por lo menos dos materias vencidas, o historial académico firmado por el director de carrera. Si el estudiante cuenta con una sola materia vencida deben tener la autorización del director de carrera para proceder a la readmisión.

Procedimiento:

Presentar a la DGAR los documentos indicados anteriormente, en originales y un juego de fotocopias una vez que los requisitos estén completos remitir la documentación a la carrera respectiva, a fin de solicitar el informe de aceptación del trámite.

Con la firma del jefe de carrera retorna el trámite a la DGAR, si el informe es favorable, el interesado recoge la orden de compra de matrícula, que lo habilita a matricularse en las fechas establecidas para el efecto.

2.2.4 Proceso de Modalidades de graduación

Podrán realizar este trámite aquellas personas que hayan culminado sus estudios en la UMSA y quieran realizar su defensa para optar al título de licenciatura o técnico superior.

Los tipos de modalidades que un estudiante puede optar son: proyecto de grado, tesis de grado, internado rotatorio, examen de grado, graduación por excelencia, contador general, bachiller superior, técnico superior, tesina. Cada tipo de modalidad tienen sus respectivos requisitos por ejemplo en el caso de tesis de grado se requieren los siguientes requisitos.

Requisitos:

- a) Fotocopia legalizada del diploma de bachiller.
- b) Certificado de conclusión de estudios original
- c) Informe de Kardex
- d) Un juego de certificado de notas originales
- e) Resoluciones decanales y/o rectorales en caso de convalidación de materias.
- f) Si la convalidación se produjo por traspaso, carrera paralela o admisión especial, adjuntar copia o fotocopias de dicho trámite.
- g) Pensum de materias firmado y sellado por el director de carrera, este documento deberá estar en correspondencia con las materias vencidas por el estudiante.
- h) Tres matrículas universitarias originales.
- i) Fotocopia de carnet de identidad.

- j) Resolución de carrera, acta o nota del director de carrera, aprobando la modalidad de graduación y autorizando su defensa, regularizando su aprobación.

Procedimiento:

Una vez que el interesado presenta todos los requisitos, el personal de la DGAR revisa cuidadosamente la documentación, si todo esta correcto entonces envían la documentación a la carrera del interesado para señalamiento de fecha y hora de defensa. Una vez que se tenga la fecha establecida para la defensa los documentos pasan al decanato de la carrera para que el trámite tenga el visto bueno.

Una vez que el trámite tenga el visto bueno de la carrera y decanato se devuelven a la DGAR para poder realizar la resolución correspondiente, como también el orden de pago para defensa.

2.2.5 Proceso de Admisiones Especiales

Se denomina admisión especial, al procedimiento académico administrativo por el cual las personas que deseen iniciar o continuar estudios universitarios quedan exentas de cumplir con la modalidad de admisión vigente, debido a que por poseer título de licenciatura, técnico superior, graduados del Colegio Militar, Academia Nacional de Policías, Normal Superior, y otros que son resultados de convenios especiales con las Universidades del Sistema⁶

Podrán optar por este trámite los profesionales con Título Académico, Oficiales de la FFAA, Oficiales de Policía, Maestros Normalistas y Titulados de Institutos Técnicos Superiores que cuenten con título en provisión nacional y el convenio con la UMSA como la Escuela Industrial Pedro Domingo Murillo.

También postulantes destacados con competencias académicas, reconocidas por la UMSA, que hubieran obtenido los tres primeros promedios. Los beneficiarios de convenios especiales también tienen derecho a esta modalidad de admisión.

17_____

⁶ Se consideran Universidades del Sistema a todas aquellas universidades que participan del X Congreso Nacional de Universidades

Requisitos:

Postulantes en condición de Profesionales

- a) Formulario de admisión especial.
- b) Fotocopia legalizada del Título de Bachiller.
- c) Fotocopia legalizada del Título Académico.
- d) Certificado de Nacimiento Original.
- e) Fotocopia de carnet de identidad.

Postulantes destacados en competencias

- a) Certificado de competencia.
- b) Resolución de Consejo de Carrera aceptando su ingreso.
- c) Título de bachiller o fotocopia legalizada.
- d) Cédula de identidad.
- e) Certificado de nacimiento original.
- f) Formulario de Admisión Especial gratuito.

Postulantes por convenios nacionales

- a) Carta de la Carrera aceptando su admisión (Técnicos medios).
- b) Resolución de la instancia correspondiente que autoriza y aprueba el ingreso del postulante (Convenio San Andrés Olimpiadas).
- c) Cédula de identidad en fotocopia legalizada
- d) Certificado de nacimiento original.
- e) Formulario de solicitud de Admisión Especial.

Postulantes por convenio Municipal mejores estudiantes de colegios de cada distrito municipal (Admisiones Directas Beca Provincial)

- a) Resolución del Honorable Consejo Universitario, aprobando la nómina de postulantes aceptados.
- b) Diploma de Bachiller o compromiso de entrega del mismo.
- c) Cédula de identidad original.

Postulantes por convenios internacionales

- a) Carta de selección otorgada por la Embajada respectiva.
- b) Carta de solicitud de la Chancillería.

- c) Documento de identidad.
- d) Certificados de estudios de la vida escolar originales.
- e) Certificado de nacimiento original.

Procedimiento:

Una vez que el postulante cuente con todos los requisitos, la aceptación a la plaza respectiva se procederá a la recepción o revisión de los documentos para otorgarles la orden de compra de matrícula previo cumplimiento de todo el procedimiento académico administrativo que se sigue en cada caso.

La selección y aprobación de los alumnos ganadores en Olimpiadas se realizará en las respectivas Facultades, tomando en cuenta que pueden existir más de tres personas seleccionadas por haber alcanzado el mismo promedio.

Las admisiones directas postulantes por Convenio Municipal mejores estudiantes de colegios de cada distrito municipal no requerirán de la orden de canje de matrícula con la resolución del Honorable Consejo Universitario procederán a la matriculación liberados del pago de valores.

2.3 METODOLOGÍA RUP

La metodología RUP captura varias de las mejores prácticas en el desarrollo moderno de software de forma que sea aplicable para un amplio rango de proyectos y organizaciones, es una guía de cómo utilizar de manera efectiva UML. Consta de las siguientes características:

- Dirigido por casos de uso
- Centrado en la arquitectura
- Es iterativo e incremental

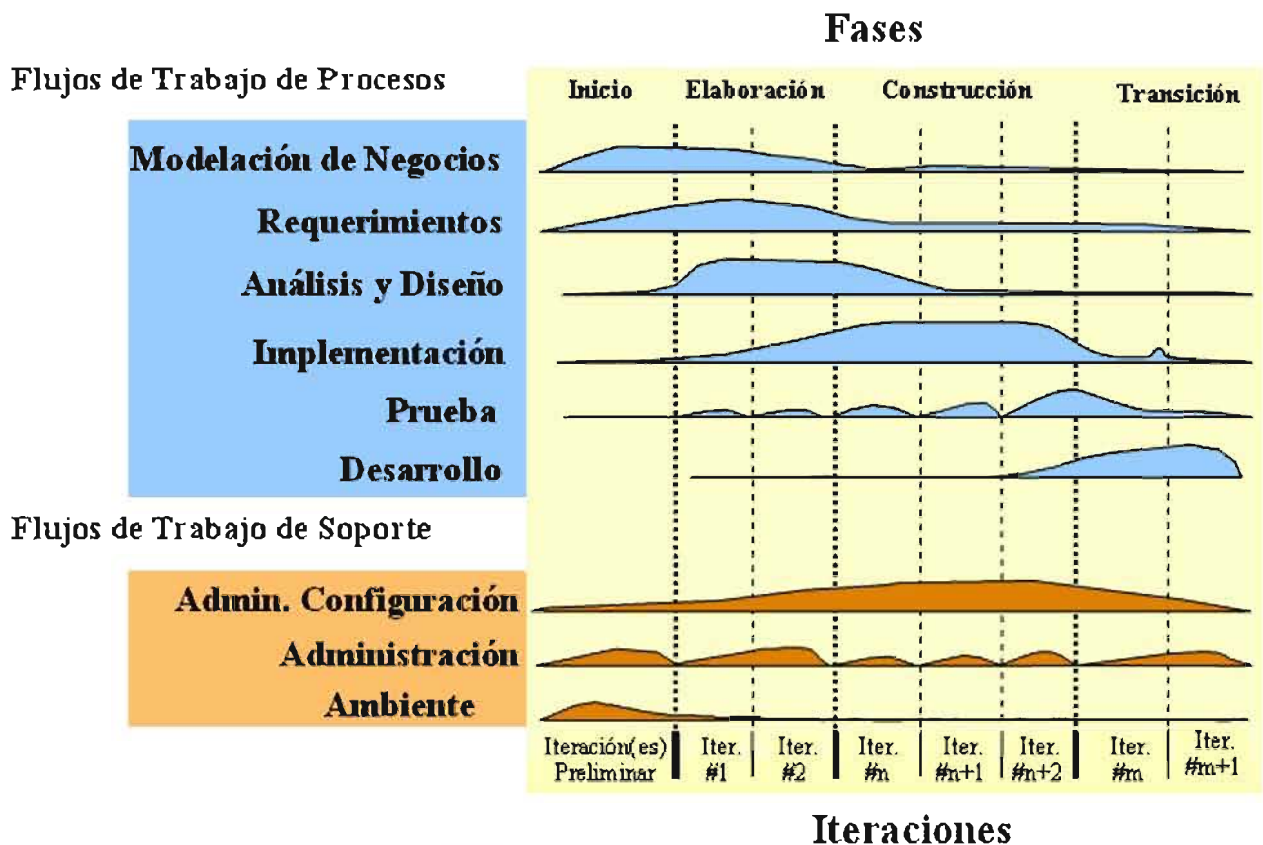


Fig. 2 Metodología RUP

Fuente: [RUP-PDF]

2.3.1 Características de la metodología RUP

2.3.1.1 Dirigido por casos de uso

Utiliza casos de uso que proporcionan los requisitos funcionales del sistema. El modelo de casos de uso que describe toda la funcionalidad del sistema. La filosofía de desarrollo que involucra los modelos de: casos de uso, análisis, diseño e implementación. La trazabilidad del sistema que implica cambios en requisitos de un caso de uso fácil de detectar las clases y componentes que afectan [Jacobson, 2001].

2.3.1.2 Centrado en la arquitectura

Involucra que con la utilización de los casos de uso, se describa la funcionalidad del sistema, la arquitectura define la forma del sistema y se describe mediante vistas.

2.3.1.3 Iterativo e incremental

El ser iterativo e incremental da beneficio de obtener un sistema robusto, reducir el riesgo de tener un mal producto, reducir el riesgo de no obtener el producto en el tiempo previsto y permite atacar problemas con requisitos incompletos.

2.3.2 Fases de la metodología RUP

La metodología RUP consta de cuatro fases que son importantes para el desarrollo del producto:

- Inicio.
- Elaboración.
- Construcción.
- Transición

2.3.2.1 Fase de inicio

Se realiza un estudio de las necesidades de la empresa u organización, ver si es viable o no el proyecto, definir las alternativas, analizar los riesgos y las oportunidades que existen. Se definen los objetivos del proyecto, se hace una estimación aproximada del costo que implicara el proyecto y la duración que el mismo tendrá.

Además en esta fase de inicio se debe obtener los primeros requerimientos, se planifica la primera iteración, se definen los casos de uso y se identifican los riesgos

Se diseña el documento Visión ver Anexos I y un prototipo del sistema a desarrollar, debe de estar bien fundamentada para garantizar la entrada en la fase de elaboración [Jacobson, 2001].

2.3.2.2 Fase de elaboración

En esta fase se definen la visión del producto y su arquitectura, se expresan con claridad los requisitos del sistema que son priorizados y utilizados para crear una sólida base arquitectónica, además se planifican las actividades y los recursos necesarios.

2.3.2.3 Fase de construcción

En esta fase se construye el producto mediante una serie de iteraciones incrementales, se lleva el software desde una base arquitectónica ejecutable hasta su disponibilidad para la comunicación de usuarios.

2.3.2.4 Fase de transición

En esta fase el producto es puesto en manos de los usuarios, en esta fase se puede considerar iniciar de nuevo un ciclo vale decir volver a empezar desde la primera fase hasta la cuarta considerando las observaciones y modificaciones a realizarse en el desarrollo del producto final.

2.4 ARQUITECTURA SPRING

Spring es un arquitectura en java que mantiene una solución para los sistemas robustos construyendo las aplicaciones de forma organizada, proporciona un armazón de MVC y consta de siete módulos como muestra la siguiente figura:

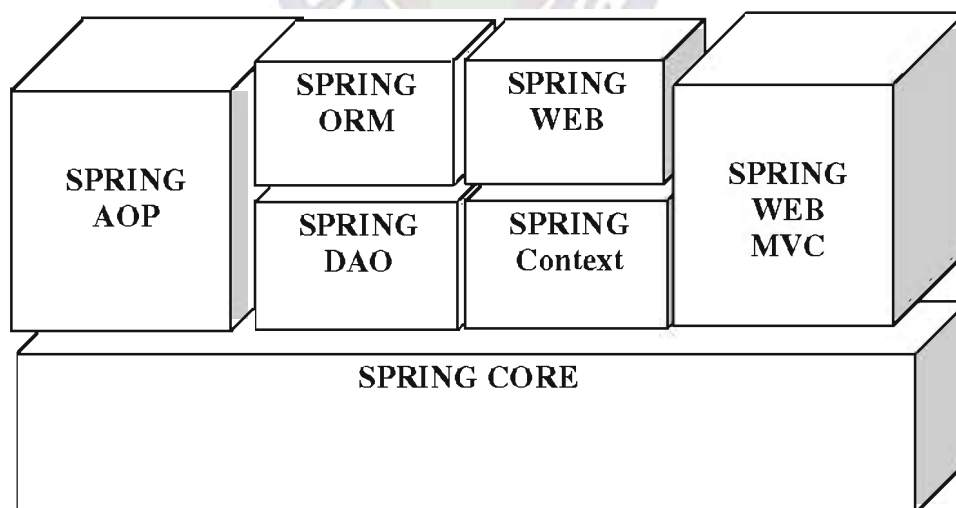


Fig. 3 Módulos del Spring

Fuente: [SPR-PDF]

Cada uno de los módulos componentes de Spring cumplen una función específica e interactúan unos con otros con el fin de lograr un sistema robusto y hacer más sencillo el uso del J2EE [Spring-PDF].

El Spring proporciona las siguientes características:

- Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control. Esto hace que la configuración de aplicaciones sea rápida y sencilla. Los beans pueden ser usados en cualquier entorno, desde applets hasta contenedores J2EE.
- Una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción accesibles y haciendo sencilla la demarcación de transacciones sin tratarlas a bajo nivel. El soporte de transacciones Spring no está ligado a entornos J2EE.
- Una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones, simplifica el manejo de errores y reduce considerablemente la cantidad de código necesario.
- Integración con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementación DAO y estrategias con transacciones, especial soporte a Hibernate. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO.
- Funcionalidad AOP totalmente integrada en la gestión de configuración de Spring. Se puede aplicar AOP a cualquier objeto gestionado por Spring, añadiendo aspectos como gestión de transacciones declarativa.
- Un framework MVC, construido sobre el núcleo de Spring, es altamente configurable vía interfaces y permite el uso de múltiples tecnologías para la capa vista como puede ser JSP, iText, Tiles. Una capa modelo realizada con Spring puede ser fácilmente utilizada con una capa web basada en cualquier otro framework MVC, como Struts, WebWork o Tapestry.

2.5 ESTÁNDAR DE IEEE 830

El estándar de IEEE 830 es un documento de requisitos donde se muestra el propósito, el ámbito, definiciones y referencias del sistema. Además de las perspectivas y características del producto, sin olvidar también las características de los usuarios del sistema.

PROPOSITO

El propósito especifica a quien va dirigido el documento, define los requerimientos y quienes interactuarán en el sistema.

ÁMBITO DEL SISTEMA

En esta parte del documento tiene como objetivo:

- Se define el nombre del sistema
- Se explicará la funcionalidad del sistema y las acciones que no podrá realizar.
- Se describen los beneficios que se obtendrán con el sistema.
- Describir los objetivos y metas que se pretender alcanzar con el desarrollo del sistema.
- Se elaborará la especificación de requisitos tanto de hardware como de software del sistema.

DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES

En esta parte del documento se describen todas las definiciones que se utilizaran en la elaboración del documento como también las abreviaciones y acrónimos.

REFERENCIAS

Para que la documentación este completa en necesario referenciar toda la documentación utilizada para su elaboración.

VISIÓN GENERAL DEL DOCUMENTO

Hace una descripción general de la documentación elaborada, describe de forma breve los contenidos y la organización del documento.

DESCRIPCIÓN GENERAL DEL SISTEMA

Se describen todos los factores que afectan al sistema como también los requisitos del sistema, se deben de detallar los siguientes puntos que son importantes:

- Las perspectivas del producto.
- Funciones del producto.
- Las características de los usuarios como también sus restricciones.
- Factores que se asumen como también los futuros requisitos que serán necesarios.

PERSPECTIVA DEL PRODUCTO

Las perspectivas del producto son importantes, en esta parte del documento se ve la relación del producto con otro producto mayor si así existiera.

Se verá la relación del sistema con otros sistemas siempre y cuando el sistema sea un módulo componente de otro sistema mayor se identificarán las interfaces que existen entre los subsistemas.

FUNCIONES DEL PRODUCTOS

En esta parte del documento se detallaran las funciones que tendrá el producto de forma resumida.

Al detallar las funciones se debe ver con una perspectiva futura del sistema en beneficio de los usuarios.

CARACTERÍSTICAS DE LOS USUARIOS

Se debe de detallar las características de los usuarios como ser el nivel educacional, experiencia de trabajo y experiencia técnica.

RESTRICCIONES

Esta parte del documento muestra las restricciones y limitaciones que el producto tendrá en su desarrollo, estas pueden ser de hardware como de software.

El documento de descripción del estándar IEEE 830 esta detallado en la parte de Anexos J

2.6 MÁQUINAS DE TURING

2.6.1 Definición

Una máquina de Turing es un autómatas que se mueve sobre una secuencia lineal de datos. En cada instante la máquina puede leer un solo dato de la secuencia (generalmente un carácter) y realiza ciertas acciones en base a una tabla que tiene en cuenta su **estado** actual (interno) y el último dato leído. Entre las acciones está la posibilidad de escribir nuevos datos en la secuencia; recorrer la secuencia en ambos sentidos y cambiar de "estado" dentro de un conjunto finito de estados posibles [Kelley, 2000].

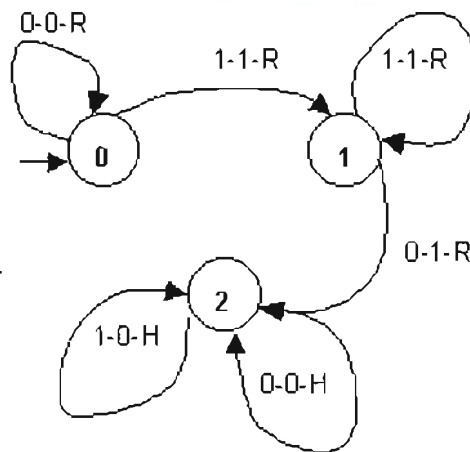


Fig. 4 Esquema de una máquina de Turing representada por un grafo

Fuente: [Máquinas de Turing - PDF]

La máquina de Turing es más una abstracción matemática que un dispositivo físico o mecánico. El hecho que se le denomine "máquina" se debe a que su funcionamiento puede ser descrito en términos de operaciones individuales muy sencillas que sugieren una implementación real muy simple.

2.6.2 Elementos de una Máquina de Turing

Una máquina de Turing consta de:

- Un alfabeto definido y finito denominado **alfabeto de la máquina**,
- Una cinta sobre la que puede desplazarse a izquierda y derecha, contiene una serie de celdas, y en cada una de ellas puede escribirse un símbolo del alfabeto.
- Un cabezal de lectura/escritura, que nos permite el desplazamiento de izquierda a derecha dentro de la cinta puede leer o escribir en cada celda.
- Tiene **registro de estado** que almacena el estado de la máquina. El número de estados posibles es finito, y no se exige ningún estado especial con el que sea iniciada la máquina.
- Consta además de una **tabla de acción**, que contiene las instrucciones de lo que hará el autómata. Estas instrucciones representan en cierta forma el "programa" de la máquina. La ejecución de cada instrucción de la tabla de acción incluye cuatro pasos:
 - ❖ Leer un carácter en la posición actual.
 - ❖ Escribir un nuevo símbolo en esta posición (puede ser el mismo que había). El símbolo a escribir es alguno del alfabeto de la máquina, y depende del carácter leído y del estado actual.
 - ❖ Desplazar el cabezal una celda a derecha o izquierda; en algunos modelos el desplazamiento puede detenerse.
 - ❖ Decidir cual será el nuevo estado en función del carácter que se acaba de leer y del estado actual. Si la tabla de acción no contiene ninguna correspondencia con el estado actual y el símbolo leído, entonces la máquina detiene su funcionamiento.

En modelos didácticos computarizados la tabla suele definirse mediante una matriz de cinco columnas que contiene: estado, carácter leído, carácter a escribir, movimiento y nuevo-estado.

Para ejemplificar a continuación se muestra el diseño de la tabla para la máquina de Turing que se muestra en la Fig. 4.

ESTADO ACTUAL	CARÁCTER LEIDO	CARÁCTER ESCRITO	DIRECCION DEL MOVIMIENTO	NUEVO ESTADO
0	0	0	R	0
0	1	1	R	1
1	0	1	H	2
1	1	1	H	1
2	0	0	H	1
2	1	0	H	2

Tabla 1: Tabla de acción de la Máquina de Turing

Fuente: [Elaboración propia]

2.6.3 La Máquina de Turing como aplicación

Es notable que el diseño de una Máquina de Turing contiene de forma implícita la idea de que el autómata puede alterar su propio programa, pero el punto más significativo de su filosofía de funcionamiento es que se comporta como la mente, en el sentido que la configuración interna de la máquina establece el entorno en el que se toman las decisiones, de forma que la acción depende de dos factores: el estado interno y la información externa que puede ver a través de su cabezal

La consecuencia es que es imposible predecir su comportamiento de la simple inspección de su tabla de acción, ya que el comportamiento depende también de la entrada recibida.

Es significativo que la cinta puede extenderse indefinidamente a derecha e izquierda, lo que hace que en la práctica sea imposible construir un modelo real de lo que se denomina un sistema de Turing completo. Es también destacable que la máquina da a la cinta tres utilizaciones distintas:

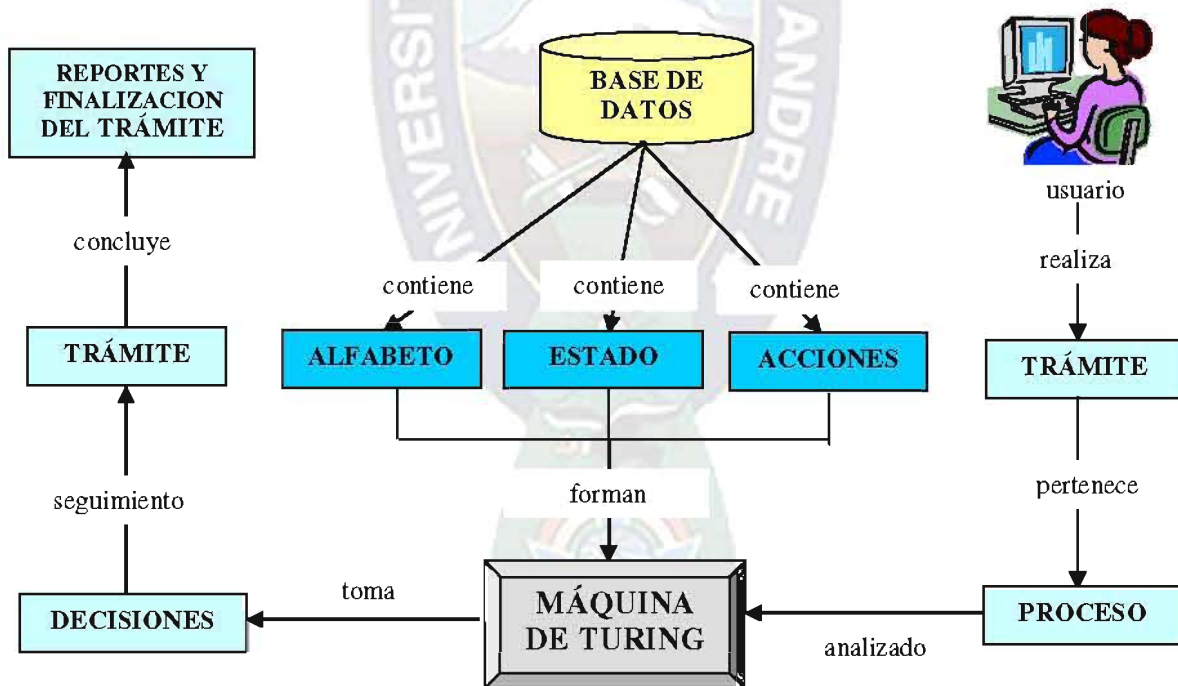
- Como elemento de almacenamiento de los datos de entrada (de capacidad potencialmente ilimitada)
- Como elemento de salida (de cualquier cantidad de datos)
- Como almacenamiento de información intermedia durante el proceso (puede ser de cualquier tamaño). son finitos, lo que confiere su potencia a la máquina de Turing [Turing-PDF].

2.6.4 Forma de implementación de la máquina de Turing en el proyecto

De acuerdo a la definición anteriormente escrita de lo que es una máquina de Turing, y llevando a práctica la teoría se diseñó una tabla de estados en la Base de Datos del sistema, en el cual se detallan los estados y las acciones que se tendrán en los diferentes trámites inmersos en los diferentes procesos. Esta tabla de estados permite que la máquina de Turing pueda tomar decisiones sobre los diferentes pasos existentes en un determinado trámite.

El hecho de tomar decisiones sobre el tratamiento de los trámites implica que la máquina de Turing inmerso en la base de datos constituya una base de datos de conocimiento ya que determina el funcionamiento del sistema.

El siguiente esquema muestra gráficamente el funcionamiento de la máquina de Turing en el sistema.



De acuerdo al gráfico anterior un usuario realizara los diferentes trámites pertenecientes a los procesos utilizando una Máquina de Turing el cual esta representado dentro la base de datos el cual contiene tablas que representan el alfabeto, estado y acciones de la máquina.

La Máquina de Turing toma decisiones sobre los diferentes pasos existentes de cada trámite pasando de un estado a otro dependiendo la decisión tomada. Dependiendo de un estado anterior y una acción anterior dependerá el estado siguiente como la acción siguiente.

La decisión que vaya a tomar la máquina de Turing en un estado particular será esencial para la realización de la siguiente acción que corresponda al trámite.

Cada trámite tiene un estado inicial y un estado final por lo cual la Máquina de Turing sabe cuando empezar un proceso de seguimiento de trámite como cuando terminarlo, eso permite a los usuarios un seguimiento estricto de los trámites existentes.

2.7 WORKFLOWS

Workflow es una palabra en inglés que significa flujo de trabajo, es el estudio de los aspectos operacionales de una actividad de trabajo como ser: cómo se estructuran las tareas, cómo se realizan, cual es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

Generalmente los problemas de flujo de trabajo se modelan con redes de Petri. Una aplicación de Flujos de Trabajo (WorkFlow) automatiza la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo

Un flujo de trabajo consiste en la automatización de los procesos del negocio, en la que la información pasa de una instancia a otro controlada por reglas o procedimientos.

Entre las ventajas podemos encontrar son: Automatización del negocio lo que hace eficiente las operaciones pues elimina pasos innecesarios, gerencia mejorada de los procesos del negocio, alcanzados con estandarizar métodos de funcionamiento y la disponibilidad de los rastros de intervención, mejoras en el servicio al cliente, adecuación y flexibilidad a las necesidades cambiantes del negocio.

Es importante destacar este concepto ya que de alguna manera el sistema controlará el seguimiento de trámites, pero cada trámite involucra un flujo de trabajo, por lo que se aplica indirectamente workflows en el desarrollo del sistema.

2.8 BASE DE DATOS DEL SISTEMA

Una base o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso, es un conjunto de información relacionada que se encuentra agrupada o estructurada.

En el sistema se utilizará dos tipos de base de datos una que nos servirá para el almacenamiento de información de los diferentes trámites y otra que será una base de datos de conocimiento ya que determinará el funcionamiento del sistema

2.8.1 Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- **Encapsulación:** Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia** Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo:** Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos

invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado.

En una base de datos orientada a objetos, la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS, object database management system).

Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o mas lenguajes de programación a los que de soporte, usan exactamente el mismo modelo que estos lenguajes de programación, son una buena elección para aquellos sistemas que necesitan un buen rendimiento en la manipulación de tipos de dato complejos, proporcionan los costes de desarrollo mas bajos y el mejor rendimiento cuando se usan objetos gracias a que almacenan objetos y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos, al almacenar exactamente el modelo de objeto usado a nivel aplicativo, lo que reduce los costes de desarrollo y mantenimiento.

2.8.2 Bases de datos relacionales

Muy ajeno a la revolución que supuso la orientación a objetos, el área de las bases de datos sigue fiel a un modelo antiguo pero que ha probado su eficacia, el llamado "modelo relacional" [Palasi, 2006].

El modelo relacional se convierte en un estándar prácticamente universal para el acceso a datos, dominando totalmente el área de las bases de datos hasta la actualidad.

El modelo relacional es muy diferente del modelo orientado a objetos. Por una parte, el modelo relacional sólo se ocupa de la parte estática de la aplicación y no de la parte dinámica.

Hay otras diferencias entre los dos modelos. Por ejemplo, la forma en la que el modelo relacional trata los datos es muy diferente a cómo lo hace el modelo orientado a objetos. Mientras en este último, los datos son modelados en forma de objetos, en el modelo relacional son modelados como registros, los cuales son una serie de datos pertenecientes a

una misma entidad de la vida real. Un registro difiere de un objeto en que sólo modela datos y que éstos no tienen estructura. Los registros similares se agrupan en tablas, el modelo relacional no refleja la estructura de la realidad.

2.8.2.1 Documentos XML

XML es una abreviación de Xtensible Markup Language que quiere decir Lenguaje Extensible de Marcas, el crear un documento XML permite que podamos ordenar la información en forma de árbol donde existe un nodo padre y varios hijos.

Escribir un documento XML es muy sencillo por ejemplo para crear un documento XML del proceso de Readmisiones se tiene que crear de la siguiente manera:

```
<?xml version = "1.0">  
  
<readmisiones>  
  
    <nombres>ALFREDO MARCOS</nombres>  
  
    <apellidos>FLORES DURAN</ apellidos >  
  
</readmisiones>
```

En el primer renglón se especifica la versión XML que tiene el documento y en el segundo se utiliza un elemento padre que contendrá dentro los elementos hijos, cada elemento esta en medio de dos tags una de inicio y otra de fin en el ejemplo anterior el nodo padre es readmisiones que tiene como nodos hijos a nombres y apellidos, para acceder a la información de cada elemento se utiliza el XPath que es un medio para seleccionar partes individuales de un documento XML.

En la base de datos del sistema se esta utilizando documentos XML para guardar información de los datos necesarios para un determinado tramite como ser número de trámite, nombre del interesado, carrera, facultad, requisitos presentados y otros.

Para el manejo de archivos XML se utilizó JDOM que es un componente de Java que genera un objeto árbol en memoria que representa al XML; el mismo utiliza un parser que es un analizador léxico semántico que traduce e interpreta al archivo XML.

2.8.2.2 Motor de Persistencia ORM

Si una aplicación esta formada por un programa y una base de datos que se comunican entre si. El programa suele estar diseñado según el modelo orientado a objetos y, por lo tanto, trabaja con datos en formato de objetos. Por el contrario, la base de datos esta diseñada según el modelo relacional y, por lo tanto trabaja con datos en formato de registros, esto introduce una dificultad importante, porque los dos formatos de datos objetos y registros son incompatibles entre sí, la solución más obvia a este problemas es hacer que uno de los componentes hable el idioma del otro. Se debe encontrar un traductor que sepa traducir de un idioma al otro, el la programación este traductor es un componente de software denominado Motor de persistencia

Un motor de persistencia traduce entre los dos formatos de datos de registros a objetos y de objetos a registros, un motor de persistencia reduce el código fuente en un 40% haciéndola menos costosa para desarrollar [Palasi, 2006].

El ORM es un método de gran alcance para diseñar modelos de la base de datos en el nivel conceptual, donde el uso se describe en los términos entendidos fácilmente por los usuarios no técnicos. En la práctica, los modelos de los datos de ORM capturan a menudo más reglas de negocio, y son más fáciles de validar y de desarrollarse que modelos de los datos en otros acercamientos.

2.8.3 Bases de datos de conocimiento

Una base de datos de conocimiento es donde se almacena el conocimiento en el campo a desarrollar.

Los conocimientos que se pueden almacenar son de cuatro tipos:

- **Conocimiento objetivo:** que describe la situación real del sistema.
- **Conocimiento de sucesos:** relacionado con los sucesos que ocurren en el tiempo de ejecución.
- **Conocimiento del funcionamiento del sistema:** de cómo se hacen las cosas
- **Metaconocimiento:** que relaciona los anteriores

La base de datos aplicada al sistema como base de conocimiento almacena la tercera clasificación ya que es la que determina el funcionamiento del sistema al momento de realizar el seguimiento de un trámite.



CAPÍTULO III

MARCO APLICATIVO

En este capítulo se presenta el análisis y diseño del proyecto mediante diagramas que esquematizan al mismo, además de definiciones que son importantes para su desarrollo, como también la aplicación de las máquinas de Turing en el sistema.

3.1 INTRODUCCIÓN

Todo sistema informático debe de ser analizado para desarrollarlo y para ello es necesario tener un modelo en el cual se pueda basar todos los pasos del análisis.

Después de hacer el análisis es necesario poder diseñar el sistema de acuerdo a los requerimientos precisados después del análisis realizado.

El análisis y diseño de este sistema se hará bajo la metodología RUP que utiliza como herramienta de descripción al UML⁷.

La metodología RUP comprende las siguientes etapas

- Modelado del Negocio
- Modelado de Requisitos
- Modelado del Análisis
- Modelado del Diseño

La siguiente figura muestra de forma gráfica la relación que tienen las diferentes etapas de la metodología RUP.

36 _____
⁷ UML Lenguaje Unificado de Modelado

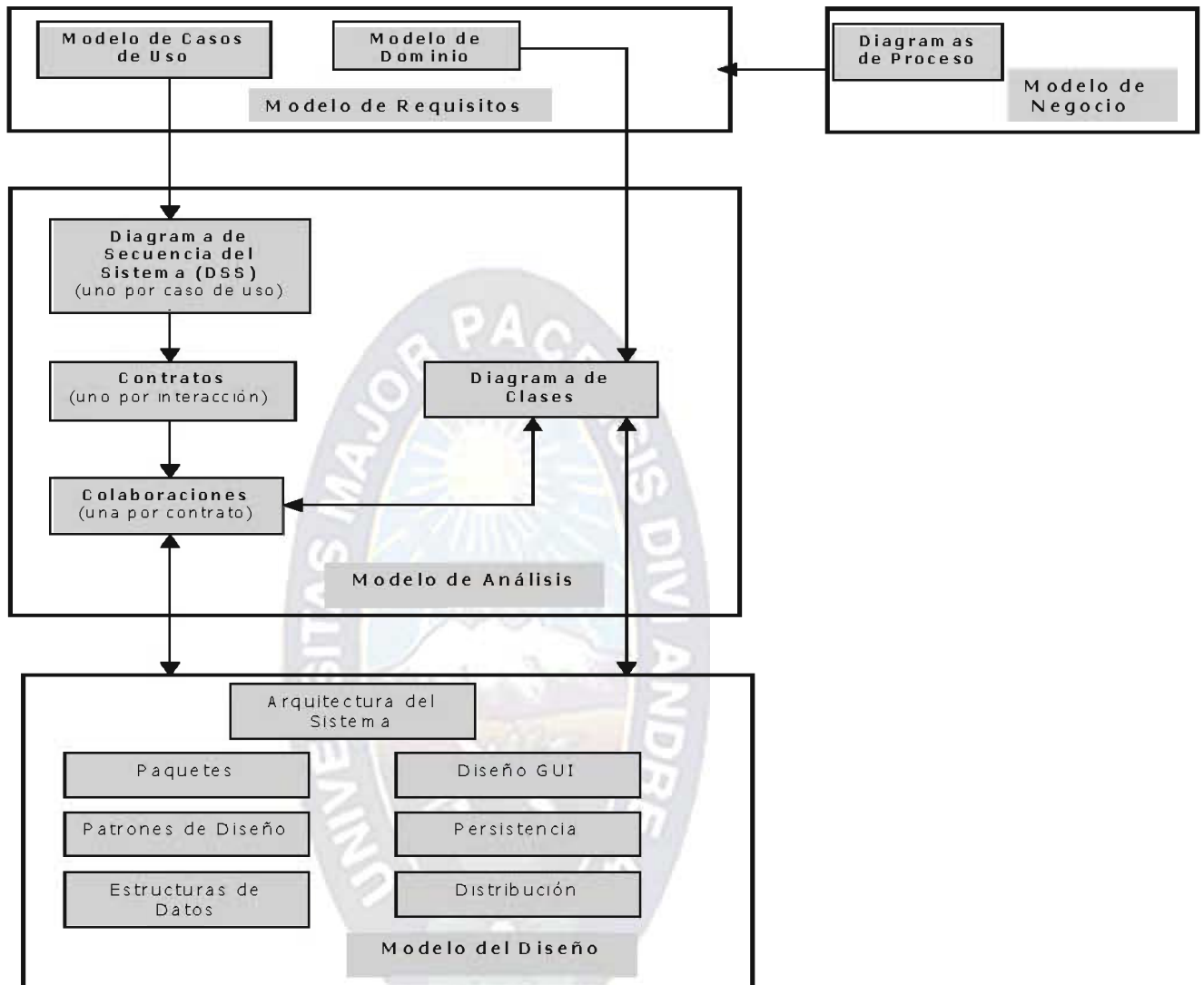


Fig. 6 Etapas de la metodología RUP

Fuente: [Proceso - Larman]

3.2 MÁQUINAS DE TURING DE LOS PROCESOS

En el capítulo del Marco Teórico se dio la definición de lo que es una máquina de Turing, como el sistema de encarga del seguimiento y control de trámites es necesario que dentro de un trámite se pueda tomar decisiones para dar curso a un trámite determinado.

De acuerdo al análisis realizado dentro la DGAR y de acuerdo a los requerimientos de los usuarios se pudo esquematizar los procesos mediante la máquina de Turing de la siguiente manera:

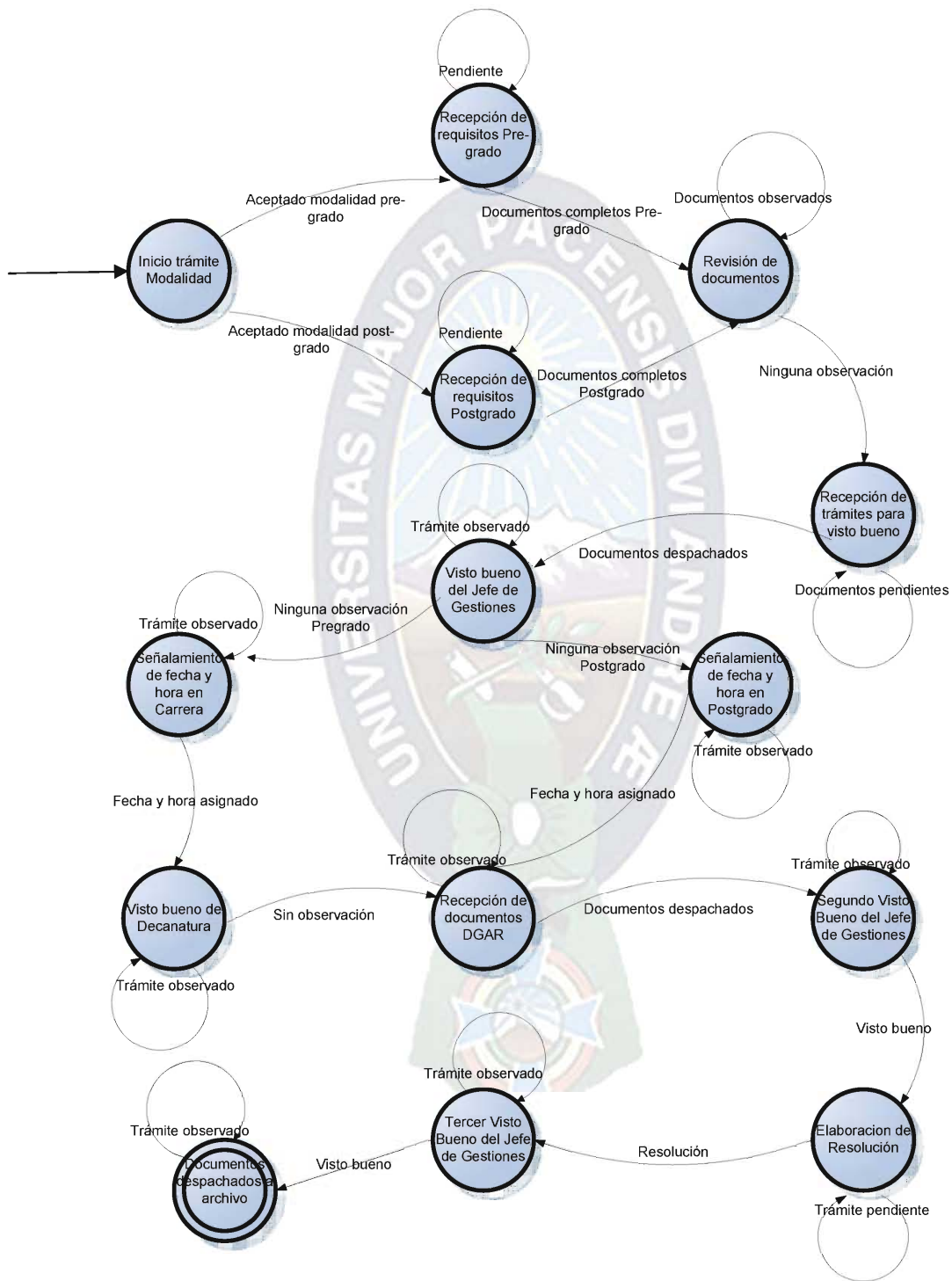


Fig.7 Máquina de Turing para Modalidades de Graduación

Fuente: Elaboración Propia

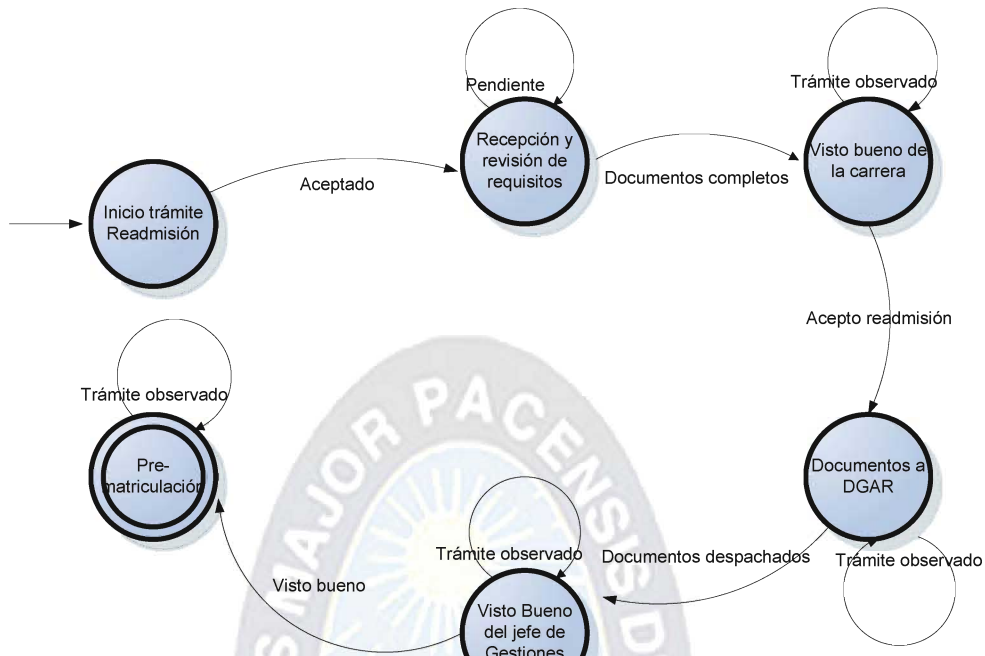


Fig. 8 Máquina de Turing para Readmisiones
Fuente: Elaboración Propia

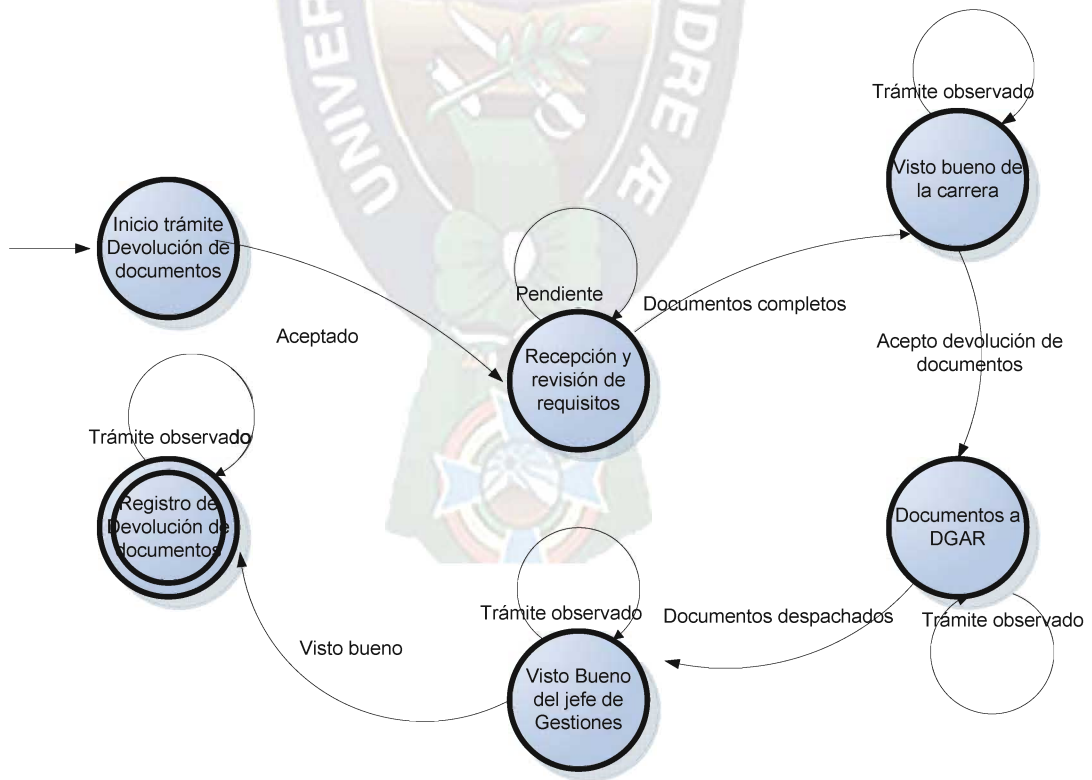
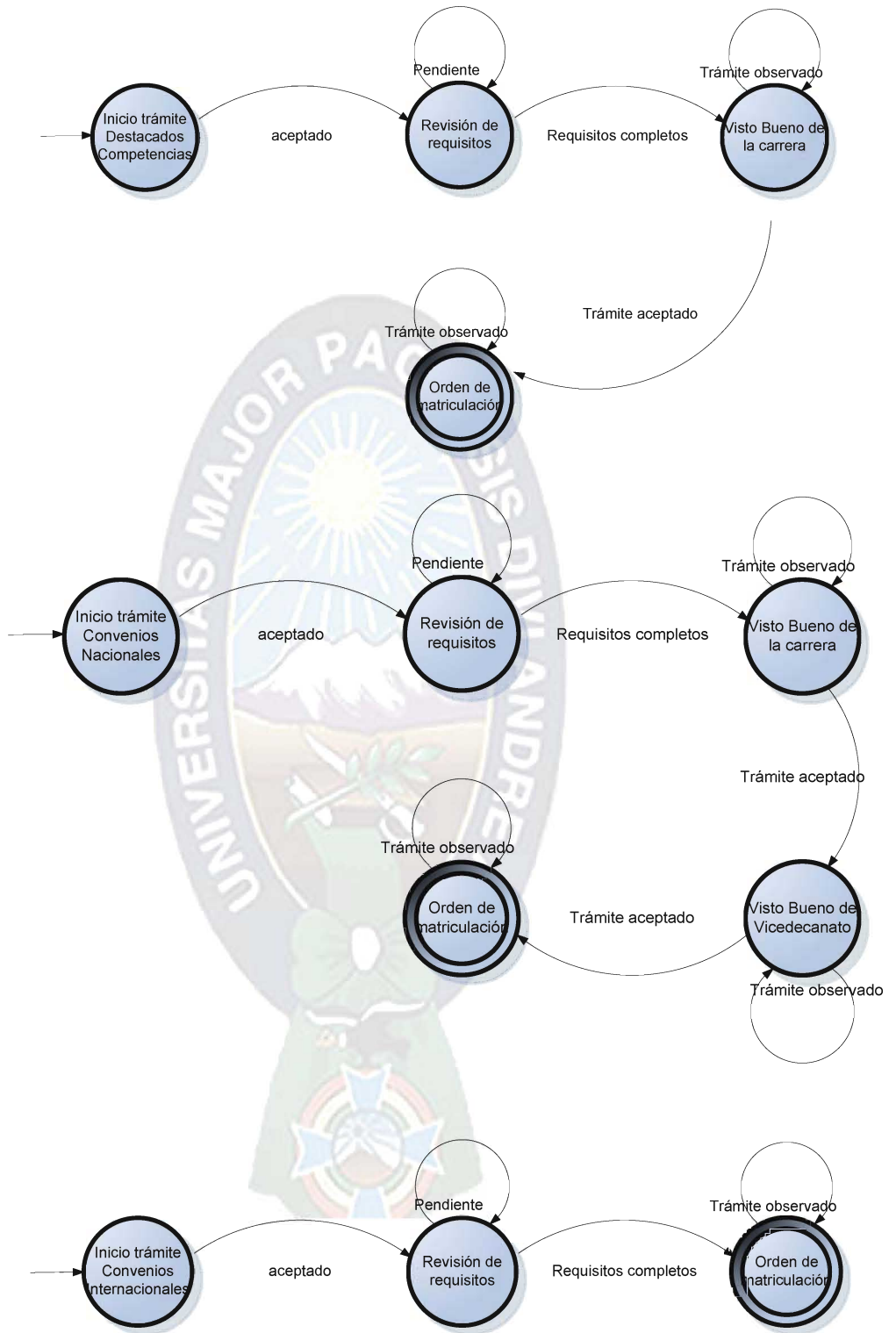


Fig. 9 Máquina de Turing para Devolución de documentos
Fuente: Elaboración Propia



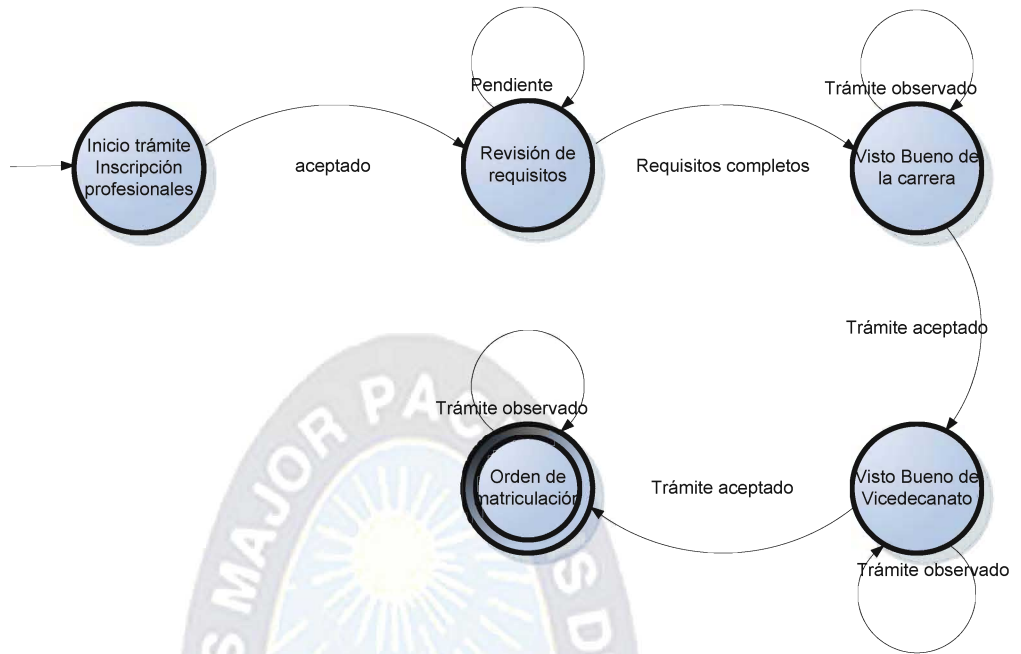


Fig. 10 Máquinas de Turing para Admisiones especiales
Fuente: Elaboración Propia

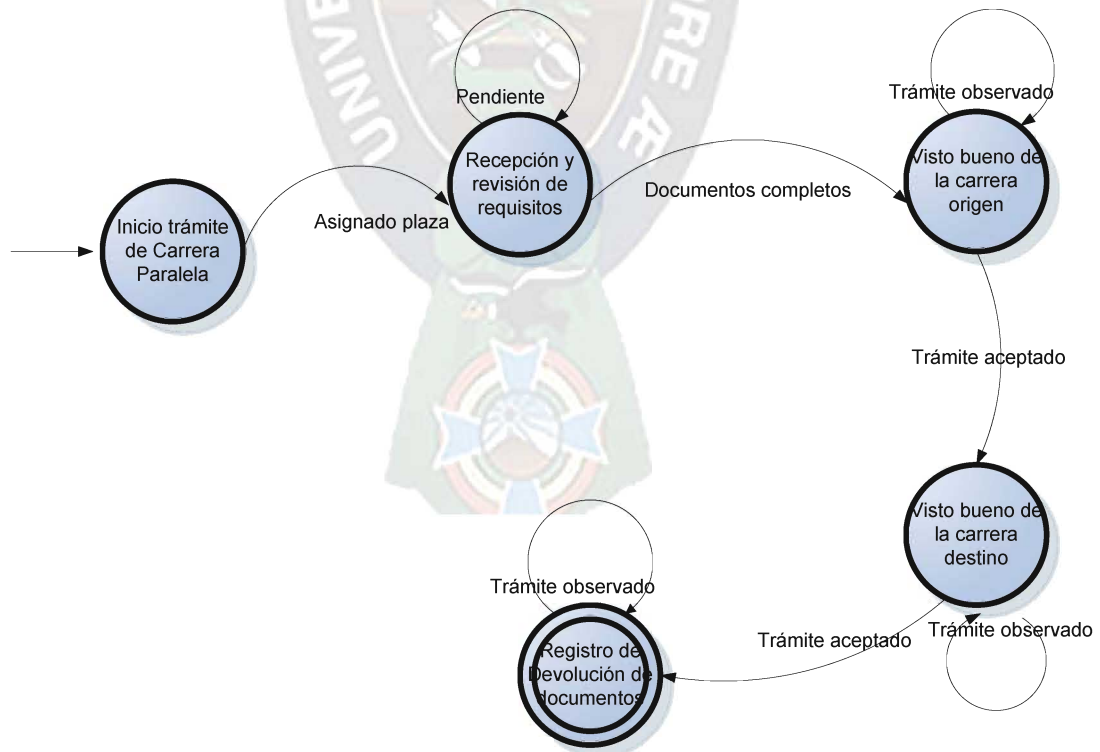


Fig. 11 Máquina de Turing para Carreras paralelas
Fuente: Elaboración Propia

La implementación de la máquina de Turing dentro de la base de datos implicó el desarrollo de una tabla de acción, esta tabla tiene el objetivo principal de controlar el seguimiento del trámite.

Dado que el seguimiento de trámites se la realiza de forma secuencial, la máquina de Turing solamente se desplaza hacia la derecha, ya que ningún trámite tiene la necesidad de retroceder hacia la izquierda como se muestra claramente en los gráficos anteriores. A continuación se muestra la tabla de acción para inscripción de profesionales, la máquina de Turing se muestra gráficamente en la figura 10.

ESTADO ACTUAL	CADENA LEIDA	CADENA ESCRITA	NUEVO ESTADO	DIRECCION
Inicio	Inicio tramite inscripciones profesionales	Revisión de requisitos	Aceptado	Derecha
Aceptado	Revisión de requisitos	Visto bueno de la carrera	Requisitos completos	Derecha
Aceptado	Revisión de requisitos	Revisión de requisitos	Pendiente	Centro
Pendiente	Revisión de requisitos	Visto bueno de la carrera	Requisitos completos	Derecha
Requisitos completos	Visto bueno de la carrera	Visto bueno de Vicedecanato	Trámite aceptado	Derecha
Requisitos completos	Visto bueno de la carrera	Visto bueno de la carrera	Trámite observado	Centro
Trámite observado	Visto bueno de la carrera	Visto bueno de Vicedecanato	Trámite aceptado	Derecha
Trámite aceptado	Visto bueno de Vicedecanato	Orden de matriculación	Trámite aceptado	Derecha
Trámite aceptado	Visto bueno de Vicedecanato	Visto bueno de Vicedecanato	Trámite observado	Centro
Trámite observado	Visto bueno de Vicedecanato	Orden de matriculación	Trámite aceptado	Derecha
Trámite aceptado	Orden de matriculación	Fin de trámite	Fin	Derecha

Tabla 2 Tabla de estados para Inscripción de Profesionales

Fuente: Elaboración Propia

Cada etapa de un trámite esta representado por un nodo como se puede apreciar en las figuras representativas de la máquina de Turing Fig. 7-11 y cada nodo representa un programa independiente que contiene dos controladores y una vista para introducir información.

3.3 ETAPAS DE LA METODOLOGÍA RUP

3.3.1 Modelado del negocio

Un modelo comprende un conjunto de paquetes que contienen los elementos del mismo, es una descripción completa de un sistema

El objetivo del modelado del negocio es el comprender el conjunto de procesos de negocio que tienen lugar dentro de una empresa, para poder establecer los requisitos del sistema a desarrollar.

Es acá donde se debe responder a la pregunta ¿Cómo es que consigue la empresa sus objetivos?

Las etapas del modelado del negocio son:

- Primeramente identificar y definir los procesos de negocio según los objetivos de la empresa. La siguiente figura muestra el diagrama del modelado de negocio del Sistema de Control de trámites utilizando Máquinas de Turing con todos los procesos que intervienen en el sistema.

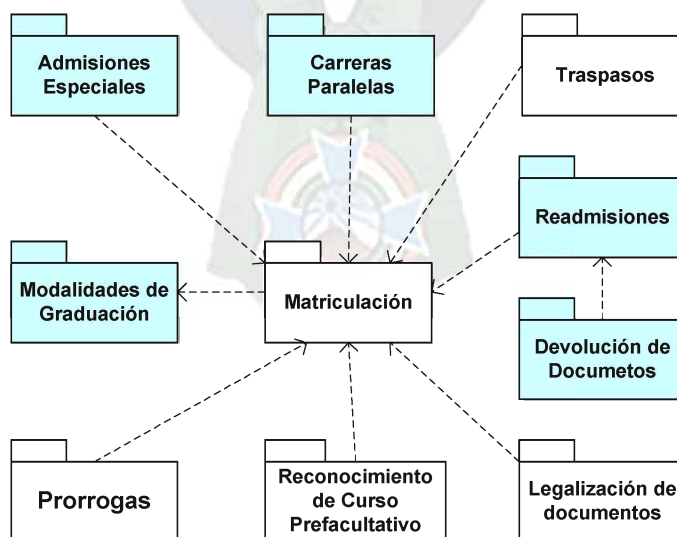


Fig. 12 Modelado del negocio

Fuente: Elaboración propia

La parte sombreada representan los módulos a ser implementados en el presente proyecto.

- Definir un caso de uso del negocio para cada proceso del negocio (esto con un diagrama de casos de uso general del negocio el mismo puede mostrar el contexto y los límites de la organización).

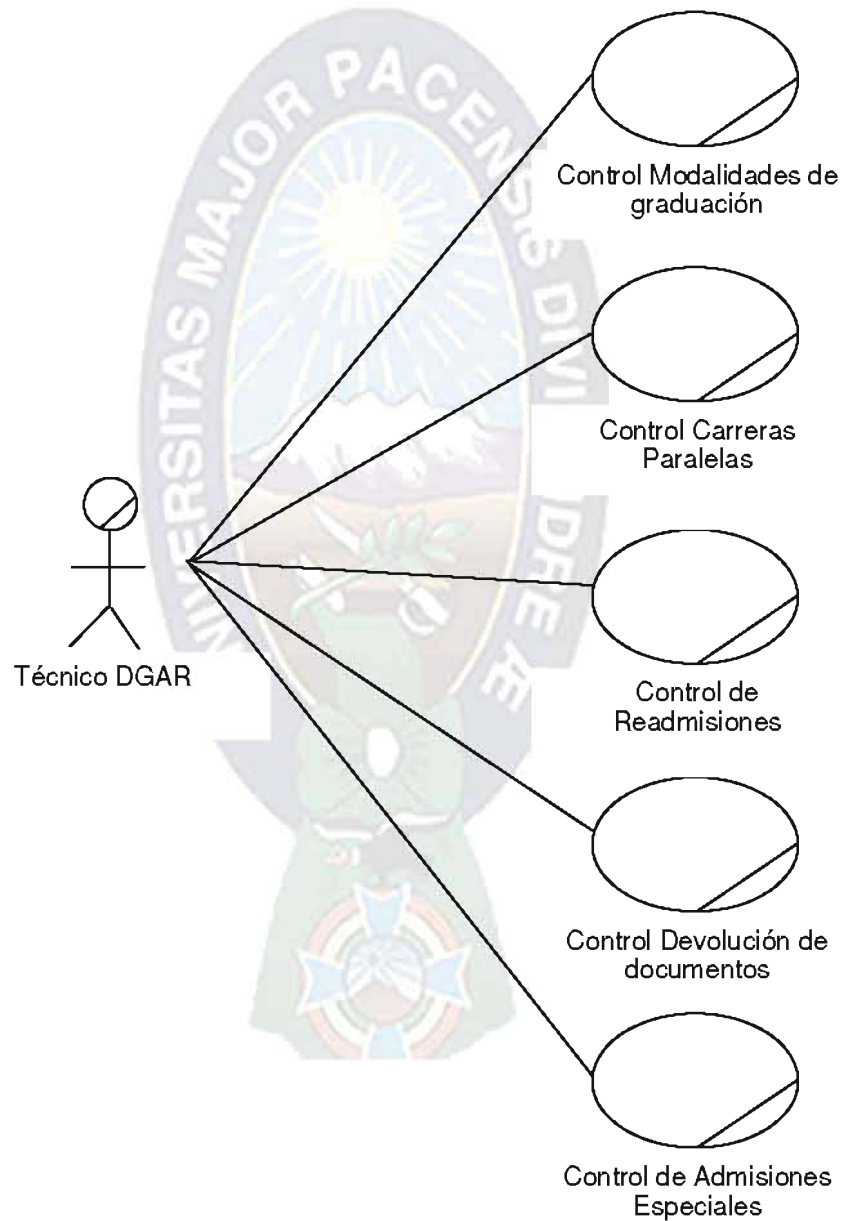


Fig. 13 Diagrama de casos de uso del negocio

Fuente: Elaboración propia

- Modelar los objetos del negocio que describa cada caso de uso del negocio con los actores participantes de cada uno. Cada modelo de objetos esta asociado a cada uno de los casos de uso del negocio.

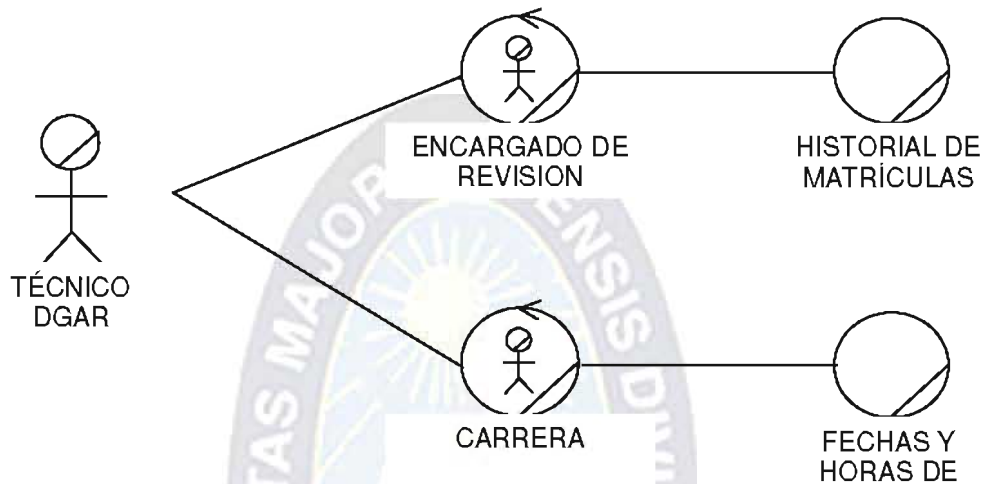


Fig. 14 Modelo de Objetos de Control de Modalidades de Graduación

Fuente: Elaboración propia

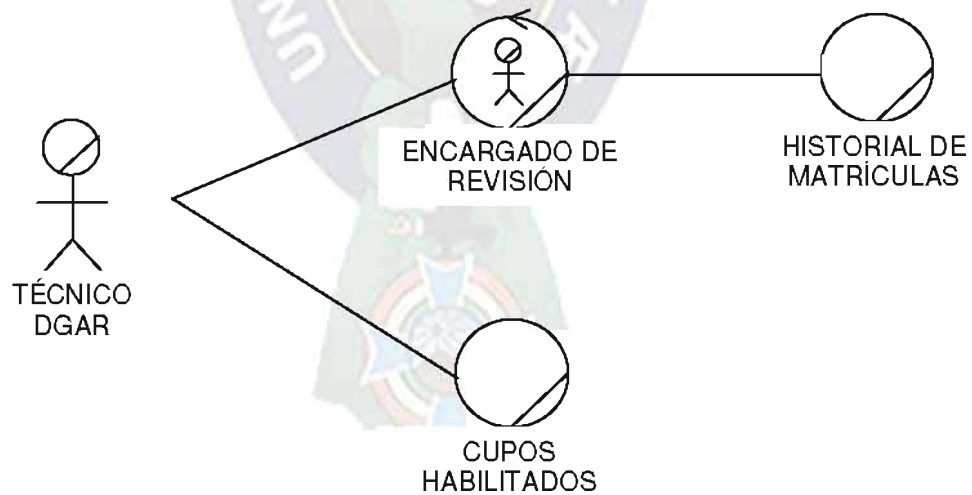


Fig. 15 Modelo de Objetos de Control Carreras Paralelas

Fuente: Elaboración propia

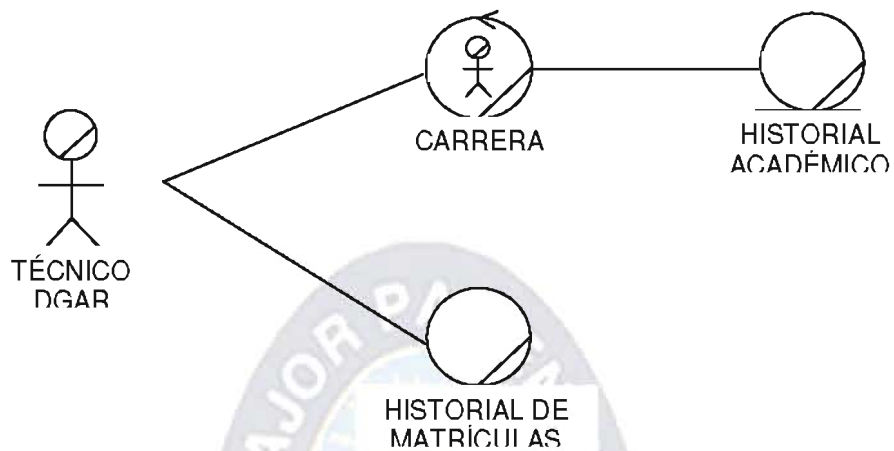


Fig. 16 Modelo de Objetos de Control de Readmisiones

Fuente: Elaboración propia



Fig. 17 Modelo de Objetos de Control Devolución de documentos

Fuente: Elaboración propia



Fig. 18 Modelo de Objetos de Control de Admisiones Especiales

Fuente: Elaboración propia

3.3.2 Modelado de Requisitos

En el modelado de requisitos se definen los requisitos funcionales y no funcionales del sistema.

El modelado de requisitos se basará básicamente en el modelo de dominio y el modelo de casos de uso.

3.3.2.1 Diagramas de Casos de Uso

Los diagramas de casos de uso representan las interacciones existentes entre el usuario y el sistema, al usuario se lo denomina también actor.

Este tipo de diagrama muestra el comportamiento del sistema hacia los usuarios en cuanto a su funcionalidad.

A continuación se muestra los diagramas de casos de uso que el sistema presenta.

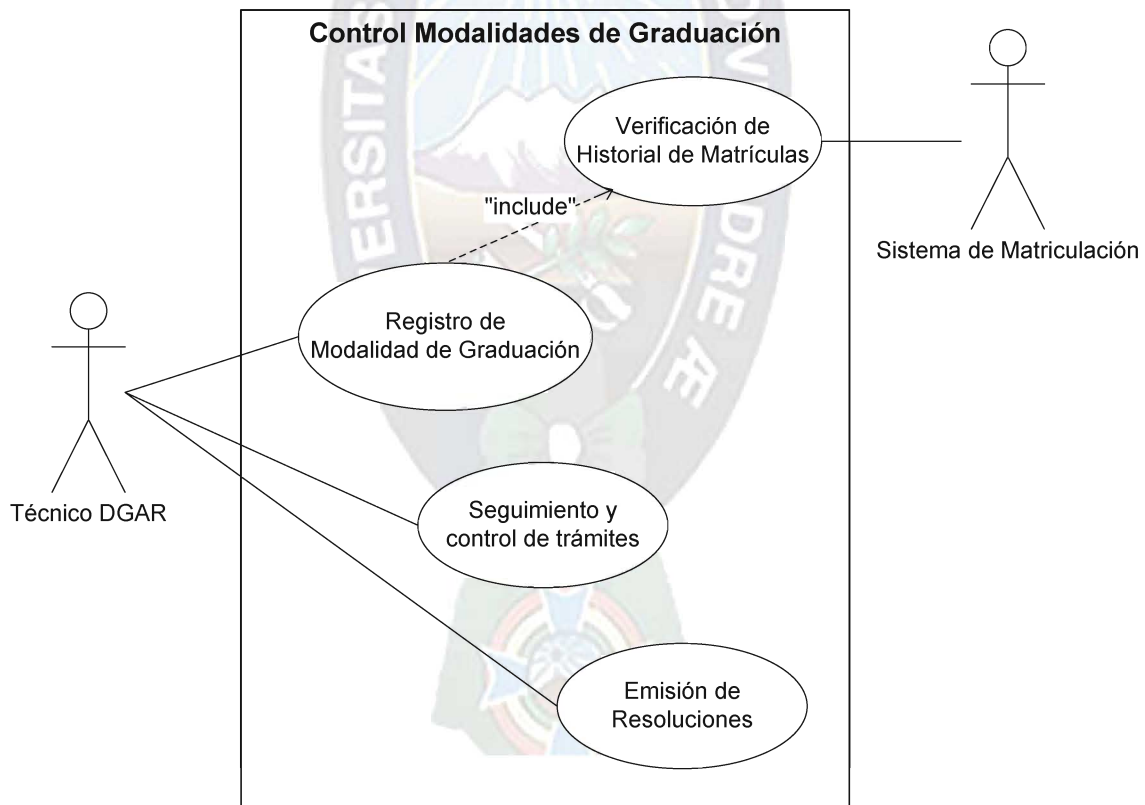


Fig. 19 Diagrama de casos de uso Control Modalidades de Graduación

Fuente: Elaboración propia

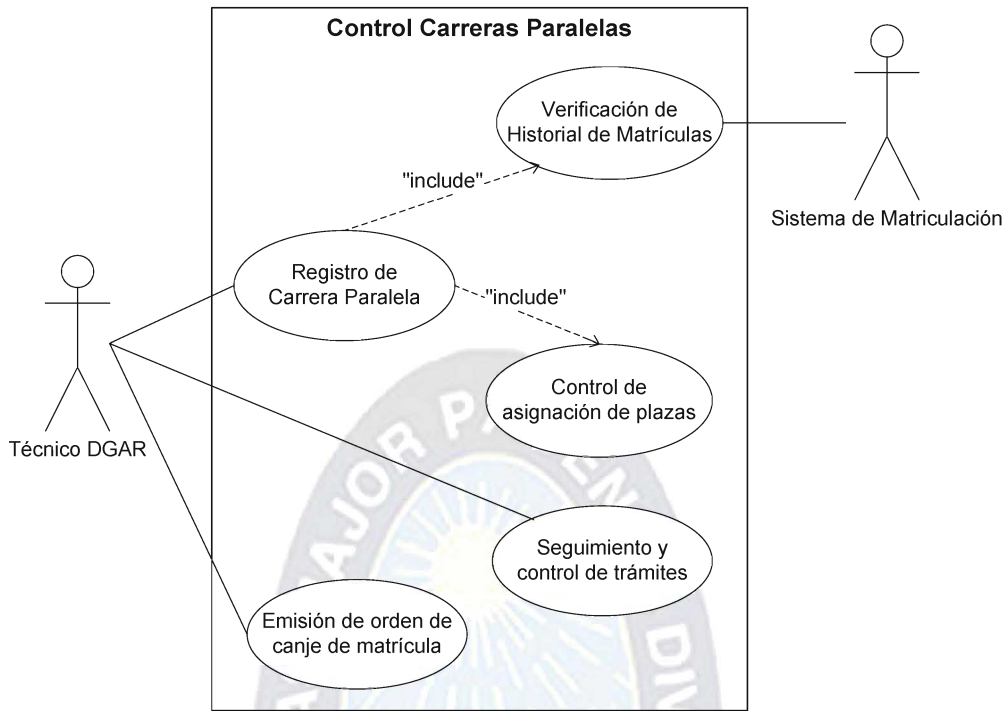


Fig. 20 Diagrama de casos de uso Control Carreras Paralelas

Fuente: Elaboración propia

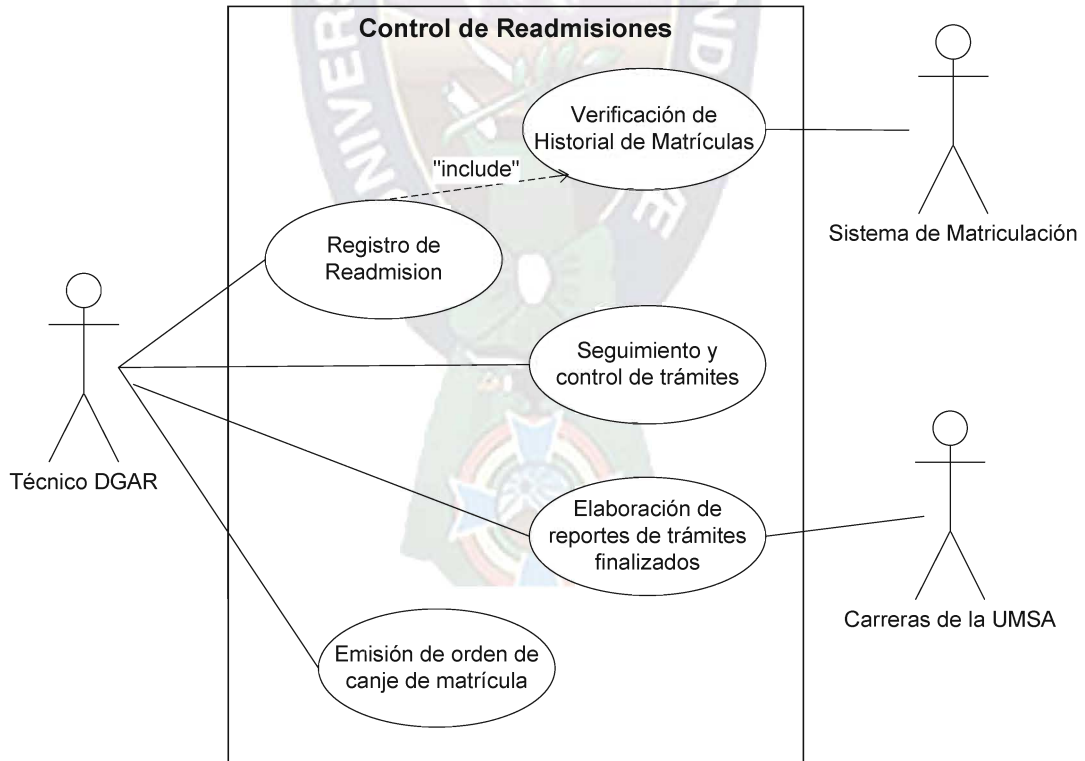


Fig. 21 Diagrama de casos de uso Control de Readmisiones

Fuente: Elaboración propia

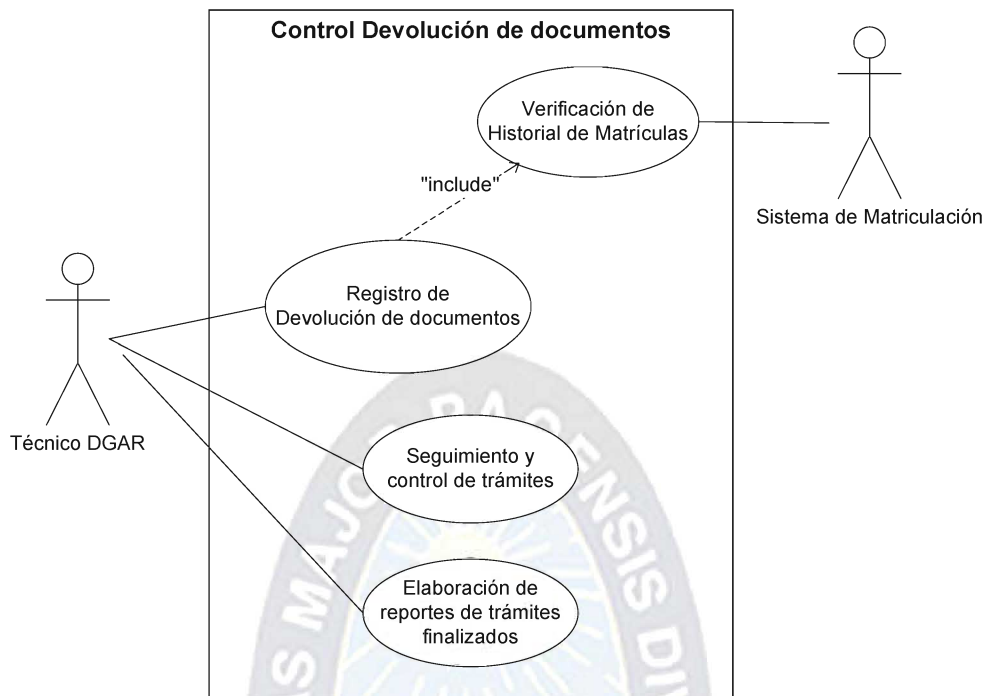


Fig. 22 Diagrama de casos de uso Control Devolución de documentos

Fuente: Elaboración propia

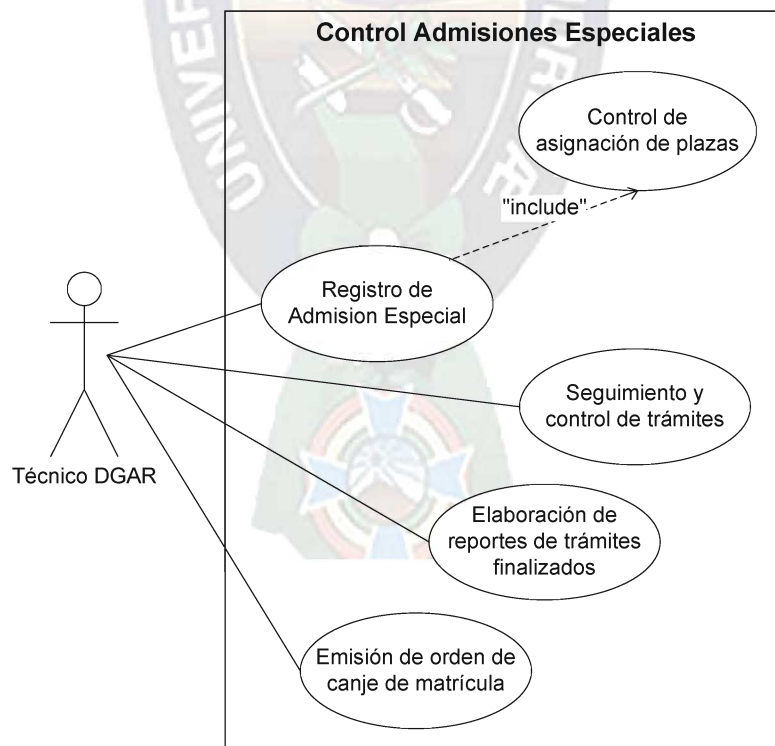


Fig. 23 Diagrama de casos de uso Control de Admisiones Especiales

Fuente: Elaboración propia

Para cada caso de uso se ha elaborado su especificación de casos de uso ver Anexos K

3.3.3 Modelado del análisis

En la parte de análisis se tiene que tomar muy en cuenta los conceptos que implican el analizar un sistema, se realiza una especificación más precisa de los requisitos.

El análisis facilita la comprensión, preparación, modificación y mantenimiento de requisitos.

Se logra una aproximación a la fase de diseño. Para el análisis se utilizarán diagramas que servirán de modelo.

Los objetivos del modelado del análisis son los siguientes:

- A partir de los casos de uso obtener el diseño preliminar del sistema que deberá ser refinado en el modelo del diseño.
- Obtener un nivel de abstracción más alto que en el diseño. Tener una visión ideal y clara del sistema.
- Se define la arquitectura del sistema.
- Para cada caso de uso se define un diagrama de secuencia del sistema que muestra los eventos que un actor genera durante la interacción con el sistema (caja negra)
- Cada evento da origen a una operación del sistema
- El efecto de las operaciones se describen mediante contratos especificados mediante una plantilla.
- Para cada operación del sistema se define un diagrama de interacción que muestra cómo deben colaborar los objetos para satisfacer las responsabilidades y poscondición expresada en el contrato de dicha operación.

Para representar mejor el modelado de análisis se utilizará los diagramas de clases de análisis.

3.3.3.1 Diagrama de clases de análisis

Los diagramas de clases de análisis representan una abstracción de una o mas clases del diseño del sistema, se centran en el tratamiento de requisitos funcionales, son evidentes en el dominio del sistema.

Los atributos, operaciones y relaciones están a un nivel mayor de abstracción, las clases se pueden clasificar en tres que son:

- Clases de entidad.
- Clases de interfaz.
- Clases de control

La siguiente grafica muestra la forma de representación gráfica de las clases ya mencionadas:

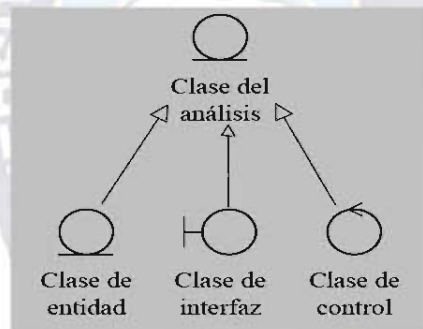


Fig. 24 Esquema de clases de diseño

Fuente: [Proceso Unificado-PDF]

La realización de clases de uso de análisis define como llevar a cabo y ejecutar un caso de uso en términos de clases de análisis y de objetos de análisis en colaboración.

La realización de un caso de uso queda definida por:

- Diagrama de clases de análisis.
- Diagrama de interacción de objetos de análisis.
- Una descripción textual del flujo de procesos.

A continuación se muestran los diferentes diagramas de clases de análisis diseñados para los diferentes procesos ya definidos para el sistema.

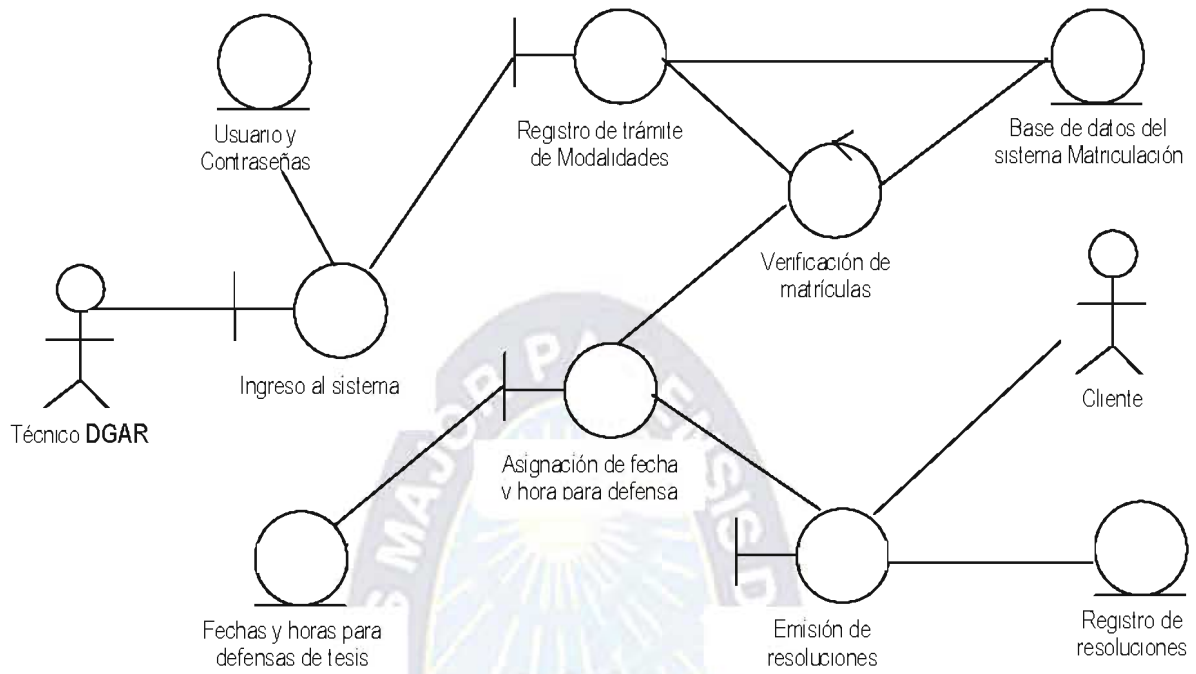


Fig. 25 Diagrama de clases del proceso: Control de Modalidades de Graduación

Fuente: Elaboración Propia

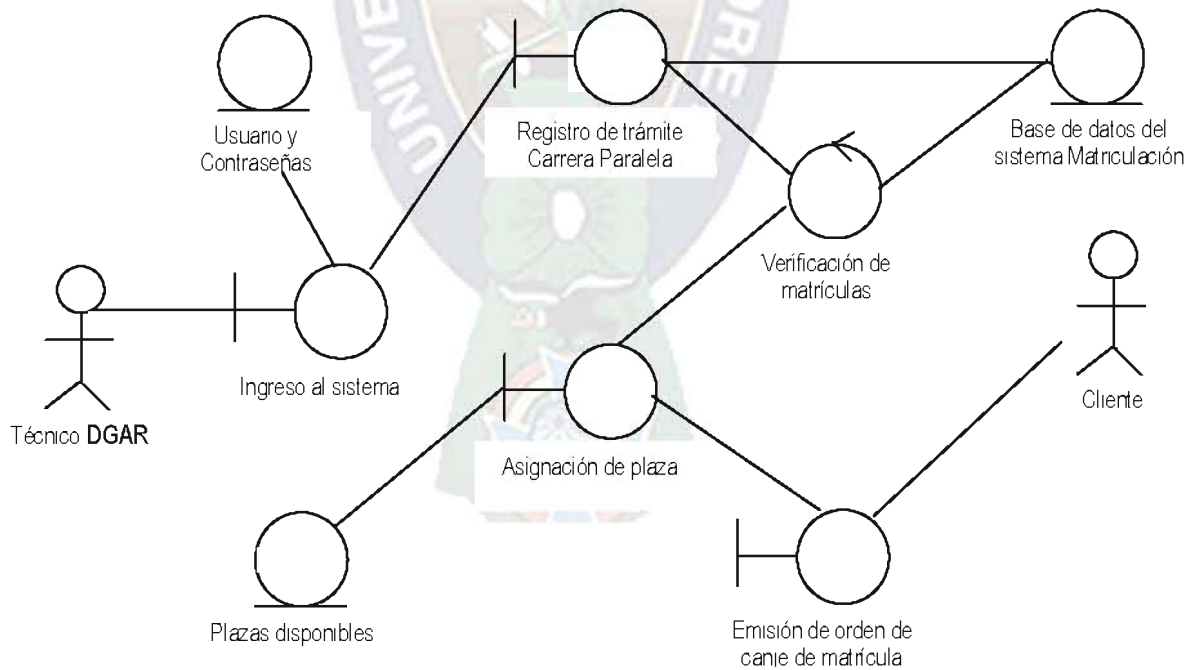


Fig. 26 Diagrama de clases del proceso: Control de Carreras Paralelas

Fuente: Elaboración Propia

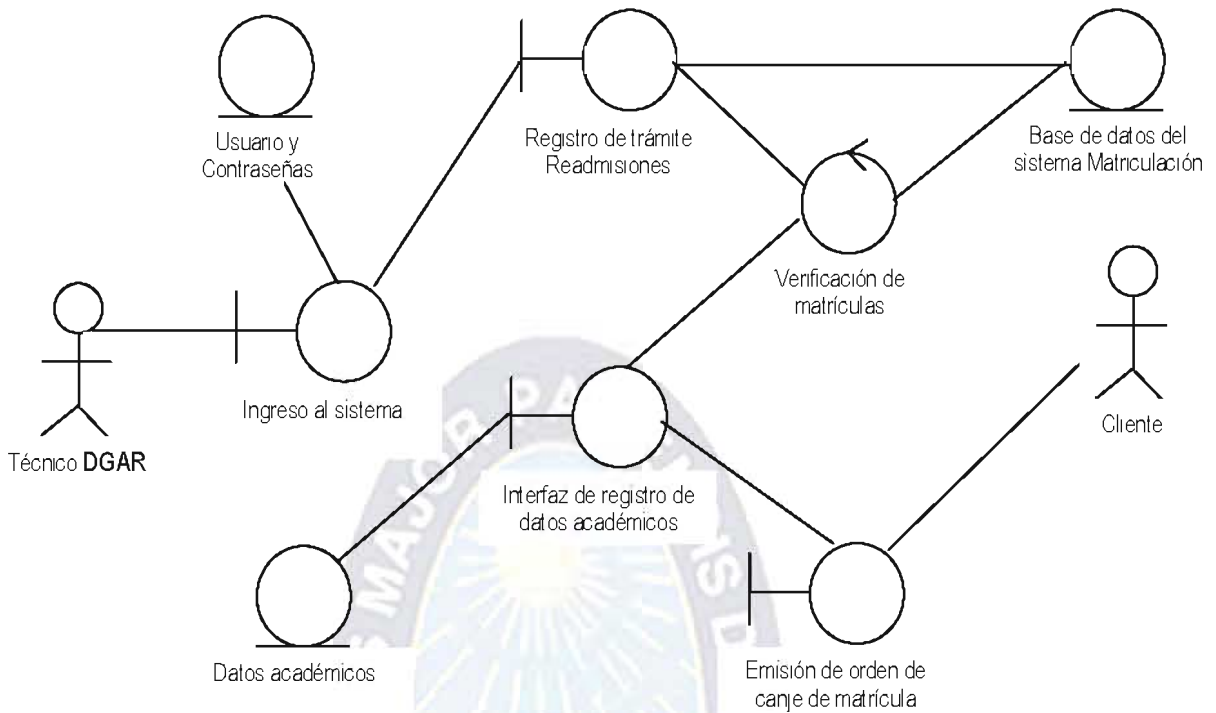


Fig. 27 Diagrama de clases del proceso: Control de Readmisiones

Fuente: Elaboración Propia

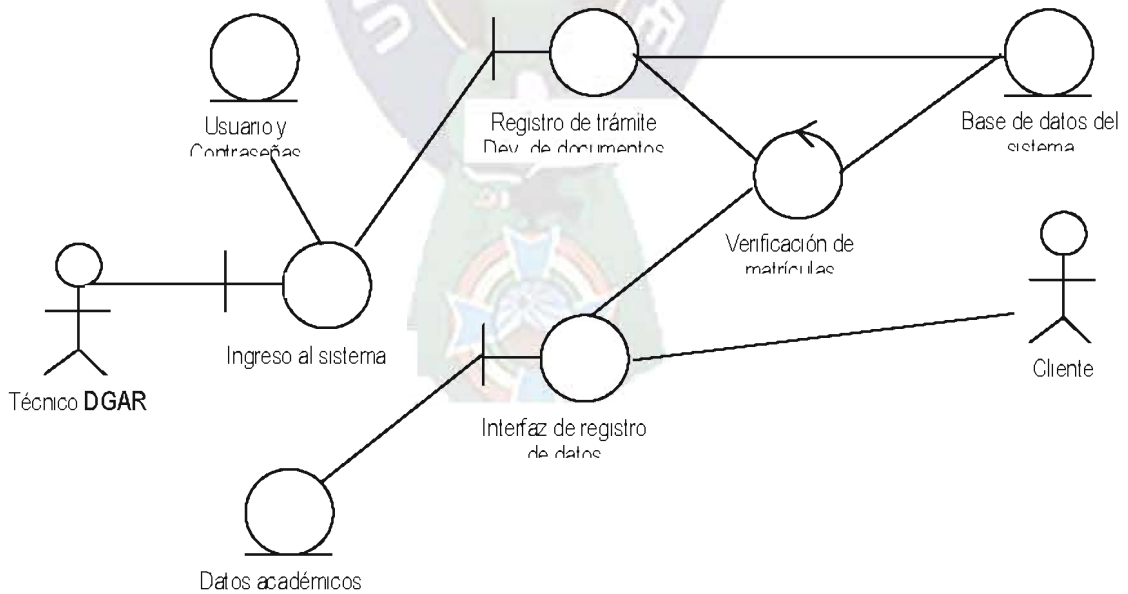


Fig. 28 Diagrama de clases del proceso: Control de Devolución de documentos

Fuente: Elaboración Propia

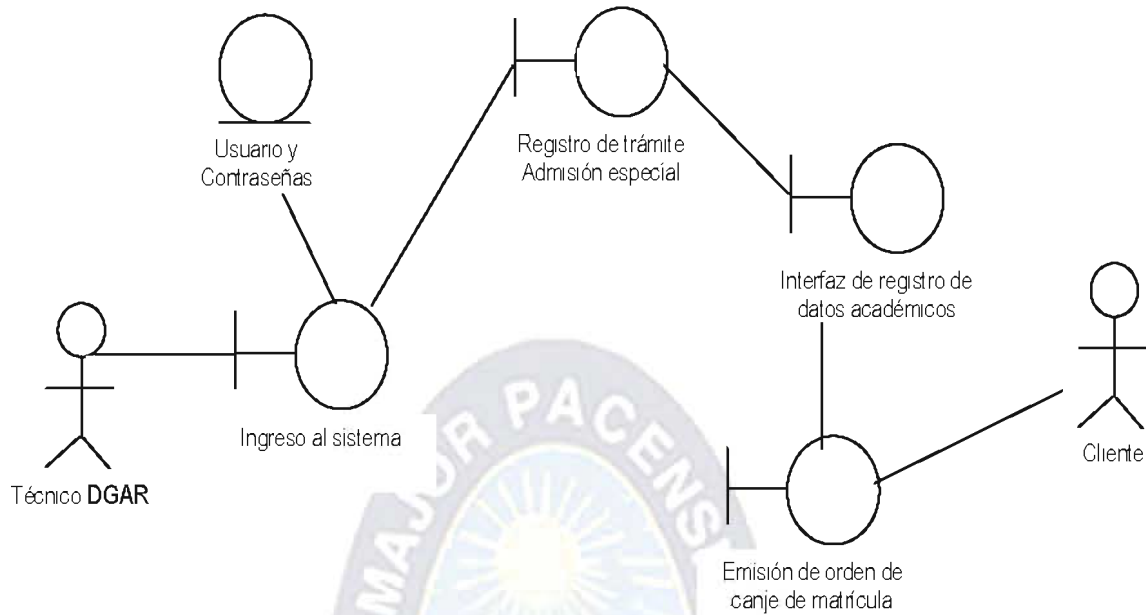


Fig. 29 Diagrama de clases del proceso: Control de Admisiones Especiales

Fuente: Elaboración Propia

3.3.3.2 Diagramas de secuencias

Después de haberse identificado cada una de las clases se debe de describir la interacción existente entre ellas para la funcionalidad de los casos de uso, para ello se elaborará los diagramas de secuencia conocidos también con el nombre de interacción o eventos, los cuales describen los casos de uso según la interacción de los objetos de la arquitectura del sistema.

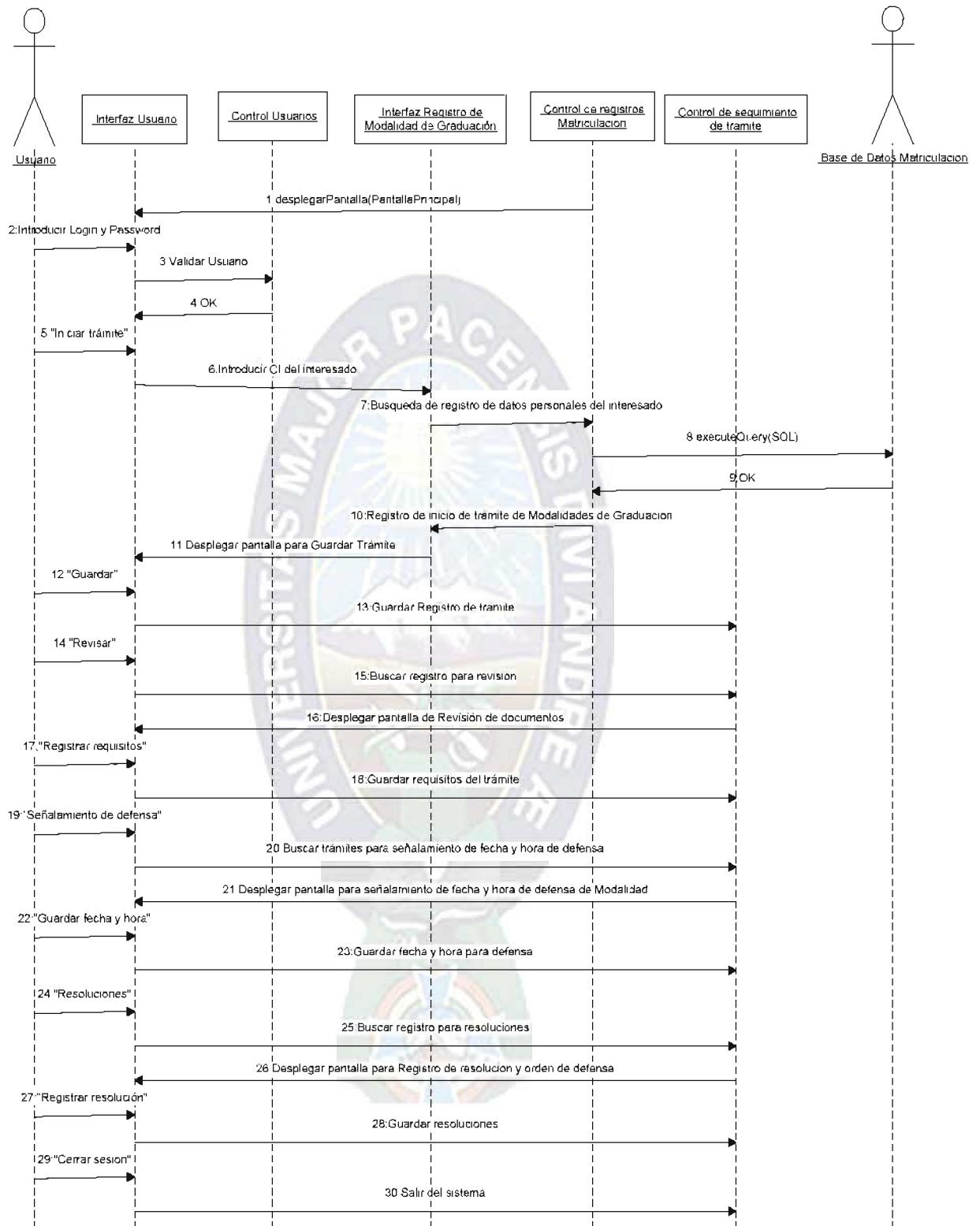


Fig. 30 Diagrama de secuencias: Control de Modalidades de Graduación

Fuente: Elaboración Propia

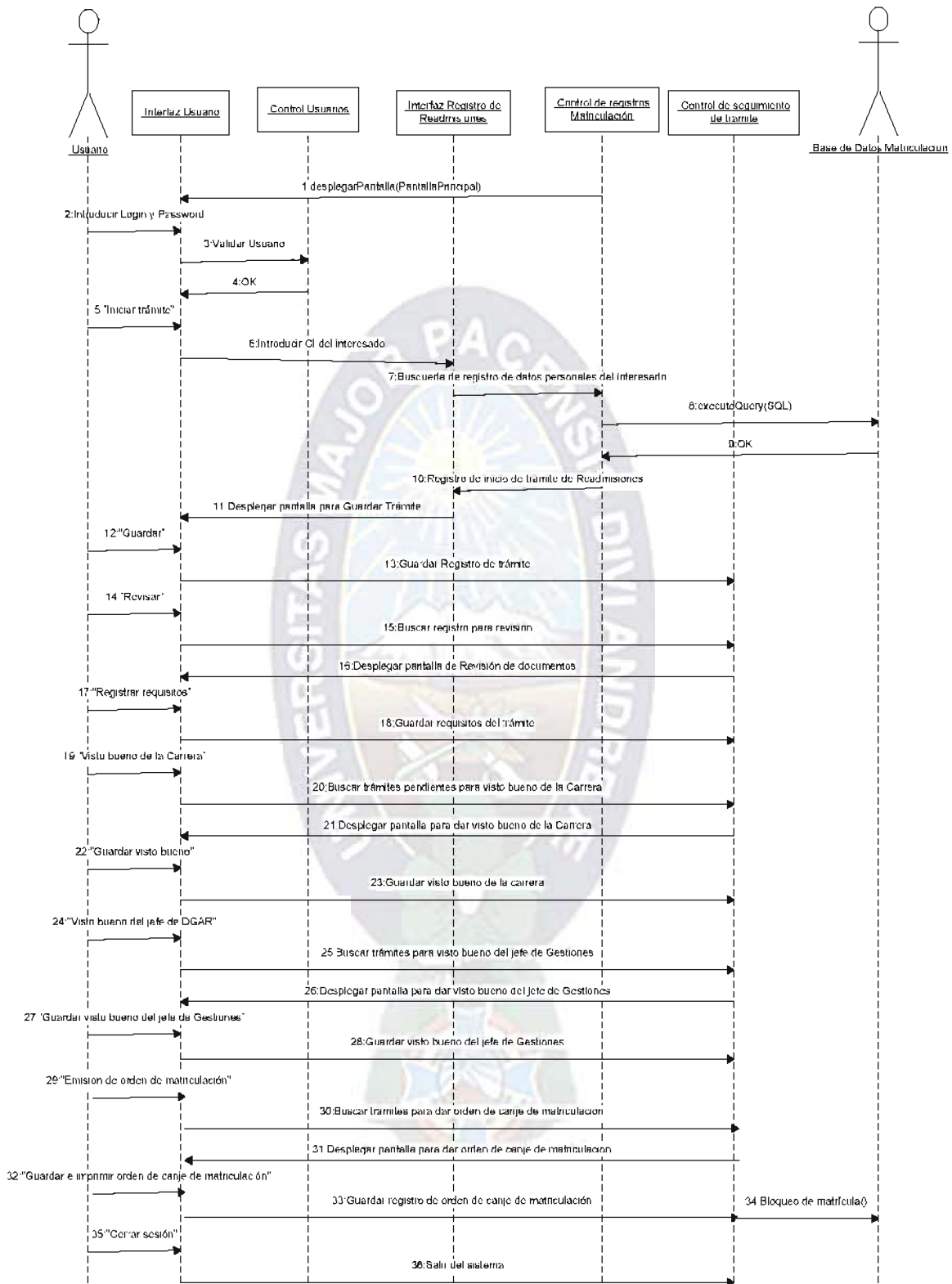


Fig. 31 Diagrama de secuencias: Control de Readmisiones

Fuente: Elaboración Propia

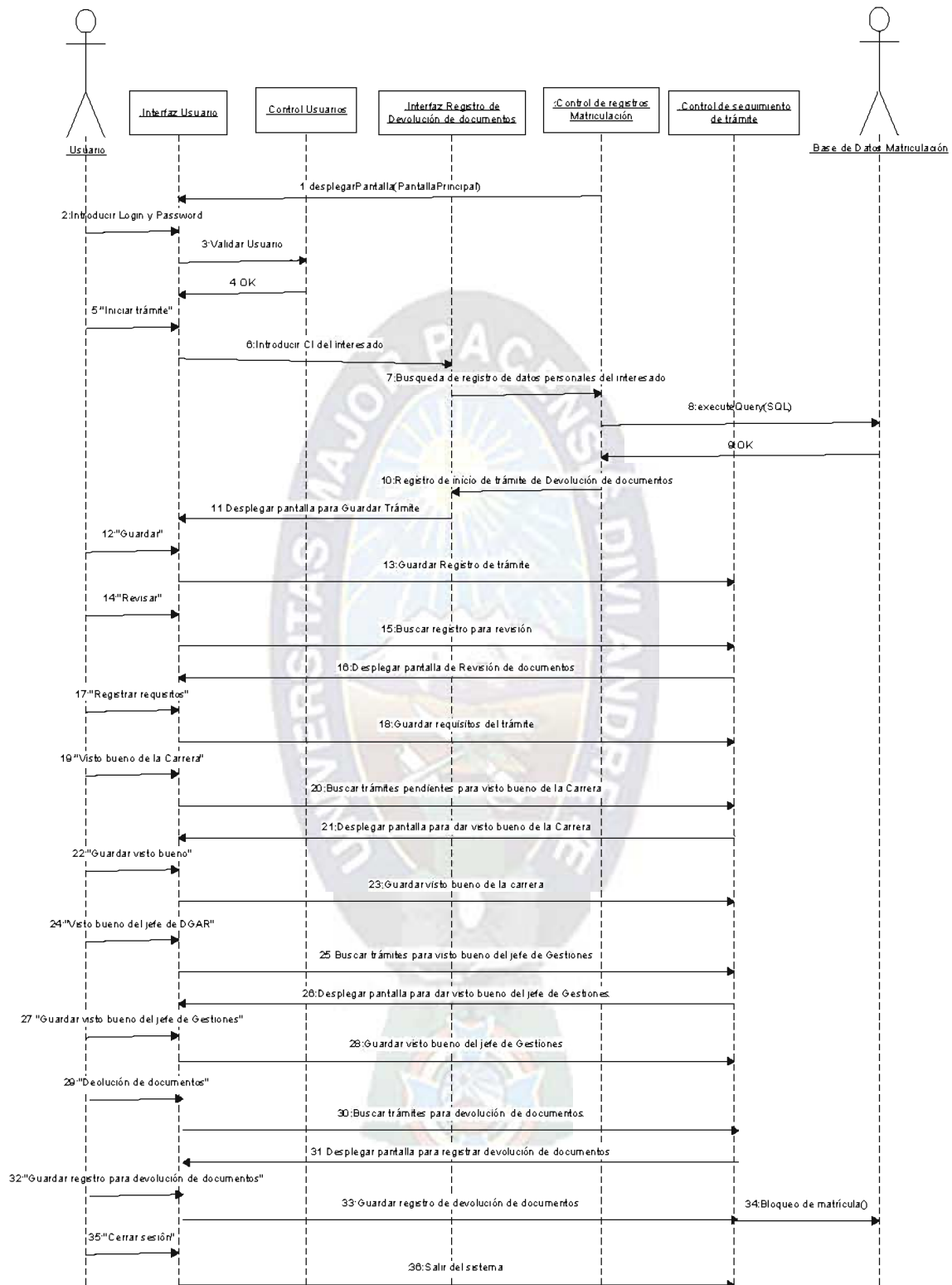


Fig. 32 Diagrama de secuencias: Control de Devolución de documentos

Fuente: Elaboración Propia

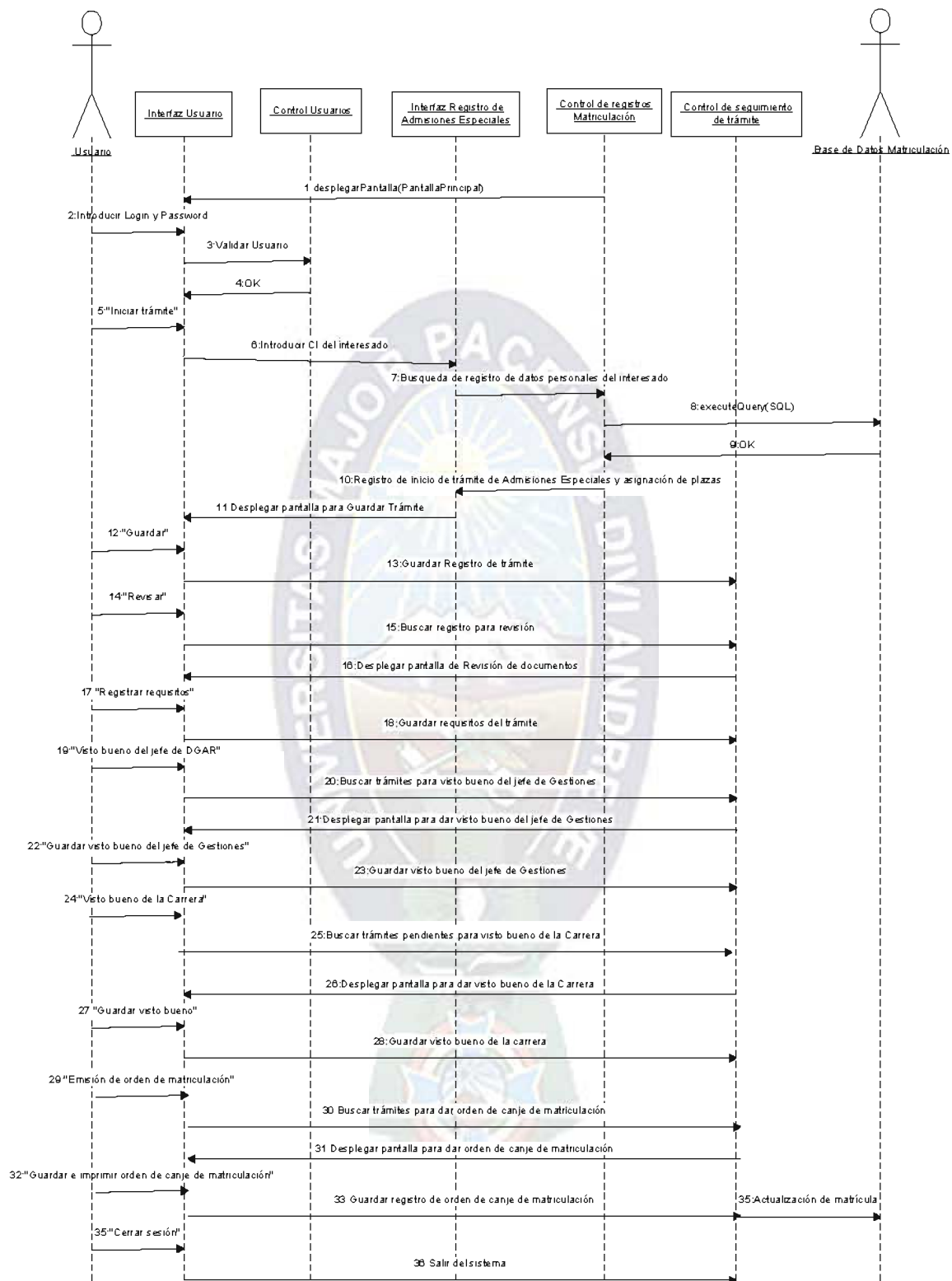


Fig. 33 Diagrama de secuencias: Control de Admisiones Especiales

Fuente: Elaboración Propia

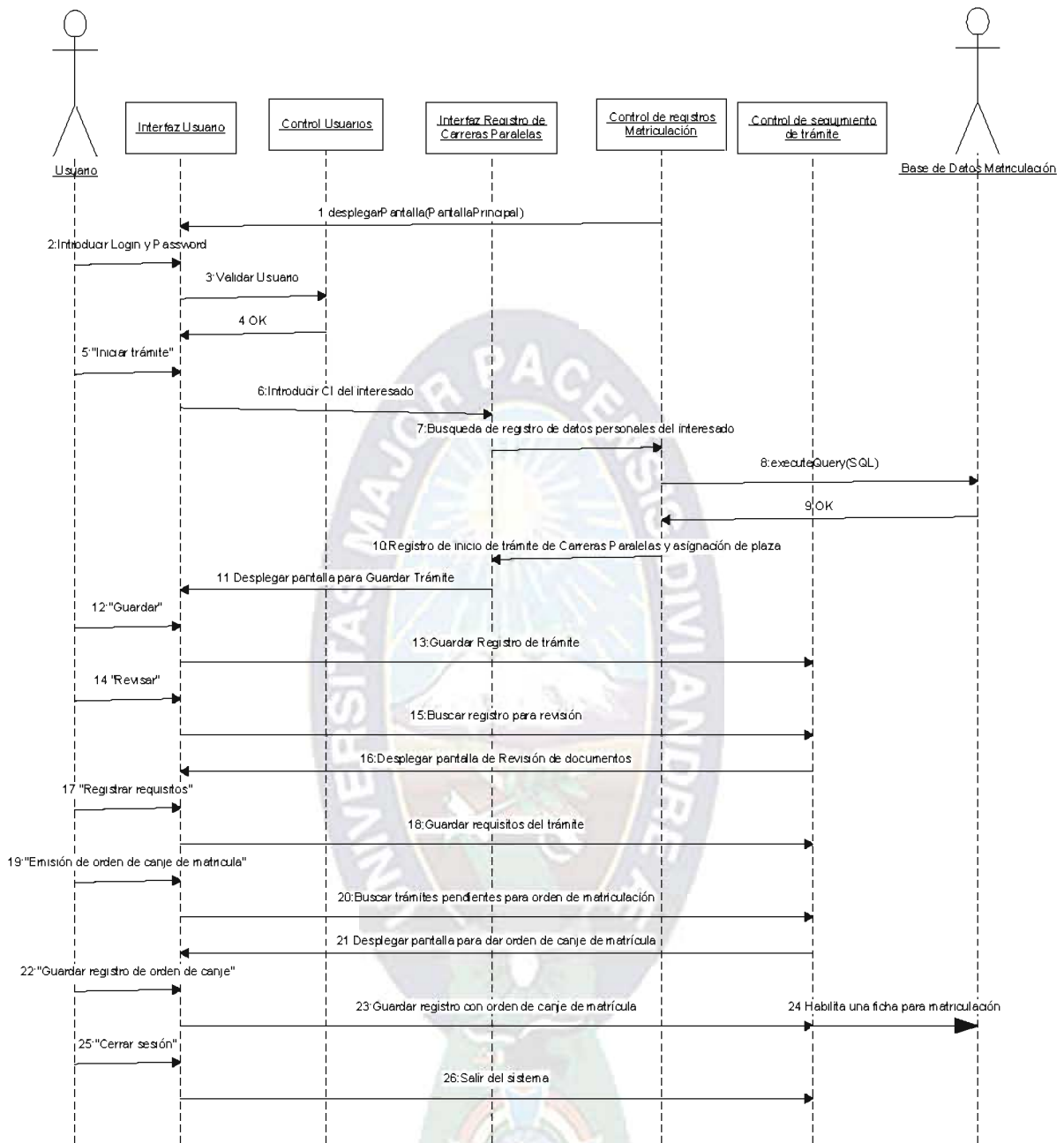


Fig. 34 Diagrama de secuencias: Control de Carreras Paralelas

Fuente: Elaboración Propia

3.3.3.3 Diagramas de actividades

Los diagramas de actividades nos permiten visualizar los aspectos de procedimiento del software, esta notación de UML es similar al diagrama de flujo y se utiliza para representar lo que sucede mientras el sistema realiza sus funciones. Los rectángulos redondeados implican

una función específica del sistema; las flechas representan el flujo a través del sistema [Pressman, 2005].

A continuación se muestran los diferentes diagramas de actividades del sistema:

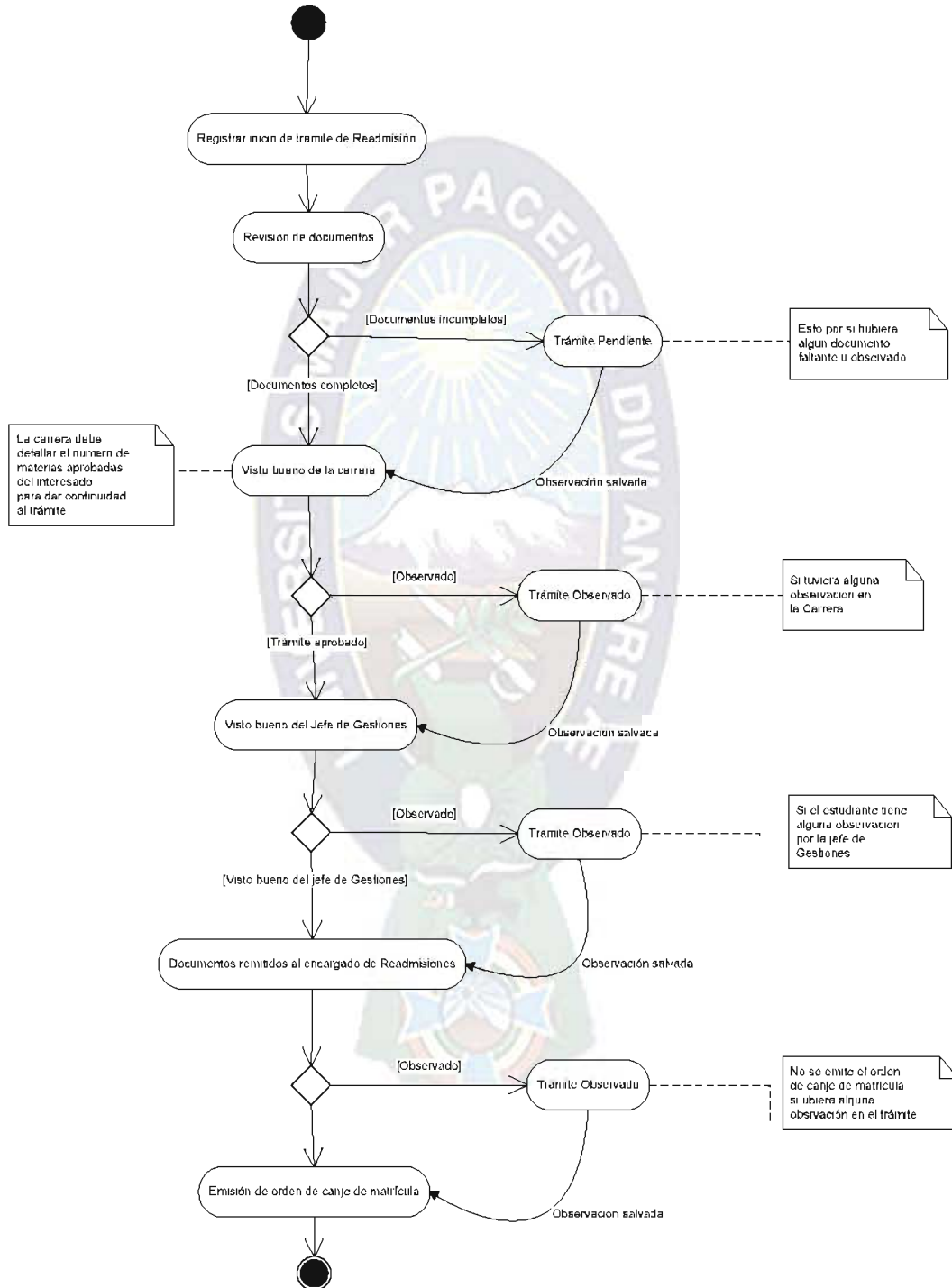


Fig. 35 Diagrama de actividades: Control de Readmisiones

Fuente: Elaboración Propia

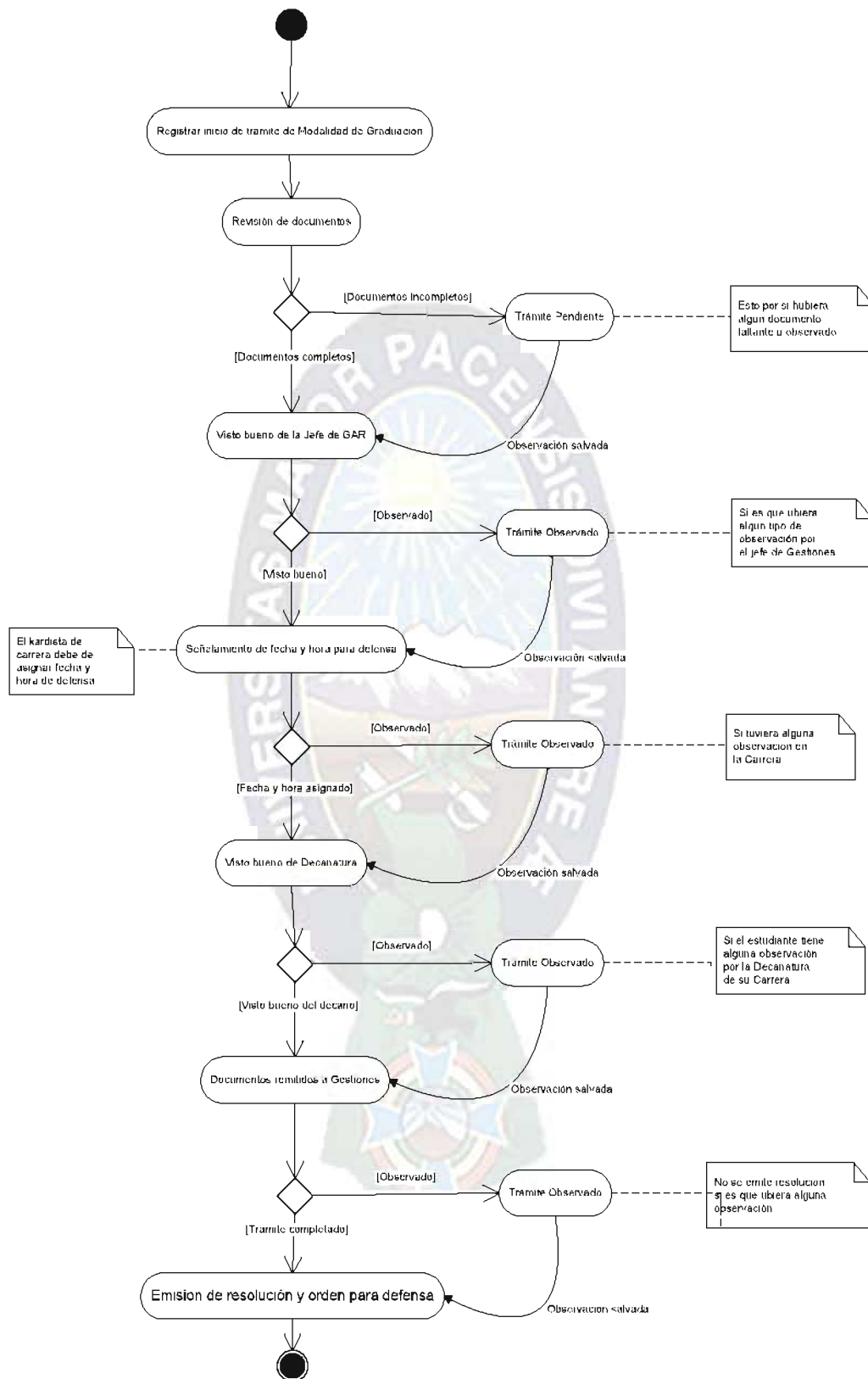


Fig. 36 Diagrama de actividades: Control de Modalidades de graduación

Fuente: Elaboración Propia

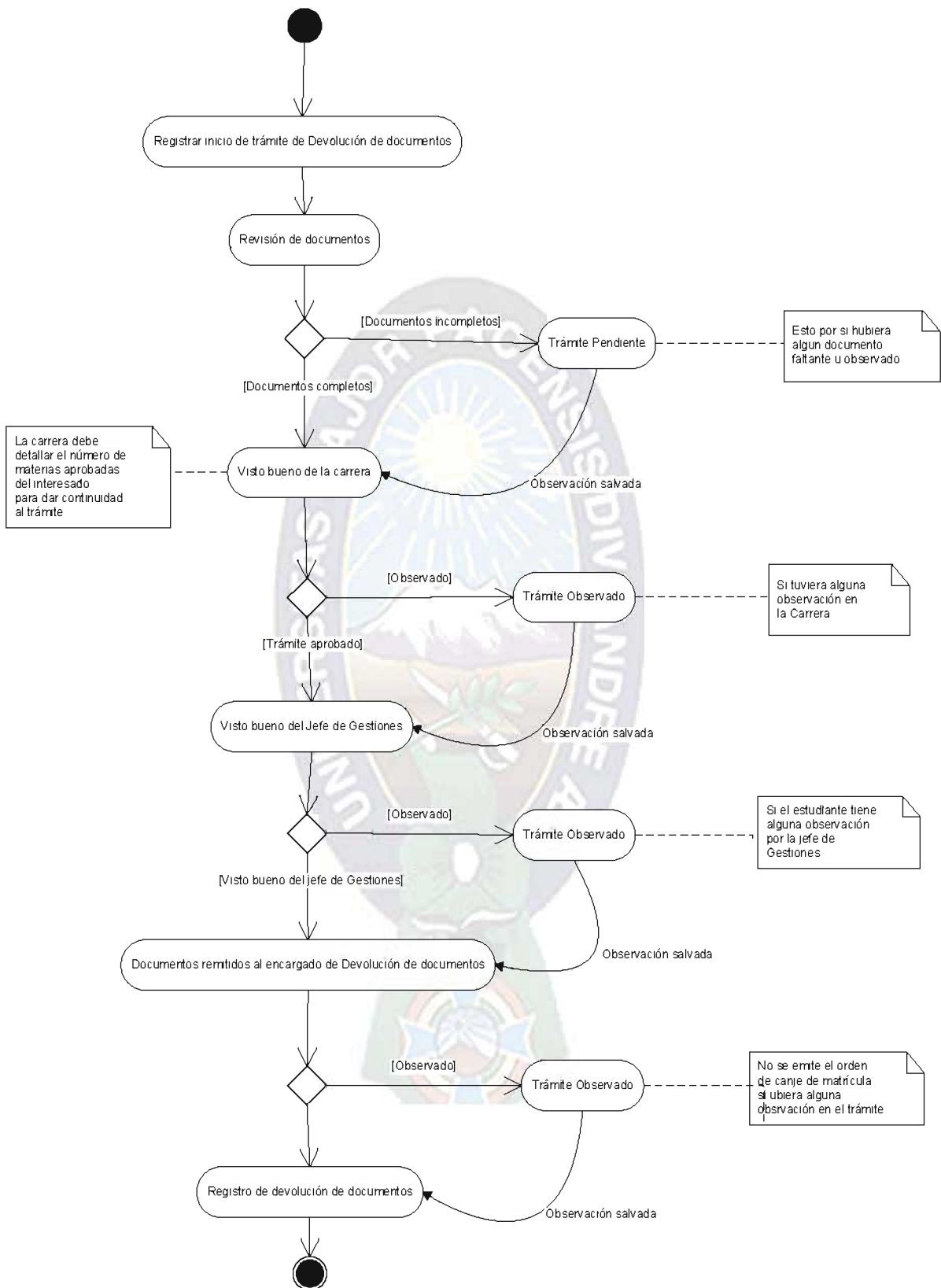


Fig. 37 Diagrama de actividades: Control de Devolución de documentos

Fuente: Elaboración Propia

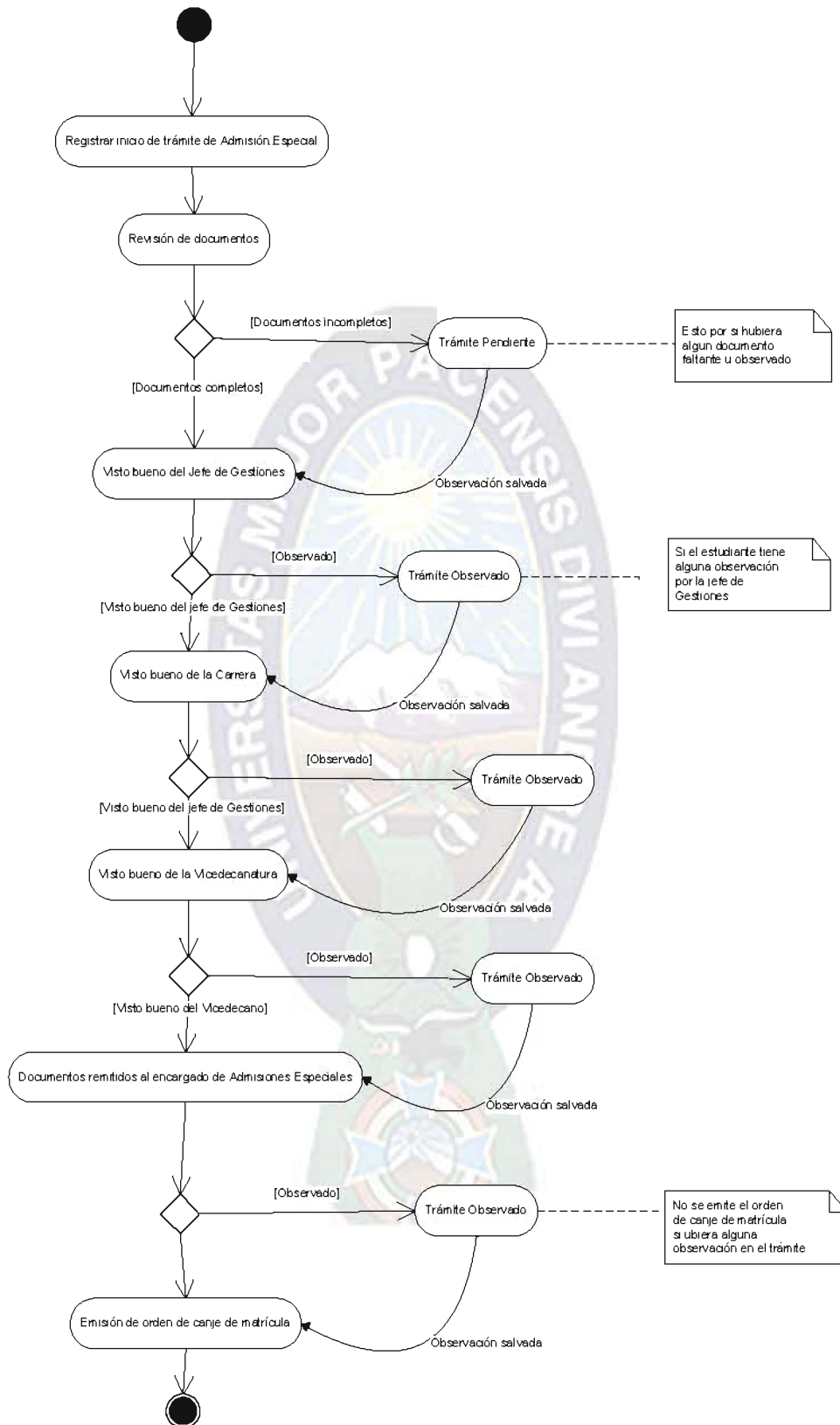


Fig. 38 Diagrama de actividades: Control de Admisiones Especiales

Fuente: Elaboración Propia

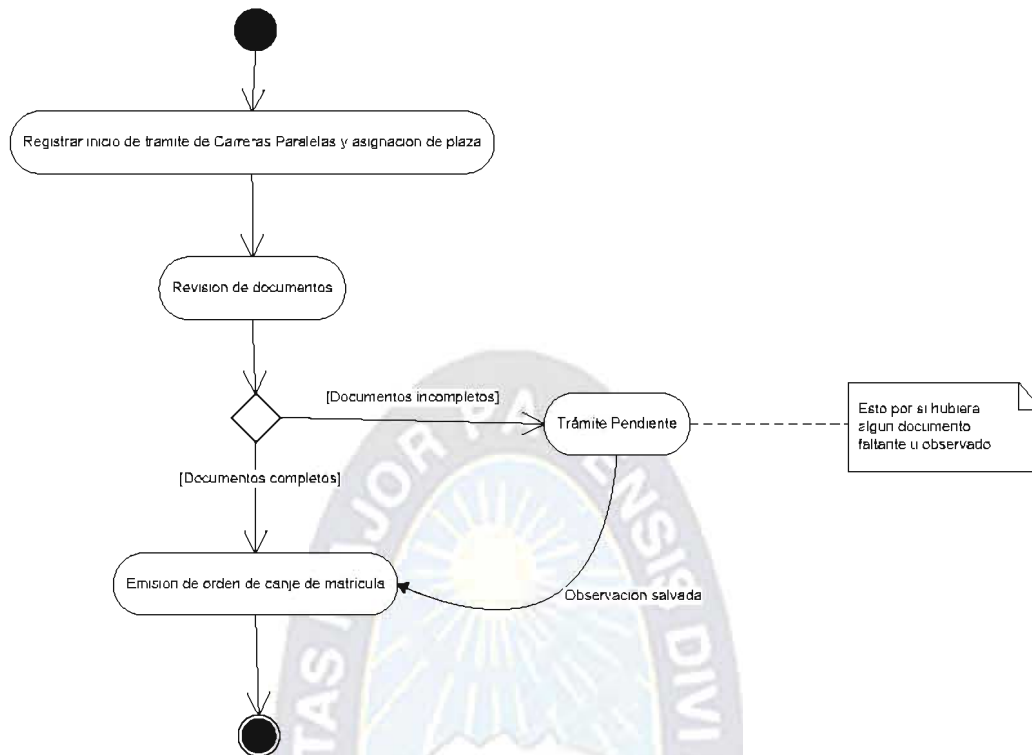


Fig. 39 Diagrama de actividades: Control de Carreras Paralelas

Fuente: Elaboración Propia

3.3.4 Modelado del diseño

El objetivo del modelado de diseño es refinar el diseño del sistema del modelo del análisis introduciendo los requisitos no funcionales y restricciones del entorno de implementación.

De manera iterativa se refina el diagrama de clase del análisis hasta obtener un diseño del sistema adecuado para pasar a la implementación.

Se crean los Diagramas de Clases del Diseño que refinan los del análisis. Se identifican clases (atributos y métodos) e interfaces que éstos vayan a tener. Establece navegabilidad para todas las asociaciones determinar visibilidad entre clases Incluyendo las relaciones de dependencia entre clases.

3.3.4.1 Diagrama de clases

Los diagramas de clases representan el modelado de la clase estática del sistema, se definen las clases, sus atributos y relaciones que puedan tener entre ellos.

El diagrama de clases definido para el sistema se puede ver en el anexo M.

3.3.4.2 Arquitectura del sistema

La arquitectura del software alude a “la estructura general del software y las formas en que la estructura proporciona una integridad conceptual para un sistema” [Pressman, 2005].

Una arquitectura de software es el producto del trabajo de desarrollo que ofrece el mayor rendimiento de la inversión con respecto a la calidad, el tiempo y el costo [Len Bass, 2005].

A continuación se muestra la arquitectura centrada en datos que muestra como es la comunicación de datos entre los módulos.

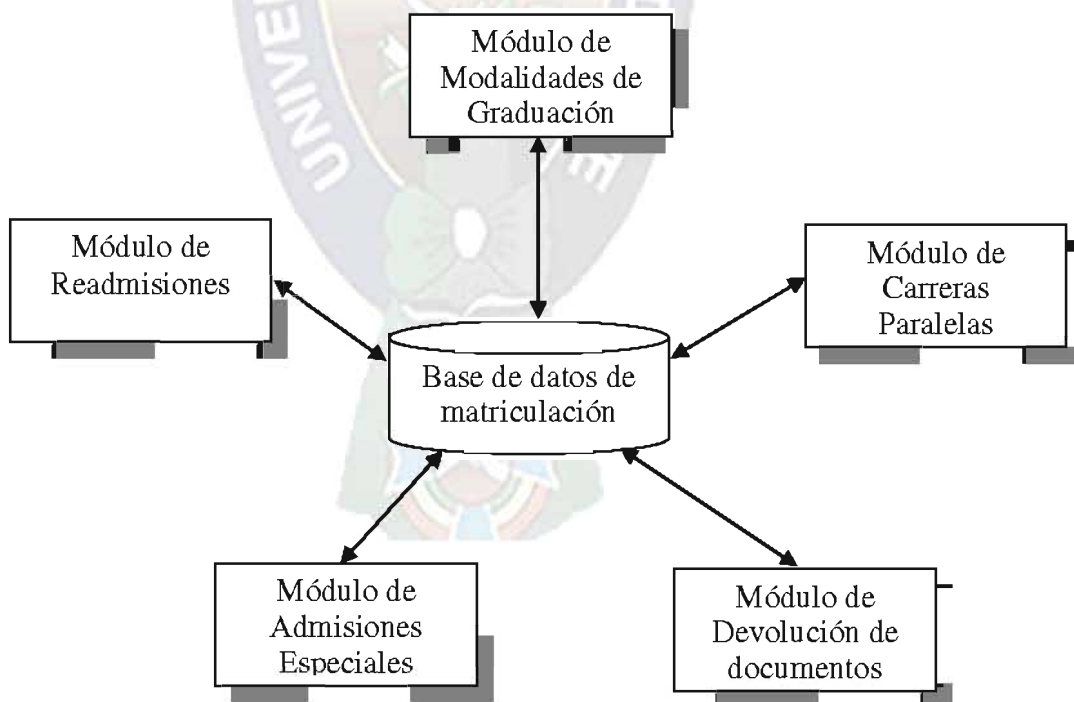


Fig. 40 Arquitectura de datos del sistema

Fuente: Elaboración Propia

La arquitectura del sistema en cuanto al programa principal y sus subprogramas es el siguiente:

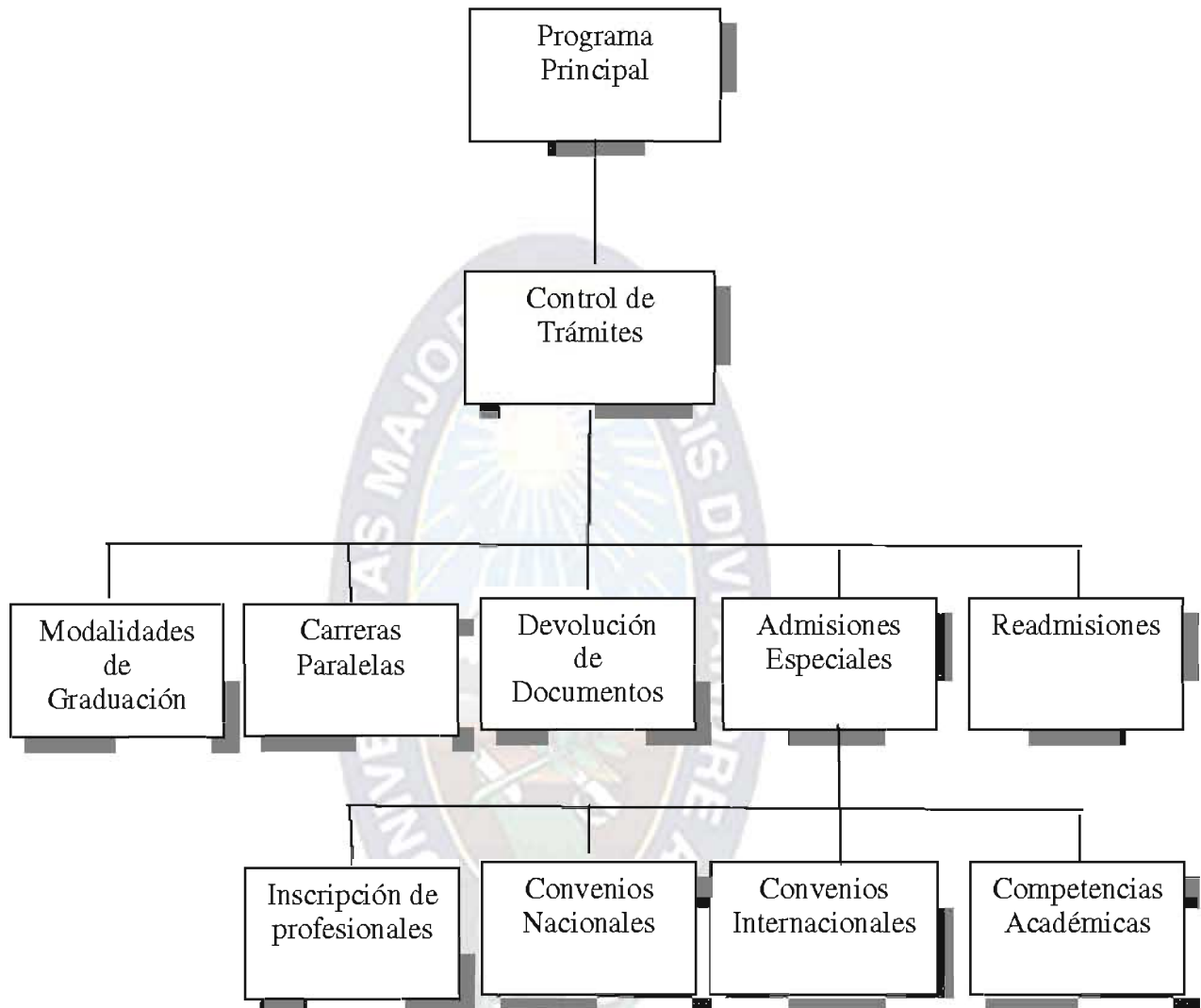


Fig. 41 Arquitectura de programa principal y subprogramas del sistema

Fuente: Elaboración Propia

3.3.4.3 Modelado de datos

El modelado de datos permite definir los objetos de datos que se procesaran dentro del sistema y sus relaciones, es una representación de cualquier información que se procesa dentro de un sistema.

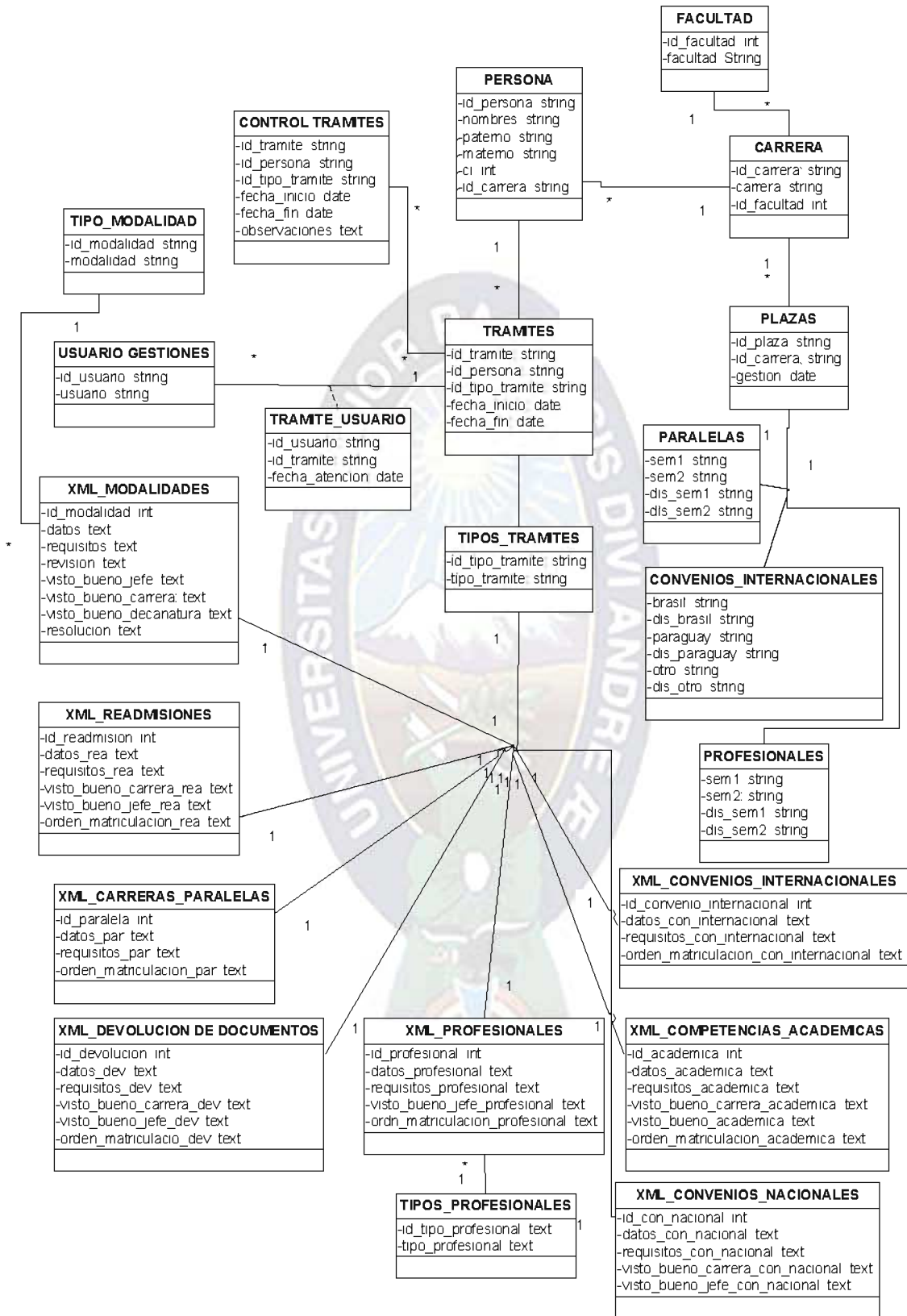


Fig. 42 Modelado de datos del Sistema

Fuente: Elaboración Propia

El modelo entidad relación de la Base de datos se muestra a continuación:

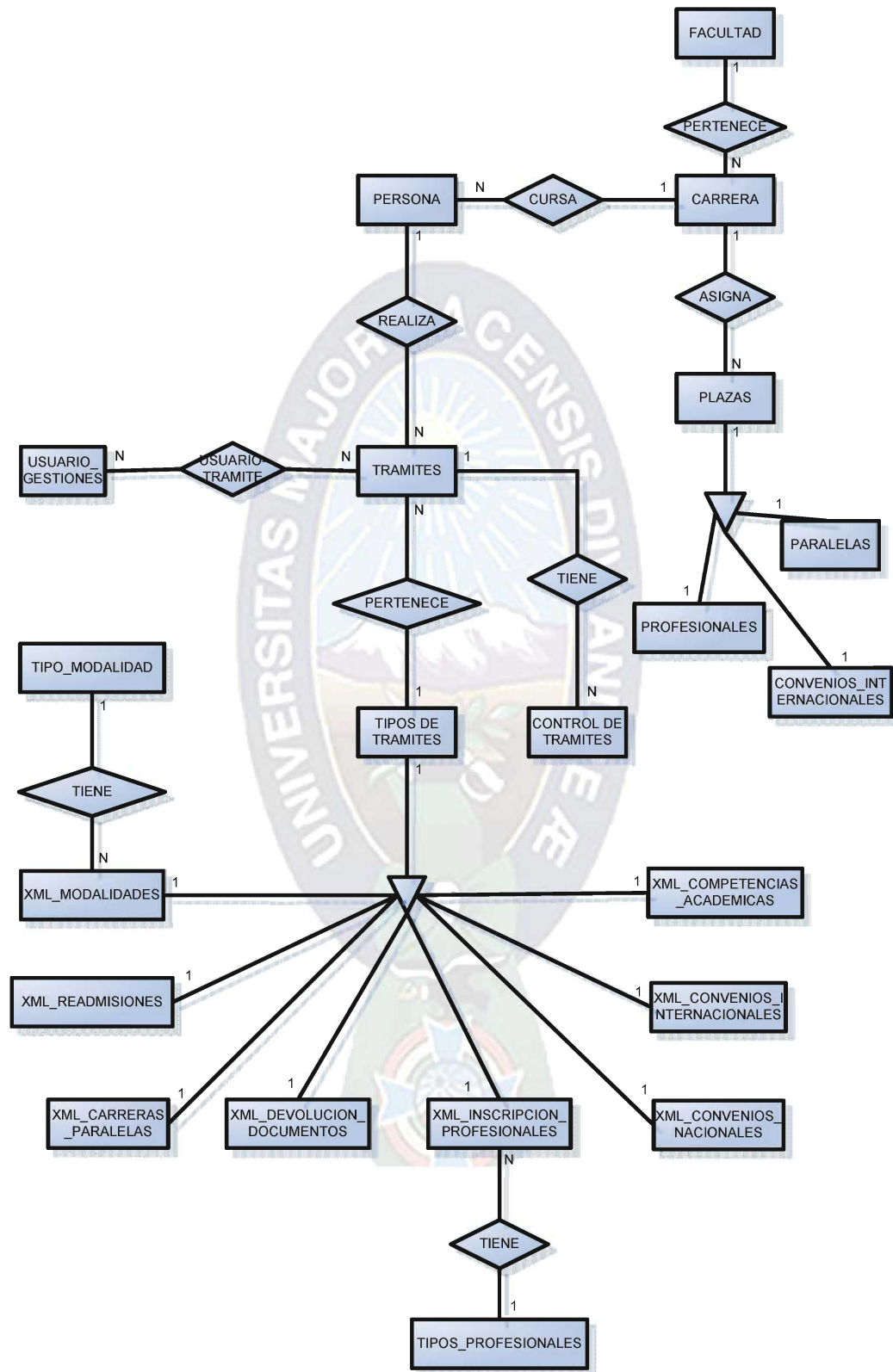


Fig. 43 Diagrama Entidad Relacion

Fuente: Elaboración Propia

3.4 IMPLEMENTACIÓN

La implementación toma como resultado del modelo de diseño para generar un código final [Weitzenfeld,2002].

Para ver la relación de componentes en la implementación ser tomará en cuenta el siguiente diagrama de componentes:

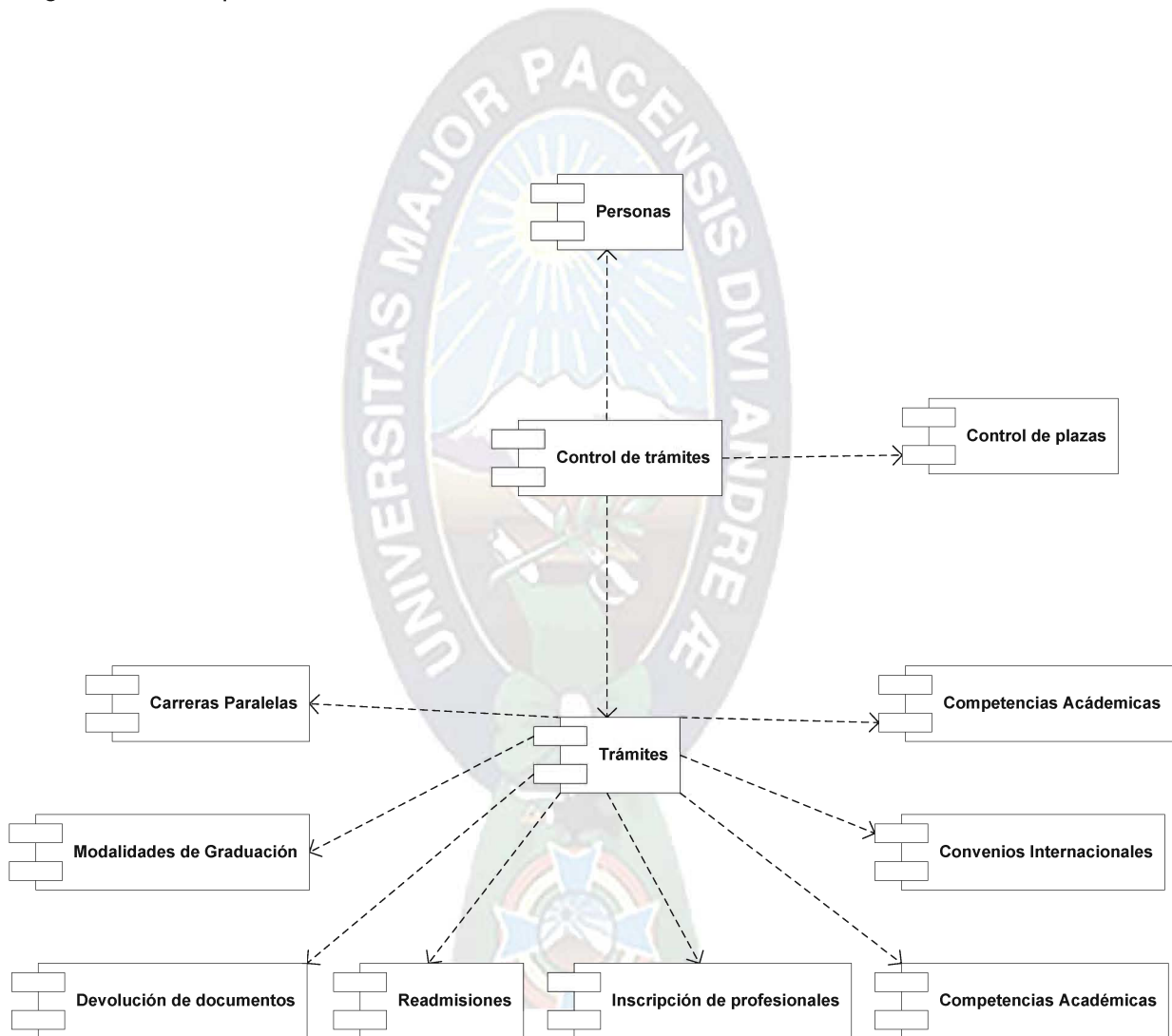


Fig. 44 Diagrama de componentes

Fuente: Elaboración Propia

3.4.1 Pantallas del sistema

La siguiente figura muestra el ingreso al sistema mediante el login y password del usuario.

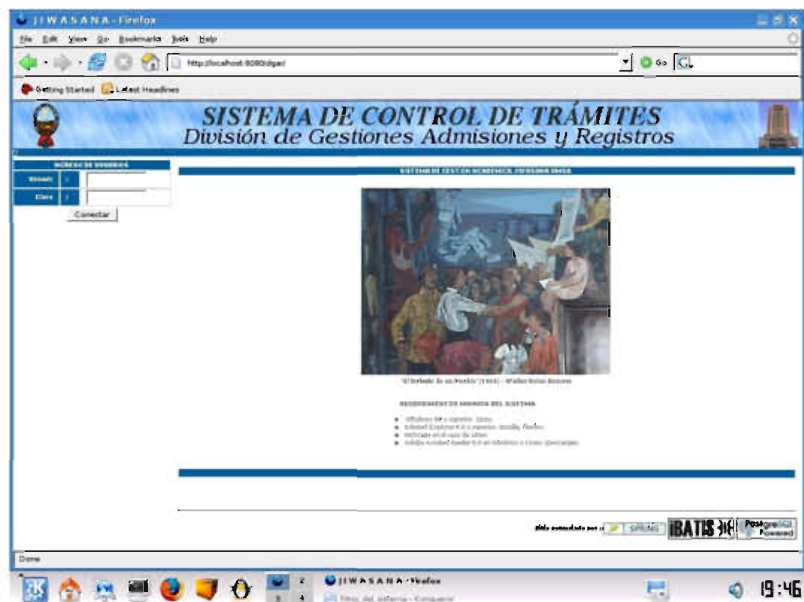


Fig. 45 Pantalla de ingreso al sistema
Fuente: Elaboración Propia

La siguiente figura muestra la búsqueda de la persona por CI.

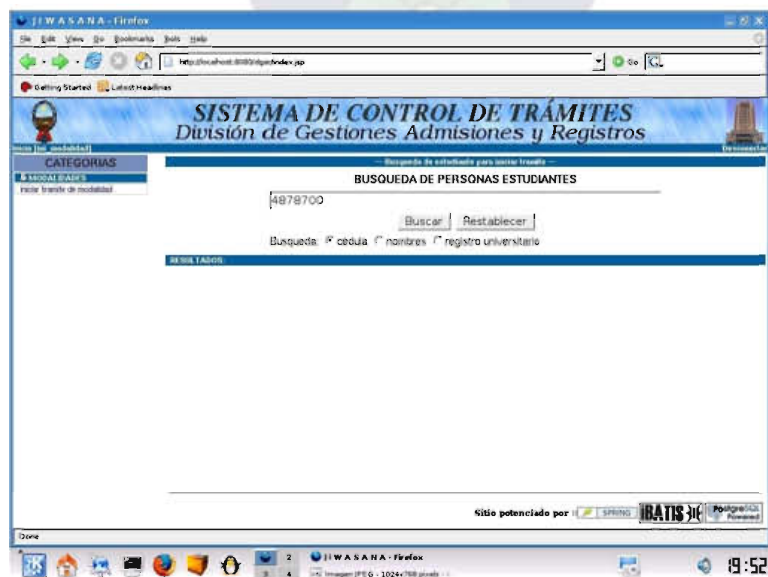


Fig. 46 Pantalla de búsqueda de la persona por CI
Fuente: Elaboración Propia

La siguiente figura muestra los resultados de la búsqueda de la persona por CI.

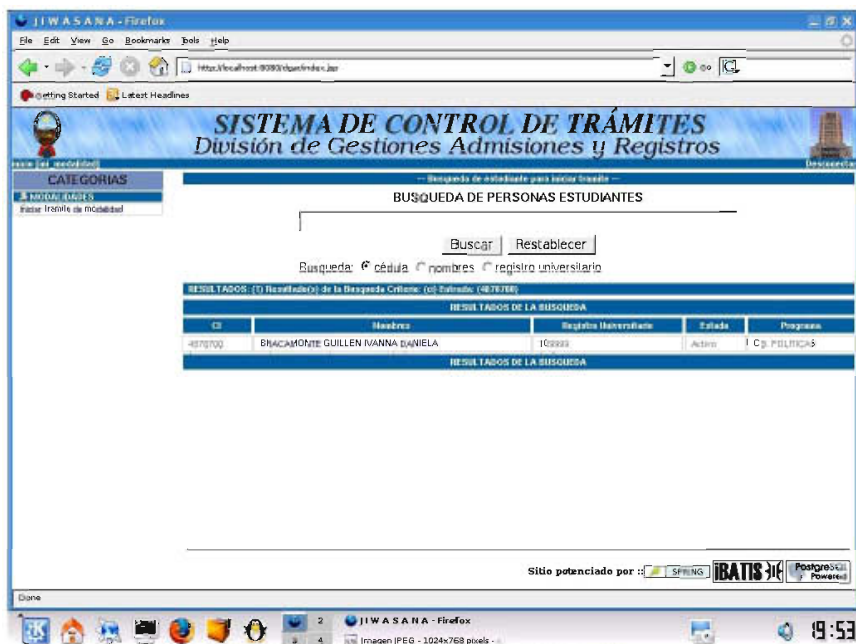


Fig. 47 Pantalla de resultados de la búsqueda de la persona por CI

Fuente: Elaboración Propia

La siguiente figura muestra el inicio de trámite para modalidades de graduación.

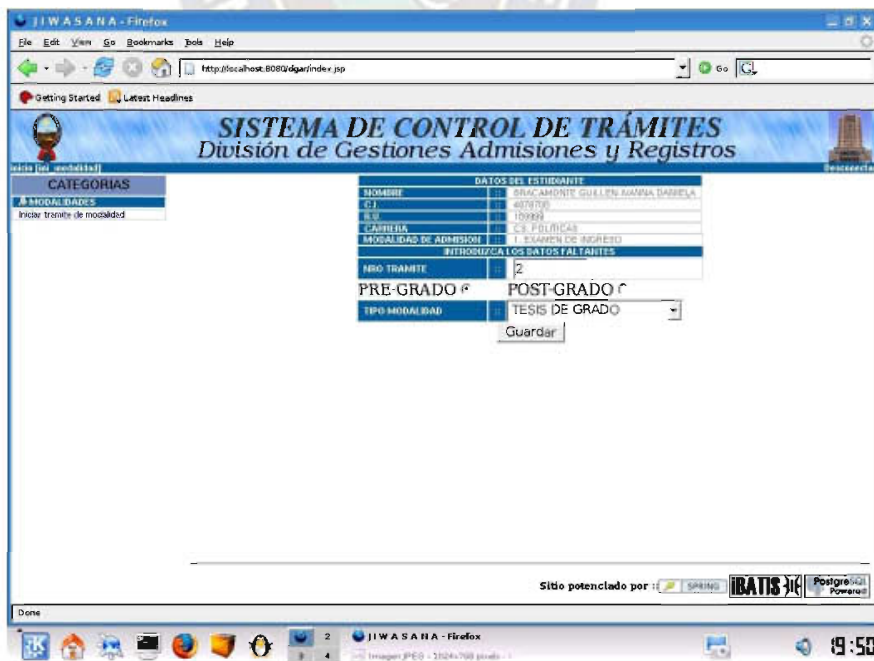


Fig. 48 Pantalla inicio de trámite de modalidades de graduación

Fuente: Elaboración Propia

La siguiente figura muestra el inicio de trámite para convenios internacionales.

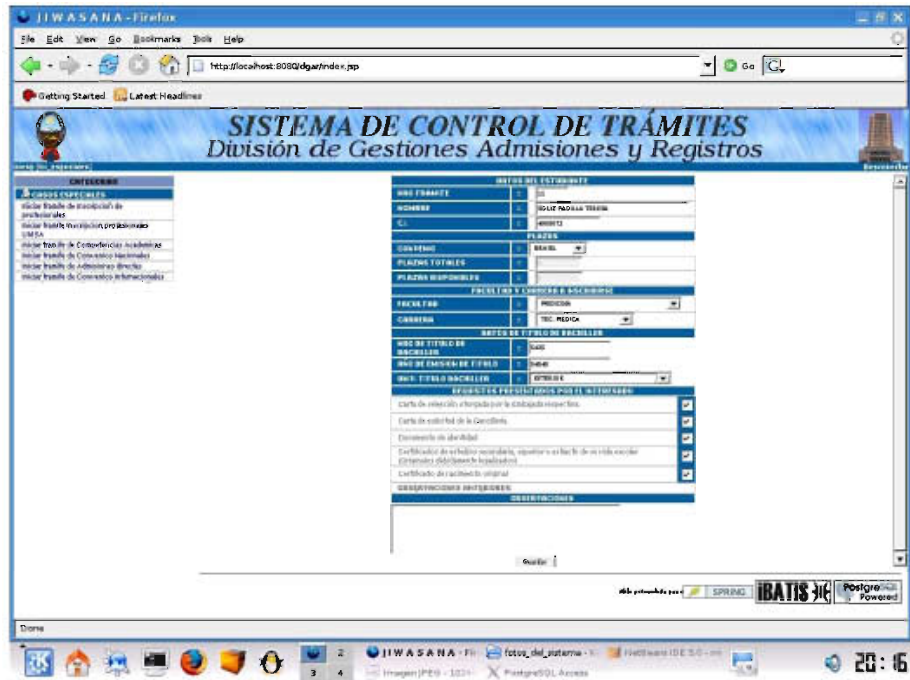


Fig. 49 Pantalla de inicio de trámite de convenio internacional

Fuente: Elaboración Propia

La siguiente figura muestra los trámites pendientes de un usuario.



Fig. 50 Pantalla de trámites pendientes

Fuente: Elaboración Propia

La siguiente figura muestra el registro de requisitos presentados por el interesado

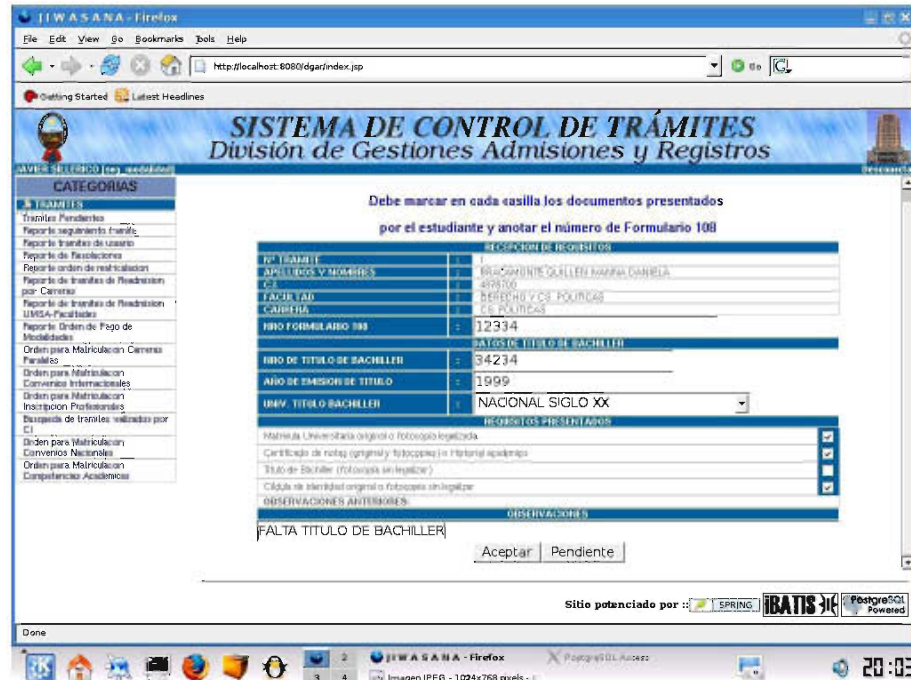


Fig. 51 Pantalla de llenado de requisitos del interesado

Fuente: Elaboración Propia

La siguiente figura muestra los trámites realizados por un estudiante

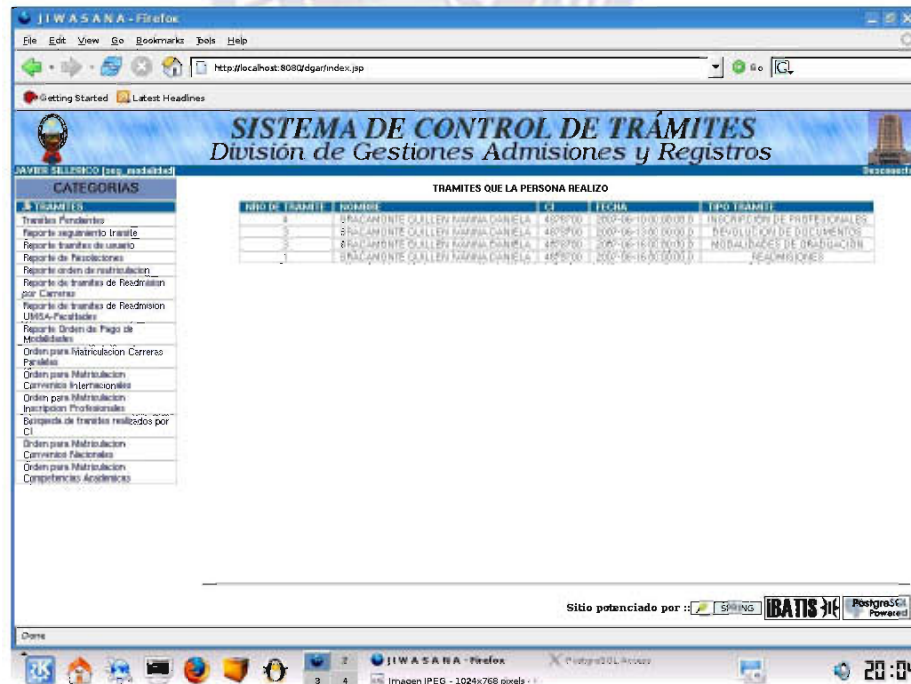


Fig. 52 Pantalla de trámites realizados por un estudiante

Fuente: Elaboración Propia

3.4.2 Prueba de la caja blanca

También llamada prueba de la caja de cristal, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba [Pressman, 2002]. Esta prueba nos permite obtener un valor que es la complejidad ciclomática.

Complejidad Ciclométrica

La complejidad ciclométrica es una métrica de software que proporciona una medida cuantitativa de la complejidad lógica de un programa.

La complejidad ciclométrica se basa en la teoría gráfica y se calcula de una de tres maneras que son:

- El número de regiones corresponde a la complejidad ciclométrica.
- La complejidad ciclométrica $V(G)$, de una gráfica de flujo, G se define como:
$$V(G) = E - N + 2$$

Donde:

 - E = Número de aristas.
 - N = Número de nodos de la gráfica de flujo.
- La complejidad ciclométrica $V(G)$, de una gráfica de flujo G también se define como:
$$V(G) = P + 1$$

Donde:

 - P = Es el número de nodos predicado incluidos en la gráfica de flujo G

Para realizar esta prueba se tomara en cuenta el diseño procedimental del módulo de carreras paralelas y el diseño del grafo respectivo como muestra la siguiente figura.

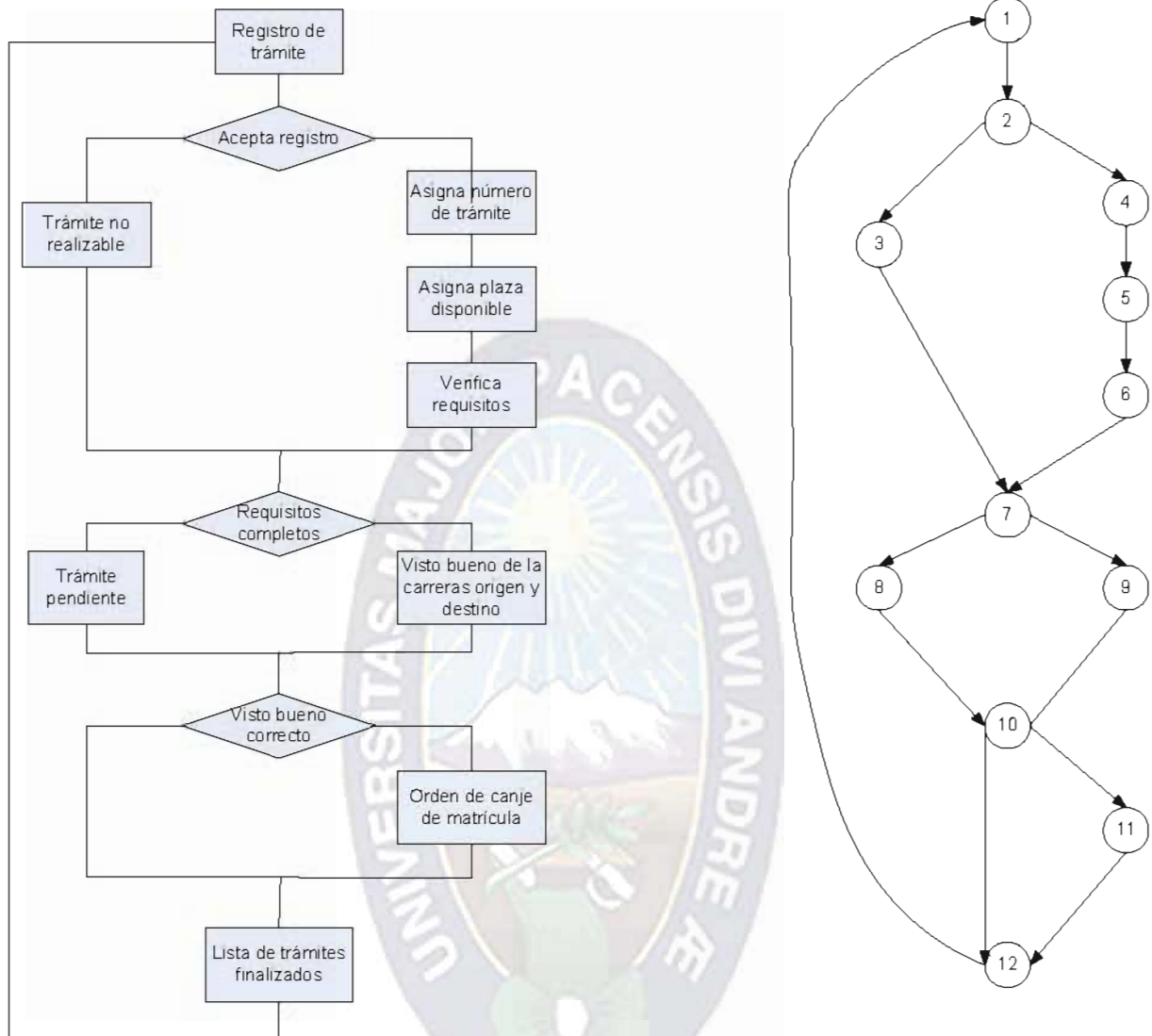


Fig. 53 Diseño procedimental y grafo de flujo

Fuente: Elaboración Propia

De acuerdo al grafo se obtienen los siguientes caminos independientes:

Camino 1: 1-2-4-5-6-7-8-10-12

Camino 2: 1-2-4-5-6-7-9-10-12

Camino 3: 1-2-4-5-6-7-8-10-11-12

Camino 4: 1-2-4-5-6-7-9-10-11-12

Camino 5: 1-2-3-7-8-10-12

Camino 6: 1-2-3-7-9-10-12

Camino 7: 1-2-3-7-8-10-11-12

Camino 8: 1-2-3-7-9-10-11-12

Aplicando las tres formas de obtener la complejidad ciclomática se tiene:

- El grafo tiene cinco regiones.
- $V(G) = 15 - 12 + 2 = 5$
- $V(G) = 4 \text{ nodos predicado} + 1 = 5$

Por lo tanto la complejidad ciclomática del grafo de flujo es 5.

3.4.3 Prueba de la caja negra

Una de las pruebas más utilizadas para verificar la funcionalidad del sistema es la prueba de la caja negra que consiste en introducir datos de entrada al sistema para luego ver la salida obtenida mediante operaciones internas del sistema.

A continuación se muestra la prueba realizada para el módulo de readmisiones cuyo diagrama de secuencia es la Fig. 31.

Caso de Prueba: Control de Readmisiones

Entradas: - Número de carnet de identidad del universitario que solicita readmisión

Resultados: - Se le asigna un número de trámite
- Se verifica datos del interesado
- Se controla el seguimiento del trámite
- Emisión de orden de canje de matrícula

Condición: - La persona interesada debe haber abandonado la universidad por más de tres gestiones

Procedimiento de Prueba

Una vez definido el caso de prueba se debe de verificar si se obtienen los resultados necesarios y para eso se debe de tomar como punto de partida el caso de uso control de readmisiones.

Caso de uso: Control de Readmisiones

1. Desplegar pantalla principal.
2. Introducir logín y password del usuario del sistema y presiona el botón "Aceptar".
3. El sistema valida al usuario.
4. Si es usuario esta autorizado entra para a otra pantalla para poder iniciar el trámite.
5. El usuario introduce el CI del estudiante interesado y presiona el botón "Buscar".
6. El sistema realiza la búsqueda de datos del interesado en la base de datos de matriculación.

7. El sistema obtiene los datos del interesado y le asigna un número de trámite y presiona el botón "Guardar".
8. Luego ingresa el usuario encargado de la revisión mediante el login y password correspondiente.
9. El sistema valida el ingreso del usuario.
10. El sistema despliega la pantalla de trámites pendientes de readmisiones para revisión de documentos.
11. El usuario hace un clic sobre el nombre de la persona a ser revisados sus requisitos.
12. El usuario entra en otra pantalla donde llena datos de bachillerato y marca cada uno de los requisitos presentados y faltara más de un requisito presiona "Pendiente".
13. Si los requisitos están completos presiona "Guardar".
14. Luego ingresa el usuario encargado del visto bueno de la carrera mediante el login y password correspondiente.
15. El sistema valida el ingreso del usuario.
16. El sistema despliega la pantalla de trámites pendientes de readmisiones para visto bueno de la carrera correspondiente.
17. El usuario hace un clic sobre el nombre de la persona a ser atendida.
18. El usuario entra en otra pantalla donde debe llenar el número de materias aprobadas por el interesado y debe marcar una casilla si es que el trámite no tiene observaciones y presiona "Pendiente".
19. Si no hay observaciones presiona "Guardar".
20. Luego ingresa el usuario encargado del visto bueno del jefe de Gestiones mediante el login y password correspondiente.
21. El sistema valida el ingreso del usuario.
22. El sistema despliega la pantalla de trámites pendientes de readmisiones para visto bueno del jefe de gestiones correspondiente.
23. El usuario hace un clic sobre el nombre de la persona a ser atendida.
24. El usuario entra en otra pantalla donde debe marcar una casilla si es que el trámite no tiene observaciones y presiona "Pendiente".
25. Si no hay observaciones presiona "Guardar".
26. Luego ingresa el usuario encargado de la emisión de orden de canje de matrícula mediante el login y password correspondiente.
27. El sistema valida el ingreso del usuario.
28. El sistema despliega la pantalla de trámites pendientes de readmisiones para emisión de orden de matrícula.

29. El usuario hace un clic sobre el nombre de la persona a ser atendida.
30. El usuario entra en otra pantalla donde debe marcar una casilla si es que el trámite no tiene observaciones y presiona "Pendiente".
31. Si no hay observaciones presiona "Guardar".
32. Luego el usuario selecciona el menú de Reporte de Orden de Canje de Matrícula.
33. El usuario introduce el número de trámite y presiona el botón "Buscar".
34. El sistema despliega otra pantalla donde muestra los datos a ser impresos en el reporte de Orden de Canje de Matrícula.
35. Luego hace clic en "Ver documento PDF" e imprime el documento.

De acuerdo a los pasos realizados en esta prueba se puede comprobar que se obtuvieron los resultados planificados por lo que el proceso funciona correctamente.

Se realizaron todas las pruebas necesarias para la implementación del sistema, con lo que se verificó que el sistema cubre todos los requerimientos de los usuarios.

3.5 Despliegue

Es la etapa del desarrollo que describe la configuración del sistema para su ejecución en un ambiente del mundo real. Para el despliegue, se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia [Rumbaugh, 1999].

3.5.1 Modelo de Despliegue

A continuación se muestra el modelo de despliegue, el cual consta de un servidor y varias terminales que serán responsables de la atención de los trámites dentro de la División de Gestiones Admisiones y Registros.

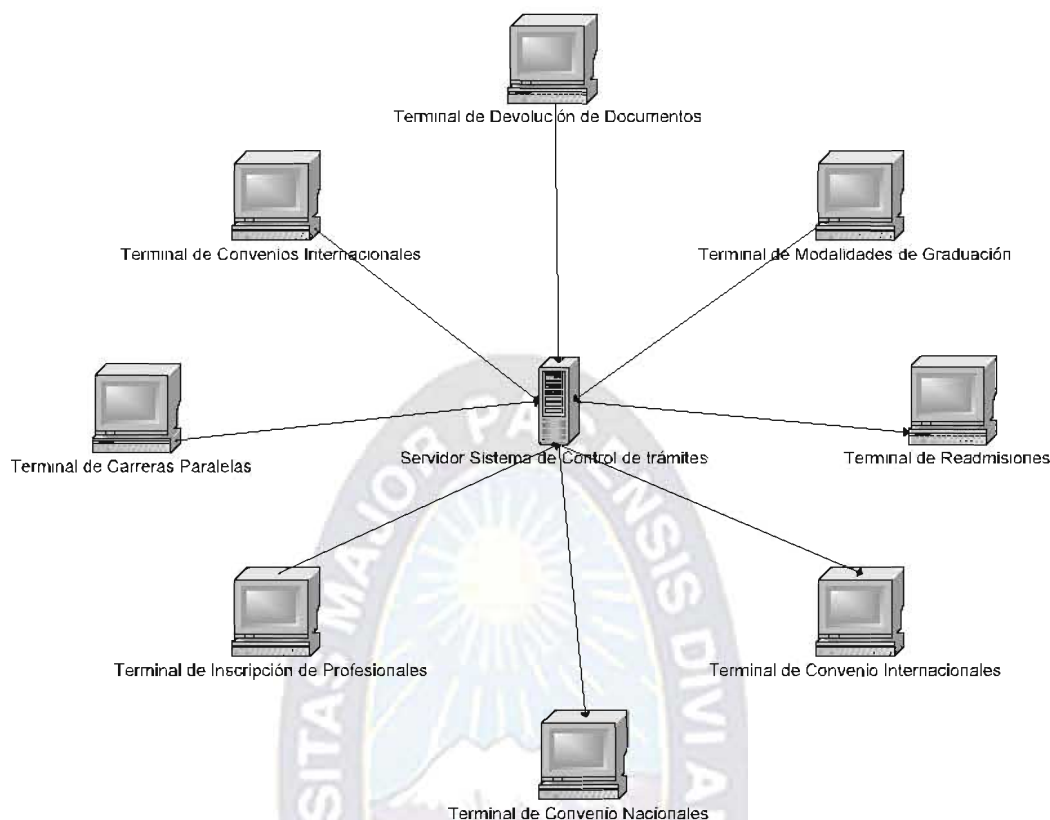


Fig. 54 Modelo de despliegue del sistema

Fuente: Elaboración Propia

3.6 SEGURIDAD DEL SISTEMA

Al momento de implementar un sistema de información es importante considerar medidas de seguridad que ayuden a prevenir diferentes ataques de intrusos al sistema, para que de esta manera se cuide la información que es vital dentro del mismo.

Existen varias formas de proteger la información pero las que se utilizan para este sistema son el TCPWrappers, permite reducir el acceso a los servicios del servidor; puede rechazar o aceptar el ingresos de IP desconocidos, ya que la implementación del sistema será dentro de una intranet entonces se configuró esta herramienta para que solamente acepte el ingreso de números IP`s autorizados rechazando absolutamente otro tipo de infiltración al servidor.

Para el resguardo de la base de datos se pusieron contraseñas tanto en la creación de la base de datos como también en cada tabla a utilizar, de tal forma que solamente pueda acceder a la base de datos la persona encargada del mantenimiento del mismo, también se obtienen backups de seguridad cada día con el objetivo de evitar pérdidas de información.

Se esta viendo la posibilidad de comprar una UPS que es un Sistema de Fuerza Ininterrumpible es un equipo cuya función principal es evitar una interrupción de voltaje en el servidor del sistema, esto para los casos de cortes de energía eléctrica, de esa forma evitamos perder la información antes y durante un corte de energía eléctrica.

3.7 CAPACITACIÓN DEL PERSONAL

Una vez terminada las pruebas del sistema se realizó la capacitación del personal, esta capacitación fue muy importante ya que con el mismo se logró familiarizar al usuario con el sistema.

La capacitación se la realizó módulo por módulo en el cual se mostró como iniciar un trámite, como realizar los pasos para el seguimiento del trámite y como obtener reportes de trámites finalizados.

Esta capacitación se la hizo a todos los funcionarios que atienden los procesos de modalidades de graduación, readmisiones, devolución de documentos, carreras paralelas y admisiones especiales.

La capacitación se realizó bajo el siguiente cronograma:

PROCESO	NRO DE PERSONAS CAPACITADAS	ABRIL		MAYO			
		SEM. 3	SEM. 4	SEM. 1	SEM. 2	SEM. 3	SEM. 4
MODALIDADES DE GRADUACIÓN	5						
CARRERAS PARALELAS	2						
READMISIONES	1						
DEVOLUCION DE DOCUMENTOS	1						
INSCRIPCION DE PROFESIONALES	1						
CONVENIOS NACIONALES	1						
CONVENIOS INTERNACIONALES	1						
COMPETENCIAS ACADÉMICAS	1						

Tabla 3: Cronograma de capacitación del personal de la División de Gestiones

Fuente: Elaboración Propia

Después de las pruebas, capacitación y a requerimiento del personal de la División de Gestiones se elaboró una opción de ayuda dinámica dentro el sistema para que en cualquier momento el usuario pueda obtener asesoramiento para la manipulación del sistema. Además se entregó el manual de usuario donde muestra los pasos para poder realizar el seguimiento de los trámites y como obtener los reportes del sistema.



CAPÍTULO IV

CONCRECIÓN DEL MODELO

En este capítulo se muestra la concreción del sistema y conceptos importantes de métricas de calidad que son necesarios para las pruebas realizadas a un sistema.

4 MÉTRICAS DE CALIDAD

Para la ingeniería de software es fundamental la medición del software, ya que ayuda al control de calidad del sistema en cada una de sus fases.

Existen muchas herramientas para la medición de calidad del software para el presente proyecto se tomaran en cuenta los siguientes:

- Métricas orientadas a la función
- Métricas de mantenimiento
- Confiabilidad
- Portabilidad

4.1 Métricas orientadas a la función

La funcionalidad es una característica muy importante dentro del desarrollo del software.

La métrica de Punto Función (PF) se usa para:

- Estimar el costo o el esfuerzo requerido para diseñar, codificar y probar el software.
- Predecir el número de errores que se encontraran durante la prueba.
- Pronosticar el número de componentes, de líneas de código proyectada, o ambas en el sistema implementado [Pressman, 2005].

La funcionalidad de un sistema se mide tomando algunas características de dominio de información, donde se le asocia un valor de complejidad. Para realizar la medición de los puntos de función se debe determinar características como: número de entradas del usuario, número de salidas del usuario, número de peticiones del usuario, número de archivos y número de interfaces externas.

De acuerdo a los anteriores parámetros se tiene la siguiente tabla de valores:

VALOR DEL DOMINIO DE INFORMACIÓN	CONTEO	FACTOR DE PONDERACION			TOTAL
		SIMPLE	PROMEDIO	COMPLEJO	
Entradas externas	10	3	4	6	30
Salidas externas	20	4	5	7	80
Consultas externas	7	3	4	6	21
Archivos de lógica interna	20	7	10	15	140
Archivos de interfaz externa	5	5	7	10	25
TOTAL					296

Tabla 4: Cálculo de Cuenta Total de la métrica de Punto Función
Fuente: Elaboración Propia

NRO	PREGUNTAS	FACTOR DE AJUSTE
1	El sistema requiere respaldo y recuperación confiables	5
2	Se requieren comunicaciones de datos especializadas para transferir información a la aplicación, u obtener de ella.	5
3	Hay funciones distribuidas de procesamiento	3
4	El desempeño es crítico	4
5	El sistema se ejecutará en un entorno existente que tiene un uso pesado de operaciones.	1
6	El sistema requiere entrada de datos en línea.	5

7	La entrada de datos en línea requiere de la transacción de entrada se construya en varias pantallas u operaciones	4
8	Los archivos lógicos internos se actualizan en línea.	4
9	Las entradas, las salidas, los archivos o las consultas son complejos	3
10	Es complejo el procesamiento interno.	3
11	El código diseñado será reutilizable	5
12	Se incluyen la conversión e instalación en el diseño.	3
13	Esta diseñado el sistema para instalaciones múltiples en diferentes organizaciones	5
14	La aplicación esta diseñada para aplicar el cambio y para que el usuario lo use fácilmente	5
TOTAL		55

Tabla 5: Cálculo de Ajuste de complejidad

Fuente: Elaboración Propia

Determinar el valor de complejidad es subjetivo, para calcular los puntos de función se realiza la siguiente operación.

$$PF = \text{Cuenta total} * (0.65 + 0.01 * \sum Fi)$$

Donde:

Cuenta total: Es la suma de los valores obtenidos en el cálculo de métricas de punto función obtenidas.

Fi : Son valores de ajuste de complejidad dadas según las respuestas obtenidas.

Reemplazando valores obtenidos se tiene el siguiente valor:

$$PF = 296 * (0.65 + 0.01 * 55)$$

$$PF = 355.2$$

Realizando una comparación con la escala de valores para el Punto Función dada por:

NRO	ESCALA	RESULTADO
1	PF > 300	Optimo
2	200 > PF > 300	Buena
3	100 > PF > 200	Suficiente
4	PF < 100	Deficiente

Tabla 6: Escala de valores para Punto Función
Fuente: Elaboración Propia

El sistema tiene una funcionalidad óptima ya que obtuvo un valor mayor a 300.

4.2 Métricas de mantenimiento

Para aplicar la métrica de mantenimiento se utilizará el Estándar IEEE 982 que nos da un índice de madurez del software que proporciona un indicador de la estabilidad de un producto software.

Esta métrica esta dada por los siguientes parámetros:

- Mt = Número de módulo de la versión actual.
- Fi = Número de módulos en la versión actual que han cambiado
- Fa = Número de módulos en la versión actual que se han añadido
- Fd = Número de módulos en la versión anterior que se han borrado en la versión actual.

El índice de madurez de software (IMS) se calcula mediante la siguiente fórmula:

$$IMS = [Mt - (Fa + Fi + Fd)]/Mt$$

Realizando las pruebas necesarias se obtuvieron los siguientes valores

$$M_t = 20$$

$$F_i = 1$$

$$F_a = 1$$

$$F_d = 0$$

Remplazando valores en la fórmula anterior se tiene:

$$IMS = [20 - (1 + 1 + 0)]/20$$

$$IMS = 0.9$$

De acuerdo al valor obtenido se deduce que el Índice de Madurez del Proyecto es del 90 %.

4.3 Portabilidad

La portabilidad no permite transportar datos de un sistema a otro, en el caso del presente sistema es portable en un 99% ya que puede ser instalado en Windows como Linux., ya que utiliza el lenguaje de programación Java y la base de datos no tiene funciones por lo que puede ser instalado para cualquier versión de Postgres.

4.4 Confiabilidad

Es la cantidad de tiempo en que el software esta disponible para usarse según los siguientes atributos: madurez, tolerancia a fallas y facilidad de recuperación.

La confiabilidad del sistema esta dado por la siguiente relación:

$$R(t) = R_1(t) * R_2(t) * R_3(t) \dots R_n(t)$$

Donde R(t) esta dado por:

$$R(t) = e^{-\lambda P(t)}$$

Donde:

R(t) = Es la confiabilidad del subsistema en un tiempo t

P(t) = Es la probabilidad de falla

λ = Es la tasa constante de fallo

Después de realizar las pruebas necesarias al sistema se obtuvieron los siguientes resultados:

R1 = Registro y control de modalidades de graduación	= $R(t) = e^{-0.8*0.2} = 0.85$
R2 = Registro y control de readmisiones	= $R(t) = e^{-0.9*0.1} = 0.91$
R3 = Registro y control de devolución de documentos	= $R(t) = e^{-0.9*0.1} = 0.91$
R4 = Registro y control de carreras paralelas	= $R(t) = e^{-0.8*0.2} = 0.85$
R5 = Registro y control de inscripción de profesionales	= $R(t) = e^{-0.9*0.1} = 0.91$
R6 = Registro y control de convenios internacionales	= $R(t) = e^{-0.8*0.2} = 0.85$
R7 = Registro y control de convenios nacionales	= $R(t) = e^{-0.9*0.1} = 0.91$
R8 = Registro y control de competencias académicas	= $R(t) = e^{-0.9*0.1} = 0.91$

La confiabilidad en paralelo se obtiene de la siguiente forma:

$$R9 = 1 - (1 - R2) * (1 - R1) = 1 - (1 - 0.91) * (1 - 0.85) = 0.98$$

$$R10 = 1 - (1 - R9) * (1 - R4) = 1 - (1 - 0.98) * (1 - 0.85) = 0.99$$

$$R11 = 1 - (1 - R5) * (1 - R6) = 1 - (1 - 0.91) * (1 - 0.85) = 0.98$$

$$R12 = 1 - (1 - R8) * (1 - R7) = 1 - (1 - 0.91) * (1 - 0.91) = 0.99$$

La confiabilidad en paralelo es

$$R13 = R9 * R7 = 0.98 * 0.91 = 0.89$$

$$R14 = R4 * R11 = 0.85 * 0.98 = 0.93$$

De acuerdo a la ecuación planteada para medir la confiabilidad se tiene:

$$R = 1 - (1 - R9) * (1 - R10) * (1 - R11) * (1 - R12) * (1 - R13) * (1 - R14)$$

$$R = 1 - (1 - 0.98) * (1 - 0.99) * (1 - 0.98) * (1 - 0.99) * (1 - 0.89) * (1 - 0.93)$$

$$R = 0.99$$

De acuerdo al resultado anterior se deduce que el sistema tiene un 99% de confiabilidad.

4.5 CONCLUSIONES

Se logró mejorar el seguimiento y control de trámites que involucran los procesos de modalidades de graduación, carreras paralelas, readmisiones, devolución de documentos, inscripción de profesionales, convenios nacionales, convenios internacionales y competencias académicas, ya que ahora existe un control estricto en cada paso del trámites, además el sistema realiza reportes del seguimiento de cada trámite con lo que se logró un óptimo control en cada momento, los reportes obtenidos con el sistema se pueden ver en anexo L, los cuales nos permiten sacar algunas estadísticas de trámites realizados en la División.

De acuerdo a las pruebas realizadas con el sistema, se pudieron obtener los siguientes resultados.

ACTIVIDAD	ANTES	CON EL SISTEMA
Verificación del estado de los trámites en cualquier momento.	De 3 a 5 minutos	10 segundos
Reportes de trámites finalizados de modalidades de graduación	De 2 a 3 días	20 segundos
Reportes de trámites finalizados de readmisiones	De 1 a 2 días	15 segundos
Reportes de trámites finalizados de carreras paralelas	De 1 a 2 días	15 segundos
Reportes de trámites finalizados de devolución de documentos	De 6 a 8 horas	15 segundos
Reportes de trámites finalizados de admisiones especiales	De 6 a 8 horas	15 segundos
Orden de canje de matrícula	De 1 a 2 minutos	5 segundos
Control automatizado de cupos	De forma manual	Automatizado
Reporte del seguimiento de trámites	No existía	10 segundos

Tabla 7: Tabla comparativa de resultados

Fuente: Elaboración Propia

De acuerdo a la anterior tabla se puede evidenciar que se logró el objetivo de mejorar el control del seguimiento de trámites y minimizar el tiempo de emisión de reportes.

Se diseñó una máquina de Turing para controlar el seguimiento de los trámites el cual se encuentra implantado en la base de datos del sistema.

La realización de la máquina de Turing y el hecho de tener una base de datos normal donde se pueda almacenar información de todos los trámites implica tener una base de datos híbrida por lo que también se logró ese objetivo.

Se elaboran reportes sobre el seguimiento de los trámites, trámites finalizados de acuerdo a intervalos de tiempo, ordenes de canje de matrícula al finalizar un trámite, resoluciones, ordenes para defensa, trámites realizados por una persona, trámites atendidos por un usuario. La emisión de los reportes es cuando el usuario así lo desee y en cualquier momento optimizando de gran manera el manejo de la información y obteniendo datos estadísticos que son importantes para proyecciones a futuro.

4.6 RECOMENDACIONES

Se puede utilizar una máquina de Turing en procesos donde exista la toma de decisiones para poder tener un control más dinámico.

Cuando se realiza la tabla de estados de una máquina de Turing se debe de tener mucho cuidado y lógica, ya que un buen diseño implica un buen funcionamiento

Sería recomendable que el sistema posteriormente interactúe con las unidades académicas de la universidad como ser los postgrados, decanaturas y carreras para tener un control aún más óptimo en el seguimiento de los trámites.

De acuerdo a los alcances mencionados en el capítulo I del presente documento el sistema solo cubre una parte de los procesos realizados por la División de Gestiones Admisiones y Registros, por lo que sería adecuado automatizar los procesos faltantes para tener un sistema completo de seguimiento de trámites en la División de Gestiones.

REFERENCIAS BIBLIOGRÁFICAS

[Booch, 2002] Booch, Grady, 2002: *Análisis y Diseño Orientado a Objetos con Aplicaciones*, 1ra edición, 420 pags, McGrawHill, Mexico.

[Castaño, 2001] Castaño, Adoración de Miguel, 2001: *Diseño de Bases de Datos*, 1ra edición, 484 pags, Alfaomega, Madrid España.

[Castro, 2001] Castro Elizabeth, 1997: *XML*, 3ra edición, 244 pags, Prentice-Hall, España

[Ceballos, 2002] Ceballos Sierra, Javier, 2002: *JAVA*, 1ra edición, 298 pags, Alfaomega, Madrid-España

[DGAR, 2002] División de Gestiones Admisiones y Registro, 2002: *Levantamiento de campo de la División de Gestiones Admisiones y Registros*, 1ra edición, 300 pags, Editorial Universitario UMSA, La Paz-Bolivia

[DGAR, 2004] División de Gestiones Admisiones y Registro, 2003: *Manual de procedimientos régimen estudiantil*, 1ra edición, 63 pags, Editorial Universitario UMSA, La Paz-Bolivia

[Froufe, 2002] Froufe Quintas, Agustín, 2002: *JavaServer Pages manual de usuario y tutorial*, 264 pags. 1ra edición, Alfaomega, Madrid-España

[Jacobson, 2001] Jacobson, 2001: *Proceso Unificado*, 1ra edición, 285 pags, Prentice-Hall, Mexico.

[Kendall, 1997] Kendall Kenneth E. y Kendall, Julie E., 1997: *Análisis y Diseño de Sistemas*, 3ra edición, 56-57 pags, Prentice-Hall, Mexico.

[Larman, 1997] Larman Craig, 1997: *UML y patrones*, 1ra edición, 507 pags, Prentice-Hall, Mexico.

[Mahalanabis, 1999] Mahalanabis, A., 2001: *Introducción a la Ingeniería de Sistemas*, 1ra edición, 530 pags, McGrawHill, Mexico.

[Pressman, 2003] Pressman, Roger S., 2003 : *Ingeniería del software un enfoque práctico*, 601 pags. 5ta edición, McGrawHill, Concepción Fernandez. Madrid España.

[Pressman, 2005] Pressman, Roger S., 2005 : *Ingeniería del software un enfoque práctico*, 958 pags. 6ta edición, McGrawHill, México.

[Senn, 2004] Senn, James A., 1992 : *Análisis y Diseño de Sistemas de Información*, 600 pags. 2da edición, McGrawHill, Mexico

[Mercer, 2002] Mercer, Dave, 2001 : *Fundamentos de programación en XML*, 436 pags. 1ra edición, McGrawHill, Colombia

[Weitzenfeld, 2002] Weitzenfeld, Alfredo, 2002 : *Ingeniería de Software Orientado a Objetos con UML, Java e Internet*, 680 pags. 1ra edición, Thomson

[web 01] Gervás, Pablo, 2002: *Proceso de desarrollo de software*, Disponible: <http://www.sei.cmu.edu/cmm/obtain.cmm.html>

[web 02] Banco Internacional de Desarrollo, 2005: *Marco Lógico*, Disponible: <http://www.iadb.org/int/rtc/ecourses/esp>