

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



PROYECTO DE GRADO

**SISTEMA DE CONTROL ADMINISTRATIVO
CASO: CLÍNICA CAJA PETROLERA DE SALUD (CPS)**

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

POSTULANTE : Angela Mélany Mejillon Tiñini

TUTOR : Lic. Nancy Orihuela Sequeiros

REVISOR : Lic. Jose Luis Zeballos Abasto

LA PAZ – BOLIVIA

2009

AGRADECIMIENTOS

En primer lugar quiero dar gracias a Dios, por darme la oportunidad de llegar a ser lo que soy hoy y llenarme de bendiciones todos los días.

Agradecer a mis padres, Jaime Mejillon y Lidia Tiñini por todo el esfuerzo que ambos orientaron durante mi formación académica y personal, dándome todo su apoyo y comprensión.

A mi tutor Lic. Nancy Orihuela S. por su apoyo, guía y comprensión.

A mi revisor Lic. Jose Luis Zeballos A. por su constante apoyo, colaboración y comprensión.

Al Lic. Juan Carlos Guzmán administrador de la Clínica CPS por su tiempo y apoyo para el desarrollo del presente proyecto.

Angela Mélany Mejillon Tiñini

Melany_a@hotmail.com

Dedicado a Dios, el ser supremo, omnipotente, omnipresente y omnisciente quien fue mi guía; a mis padres y hermano, por su aliento y apoyo incondicional en la realización de este proyecto.

RESUMEN

En la actualidad y debido a que la información generada en la mayoría de las instituciones de mediana o gran envergadura, llega a tener dimensiones que son difícilmente manejables por medios manuales, es necesario utilizar herramientas que permitan a las organizaciones, manejar la información que generan de forma eficiente, para brindar mejores servicios a la sociedad o convertirse en entes competitivos en el medio.

En la clínica CPS de la ciudad de La Paz no existe un manejo de la información administrativa referida al control de recursos materiales y personal, que permita el seguimiento oportuno de cada una de las actividades que desarrolla la institución.

Es por esta razón que el presente proyecto implementa un sistema de información automatizado que permita un manejo adecuado y eficiente de la información de la institución con respecto a la parte administrativa correspondiente al control de recursos materiales (reposición de materiales, mantenimiento de equipos) y el control de personal (vacaciones, permisos, bajas, comisiones, cambio voluntario de turno).

Para el desarrollo del presente proyecto se utiliza la metodología de desarrollo de software RUP con ayuda de la Herramienta Case (Enterprise Architect) para el diseño y construcción del sistema. Debido a la complejidad de los datos y la relación entre estos se utiliza para la implementación de la base de datos el modelo de datos objeto relacional.

Se ha concluido que la utilización del Sistema de Control Administrativo implementado en la Clínica de la Caja Petrolera de Salud contribuye y facilita los procesos al departamento de administración de la Clínica logrando integrar los módulos de Control de recursos y Control de personal que mejora el manejo de la información generada por la institución.

ÍNDICE

CAPÍTULO I MARCO CONTEXTUAL

1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	1
1.2.1 Antecedentes de la Institución.....	2
1.2.1.1 Misión.....	3
1.2.1.2 Visión.....	3
1.2.1.3 Ubicación.....	4
1.2.2 Antecedentes del Proyecto.....	4
1.3 PROBLEMÁTICA.....	5
1.4 OBJETO DE ESTUDIO.....	6
1.5 OBJETIVO GENERAL.....	7
1.6 OBJETIVOS ESPECÍFICOS.....	7
1.7 JUSTIFICACIÓN.....	8
1.7.1 Justificación Técnica.....	8
1.7.2 Justificación Económica.....	8
1.7.3 Justificación Social.....	9
1.8 APORTE TEÓRICO.....	9
1.9 APORTE PRÁCTICO.....	9
1.10 LÍMITES Y ALCANCES.....	10
1.11 PLANIFICACIÓN DEL PROYECTO.....	10

CAPÍTULO II MARCO TEÓRICO

2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	12
2.2 MODELO.....	12
2.3 RUP (PROCESO UNIFICADO RACIONAL).....	15
2.3.1 Ciclo de vida de RUP.....	15
2.3.1.1 Etapas de RUP.....	16
2.3.1.2 Etapa de ingeniería.....	16
2.3.1.3 Etapa de producción.....	17

2.3.2 Fases del RUP.....	17
2.3.2.1 Primera fase de RUP: Inicio.....	18
2.3.2.2 Segunda fase de RUP: Elaboración.....	20
2.3.2.3 Tercera fase de RUP: Construcción.....	20
2.3.2.4 Cuarta fase de RUP: Transición.....	21
2.4 LENGUAJE DE MODELADO UNIFICADO UML	22
2.4.1 Características de UML.....	22
2.4.2 Diagramas de UML.....	23
2.4.2.1 Diagrama de casos de uso.....	23
2.4.2.2 Diagrama de clases.....	25
2.4.2.3 Diagrama de interacción.....	25
2.4.2.3.1 Diagrama de secuencia	26
2.4.2.3.2 Diagrama de colaboración.....	27
2.4.2.4 Diagramas de comportamiento.....	27
2.4.2.4.1 Diagrama de estados.....	28
2.4.2.4.1 Diagrama de actividades.....	29
2.4.2.5 Diagrama de implementación.....	29
2.4.2.5.1 Diagrama de componentes.....	29
2.4.2.5.2 Diagrama de despliegue.....	30
2.5 HERRAMIENTA CASE.....	31
2.5.1 Definición.....	31
2.5.2 Enterprise Architect (EA).....	31
2.5.2.1 Características de EA.....	31
2.5.2.2 Ingeniería inversa.....	33
2.6 BASE DE DATOS OBJETO RELACIONAL.....	35
2.7 LAN.....	36
2.8 SEGURIDAD.....	37
2.9 CALIDAD DE SOFTWARE.....	38
2.9.1 Métricas basadas en la función.....	39

CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 FASE INICIAL.....	41
3.1.1 Requerimientos de Usuario.....	41
3.2 FASE DE ELABORACIÓN.....	42
3.2.1 Modelo del negocio.....	42
3.2.2 Modelo de Caso de Uso.....	44
3.2.2.1 Actores.....	45
3.2.3 Diagramas de Caso de Uso.....	46
3.2.4 Diagrama de Secuencia.....	62
3.2.5 Diagrama de Estados.....	67
3.2.6 Diagrama de Clases.....	73
3.3 FASE DE CONSTRUCCIÓN.....	74
3.3.1 Diagrama de Componentes.....	74
3.3.2 Implementación de la Base de Datos.....	75
3.3.3 Interfaz de Usuario.....	76
3.4 FASE DE TRANSICIÓN.....	80

CAPÍTULO IV CALIDAD DE SOFTWARE

4.1 INTRODUCCIÓN.....	81
4.2 FUNCIONALIDAD.....	81
4.3 FACILIDAD DE MANTENIMIENTO.....	86
4.4 PORTABILIDAD.....	87
4.5 FACILIDAD DE USO.....	88

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES.....	90
5.2 RECOMENDACIONES.....	91

REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

ÍNDICE DE FIGURAS

Figura 1.1: Organigrama clínica CPS	2
Figura 1.2: Cronograma de actividades	11
Figura 2.1: Un ciclo de RUP	15
Figura 2.2: Fases del RUP	18
Figura 2.3: Caso de Uso (extends)	24
Figura 2.4: Caso de Uso (uses)	24
Figura 2.5: Diagrama de Clases	25
Figura 2.6: Diagrama de Secuencia	26
Figura 2.7: Diagrama de Colaboración	27
Figura 2.8: Diagrama de Estados	28
Figura 2.9: Diagrama de Actividades	29
Figura 2.10: Diagrama de Componentes	30
Figura 2.11: Diagrama de Despliegue	30
Figura 2.12: Modelo objeto-relacional	35
Figura 3.1: Modelado de negocio (Módulo Control de Recursos)	42
Figura 3.2: Modelado de negocio (Módulo Control de Personal)	43
Figura 3.3: Modelo de Caso de Uso (Módulo Control de Recursos)	44
Figura 3.4: Modelo de Caso de Uso (Módulo Control de Personal)	45
Figura 3.5: Caso de Uso: Control de reposición de materiales	46
Figura 3.6: Caso de Uso: Control de mantenimiento	48
Figura 3.7: Caso de Uso: Control de Transporte	49
Figura 3.8: Caso de Uso: Control de Descargo de Transporte	51
Figura 3.9: Caso de Uso: Control de Permisos	53
Figura 3.10: Caso de Uso: Control de Cambio Voluntario de Turno	54
Figura 3.11: Caso de Uso: Control de Bajas	56
Figura 3.12: Caso de Uso: Control de Comisiones	57
Figura 3.13: Caso de Uso: Control de Vacaciones	59
Figura 3.14: Caso de Uso: Control de datos de Personal	60
Figura 3.15: Diagrama de Secuencia: Control de Reposición de Materiales	62
Figura 3.16: Diagrama de Secuencia: Control de Mantenimiento	63

Figura 3.17: Diagrama de Secuencia: Control de Transporte	63
Figura 3.18: Diagrama de Secuencia: Control de Descargo de Transporte	64
Figura 3.19: Diagrama de Secuencia: Control de Permisos	64
Figura 3.20: Diagrama de Secuencia: Control de cambio Voluntario de Turno	65
Figura 3.21: Diagrama de Secuencia: Control de Bajas	65
Figura 3.22: Diagrama de Secuencia: Control de Comisiones	66
Figura 3.23: Diagrama de Secuencia: Control de Vacaciones	66
Figura 3.24: Diagrama de Secuencia: Control de datos de Personal	67
Figura 3.25: Diagrama de Estados: Control de reposición de materiales	68
Figura 3.26: Diagrama de Estados: Control de mantenimiento	68
Figura 3.27: Diagrama de Estados: Control de transporte	69
Figura 3.28: Diagrama de Estados: Control de descargo de transporte	69
Figura 3.29: Diagrama de Estados: Control de permisos	70
Figura 3.30: Diagrama de Estados: Control de cambio voluntario de turno	70
Figura 3.31: Diagrama de Estados: Control de bajas	71
Figura 3.32: Diagrama de Estados: Control de comisiones	71
Figura 3.33: Diagrama de Estados: Control de vacaciones	72
Figura 3.34: Diagrama de Estados: Control de datos de personal	72
Figura 3.35: Diagrama de Clases	73
Figura 3.36: Diagrama de Componentes	74
Figura 3.37: Tabla objeto	75
Figura 3.38: Pantalla Principal	76
Figura 3.39: Control de ingresos	76
Figura 3.40: Pantalla control de recursos	77
Figura 3.41: Ventana Introducir Reposición	77
Figura 3.42: Ventana Reportes Reposición	78
Figura 3.43: Ventana imprimir	78
Figura 3.44: Ventana exportar Excel	79
Figura 3.45: Ventana maximizar tabla	79
Figura 3.46: Ventana eliminar registro	80
Figura 3.47: Ventana control de personal	80

ÍNDICE DE TABLAS

Tabla 3.1: Caso de Uso: Control de reposición de materiales	47
Tabla 3.2: Caso de Uso: Control de mantenimiento	48
Tabla 3.3: Caso de Uso: Control de Transporte	50
Tabla 3.4: Caso de Uso: Control de Descargo de Transporte	51
Tabla 3.5: Caso de Uso: Control de Permisos	53
Tabla 3.6: Caso de Uso: Control de Cambio Voluntario de Turno	55
Tabla 3.7: Caso de Uso: Control de Bajas	56
Tabla 3.8: Caso de Uso: Control de Comisiones	58
Tabla 3.9: Caso de Uso: Control de Vacaciones	59
Tabla 3.10: Caso de Uso: Control de datos de Personal	61
Tabla 4.1: Parámetro de medición	82
Tabla 4.2: Cálculo de Conteo Total	83
Tabla 4.3: Ajuste de complejidad del punto de función	84
Tabla 4.4: Escala de ajuste	85
Tabla 4.5: Relación de cambios y modificaciones	87
Tabla 4.6: Evaluación de facilidad de uso	88
Tabla 4.7: Escala de evaluación	88

MARCO CONTEXTUAL

1.1 INTRODUCCIÓN

Hoy en día la información desempeña un papel esencial en el funcionamiento de toda institución y es por este motivo que se hace necesario contar con sistemas de información que brinden confiabilidad, seguridad y disponibilidad de toda la información generada dentro del ente en cuestión.

Por su importancia, el control administrativo de toda institución debe ser optimizado al máximo; con este objetivo se utilizan sistemas automatizados para el manejo de información generalmente aplicados en la clasificación de datos, búsqueda de información y generación de reportes.

La Clínica de la Caja Petrolera de Salud (CPS), si bien cumple un objetivo específico dentro de la sociedad boliviana, no escapa a la condición de empresa, ni a la necesidad de contar con sistemas automatizados de 5ª generación acordes con la implementación de tecnología de punta, para mejorar la calidad de servicio que brinda a sus afiliados.

El uso de herramientas case para el diseño y posterior implementación de productos software se constituye en un aporte invaluable para los analistas y programadores simplificando gran parte de sus labores, es en ese sentido que el presente proyecto utiliza la Herramienta Case Enterprise Architect con el objetivo de demostrar su gran funcionalidad en la aplicación de ingeniería inversa.

1.2 ANTECEDENTES

1.2.1 Antecedentes de la Institución

La Caja Petrolera de Salud (CPS), fué creada mediante D.S. No. 5083 de 10 de noviembre de 1958, como institución pública descentralizada, con autonomía de

gestión y patrimonio independiente, para la administración de los regímenes de corto y largo plazo, establecidos en el código de seguridad social, cumpliendo en esta gestión 50 años al servicio de la comunidad.

Posteriormente, en virtud de la Ley 924 de 1987, se separa la administración de los seguros a largo plazo; invalidez, vejez, muerte y riesgos profesionales a largo plazo y la administración de los seguros de corto plazo: enfermedad, maternidad y riesgos profesionales a corto plazo.

La CPS cuenta con varios centros de atención uno de los más importantes es la Clínica de la Caja Petrolera de Salud, que ofrece servicios de emergencias y de hospitalización, atendiendo medicina externa, interna y especialidades clínicas.

La estructura que tiene la Clínica de la Caja Petrolera de Salud está dividida en diferentes departamentos cada uno con responsabilidades bien definidas (Figura 1.1).



Figura 1.1 Organigrama Clínica CPS
Fuente: Departamento Administrativo CCPS

Las unidades con las que cuenta la Clínica son las siguientes:

- Emergencias
- Nutrición
- Epidemiología CCIN

- Patología
- Unidad de Terapia Intensiva
- Laboratorio Clínico
- Rayos X
- Maternidad
- Pediatría
- Cirugía
- Proctología
- Hemodiálisis
- Ecografía
- Endocrinología
- Oncología
- Nefrología
- Quirófano
- Gastroenterología
- Ginecología
- Endoscopia
- Traumatología

1.2.1.1 Misión

Contribuir a mejorar la calidad de vida de su población protegida, mejorando los niveles de salud, mediante acciones integrales de promoción, prevención, reparación y rehabilitación, otorgadas en forma oportuna con calidad y calidez.

1.2.1.2 Visión

Ser una institución ampliamente solidaria, que con sentido de equidad social, incorpore al campo de aplicación de los seguros a corto plazo que administra, no solo a trabajadores con relación de dependencia, sino a otros sectores desprotegidos, adoptando políticas de captación de aportes.

1.2.1.3 Ubicación

La Clínica de la Caja Petrolera de Salud se encuentra ubicada en la Av. Arce Plaza Isabel la Católica N° 2525 de la ciudad de La Paz – Bolivia.

1.2.2 Antecedentes del Proyecto

Existe gran cantidad de proyectos sobre administración de centros de salud, control de recursos humanos, control de inventarios; a continuación mencionaremos algunos trabajos que son de interés para la realización del presente proyecto.

- *“Sistema de Seguimiento Clínico Hospital del Niño Dr. Ovidio Aliaga Uría”* Luis Rodrigo Gutierrez Miranda (2005) Sistema que realiza el control adecuado eficiente y correcto de toda la información referente a los pacientes, bajo medidas de seguridad en la administración de los procedimientos de información e indicadores estadísticos.
- *“Sistema de Información Integrado de Control y Seguimiento Médico Caso: Policonsultorio 20 de Octubre y Laboratorio de la Clínica de la Caja Petrolera de Salud Regional de La Paz”* Orlando Franz Lima Machicado (2004) Sistema que realiza el control de historias y exámenes clínicos.
- *“Sistema de control y administración SARIRI Bolivian Bus”* Paredes Mamani Juan (2007). Sistema que realiza el control y administración de buses y de recursos humanos en la empresa.
- *“Sistema de Administración y control de clubes Caso: Iglesia Adventista del 7º día”* Torrez Itusaca Arturo (2008). Sistema que realiza el control adecuado de inventarios en los clubes.

El proyecto propuesto se enfoca en el control de recursos materiales (reposición de materiales, mantenimiento de equipos) y en el control de personal (vacaciones, permisos, bajas, comisiones, cambio voluntario de turno) que son prioridad para la dirección administrativa de la institución.

1.3 PROBLEMÁTICA

El registro y control de la información en el departamento administrativo es inestable debido al manejo manual de esta, causando varios problemas tanto en el control de recursos materiales, como en el control de personal de la Clínica.

Algunos de los problemas actuales que atraviesa el departamento administrativo con relación al manejo de la información, se detallan a continuación:

1. Emisión de reportes incompletos en lapsos de tiempo muy largos.
2. Pérdida de información.
3. Inexistencia de información actualizada con respecto a recursos materiales y control de personal.
4. Atención deficiente a las distintas unidades de la institución debido al manejo manual de formularios.

Las causas identificadas para la problemática expuesta son:

- **Gran volumen de información no sistematizada:** Los distintos requerimientos de las unidades llegan al departamento administrativo en formularios físicos, generando el almacenamiento masivo de información física.
- **No existen copias necesarias de formularios:** Al no estar establecidas políticas claras con respecto al manejo de la información dentro la institución, la pérdida de un formulario ocasiona que el control administrativo no sea óptimo.

- **Registro incompleto de formulario para la reposición de materiales y pedido de mantenimiento de equipos:** Ocasionado por el llenado y control manual de los mismos.

- **Existe pocos documentos de insumos en transporte:** No se tiene certeza de los insumos requeridos para esta actividad porque generalmente no se registra gastos de transporte.

- **Manejo manual de la información en el área de personal:** Registros almacenados incorrectamente de las solicitudes de permisos y vacaciones; solicitud de cambio voluntario de turno y otros.

Se ha identificado como problema principal de este proyecto el siguiente:

No existe un manejo sistematizado de la información administrativa referida al control de recursos materiales y personal dentro la clínica de la CPS, que permita el seguimiento oportuno de cada una de las actividades que desarrolla la institución, lo que ocasiona pérdidas económicas y disconformidad en las distintas unidades dentro la clínica.

1.4 OBJETO DE ESTUDIO

En la actualidad la información generada en la mayoría de las instituciones de mediana o gran envergadura llega a tener dimensiones que son difícilmente manejables por medios manuales; es necesario utilizar herramientas que permitan manejar la información de forma eficiente, para brindar mejores servicios a la sociedad o convertirse en entes competitivos en el medio.

Por cuanto el estudio del “Sistema de Control Administrativo Caso: Clínica Caja Petrolera de Salud”, contribuirá a la labor de administración de la información que se maneja en la institución y a las tareas del departamento administrativo, reduciendo

de esta manera, el tiempo que se empleaba en clasificar la información, realizar búsquedas y generar reportes entre otros.

1.5 OBJETIVO GENERAL

Implementar un sistema de información automatizado que permita un manejo adecuado y eficiente de la información de la institución con respecto a la parte administrativa, correspondiente al control de recursos materiales (reposición de materiales, mantenimiento de equipos) y el control de personal (vacaciones, permisos, bajas, comisiones, cambio voluntario de turno), bajo medidas de seguridad.

1.6 OBJETIVOS ESPECÍFICOS

Para la concretización del objetivo principal se establecieron los siguientes objetivos secundarios.

- Usar Enterprise Architect (Herramienta case) para el desarrollo del software.
- Emplear la metodología RUP para el análisis y diseño del sistema.
- Desarrollar el producto software resultado del estudio propuesto para ser aplicado en un entorno LAN.
- Desarrollar el subsistema de registro y control de recursos materiales tomando en cuenta los siguientes tópicos
 - Reposición de materiales.
 - Permisos para mantenimiento de equipos.
 - Control de insumos en transporte.
- Desarrollar el subsistema de registro y control de personal considerando vacaciones, permisos, bajas, cambio voluntario de turno.
- Emitir reportes según las especificaciones de la institución en tiempo real.
- Establecer mecanismos automatizados de copias de seguridad.

- Usar normas de seguridad y calidad para el diseño del sistema de información.

1.7 JUSTIFICACIÓN

1.7.1 Justificación Técnica

La Clínica de la Caja Petrolera de Salud cuenta con tres equipos informáticos en el departamento administrativo cuyas características son:

2(PC´s)

- Microprocesadores PENTIUM IV.
- Tarjeta de Video Con resolución de 800 x 600 pixeles.
- 1 GB de memoria RAM
- Sistema Operativo Windows XP

1(PC)

- Microprocesador Core 2 DUO
- Tarjeta de Video Con resolución de 1280 x 768 pixeles.
- 16 Gb de memoria RAM
- Sistema Operativo Windows Vista

Con estos equipos informáticos es posible implementar el sistema propuesto, que permitirá aprovechar la tecnología de punta para un mejor manejo de la información de esta institución. Complementado en su estudio métodos apropiados de ingeniería de software, para dar solución a los problemas, contando con herramientas para la recolección de datos, análisis, diseño, calidad, seguridad, codificación documentación y pruebas.

1.7.2 Justificación Económica

La disposición de información oportuna en el departamento administrativo genera ahorro de tiempo por parte del personal, además de reducir de manera significativa pérdidas económicas a causa del uso innecesario e incorrecto del material de

escritorio, deficiente control de recursos materiales, equipos dañados, transporte y control de personal.

1.7.3 Justificación Social

Como el departamento administrativo de la Clínica CPS tiene un flujo constante de información; un mejor seguimiento de la información en las distintas unidades de la Clínica, ocasionará un impacto social beneficioso para la población protegida de la Clínica y para sus funcionarios, debido a que permitirá realizar un adecuado análisis, planificación y toma de decisiones dentro de la institución.

1.8 APOORTE TEÓRICO

Enterprise Architect es una herramienta basada en Windows, diseñada para ayudar a construir software robusto y fácil de manejar.

Además de modelar, permite realizar ingeniería inversa de código en muchos lenguajes y herramientas de programación como ser: C++, Java, Delphi, VB.Net, Visual Basic y PHP. Las plantillas de generación de código ayudan a personalizar el código fuente generado según las necesidades del usuario.

1.9 APOORTE PRÁCTICO

El principal aporte del proyecto es el desarrollo de un software que contribuirá con la labor de administración de la información que se maneja en la institución.

- Con este proyecto se evita la duplicación o pérdida de la información
- Generación de reportes en una interfaz amigable para el usuario según sus necesidades.
- Se obtiene mayor control en las áreas de recursos de materiales y de personal.

- Aplicar métodos de seguridad para hardware y software al sistema.
- Elaboración de manuales de usuario del sistema.

1.10 LÍMITES Y ALCANCES

El presente estudio se limita al análisis, diseño, implementación y mantenimiento de un Sistema de Información enfocado a coadyuvar con el desarrollo de las actividades administrativas de la institución.

El sistema proporciona a los usuarios información confiable, en tiempo real, de todas las actividades desarrolladas en la gestión.

En el módulo control de personal se tomará en cuenta los permisos, bajas, cambio voluntario de turno, comisiones y vacaciones.

En el módulo control de recursos se tomará en cuenta reposición de materiales, transporte y control de mantenimiento de equipos.

1.11 PLANIFICACIÓN DEL PROYECTO

Para poder establecer y controlar de forma clara el desarrollo de las actividades involucradas con el estudio, análisis e implementación del sistema propuesto se utiliza el ciclo de vida de desarrollo de software [Kendall, Kendall] expresado en un diagrama GANT (Figura 1.2).

CRONOGRAMA DE ACTIVIDADES

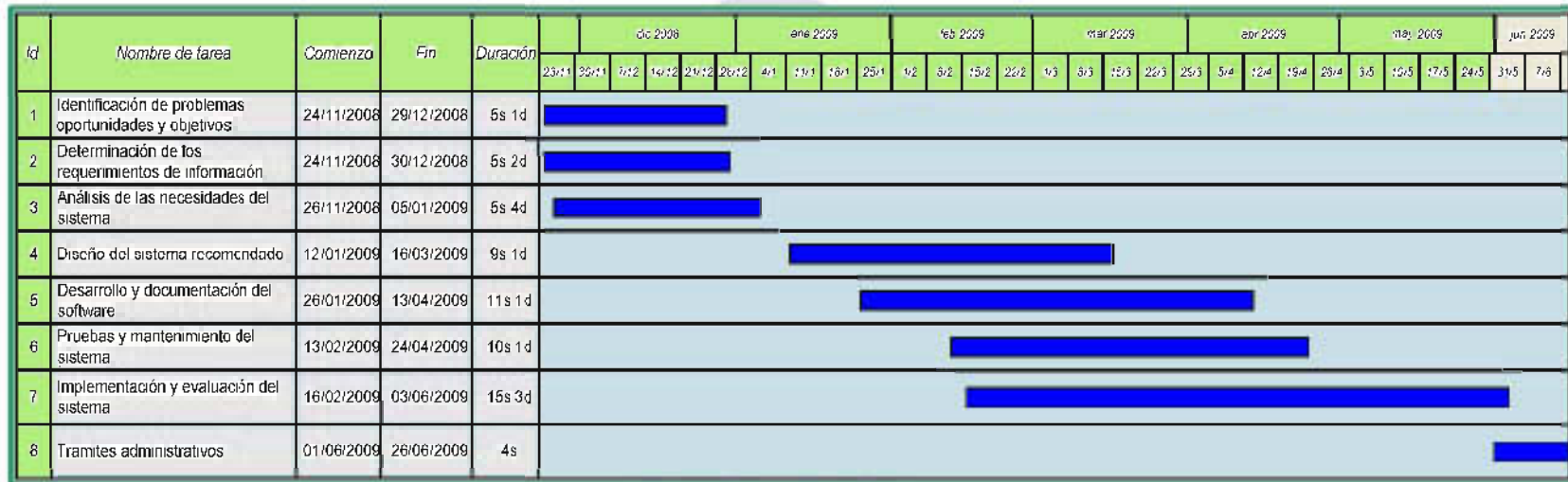


Figura 1.2: Cronograma de Actividades
Fuente: Elaboración Propia

MARCO TEÓRICO

2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Para el desarrollo del presente proyecto se utiliza la metodología de desarrollo de software RUP (Proceso Unificado Racional) que tiene como pilar fundamental el uso de los diagramas que define la herramienta UML (Lenguaje de Modelado Unificado).

El uso de la herramienta case Enterprise Architect para el diseño y posterior implementación del producto software.

A continuación se detallan los fundamentos teóricos de las metodologías y herramientas anteriormente mencionadas.

2.2 MODELO [JAC, 2000]

Si realmente se quiere construir software, es equivalente a construir una casa o un rascacielos, el problema es algo más que una cuestión de escribir grandes cantidades de código (software).

De hecho, el truco está en crear el software apropiado y en imaginar cómo escribir menos código. Esto convierte al desarrollo de software de calidad una cuestión de arquitectura, procesos y herramientas. Asimismo existen muchas empresas de desarrollo de software que construyen software (sistemas) enfocando el problema como si estuvieran enfrentándose a la construcción de una casita de perro es decir, un martillo, clavos, un serrucho y maderas, algo así de sencillo, sin planificar y sin planos. Pero luego este sistema crece hasta el tamaño de un rascacielos y simplemente porque son víctimas de su propio éxito llegan a un momento en el que, si no se ha tenido en cuenta la arquitectura, el proceso o las herramientas, la casita

de perro ahora convertida en un rascacielos, se colapsa bajo su propio peso. El derrumbamiento puede molestar al perrito, el fallo de un gran edificio afectará materialmente a sus inquilinos.

Los proyectos que fracasan lo hacen por circunstancias propias, pero todos los proyectos con éxito se parecen en muchos aspectos. Hay muchos elementos que contribuyen a una empresa de software de éxito; uno en común es el uso del MODELADO.

El modelado es una técnica de ingeniería probada y bien aceptada. Se construye modelos arquitectónicos de casas y rascacielos para ayudar a sus usuarios a visualizar el producto final. Incluso podemos construir modelos matemáticos para analizar los efectos de vientos y terremotos sobre nuestros edificios (sistema).

El modelo no es sólo parte de la industria de la construcción. Sería inconcebible crear una nueva aeronave o automóvil (hasta sistemas de software) sin construir previamente modelos, desde modelos informáticos a modelos físicos, los nuevos dispositivos eléctricos, desde los microprocesadores a las centrales telefónicas requieren algún grado de modelado para comprender mejor el sistema y comunicar la idea a otros.

¿Entonces qué es un **modelo**? Para responder de forma sencilla: un modelo es una simplificación de la realidad.

Un modelo proporciona los planos detallados y generales que ofrecen una visión global del sistema en consideración. Todo sistema puede ser descrito desde diferentes perspectivas y puntos de vista utilizando diferentes modelos.

La visualización, especificación, construcción y documentación de un sistema requiere ser vista desde varias perspectivas, diferentes usuarios (usuarios finales, analistas, desarrolladores, integradores de sistemas, encargados de test, encargados

de la documentación técnica y jefes de proyectos) que siguen diferentes agendas en relación al proyecto, y cada uno mira ese sistema de formas diferentes en diversos momentos a lo largo de la vida del proyecto.

La arquitectura de un sistema es quizás el artefacto más importante que puede emplearse para manejar estos diferentes puntos de vista y controlar el desarrollo iterativo e incremental de un sistema a lo largo de su ciclo de vida.

La **arquitectura** de un sistema (los planos de un sistema) puede describirse mejor a través de **cinco vistas** interrelacionadas, cada vista es una proyección de la organización y la estructura del sistema, centrada a un aspecto particular de ese sistema.

Estas cinco vistas son:

- La vista **de casos de uso**, que muestra los requisitos del sistema y describe el comportamiento del mismo.
- La vista **de diseño**, que captura el vocabulario del espacio del problema y del espacio de la solución, modelando los requisitos funcionales del sistema.
- La vista **de proceso**, que modela la distribución de los procesos del sistema.
- La vista **de implementación**, que se ocupa de la realización física del sistema.
- La vista **de despliegue**, que se centra en cuestiones de ingeniería, contenido los nodos que forman la topología hardware donde se ejecuta el sistema.

Estas cinco vistas involucran visualizar, modelar aspectos estáticos y dinámicos de los sistemas.

2.3 RUP (PROCESO UNIFICADO RACIONAL)

El Proceso Unificado Racional (RUP) es un proceso iterativo de desarrollo de software creado por *Rational Software Corporation*, ahora división de la IBM. El RUP no es solo un proceso preceptivo¹ concreto, sino un marco de proceso adaptable. Como tal, RUP describe cómo desarrollar el software utilizando con eficacia técnicas probadas, en el sentido de seleccionar los procesos de desarrollo apropiados a una organización o institución en particular, del proyecto o del desarrollo del software.

RUP es un **marco genérico** que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

2.3.1 Ciclo de vida de RUP

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes (Figura 2.1).

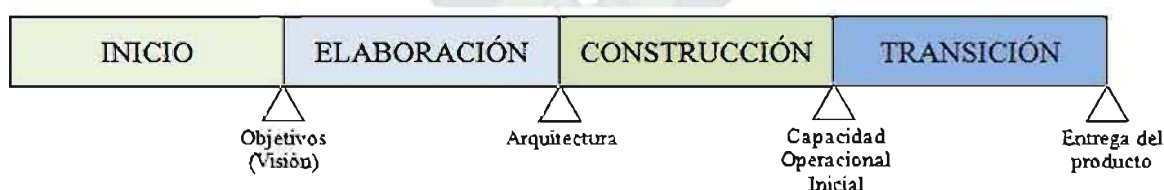


Figura 2.1: Un Ciclo de RUP
Fuente: [JGB, 2000]

¹Proceso preceptivo: Proceso que debe cumplirse obligatoriamente

Cada ciclo consta de cuatro fases: inicio (inicipión), elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones.

2.3.1.1 Etapas de RUP

Estructuralmente RUP está dividido en dos etapas: de ingeniería y de producción. Donde la etapa de ingeniería cubre las fases de inicio y elaboración y la etapa de producción cubre las etapas de construcción y transición. [WEB, 03].

2.3.1.2 Etapa de ingeniería

Esta etapa agrupa las fases de inicio y de elaboración, lo que básicamente da por objetivo la conceptualización del sistema y el diseño inicial de la solución del problema.

Se identifican los riesgos y se establece su plan de manejo, se ajusta ese plan según la priorización de riesgo y la de casos de usos vs. riesgos, para determinar en qué orden y en que iteraciones se desarrollarán los artefactos de software que son la solución a los casos de uso.

Se identifican los recursos necesarios, tanto económicos como humanos, acordes con las necesidades del proyecto y se da comienzo al proceso de estimación y planificación inicial a un nivel macro para todo el proyecto, posteriormente se realiza una estimación detallada de tiempo y recursos de las fases de concepción y elaboración. [WEB, 01].

Fase 1: Preparación inicial (Inicipión)

Su función principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber que se cubrirá y delimitar el alcance del proyecto. [WEB, 01]

Fase 2: Preparación detallada (Elaboración)

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio. [WEB, 01].

2.3.1.3 Etapa de producción

En esta etapa se realiza un proceso de refinamiento de las estimaciones de tiempos y recursos para las fases de construcción y transición, se define un plan de mantenimiento para los productos entregados en la etapa de ingeniería, se implementa los casos de uso pendientes y se entrega el producto al cliente, garantizando la capacitación y soporte adecuados.

Fase 3: Construcción

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado operación beta. [WEB, 01].

Fase 4: Transición

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos. [WEB, 01].

2.3.2 Fases del RUP

Cualquier proceso de desarrollo de software debe pasar por cuatro fases, cada fase es construida con hitos (un punto en el tiempo, en el cual ciertas decisiones críticas

deben ser tomadas) bien definidos. Cada fase tiene un propósito específico (Figura 2.2).

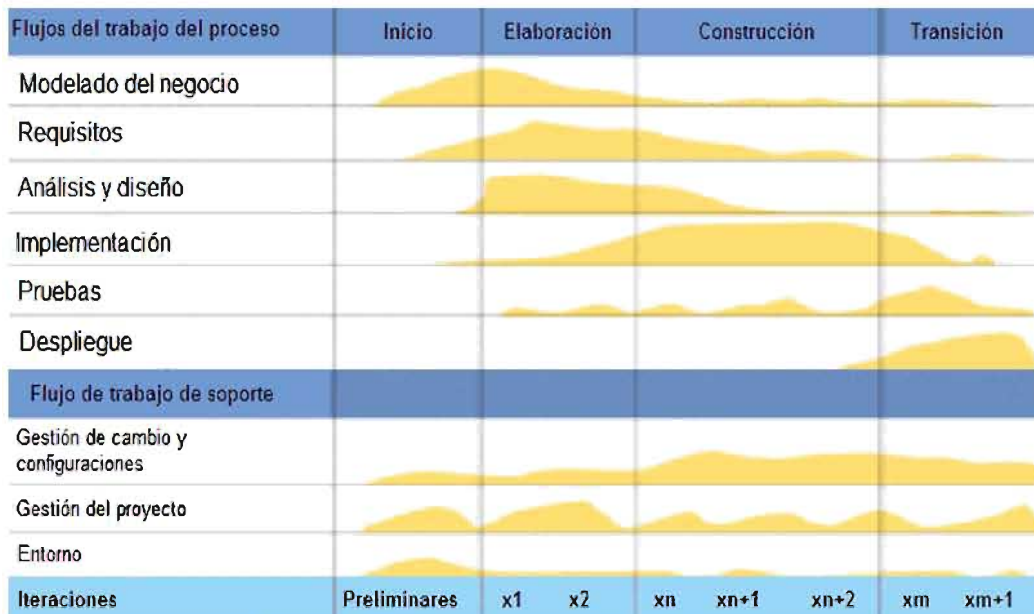


Figura 2.2: Fases del RUP

Fuente: [WEB, 04]

2.3.2.1 Primera fase de RUP: Inicio

La fase de inicio puede tomar varias formas. Para algunos proyectos, es una plática en la máquina de café: “Por favor pongan los catálogos de servicio en la Web”. Para proyectos mayores, puede ser un estudio de factibilidad completo que tome meses. [RAN, 2000]

Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio. Esta fase responde las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema para los usuarios más importantes?
- ¿Cómo podría ser la mejor arquitectura del sistema?
- ¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?

El objetivo de esta fase es ayudar a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles.

Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento del equipo.

Los artefactos que típicamente sobreviven a esta fase son:

- Un modelo del negocio.
- Un enunciado de los mayores requerimientos, generalmente como casos de uso.
- Un boceto inicial de la arquitectura.
- Una descripción de los objetivos del proyecto.
- Una versión muy preliminar del plan del proyecto.

Debe poder responder las siguientes cuestiones:

- ¿Se ha llegado a un acuerdo con todas las personas involucradas sobre los requisitos funcionales y no funcionales del sistema?
- ¿Se ha determinado con claridad el ámbito del sistema?
- ¿Se ha determinado lo que va a estar dentro del sistema y fuera del sistema?
- ¿Se vislumbra una arquitectura que pueda soportar estas características?
- ¿Se identifican los riesgos críticos?
- ¿Es factible para su organización llevar adelante el proyecto? [WEB, 02].

Al final de la fase de inicio, se examinan los objetivos del ciclo de vida del proyecto y se decide si se procede o no con el desarrollo.

2.3.2.2 Segunda fase de RUP: Elaboración [JGB, 2000]

Durante esta fase se especifican en detalle la mayoría de los casos de uso del producto (resultado del análisis a los requisitos) y se diseña la arquitectura del sistema.

La arquitectura se expresa en forma de planos (modelados) trazados desde diferentes vistas, los cuales juntos representan el sistema entero. Esto implica que hay vistas arquitectónicas expresadas mediante los diagramas, del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y modelo de despliegue.

Las iteraciones en la fase de elaboración:

- Establecen una firme comprensión del problema a solucionar.
- Eliminan los mayores riesgos.
- Establecen la fundación arquitectural para el software.
- Forman un plan detallado para las siguientes iteraciones.

2.3.2.3 Tercera fase de RUP: Construcción

Durante la fase de construcción se crea el producto, se añaden los músculos (software terminado) al esqueleto (la arquitectura). En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. [JGB, 2000]

Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo puede que no esté libre de defectos. Los artefactos producidos durante esta fase son:

- El sistema software.
- Los casos de prueba.
- Los manuales de usuario.

Hay proyectos en los cuales las pruebas y la integración son dejadas al final. Las pruebas y la integración son tareas grandes, y siempre toman más tiempo de lo que la gente piensa. Atrás en el tiempo, en los días de OS/360, se estimaba que la mitad de un proyecto era pruebas y corrección de errores. Las pruebas y la integración son más difíciles cuando se dejan al final, y más desmoralizadoras. Todo este esfuerzo lleva a un gran riesgo. Con el desarrollo iterativo se realiza el proyecto entero en cada iteración, lo que conduce al hábito de lidiar con todo los problemas cada vez. [RAN, 2000]

2.3.2.4 Cuarta fase de RUP: Transición

La fase de transición cubre el periodo durante el cual el producto se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el productor e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios. La fase de transición conlleva actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos que se encuentran tras la entrega. El equipo de mantenimiento suele dividir esos defectos en dos categorías: los que tienen suficiente impacto en la operación para justificar una versión incrementada y los que pueden corregirse en la siguiente versión normal. La fase de transición finaliza con el hito de lanzamiento del producto [JGB, 2000].

También al final de la fase de transición se decide si los objetivos del ciclo de vida han sido cumplidos, y posiblemente si se debe iniciar otro ciclo de desarrollo. Éste es también un punto donde se empacan algunas de las lecciones aprendidas en este proyecto para mejorar el proceso.

2.4 LENGUAJE DE MODELADO UNIFICADO UML

El lenguaje unificado de modelado UML es una notación que es el resultado de la evolución de las notaciones previas en ingeniería de software, toma los aspectos fuertes de tres metodologías anteriores: OMT, Booch y OOSE. El Lenguaje de Modelado Unificado UML es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

UML es apropiado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la WEB, e incluso para sistemas empotrados en tiempo real muy exigentes. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas.

UML puede ser utilizado por cualquier metodología de análisis y diseño orientado a objetos para expresar los diseños.

2.4.1 Características de UML [JAC, 2000]

UML se ha convertido en un estándar de facto que tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas objetos (O.O).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubren las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.

- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

2.4.2 Diagramas de UML

UML presenta varios diagramas donde cada uno representa un aspecto del sistema. Los diagramas a utilizar en las diferentes etapas del desarrollo de los sistemas de información pueden variar dependiendo del tamaño y tipo de sistema, por lo que es necesario organizarlos según las fases del proceso unificado [JGB, 2000].

2.4.2.1 Diagrama de casos de uso [JAC, 2000]

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un acto sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la relación y la generalización son relaciones.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al demostrar cómo reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagrama intervienen algunos conceptos nuevos: un actor es una entidad externa al sistema que se modela y que puede interactuar con el sistema, un ejemplo de actor podría ser un usuario o cualquier otro sistema. Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.

Se usa “*extends*” cuando se tiene un caso de uso similar pero hace algo más. Generalmente para situaciones imprevistas de acuerdo a las políticas del negocio. Como muestra la siguiente figura 2.3.

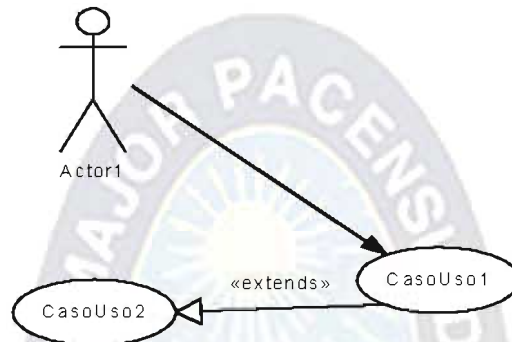


Figura 2.3: Caso de uso (extends)
Fuente: [LARMAN, 1999]

- Un caso de uso usa otro caso de uso.
- La relación “*uses*” ocurre cuando se tiene una porción de comportamiento que es similar en más de un caso de uso y no quiere copiar la descripción de tal conducta. Como muestra la siguiente figura 2.4.

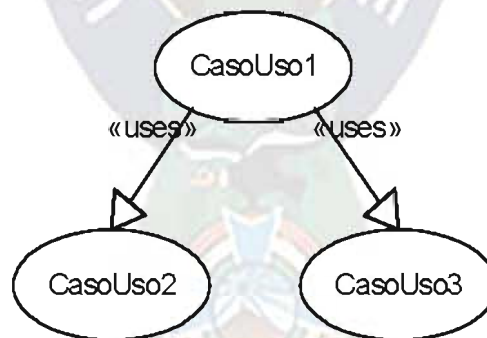


Figura 2.4: Caso de uso (Uses)
Fuente: [LARMAN, 1999]

2.4.2.2 Diagrama de clases

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones de (incluyendo herencia, agregación, asociación, etc.). Los diagramas de clases son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). [JAC, 2000] (Figura 2.5)

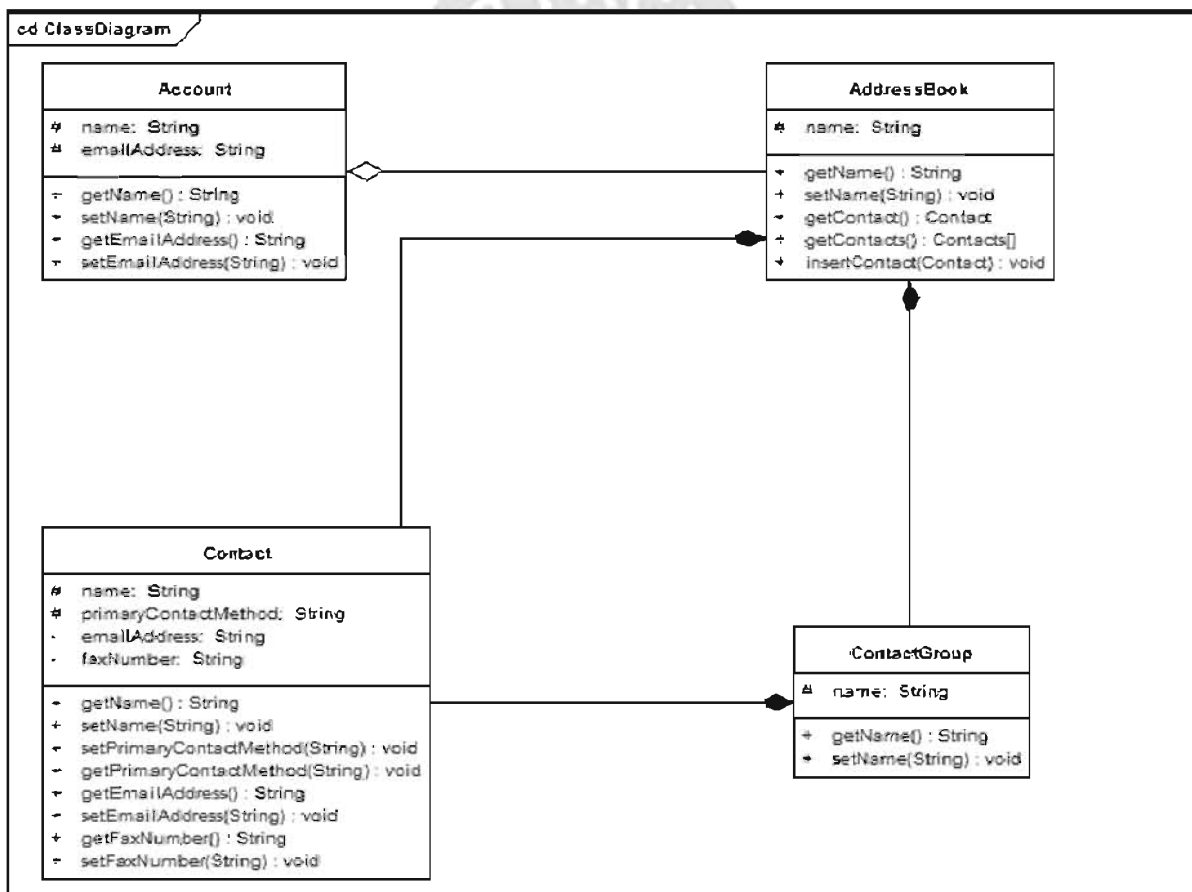


Figura 2.5: Diagrama de Clases

Fuente: [WEB, 05]

2.4.2.3 Diagrama de interacción

Son modelos que describen la manera en que colaboran grupos de objetos para ciertos comportamientos. Existen dos tipos:

2.4.2.3.2 Diagrama de colaboración

Diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes. [JAC, 2000]

Un diagrama de colaboración es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de sólo clasificadores y asociaciones. Los roles de clasificador y los de asociación describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la colaboración. Cuando se instancia una colaboración, los objetos están ligados a los roles de clasificador y los enlaces a los roles de asociación.

El rol de asociaciones puede ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimiento o variables locales del procedimiento. Los símbolos de enlace pueden llevar estereotipos para indicar enlaces temporales (Figura 2.7).



Figura 2.7: Diagrama de Colaboración

Fuente: [LAR. 1999]

2.4.2.4 Diagramas de comportamiento

Dentro de estos diagramas se encuentran:

2.4.2.4.1 Diagrama de estados

Representan la secuencia de estados posibles por los que un objeto pasa durante su tiempo de vida en respuesta a estímulos (eventos) recibidos. Representa lo que podemos denominar en conjunto una máquina de estados. Un estado en UML es cuando un objeto o una interacción satisfacen una condición, desarrollo a alguna acción o se encuentra esperando un evento. [JAC, 2000]

Cuando un objeto o una interacción pasar de un estado a otro por la concurrencia de un evento se dice que ha sufrido una transición, existen varios tipos de transiciones entre objetos: simples (normales y reflexivas) y complejas. Además una transición puede ser interna si el estado que parte el objeto o interacción es el mismo que al que llega, no se provoca un cambio de estado y se representan dentro de estado, no de la transición. Como en todas las metodologías OO se envían mensajes, en este caso es la acción de la que puede enviar mensajes a uno o varios objetos destino (Figura 2.8).

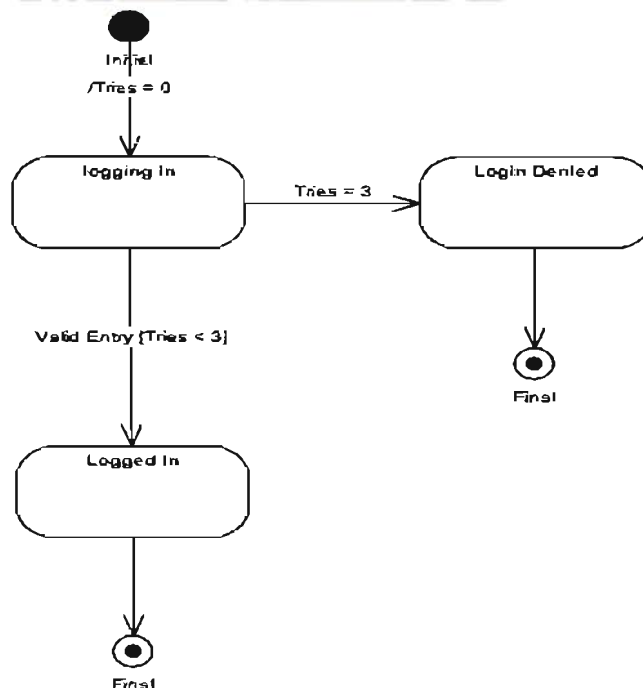


Figura 2.8: Diagrama de Estados

Fuente: [WEB, 05]

2.4.2.4.2 Diagrama de actividades

Un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades (Figura 2.9).

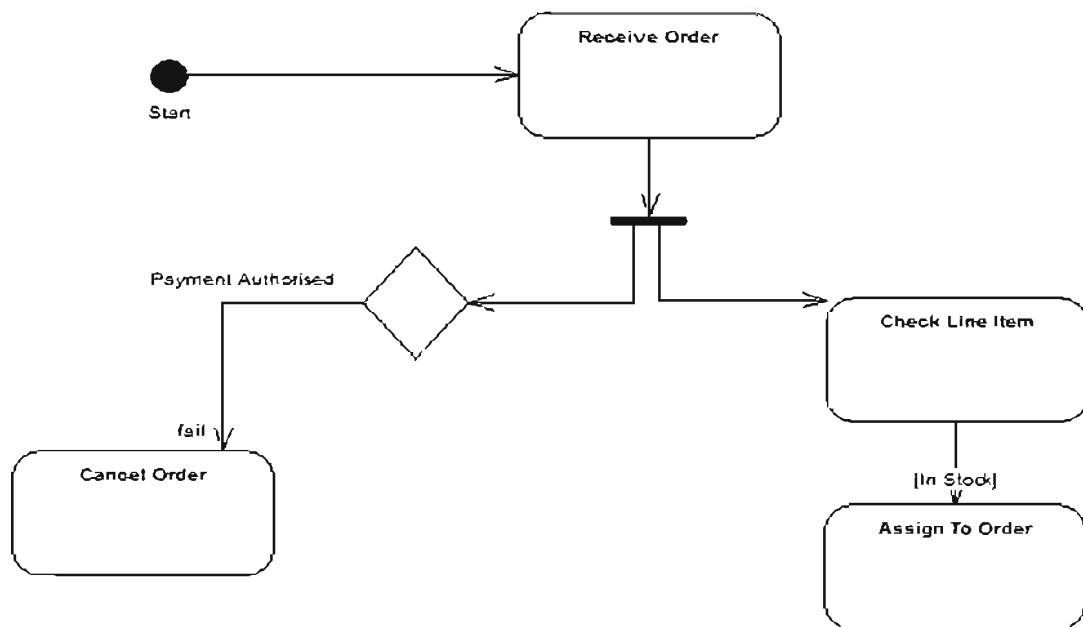


Figura 2.9: Diagrama de Actividades

Fuente: [WEB, 05]

2.4.2.5 Diagramas de implementación

2.4.2.5.1 Diagrama de componentes

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. [JAC, 2000].

Permite representar la estructura de ciertos aspectos físicos de los sistemas orientados a objetos, concretamente la organización y dependencias entre los elementos físicos que residen en un nodo, es decir ficheros que representan el empaquetamiento físico de clases, interfaces y colaboraciones (Figura 2.10).

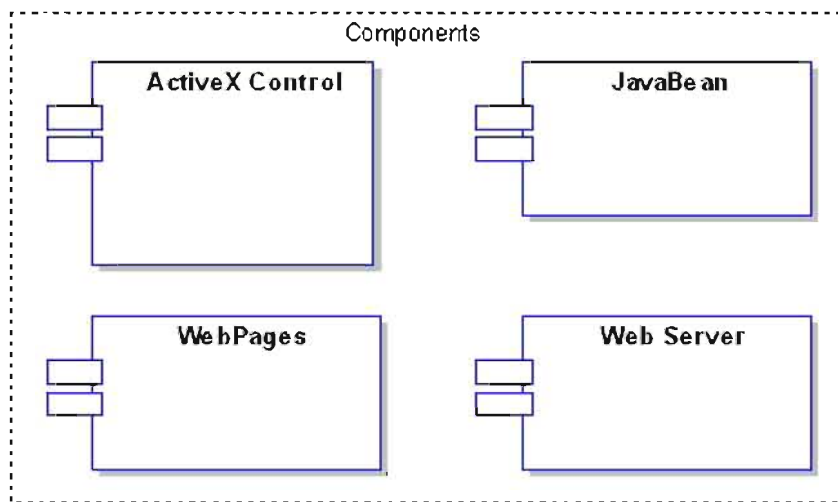


Figura 2.10: Diagrama de componentes

Fuente: [WEB, 05]

2.4.5.2 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Se relacionan con los diagramas de componentes en que un nodo incluye, por lo común uno o más componentes. Los diagramas de despliegue se utilizan para modelar la vista de despliegue estática de un sistema, que implica modelar la topología de hardware sobre el que se ejecuta el sistema (2.11).

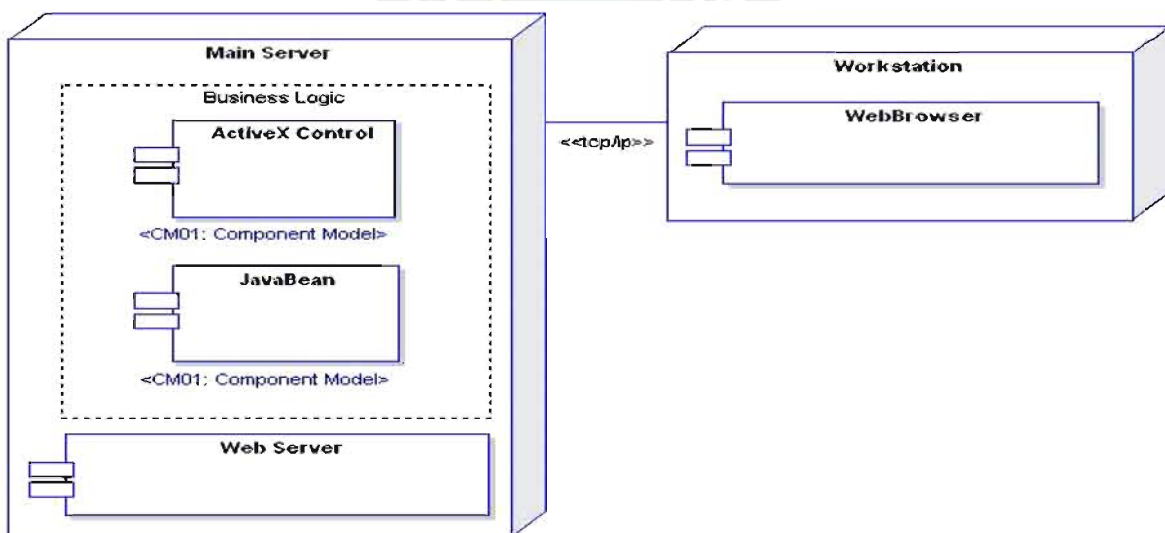


Figura 2.11: Diagrama de despliegue

Fuente: [WEB, 05]

2.5 HERRAMIENTA CASE

2.5.1 Definición

Se puede definir a las herramientas case como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos de ciclo de vida de desarrollo de un software (Investigación preliminar, análisis, diseño, implementación e instalación).

Se puede ver a la herramienta case como la unión de herramientas automáticas de software y las metodologías de desarrollo de software formales.

2.5.2 Enterprise Architect (EA) [WEB, 05]

Enterprise Architect es una herramienta case para el diseño y construcción de sistemas de software para el modelado de procesos de negocios y para objetivos de modelado más generalizados.

EA es una herramienta de análisis y diseño UML comprensivo, cubriendo el desarrollo de software desde la obtención de los requisitos, diseño del modelo, pruebas, cambio de control y mantenimiento para la implementación, con completa trazabilidad. EA combina el poder de la última especificación UML 2.1 con alta performance, una interfaz intuitiva, para brindar un modelado avanzado a todo el equipo de desarrollo.

Además de modelar, permite realizar ingeniería inversa de código en muchos lenguajes y herramientas de programación como ser: C++, Java, Delphi, VB.Net, Visual Basic y PHP.

2.5.2.1 Características de EA

Enterprise Architect es renombrado por su enriquecida lista de características.

- Comprensivo soporte UML 2.1 para los 13 diagramas UML.
- Modelado Extendido para requisitos, diseño de la interfaz de usuario, mapeo de ideas, modelado de datos y más.
- **Velocidad:** Enterprise Architect es rápido para cargar y un actor espectacularmente rápido, inclusive con grandes modelos.
- **Escalabilidad:** Enterprise Architect soporta usuarios simples y el desarrollo de pequeños modelos, o muchos usuarios concurrentes desarrollando modelos extremadamente grandes, con igual facilidad.
- Soporte para ingeniería directa e inversa para muchos lenguajes: ActionScript 3.0, Java, C#, C++, VB.Net, Delphi, Visual Basic, Python y PHP.
- Amplio rango de barras de herramientas, ventanas acoplables, y estilos visuales.
- Modifica y personaliza las barras de herramientas y menús.
- "Desplaza" las ventanas acopladas para maximizar el espacio de las ventanas y mejorar la eficiencia de su trabajo.
- Los accesos rápidos permiten la creación de elementos de diagramas y conexiones sensibles al contexto.
- Documentación compatible de Word para editar posteriormente y vincular a los documentos maestros de Word.
- Los documentos en texto enriquecido se pueden vincular a los elementos del modelo y editados directamente usando el editor de texto enriquecido incorporado.
- Modela tablas de base de datos, columnas, claves, claves foráneas, y relaciones complejas usando UML y perfiles de modelado de datos incorporados.
- Opciones flexibles de apariencia, color y disposición.
- Corrector ortográfico.
- Diversos formatos de imágenes para guardar diagramas.

2.5.2.2 Ingeniería inversa

La ingeniería inversa se ha definido como el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema software o cualquier otro producto (se puede utilizar como punto de partida cualquier otro elemento de diseño, etc.). Estas especificaciones pueden volver a ser utilizadas para construir una nueva implementación del sistema utilizando, por ejemplo, técnicas de ingeniería directa.

La ingeniería inversa permite obtener los siguientes beneficios:

- Reducir la complejidad del sistema: al intentar comprender el software se facilita su mantenimiento y la complejidad existente disminuye.
- Generar diferentes alternativas: del punto de partida del proceso, principalmente código fuente, se generan representaciones gráficas lo que facilita su comprensión.
- Recuperar y/o actualizar la información perdida (cambios que no se documentaron en su momento): en la evolución del sistema se realizan cambios que no se suele actualizar en las representaciones de nivel de abstracción más alto, para lo cual se utiliza la recuperación de diseño.
- Detectar efectos laterales: los cambios que se puedan realizar en un sistema puede conducirnos a que surjan efectos no deseados, esta serie de anomalías puede ser detectados por la ingeniería inversa.
- Facilitar la reutilización: por medio de la ingeniería inversa se pueden detectar componentes de posible reutilización de sistemas existentes, pudiendo aumentar la productividad, reducir los costes y los riesgos de mantenimiento.
- La finalidad de la ingeniería inversa es la de desentrañar los misterios y secretos de los sistemas en uso a partir del código.

La Ingeniería Inversa en EA permite al usuario importar un código fuente ya existente desde una variedad de lenguajes de código en un modelo del UML. Las estructuras de código fuente ya existentes serán asignadas en las representaciones, por ejemplo

una clase de Java será asignada en un elemento de clase del UML con las variables siendo definidas como los atributos, métodos que son modelados como operaciones y las interacciones entre las clases de Java que se muestran en el diagrama de clase del modelo del UML con los conectores apropiados. [WEB, 05].

A continuación se mencionan algunas situaciones sobre las que se puede realizar ingeniería inversa de código fuente:

- Código fuente que ya ha desarrollado.
- Código fuente que es parte de una biblioteca de terceras partes para la cual ha obtenido permiso para usar.
- Código que está siendo desarrollado diariamente por sus desarrolladores.

Enterprise Architect actualmente soporta ingeniería inversa en los siguientes lenguajes de programación.

- Actionscript
- C++
- C#
- Delphi
- Java
- PHP
- Visual Basic
- Visual Basic .NET

Enterprise Architect también puede realizar ingeniería inversa de ciertos tipos de archivos binarios: archivos Java .jar y archivos .NET

2.6 BASE DE DATOS OBJETO - RELACIONAL

El modelo de datos objeto relacional es usado en sistemas basados en el modelo de datos relacional incorporando procedimientos, objetos, versiones y otras nuevas capacidades (herencia, agregación, composición). El desarrollo del modelo objeto/relacional deriva del hecho que las ventajas del relacional se conservan y pueden integrarse aspectos de diseño que agregan claridad a la representación de los datos soportando el paradigma orientado a objetos.

La forma de trabajo del modelo objeto relacional se muestra en la siguiente figura 2.12.

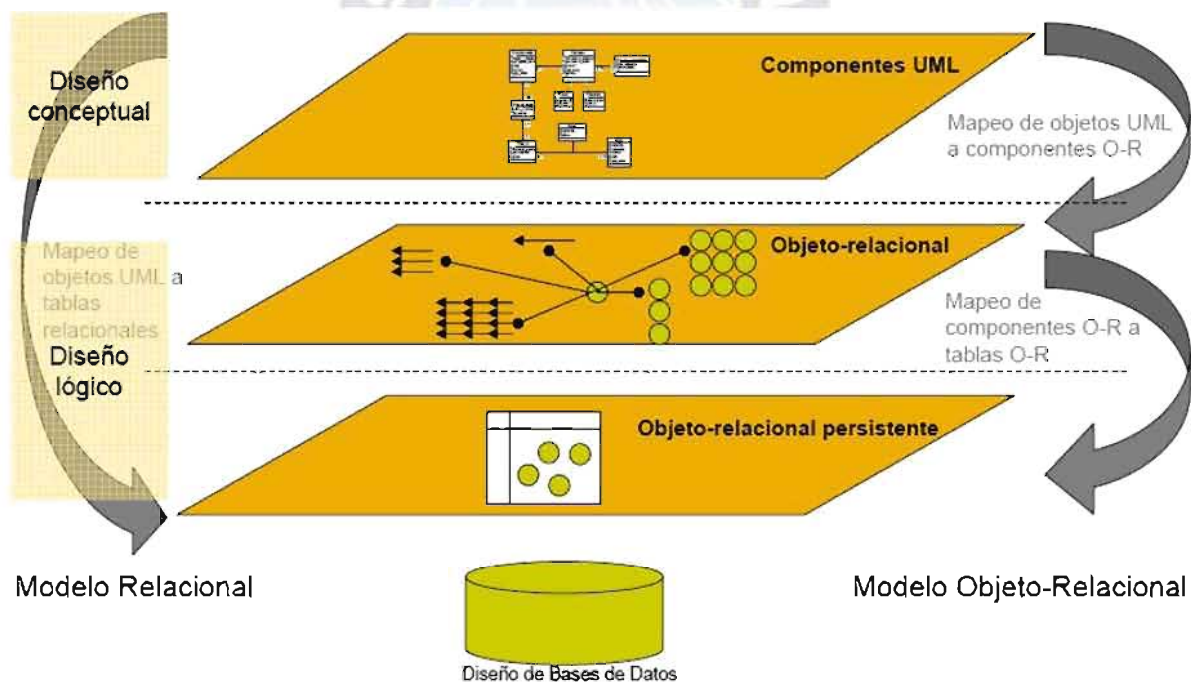


Figura 2.12: Modelo objeto relacional

Fuente: [WEB, 08]

Las características de las base de datos objeto relacionales son :

- Almacena objetos complejos.
- Soportan el paradigma orientado a objetos.

- Utiliza OID's (Identificadores de objetos)
- Reutilización futura
- Reducción de los posteriores errores
- Futuro mantenimiento.

2.7 LAN

LAN son las siglas de *Local Area Network*, Red de área local. Una LAN es una red que conecta los ordenadores en un área geográficamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios), generalmente con la misma tecnología.[WEB,07]

Las estaciones de trabajo y los ordenadores personales en oficinas normalmente están conectados en una red LAN, lo que permite que los usuarios envíen o reciban archivos y compartan el acceso a los archivos y a los datos. Cada ordenador conectado a una LAN se llama un nodo.

Cada nodo (ordenador individual) en un LAN tiene su propia CPU con la cual ejecuta programas, pero también puede tener acceso a los datos y a los dispositivos en cualquier parte en la LAN. Esto significa que muchos usuarios pueden compartir dispositivos caros, como impresoras laser, así como datos. Los usuarios pueden también utilizar la LAN para comunicarse entre ellos, enviando E-mail o chateando.

Una red de área local es una red en su versión más simple. La velocidad de transferencia de datos en una red de área local puede alcanzar hasta 10 Mbps (por ejemplo, en una red Ethernet) y 1 Gbps (por ejemplo, en FDDI o Gigabit Ethernet). Una red de área local puede contener 100, o incluso 1000, usuarios.

Al extender la definición de una LAN con los servicios que proporciona, se pueden definir dos modos operativos diferentes:

- En una red "de igual a igual", la comunicación se lleva a cabo de un equipo a otro sin un equipo central y cada equipo tiene la misma función.
- En un entorno "cliente/servidor", un equipo central brinda servicios de red para los usuarios. [WEB, 06].

2.8 SEGURIDAD

La encriptación es el proceso para volver ilegible información considerada importante. La información una vez encriptada sólo puede leerse aplicándole una clave.

Se trata de una medida de seguridad que es usada para almacenar o transferir información delicada que no debería ser accesible a terceros. Pueden ser contraseñas, números de tarjetas de crédito, conversaciones privadas, etc.

Para encriptar información se utilizan complejas fórmulas matemáticas y para desencriptar, se debe usar una clave como parámetro para esas fórmulas. El texto plano que está encriptado o cifrado se llama criptograma.

¿Cómo se mantiene indescifrable una cadena codificada?

Se trata de combinar la clave de encriptación con el mensaje de manera que las probabilidades de decodificar el mensaje, sin conocer la clave, sean virtualmente ínfimas, es decir, es tan prolongado el trabajo de desciframiento que no existen esperanzas próximas. Por ejemplo cuando un cristal ornamental se rompe violentamente, es casi imposible volver a juntar las piezas para obtener la pieza original.

2.9 CALIDAD DE SOFTWARE

Calidad del software es el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas que se esperan de todo software desarrollado profesionalmente. [PRESS, 05].

El estándar ISO 9126 se desarrolló como un intento por identificar los atributos de calidad para el software de computadora. El estándar identifica seis atributos clave de calidad:

Funcionalidad: El grado en que el software satisface las necesidades que indican los siguientes subatributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

Confiabilidad: La cantidad de tiempo en que el software está disponible para usarlo según los siguiente subatributos: madurez, tolerancia a fallas y facilidad de recuperación.

Facilidad de uso: La facilidad con que se usa el software de acuerdo con los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operabilidad.

Eficiencia: El grado en que el software emplea en forma óptima los recursos del sistema, como lo indican los siguientes subatributos: comportamiento en el tiempo, comportamiento de los recursos.

Facilidad de mantenimiento: La facilidad con que se separa el software de acuerdo con los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Portabilidad: La facilidad con que se lleva el software de un entorno a otro según los siguientes subatributos: adaptabilidad, facilidad para instalarse, cumplimiento, facilidad para reemplazarse.

2.9.1 Métricas Basadas en la Función

La métrica de punto función (**PF**) propuesta inicialmente por Albretch, se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. Empleando datos históricos, el 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 encontrarán durante la prueba y pronosticar el número de componentes, de líneas de código proyectadas, o ambas, en el sistema implementado. [PRESS, 05]

Se determina cinco características de dominios de información y se proporcionan las cuentas en posición apropiada de una tabla. Los valores del dominio de la información se definen de la siguiente manera:

Número de entradas de usuario (EU): Cada entrada externa se origina en un usuario o es transmitida desde otra aplicación y proporciona distintos datos orientados a la aplicación o información de control. Las entradas suelen emplearse para actualizar archivos lógicos internos (ALI). Las entradas deben distinguirse de las consultas, que se cuentan por separado.

Número de salidas de usuario (SU): Cada salida externa se deriva en el interior de la aplicación y proporciona información al usuario. En este contexto, salida externa alude a informes, pantallas, mensajes de error, etc. Los elementos de datos individuales dentro de un informe no se cuentan por separado.

Número de peticiones de usuario (PU): Una petición de usuario se define como la entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software, en la forma de salida en línea.

Número de archivos (ALI): Cada archivo lógico interno es un agrupamiento lógico de datos que residen dentro de los límites de las aplicaciones y que se mantienen mediante entradas externas.

Número de archivos de interfaz externos (AIE): Cada archivo de interfaz externo es un agrupamiento lógico de datos externos a la aplicación pero que proporciona datos que podrían usarse en esta.

Una vez que se ha recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Las organizaciones que usan métodos de punto función desarrollan criterios para determinar si una entrada determinada es simple, promedio o compleja. No obstante, la determinación de la complejidad es un poco subjetiva.

Para calcular puntos de función (PF) se usa la siguiente relación:

$$PF = \text{Conteo_Total} \times ([X + Y * \sum [Fi])$$

Donde:

PF: Medida de funcionalidad entregada a la aplicación como valor de normalización.

Conteo_Total: Es la suma de todas las entradas obtenidas.

X: Nivel de confiabilidad del sistema.

Y: Nivel de significación de error.

Fi: Valores de ajuste de complejidad.

ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo, se desarrolla las fases correspondientes al análisis, diseño, implementación y pruebas, siguiendo el proceso de desarrollo RUP detallado en el Marco Teórico.

Tomando en cuenta los objetivos del proyecto se desarrolla los módulos de control de recursos y control de personal de la institución.

3.1 FASE INICIAL

En esta primera fase se analiza todos los problemas que existían en la administración de la Clínica Caja Petrolera de Salud, posteriormente se hizo un estudio de los requerimientos de los usuarios mediante entrevistas.

3.1.1 Requerimientos de Usuario

El paso fundamental dentro del análisis de sistemas es el análisis de requisitos, para así comprender las necesidades del usuario, además permite tener una mejor visión del problema y encontrar el camino hacia la solución, que satisfaga esas necesidades. Dentro de este contexto, se identifica una serie de requisitos que son descritos a continuación:

- Registrar los pedidos de reposición de materiales, mantenimiento de equipos, transporte, descargo de transporte hechos por el personal.
- Generar reportes de recursos y personal según las especificaciones de la institución en tiempo real.
- Controlar el número de pedidos de materiales y mantenimiento.
- Captura de datos del personal de la clínica.
- Registrar los permisos, bajas, cambio voluntario de turno, comisiones, vacaciones, hechos por el personal de la clínica.

3.2 FASE DE ELABORACIÓN

En esta fase se establece el contexto del sistema mediante el modelo del negocio a través de casos de uso logrando definir el alcance del proyecto.

3.2.1 Modelo del negocio

El modelo del negocio es una técnica para comprender los procesos de la organización. El modelo del negocio describe los procesos en términos de casos de uso y actores del contexto que se corresponden con los procesos del sistema y los usuarios.

El modelado del negocio detallado a continuación (Ver Figura 3.1) identifica los procesos más importantes del contexto del sistema y describe los procesos exactos relacionados con los actores y casos de uso encontrados.

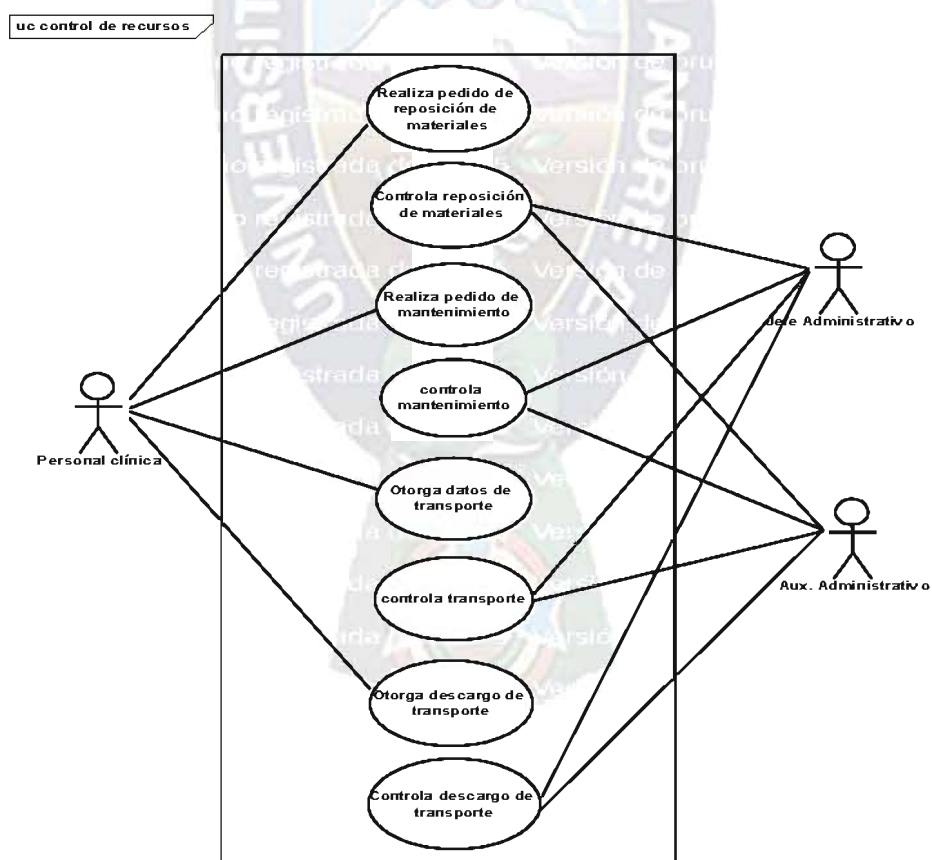


Figura 3.1: Modelado de negocio (Módulo Control de Recursos)

Fuente: Elaboración Propia

En la Figura 3.2 Se describe el modelo de negocio del módulo Control de Personal, se muestra todos los procesos y la relación que existen con los roles externos.

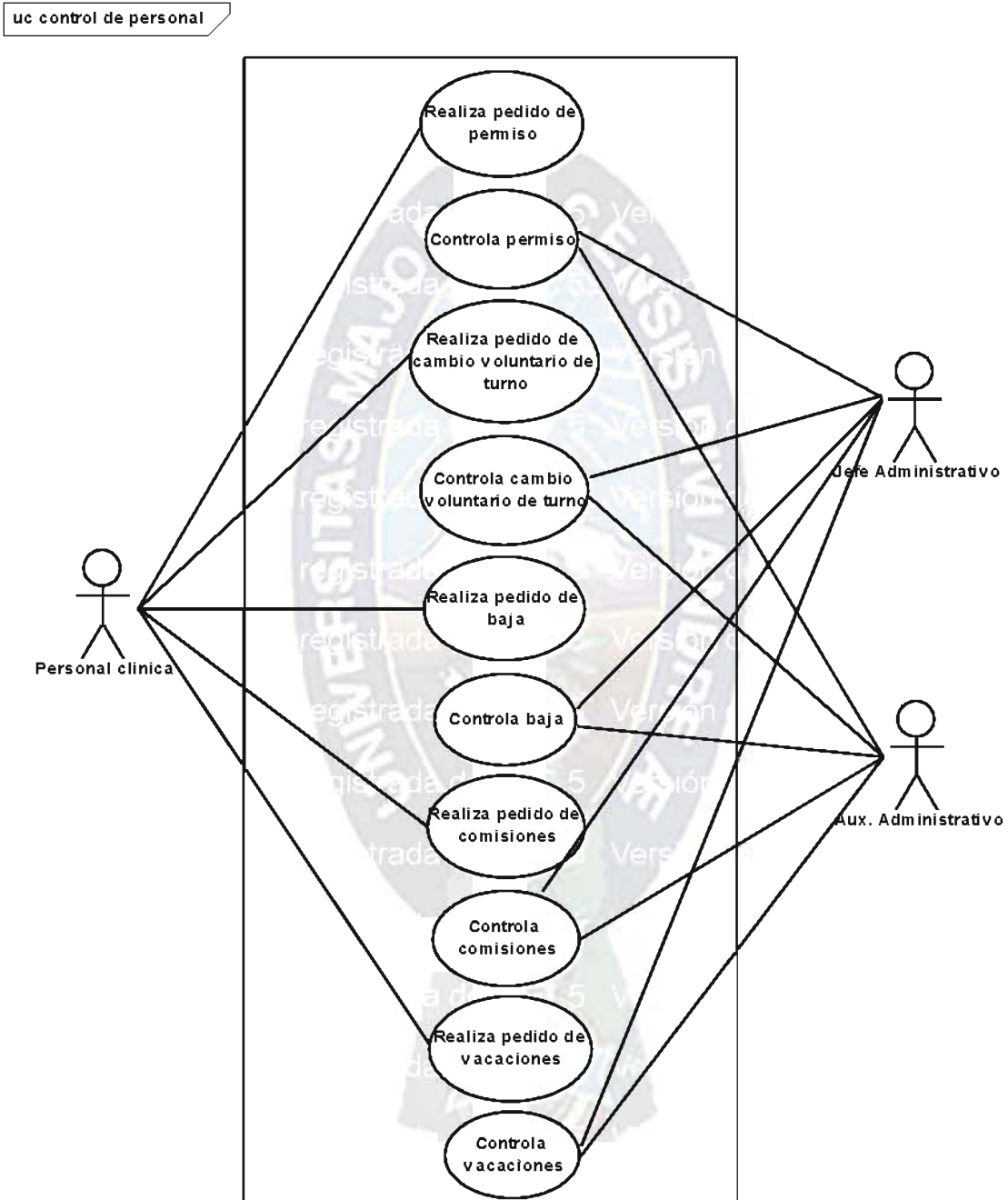


Figura 3.2: Modelado de negocio (Módulo Control de Personal)

Fuente: Elaboración Propia

3.2.2 Modelo de Caso de Uso

El modelo de Caso de Uso nos permite representar a los diferentes actores y los casos de uso de la institución. En los siguientes diagramas podemos observar a los principales actores que interactúan con el sistema.

En la Figura 3.3 se presenta el Caso de Uso del módulo control de recursos.

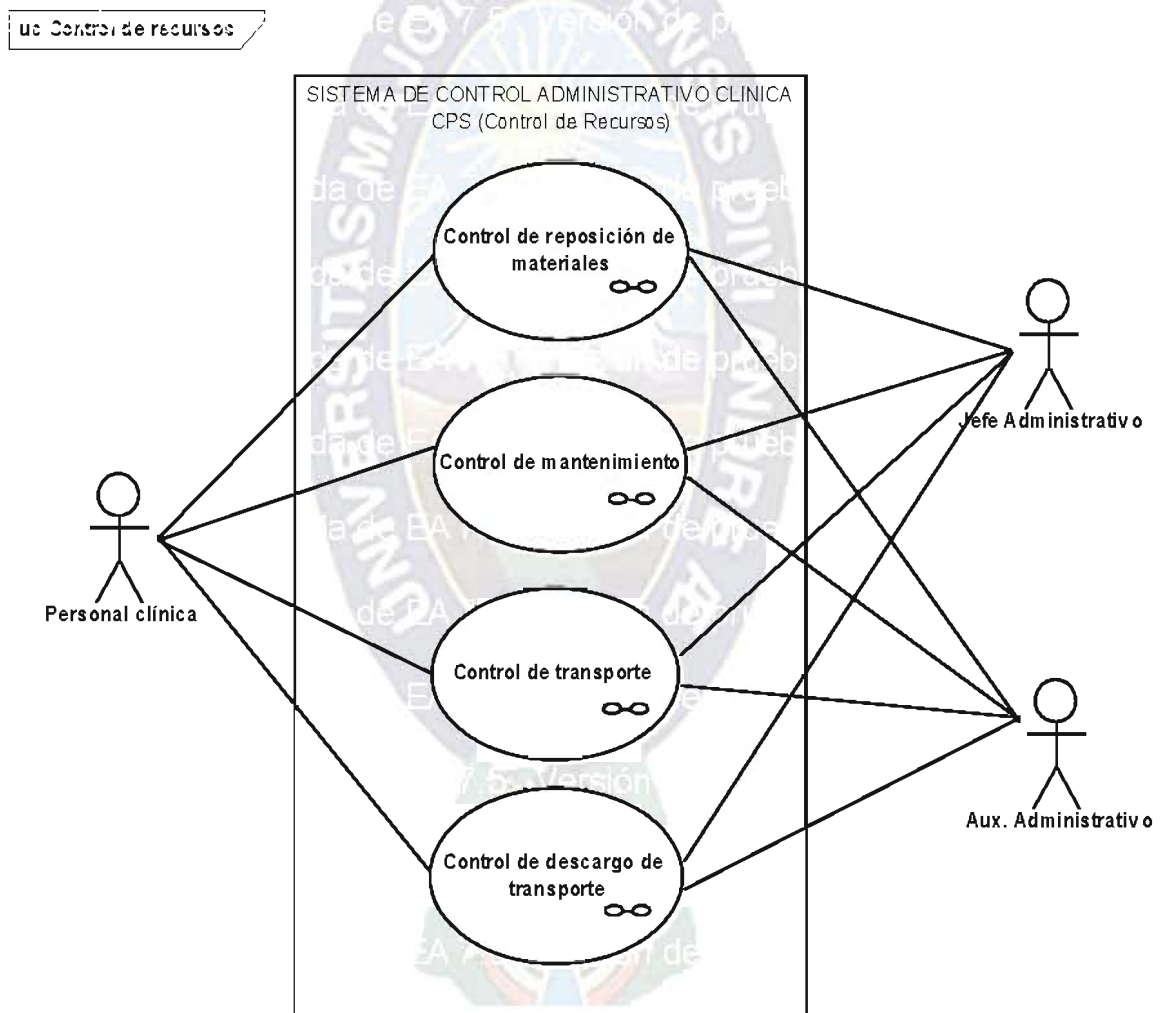


Figura 3.3: Modelo de Caso de Uso (Módulo Control de Recursos)
Fuente: Elaboración Propia

En la Figura 3.4 se presenta el Caso de Uso del módulo control de personal.

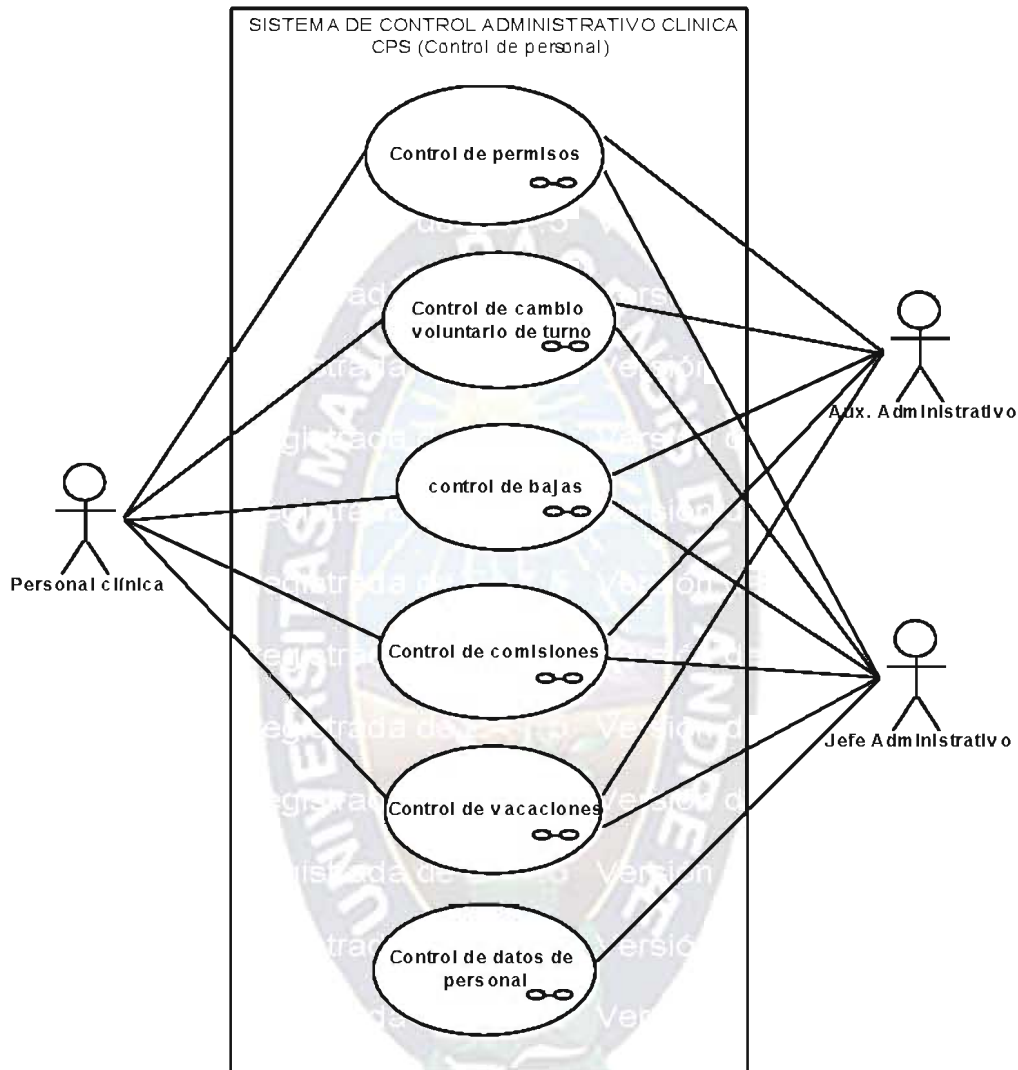


Figura 3.4: Modelo de Caso de Uso (Módulo Control de Personal)

Fuente: Elaboración Propia

3.2.2.1 Actores

Personal de clínica

El personal de la clínica es un empleado perteneciente a cualquier unidad de la institución, este dependiendo de las necesidades que tenga realiza el pedido de materiales, mantenimiento, permisos, comisiones, etc.

Auxiliar Administrativo

El auxiliar administrativo realiza el registro de los pedidos del personal correspondiente a recursos y a personal.

Jefe Administrativo

El jefe administrativo realiza todo el control que se refiere a recursos y a personal, realizando el registro de datos, generando reportes, realizando modificaciones, dependiendo de las necesidades que tenga.

3.2.3 Diagramas de Caso de Uso

Para el módulo control de recursos se detalla los diagramas de caso de uso más representativos.

➤ Caso de Uso: Control de Reposición de materiales

uc Control de reposición de materiales

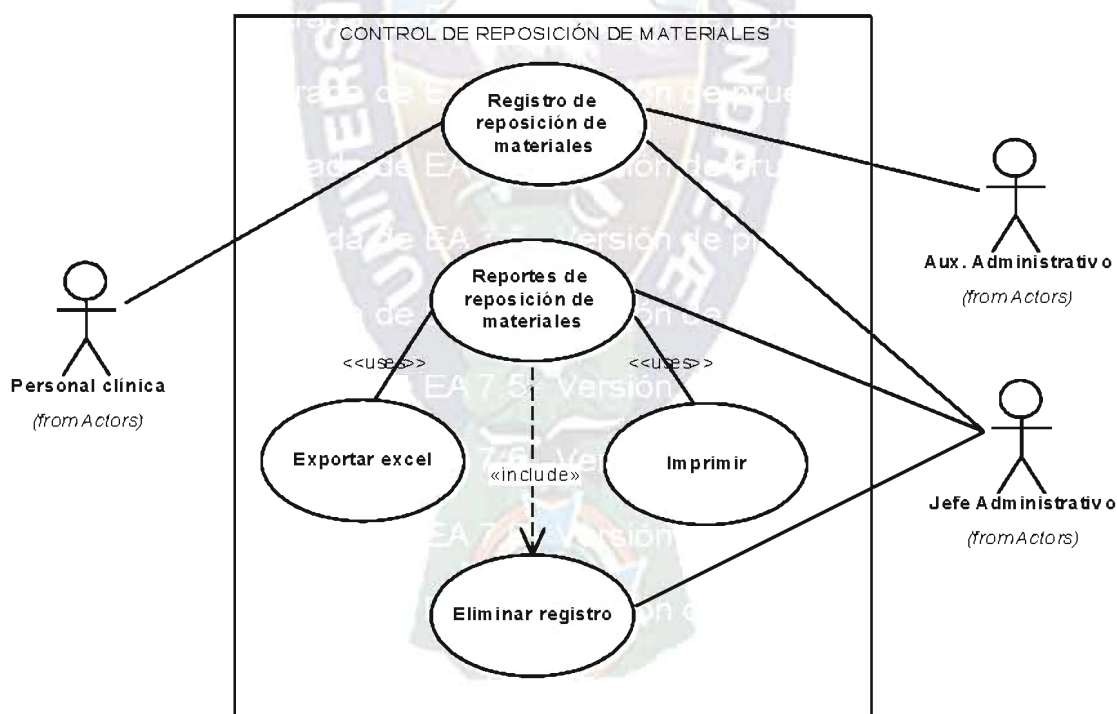


Figura 3.5: Caso de Uso: Control de reposición de materiales
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de reposición de materiales.

Tabla 3.1: Caso de Uso: Control de reposición de materiales
Fuente: Elaboración Propia

Caso de Uso:	Control de Reposición de materiales	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de materiales	
Descripción:	El personal de la clínica realiza el pedido de materiales, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de reposición de materiales.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario	
6.- El jefe administrativo elige la opción reportes de reposición de materiales	7.- Muestra reportes de acuerdo a especificaciones del usuario	
8.- El jefe administrativo elige la opción eliminar registro.	9.- Muestra mensaje de alerta.	
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.	
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.	
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.	
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.	
CURSOS ALTERNOS		
Línea 1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.		
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.		

➤ **Caso de Uso: Control de Mantenimiento**

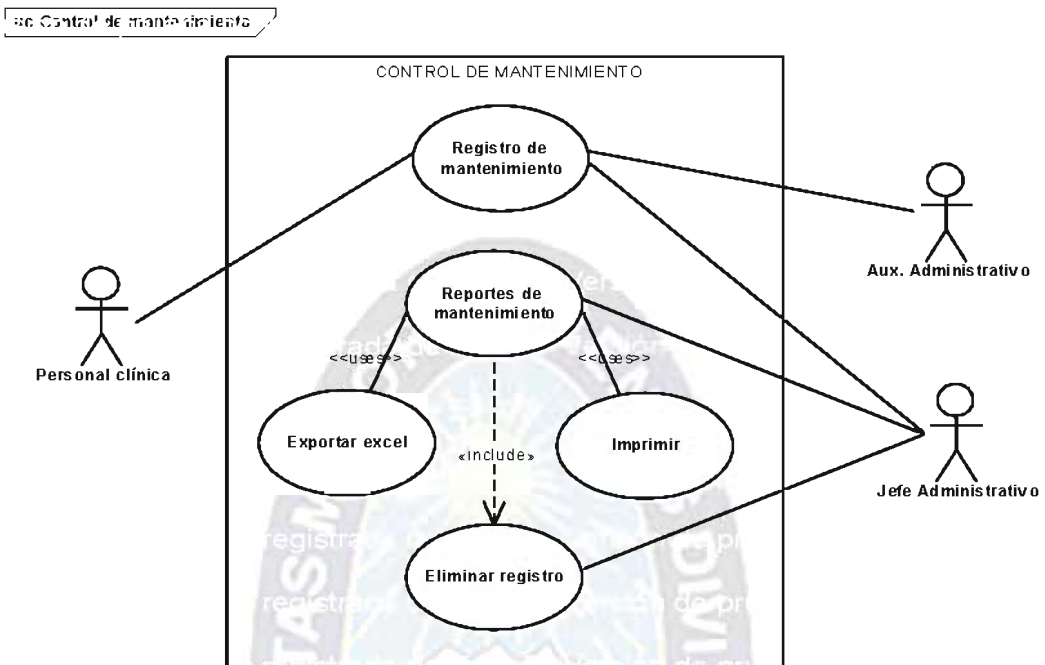


Figura 3.6: Caso de Uso: Control de mantenimiento
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de mantenimiento.

Tabla 3.2: Caso de Uso: Control de mantenimiento
Fuente: Elaboración Propia

Caso de Uso:	Control de mantenimiento	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de mantenimiento	
Descripción:	El personal de la clínica realiza el pedido de mantenimiento, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	

4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de mantenimiento.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario
6.- El jefe administrativo elije la opción reportes de mantenimiento.	7.- Muestra reportes de acuerdo a especificaciones del usuario.
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	

➤ Caso de Uso: Control de Transporte

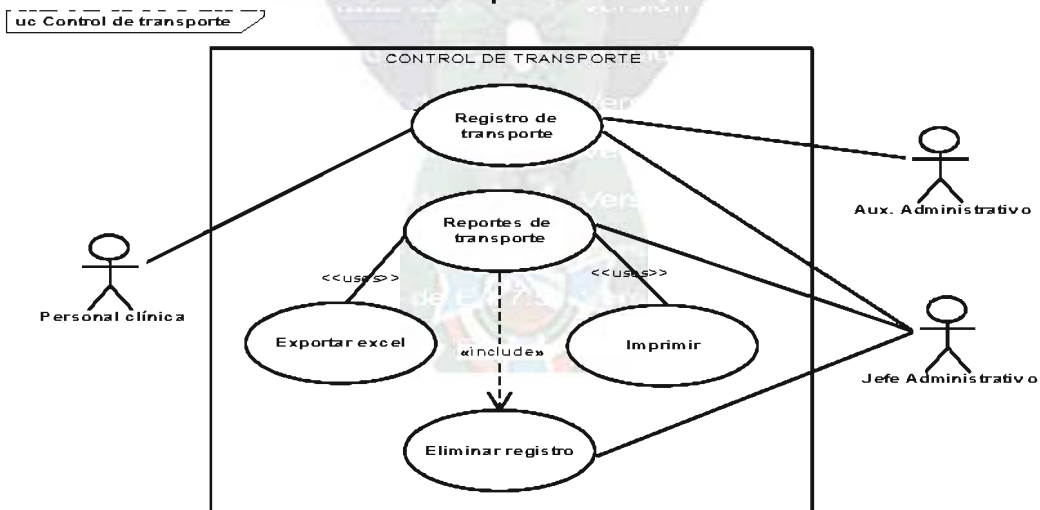


Figura 3.7: Caso de Uso: Control de Transporte
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Transporte.

Tabla 3.3: Caso de Uso: Control de Transporte
Fuente: Elaboración Propia

Caso de Uso:	Control de Transporte	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de transporte	
Descripción:	El personal de la clínica realiza el reporte de insumos de transporte, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de transporte.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario	
6.- El jefe administrativo elije la opción reportes de transporte.	7.- Muestra reportes de acuerdo a especificaciones del usuario.	
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.	
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.	
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.	
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.	
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.	
CURSOS ALTERNOS		

Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.
 Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.
 Línea 3: Si el saldo de la tarjeta de gasolina es menor a 500 bolivianos da mensaje de alerta.

➤ **Caso de Uso: Control de Descargo de transporte**

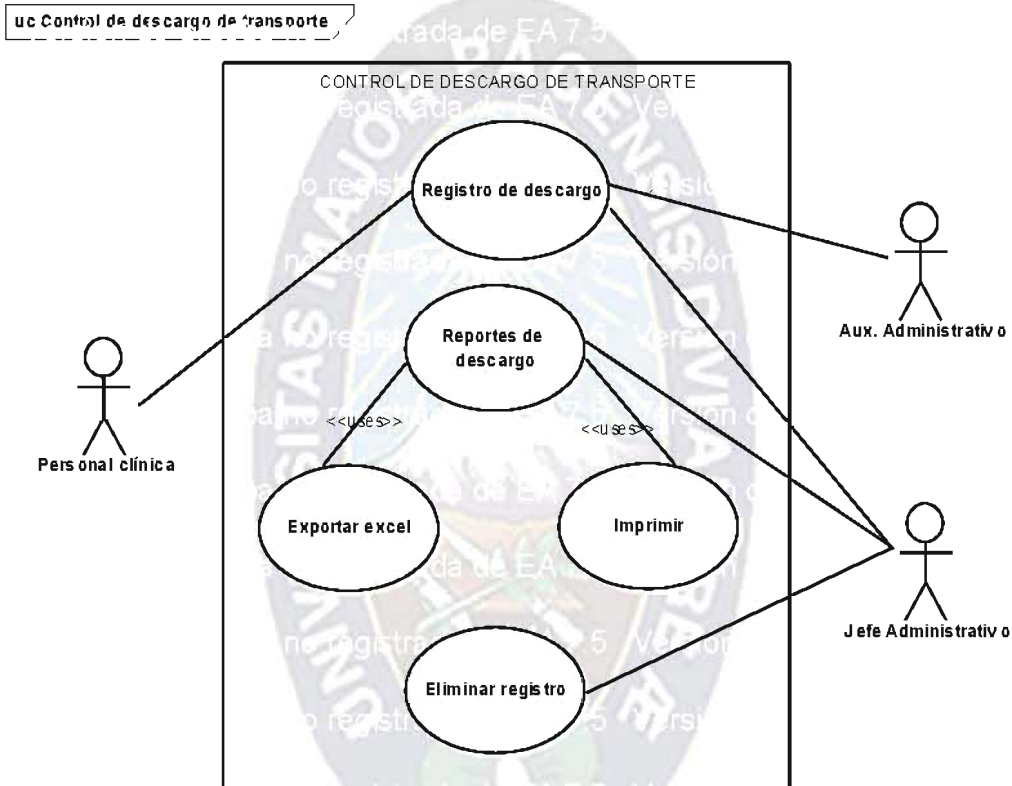


Figura 3.8: Caso de Uso: Control de Descargo de Transporte
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Descargo de Transporte.

Tabla 3.4: Caso de Uso: Control de Descargo de Transporte
Fuente: Elaboración Propia

Caso de Uso:	Control de Descargo de Transporte
Actores:	Personal clínica, Aux. administrativo, jefe administrativo
Propósito:	Tener datos actualizados del control de descargo de transporte
Descripción:	El personal de la clínica realiza el descargo de transporte, el auxiliar administrativo o el jefe administrativo registra los datos,

	posteriormente el jefe administrativo realiza el control de esta área con los reportes.
Tipo:	Primario y esencial
Referencias cruzadas:	
CURSO NORMAL DE EVENTOS	
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA
	1.- Solicita código y contraseña
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.
4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de descargo de transporte.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario
6.- El jefe administrativo elige la opción reportes de descargo de transporte.	7.- Muestra reportes de acuerdo a especificaciones del usuario.
8.- El jefe administrativo elige la opción eliminar registro.	9.- Muestra mensaje de alerta.
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea 1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	
Línea 3: Si el saldo de la tarjeta de gasolina es menor a 500 bolivianos da mensaje de alerta.	

Para el módulo Control de Personal se presenta los siguientes diagramas de Caso de Uso.

➤ Caso de Uso: Control de Permisos

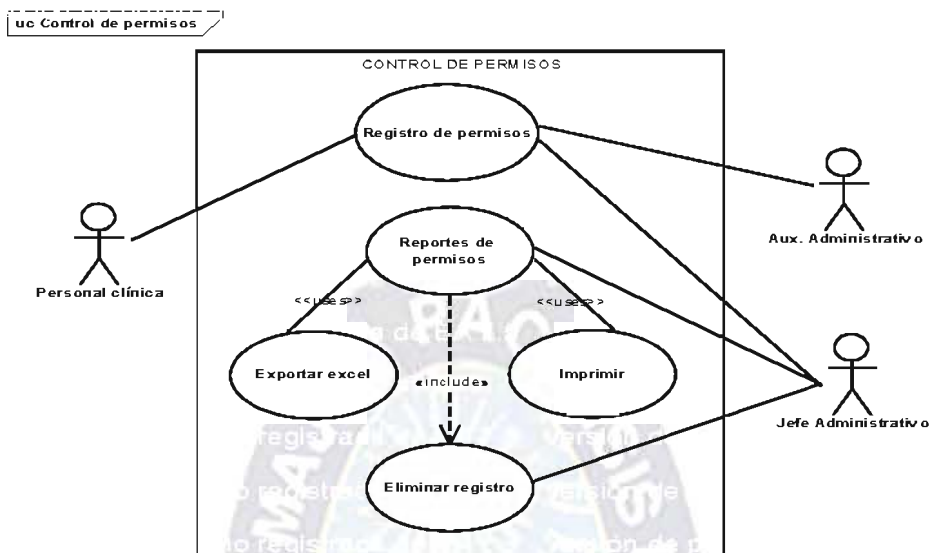


Figura 3.9: Caso de Uso: Control de Permisos
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Permisos.

Tabla 3.5: Caso de Uso: Control de Permisos
Fuente: Elaboración Propia

Caso de Uso:	Control de Permisos	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de permisos	
Descripción:	El personal de la clínica realiza el pedido de permiso, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de permiso.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento	

	de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario
6.- El jefe administrativo elije la opción reportes de permiso.	7.- Muestra reportes de acuerdo a especificaciones del usuario.
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	

➤ **Caso de Uso: Control de Cambio Voluntario de Turno**

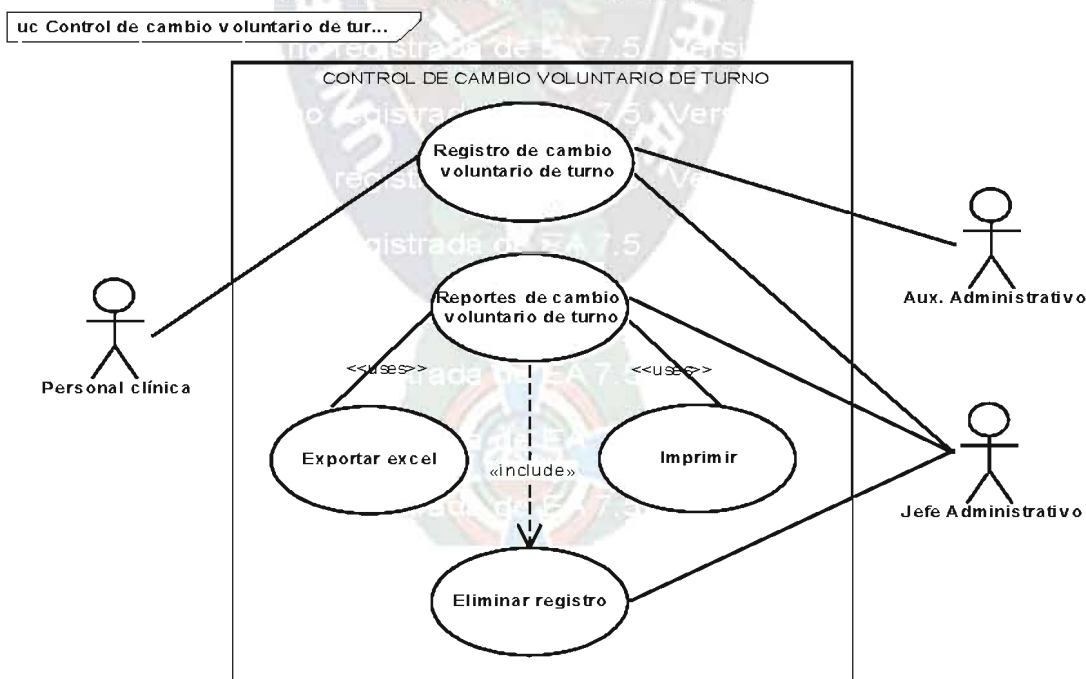


Figura 3.10: Caso de Uso: Control de Cambio Voluntario de Turno
Fuente: Elaboración Propia

Seguidamente se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Cambio Voluntario de Turno.

Tabla 3.6: Caso de Uso: Control de Cambio Voluntario de Turno

Fuente: Elaboración Propia

Caso de Uso:	Control de Cambio Voluntario de Turno	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de cambio voluntario de turno	
Descripción:	El personal de la clínica realiza el pedido de cambio voluntario de turno, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce datos de cambio voluntario de turno.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario	
6.- El jefe administrativo elije la opción reportes de cambio voluntario de turno.	7.- Muestra reportes de acuerdo a especificaciones del usuario.	
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.	
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.	
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.	
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.	

16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	

➤ **Caso de Uso: Control de Bajas**

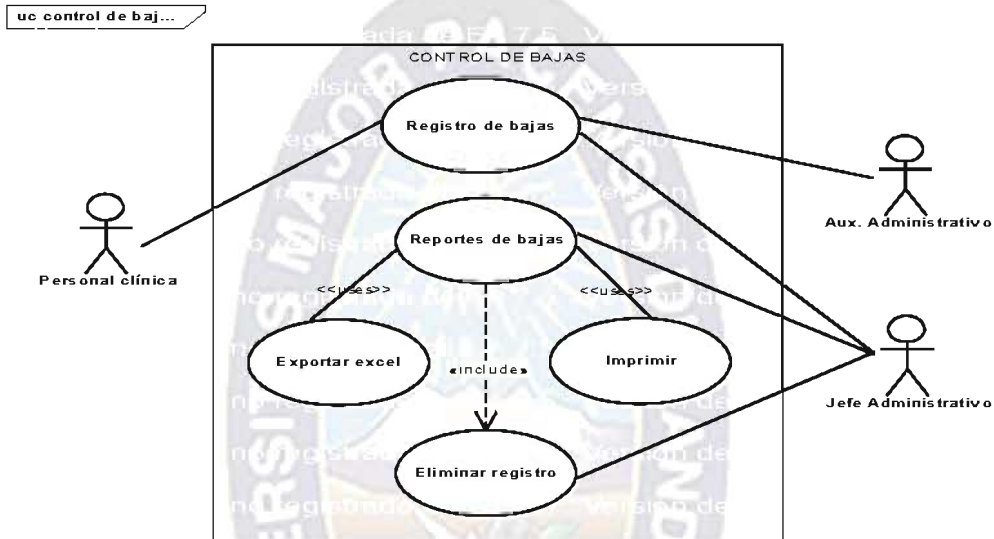


Figura 3.11: Caso de Uso: Control de Bajas
Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Bajas.

Tabla 3.7: Caso de Uso: Control de Bajas
Fuente: Elaboración Propia

Caso de Uso:	Control de Bajas	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de bajas	
Descripción:	El personal de la clínica realiza el pedido de baja, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al	

	nivel de acceso.
4.- El usuario (Aux. administrativo o jefe administrativo) introduce bajas.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario
6.- El jefe administrativo elije la opción reportes de bajas.	7.- Muestra reportes de acuerdo a especificaciones del usuario.
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	

➤ Caso de Uso: Control de Comisiones

uc Control de comisiones

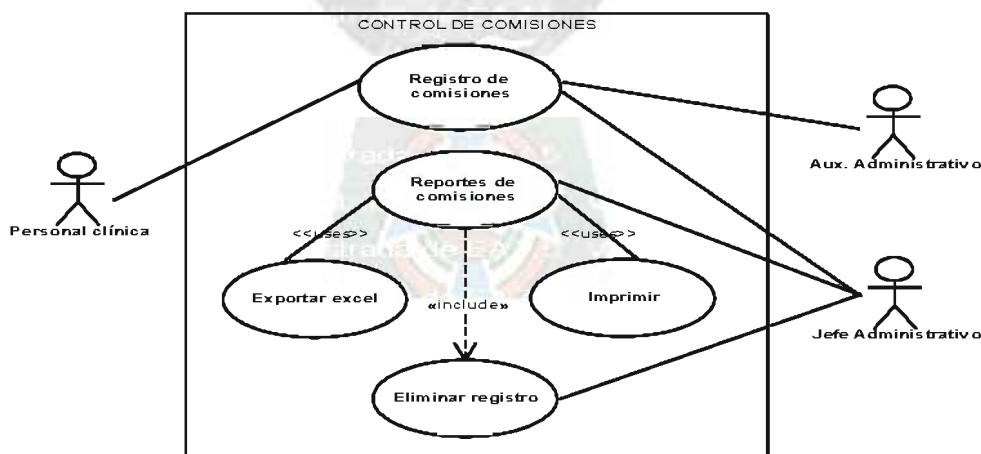


Figura 3.12: Caso de Uso: Control de Comisiones
Fuente: Elaboración Propia

Seguidamente se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Comisiones.

Tabla 3.8: Caso de Uso: Control de Comisiones

Fuente: Elaboración Propia

Caso de Uso:	Control de Comisiones	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de comisiones	
Descripción:	El personal de la clínica realiza el pedido de comisión, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce comisiones.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario	
6.- El jefe administrativo elije la opción reportes de comisiones.	7.- Muestra reportes de acuerdo a especificaciones del usuario.	
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.	
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.	
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.	
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.	
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.	
CURSOS ALTERNOS		
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva		

solicitud.
 Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.

➤ **Caso de Uso: Control de Vacaciones**

Un Control de Vacación...

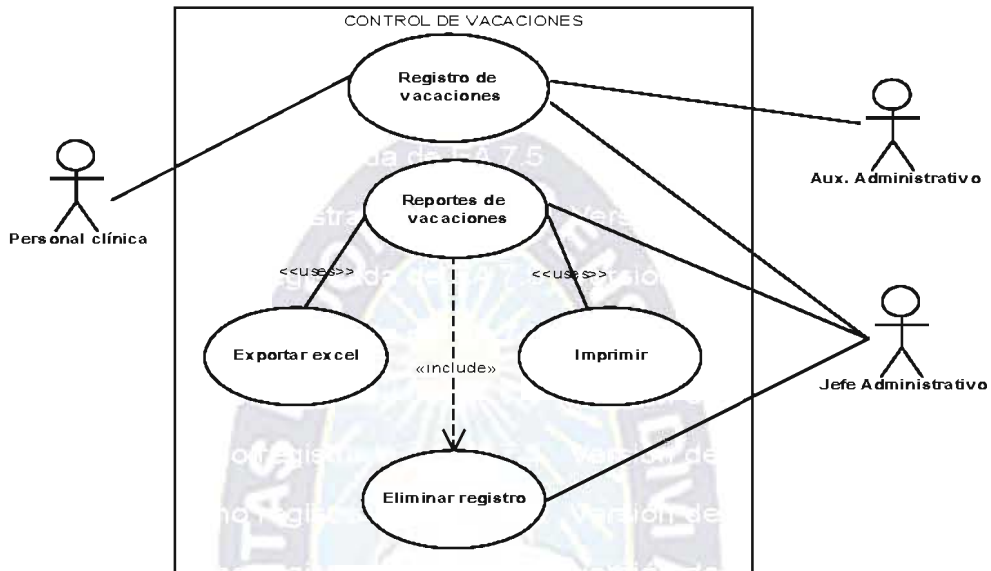


Figura 3.13: Caso de Uso: Control de Vacaciones
 Fuente: Elaboración Propia

A continuación se muestra una tabla que describe a un nivel más detallado el caso de uso: control de Vacaciones.

Tabla 3.9: Caso de Uso: Control de Vacaciones
 Fuente: Elaboración Propia

Caso de Uso:	Control de Vacaciones	
Actores:	Personal clínica, Aux. administrativo, jefe administrativo	
Propósito:	Tener datos actualizados del control de vacaciones	
Descripción:	El personal de la clínica realiza el pedido de vacación, el auxiliar administrativo o el jefe administrativo registra los datos, posteriormente el jefe administrativo realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El usuario (Aux. administrativo o jefe administrativo) introduce vacaciones.	5.- El sistema constata si los datos introducidos son correctos:	

	<p>a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos.</p> <p>b) Los datos son incorrectos, se muestra mensaje de alerta al usuario</p>
6.- El jefe administrativo elije la opción reportes de vacaciones.	7.- Muestra reportes de acuerdo a especificaciones del usuario.
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.
CURSOS ALTERNOS	
Línea1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.	
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.	

➤ **Caso de Uso: Control de datos de Personal**

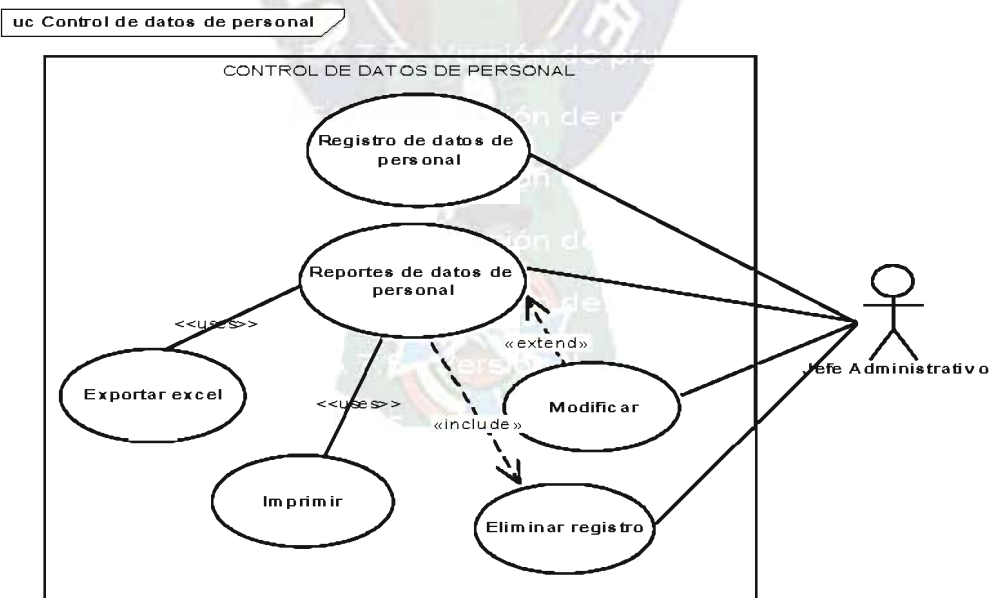


Figura 3.14: Caso de Uso: Control de datos de Personal
Fuente: Elaboración Propia

Seguidamente se muestra una tabla que describe a un nivel más detallado el caso de uso: control de datos de Personal.

Tabla 3.10: Caso de Uso: Control de datos de Personal

Fuente: Elaboración Propia

Caso de Uso:	Control de datos de Personal	
Actores:	Jefe administrativo	
Propósito:	Tener datos actualizados del control de datos de personal	
Descripción:	El jefe administrativo registra los datos, actualización, posteriormente realiza el control de esta área con los reportes.	
Tipo:	Primario y esencial	
Referencias cruzadas:		
CURSO NORMAL DE EVENTOS		
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA	
	1.- Solicita código y contraseña	
2.- Usuario introduce contraseña.	3.- El sistema identifica el tipo de usuario y retorna pantalla con opciones de acuerdo al nivel de acceso.	
4.- El jefe administrativo introduce datos de personal.	5.- El sistema constata si los datos introducidos son correctos: a) Los datos son correctos, el sistema registra datos y muestra mensaje para confirmar el almacenamiento de datos. b) Los datos son incorrectos, se muestra mensaje de alerta al usuario	
6.- El jefe administrativo elije la opción reportes de datos de personal.	7.- Muestra reportes de acuerdo a especificaciones del usuario.	
8.- El jefe administrativo elije la opción eliminar registro.	9.- Muestra mensaje de alerta.	
10.- El jefe administrativo confirma la eliminación de registro.	11.- Se elimina el registro.	
12.- El jefe administrativo selecciona la opción exportar a Excel.	13.- Muestra una ventana para introducir el nombre con el que se guardará el archivo.	
14.- El jefe administrativo ingresa el nombre y presiona aceptar.	15.- muestra el reporte en Excel.	
16.- El jefe administrativo selecciona opción imprimir.	17.- Muestra el reporte en hoja para imprimir.	
18.- El jefe administrativo selecciona la opción modificar.	19.- Muestra una nueva ventana con la opción de datos a modificar.	
20.- modifica los datos que se requiera y guarda.	21.- Guarda y actualiza los datos modificados.	

CURSOS ALTERNOS
Línea 1: Introducción de password no valido, devuelve mensaje de error y realiza nueva solicitud.
Línea 2: Introducción de datos incompletos, devuelve mensaje de advertencia.
Línea 3: Introduce datos ya registrados, devuelve mensaje de alerta.

3.2.4 Diagrama de Secuencia

Mediante los Diagramas de Secuencia se explica los pasos a seguir en forma secuencial de los diferentes procesos que existen en el sistema actual y que serán desarrollados de acuerdo a los casos de uso esenciales, comenzando de la acción del actor y la respuesta del sistema mediante la acción de las clases externas.

Para el módulo control de recursos se tienen los siguientes diagramas:

Control de Reposición de Materiales. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.15

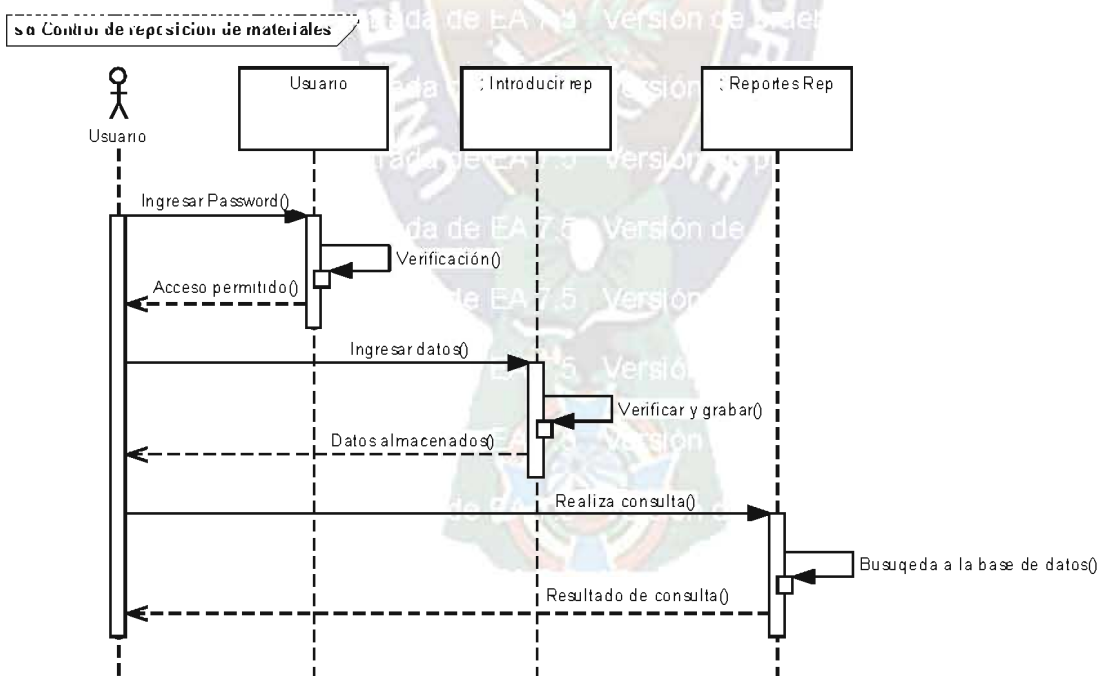


Figura 3.15: Diagrama de Secuencia: Control de Reposición de Materiales
Fuente: Elaboración Propia

Control de Mantenimiento. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.16

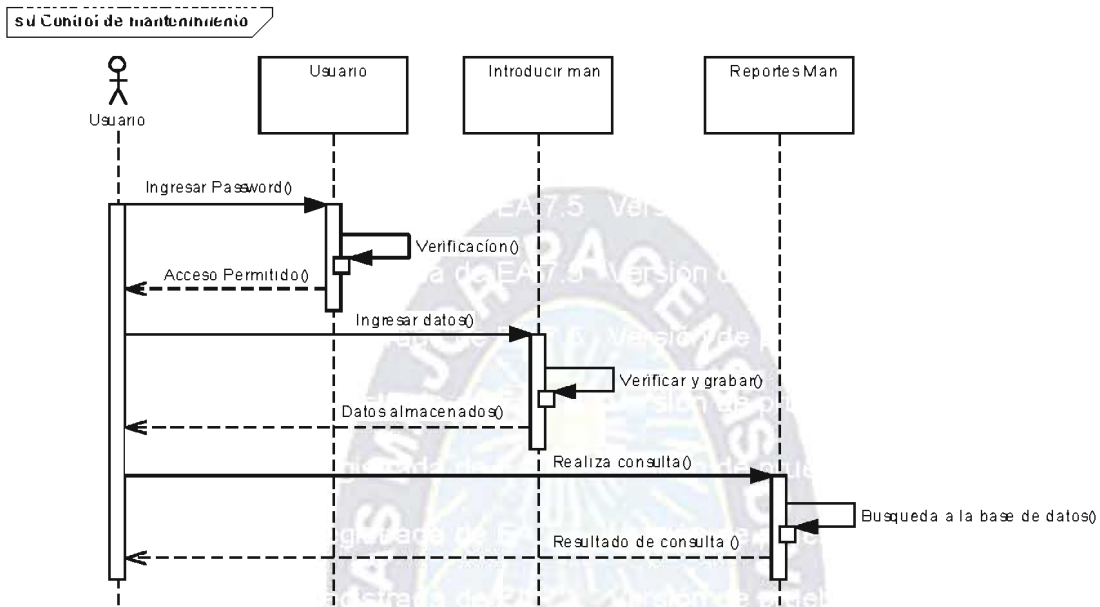


Figura 3.16: Diagrama de Secuencia: Control de Mantenimiento
Fuente: Elaboración Propia

Control de Transporte. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.17

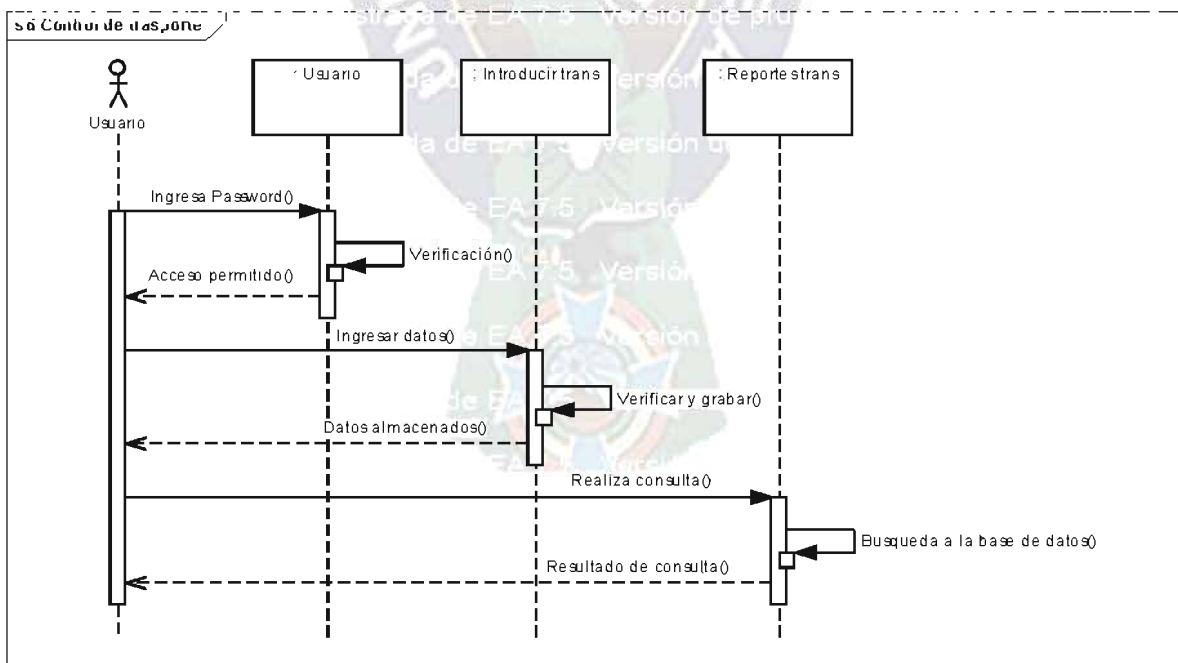


Figura 3.17: Diagrama de Secuencia: Control de Transporte
Fuente: Elaboración Propia

Control de Descargo de Transporte. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.18

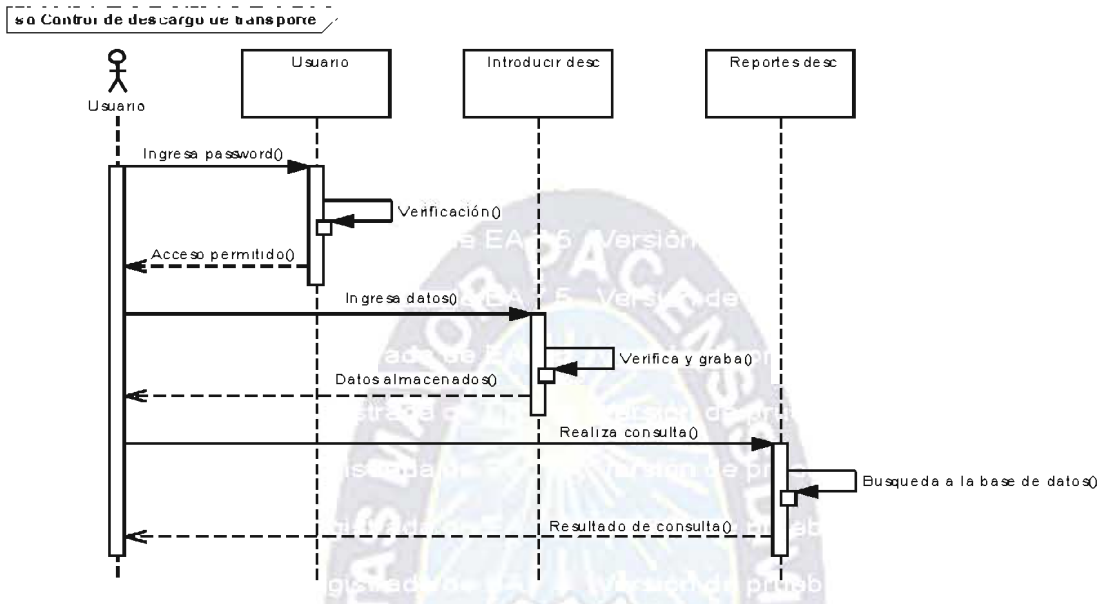


Figura 3.18: Diagrama de Secuencia: Control de Descargo de Transporte
Fuente: Elaboración Propia

Para el módulo control de personal se tienen los siguientes diagramas:

Control de Permisos. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.19

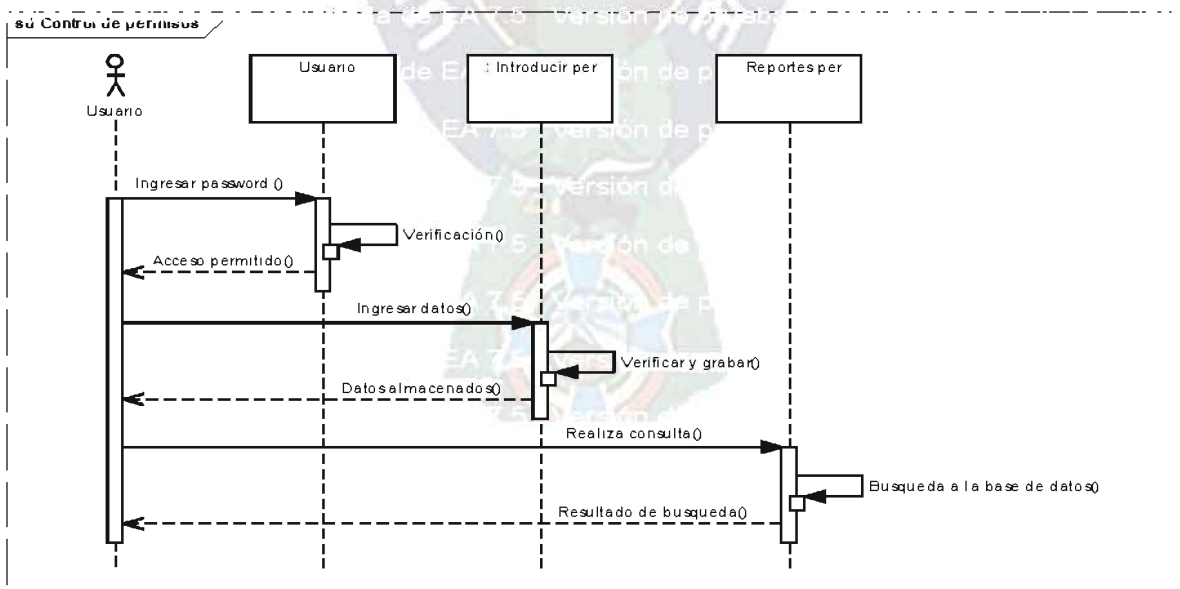


Figura 3.19: Diagrama de Secuencia: Control de Permisos
Fuente: Elaboración Propia

Control de Cambio Voluntario de Turno. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.20

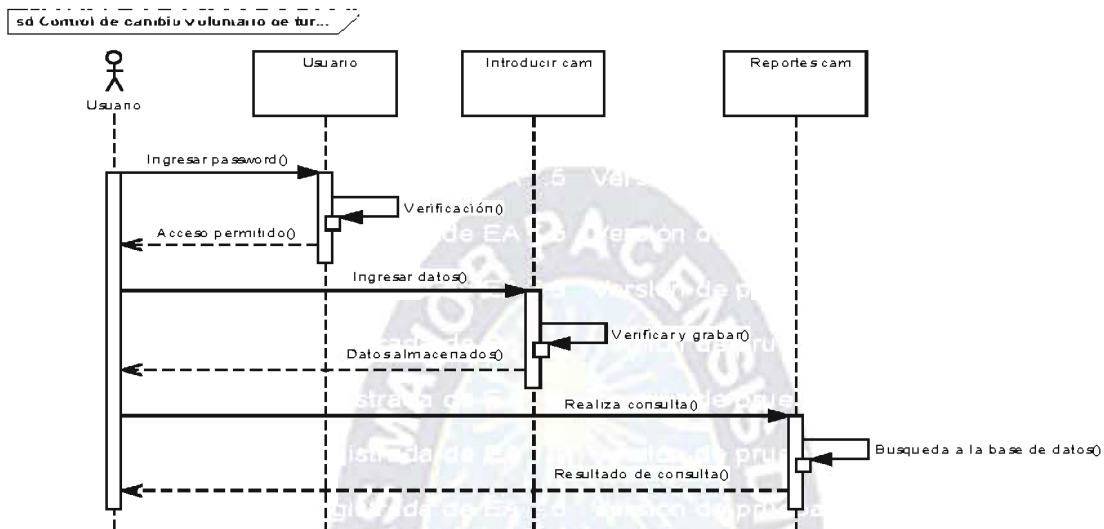


Figura 3.20: Diagrama de Secuencia: Control de cambio Voluntario de Turno
Fuente: Elaboración Propia

Control de Bajas. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.21

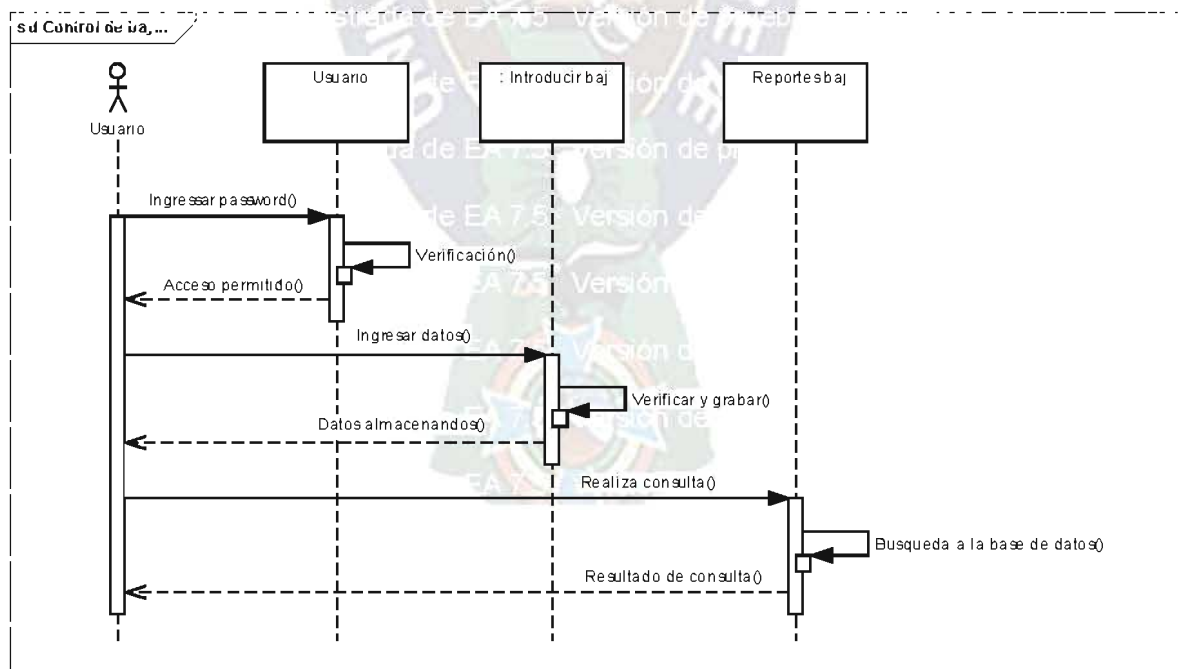


Figura 3.21: Diagrama de Secuencia: Control de Bajas
Fuente: Elaboración Propia

Control de Comisiones. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.22

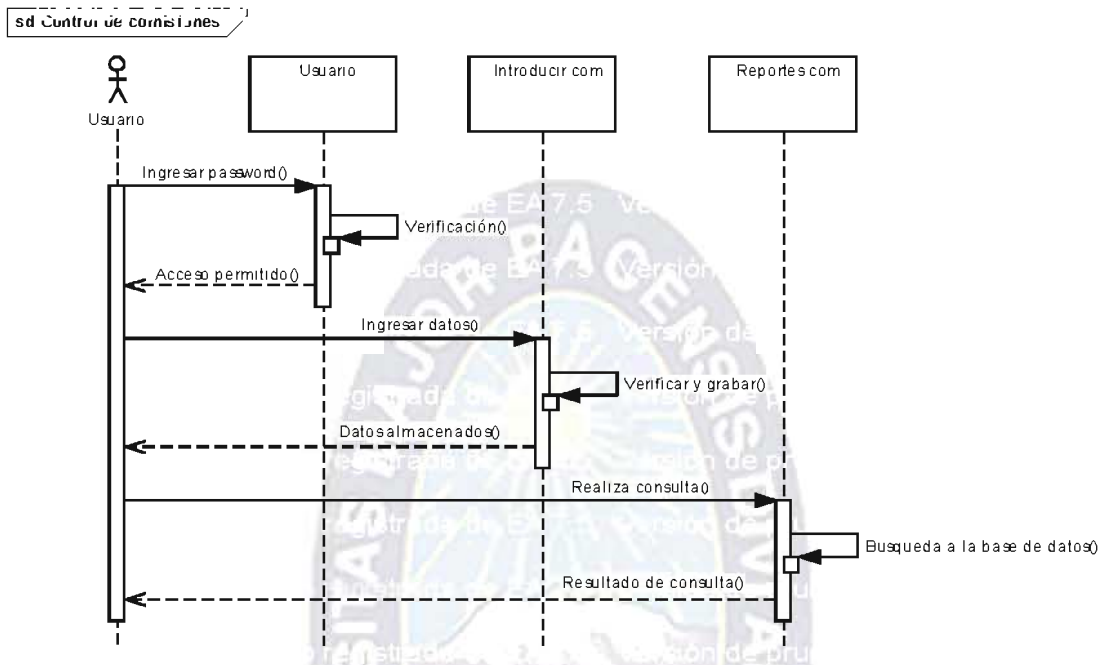


Figura 3.22: Diagrama de Secuencia: Control de Comisiones
Fuente: Elaboración Propia

Control de Vacaciones. Los usuarios son Aux. administrativo y jefe administrativo que se encuentran en la Figura 3.23

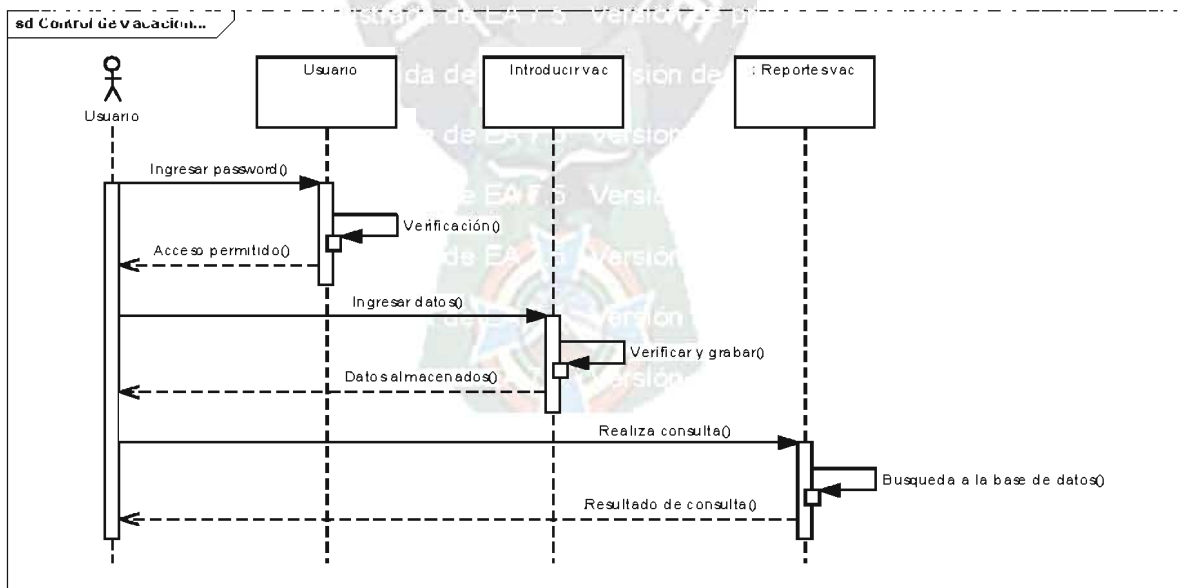


Figura 3.23: Diagrama de Secuencia: Control de Vacaciones
Fuente: Elaboración Propia

Control de datos de Personal. El usuario será jefe administrativo que se encuentra en la Figura 3.24

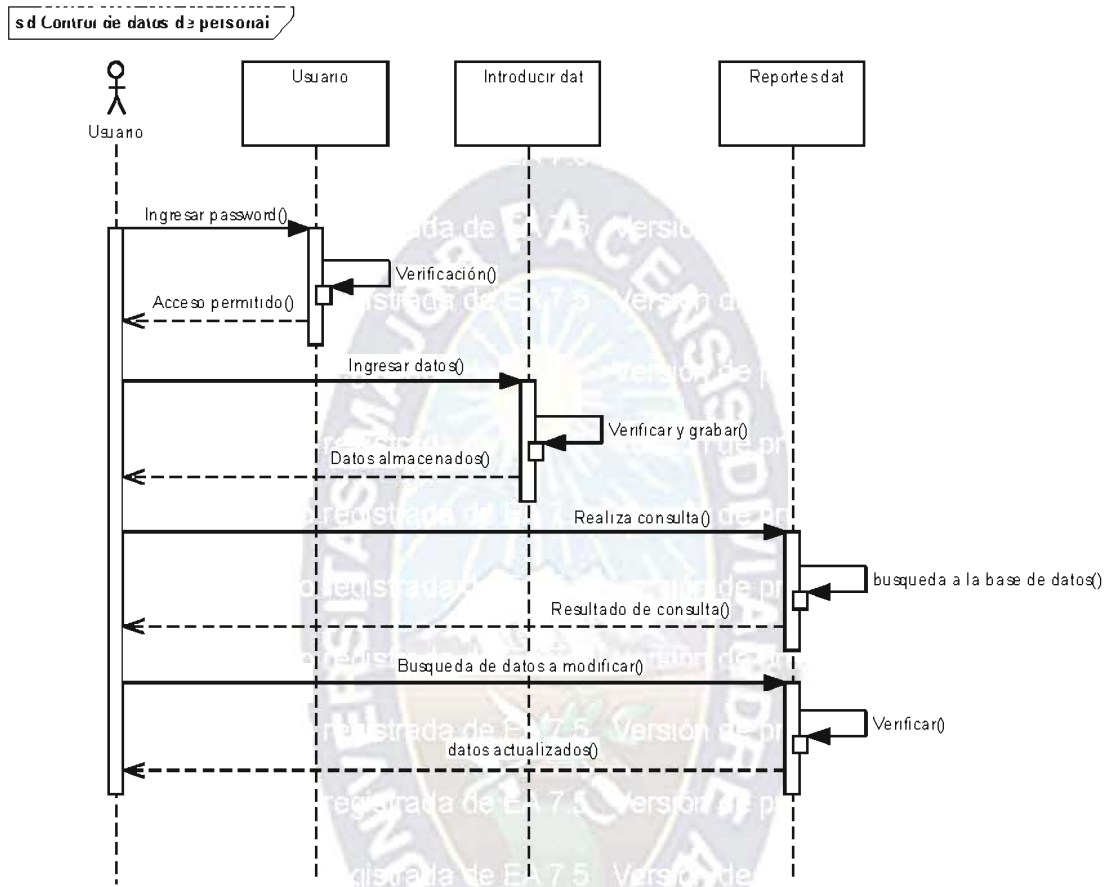


Figura 3.24: Diagrama de Secuencia: Control de datos de Personal
Fuente: Elaboración Propia

3.2.5 Diagrama de Estados

Estos Diagramas son elaborados de acuerdo a los casos de uso esenciales y a los diagramas de secuencia, para saber las transiciones y estados que tienen cada caso de uso.

Para el módulo control de recursos se tienen los siguientes diagramas:

Control de Reposición de Materiales. Los usuarios son Aux. administrativo y jefe administrativo.

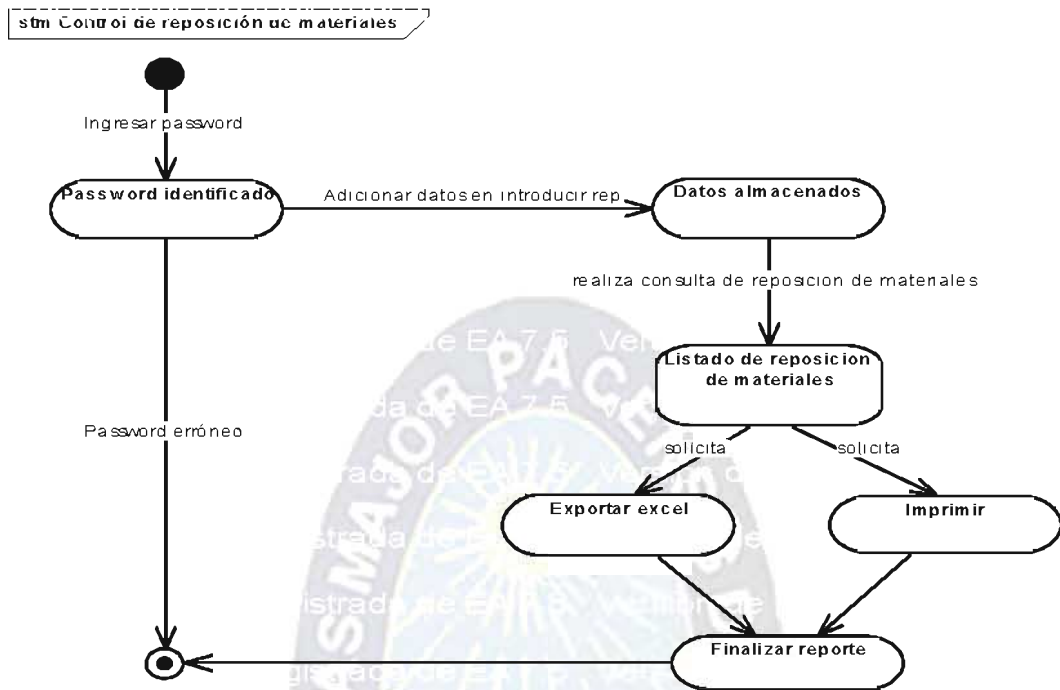


Figura 3.25: Diagrama de Estados: Control de reposición de materiales
Fuente: Elaboración Propia

Control de Mantenimiento. Los usuarios son Aux. administrativo y jefe administrativo.

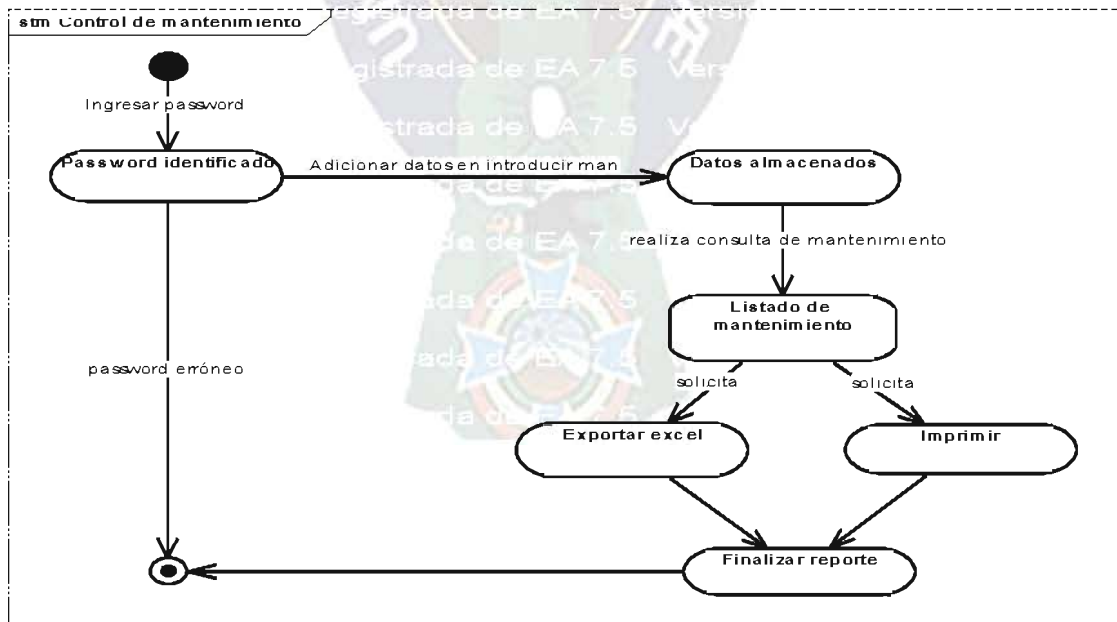


Figura 3.26: Diagrama de Estados: Control de mantenimiento
Fuente: Elaboración Propia

Control de Transporte. Los usuarios son Aux. administrativo y jefe administrativo.

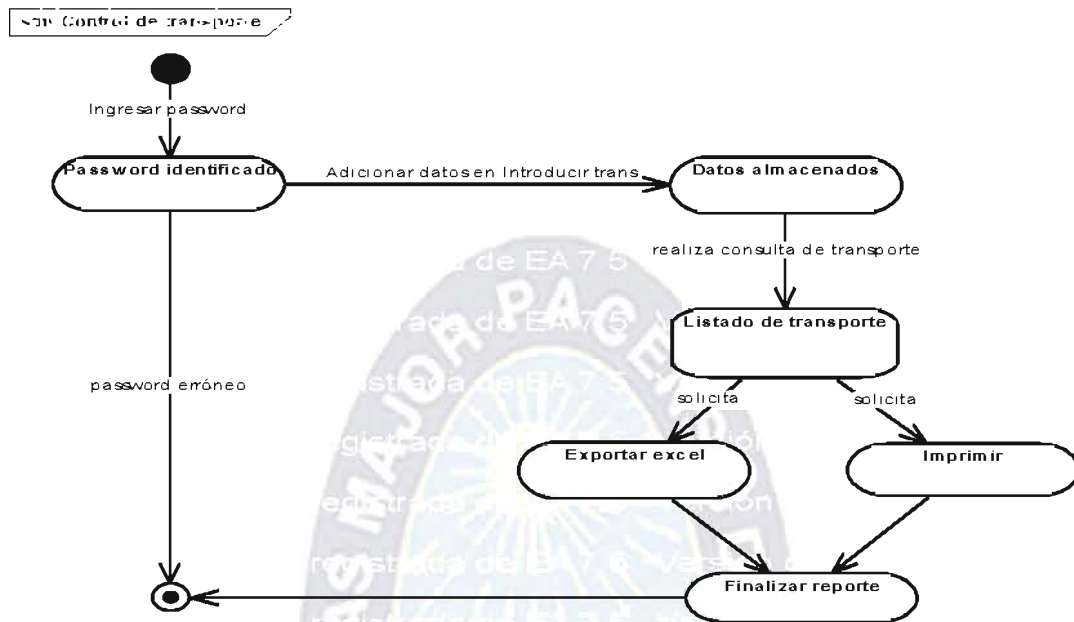


Figura 3.27: Diagrama de Estados: Control de transporte
Fuente: Elaboración Propia

Control de Descarga de Transporte. Los usuarios son Aux. administrativo y jefe administrativo.

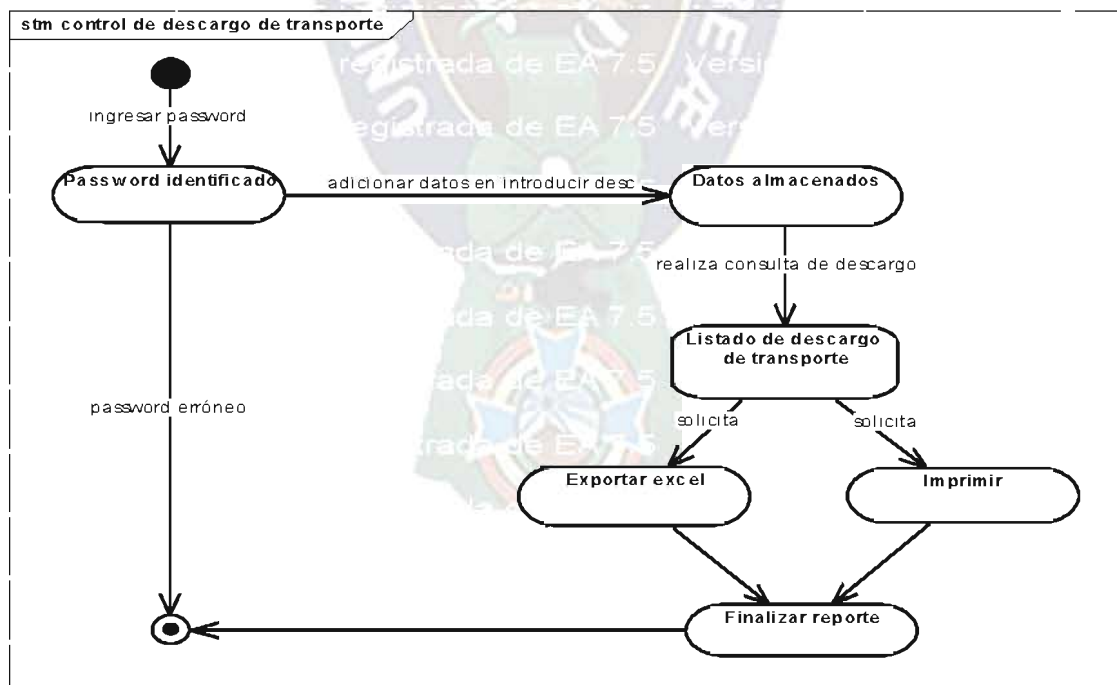


Figura 3.28: Diagrama de Estados: Control de descarga de transporte
Fuente: Elaboración Propia

Para el módulo control de recursos se tienen los siguientes diagramas:

Control de Permisos. Los usuarios son Aux. administrativo y jefe administrativo.

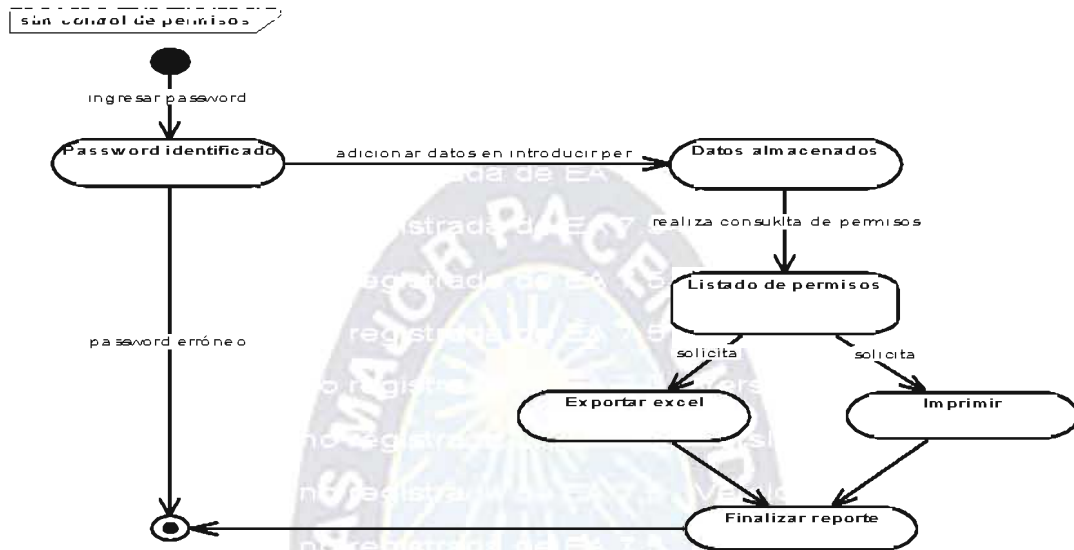


Figura 3.29: Diagrama de Estados: Control de permisos
Fuente: Elaboración Propia

Control de Cambio Voluntario de Turno. Los usuarios son Aux. administrativo y jefe administrativo.

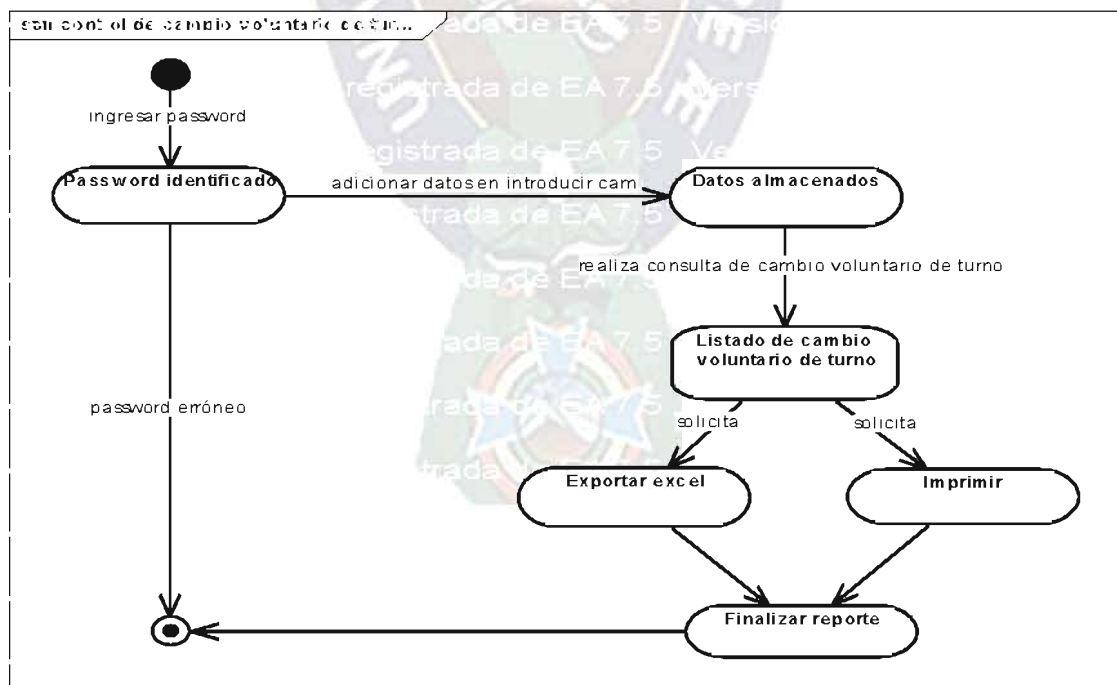


Figura 3.30: Diagrama de Estados: Control de cambio voluntario de turno
Fuente: Elaboración Propia

Control de Bajas. Los usuarios son Aux. administrativo y jefe administrativo.

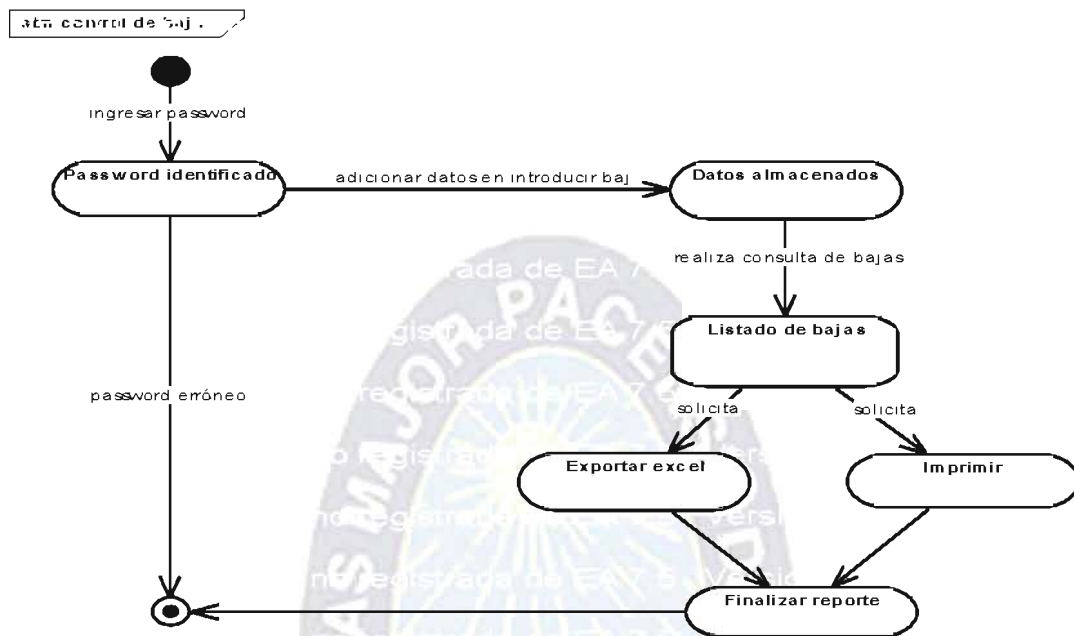


Figura 3.31: Diagrama de Estados: Control de bajas
Fuente: Elaboración Propia

Control de Comisiones. Los usuarios son Aux. administrativo y jefe administrativo.

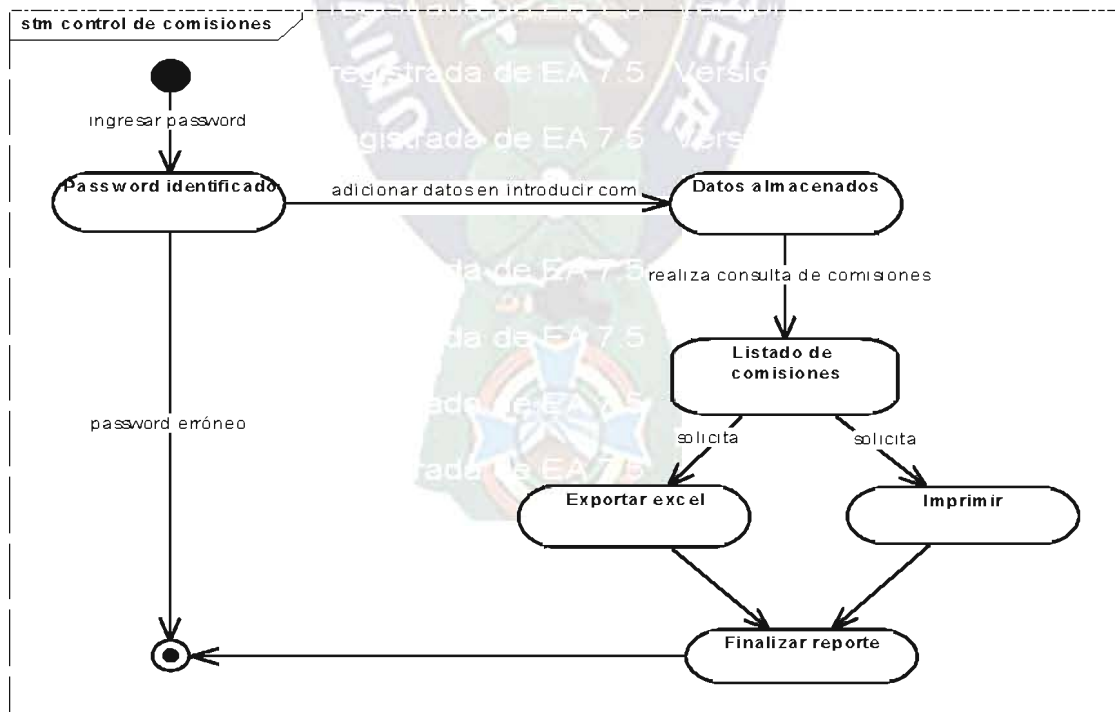


Figura 3.32: Diagrama de Estados: Control de comisiones
Fuente: Elaboración Propia

Control de Vacaciones. Los usuarios son Aux. administrativo y jefe administrativo.

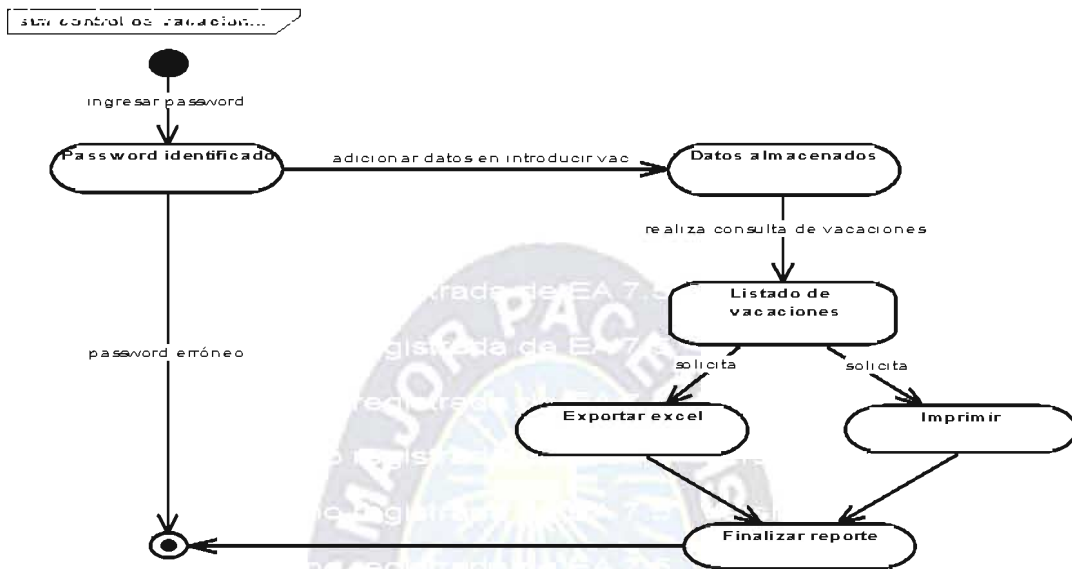


Figura 3.33: Diagrama de Estados: Control de vacaciones
Fuente: Elaboración Propia

Control de datos de Personal. El usuario es el jefe administrativo.

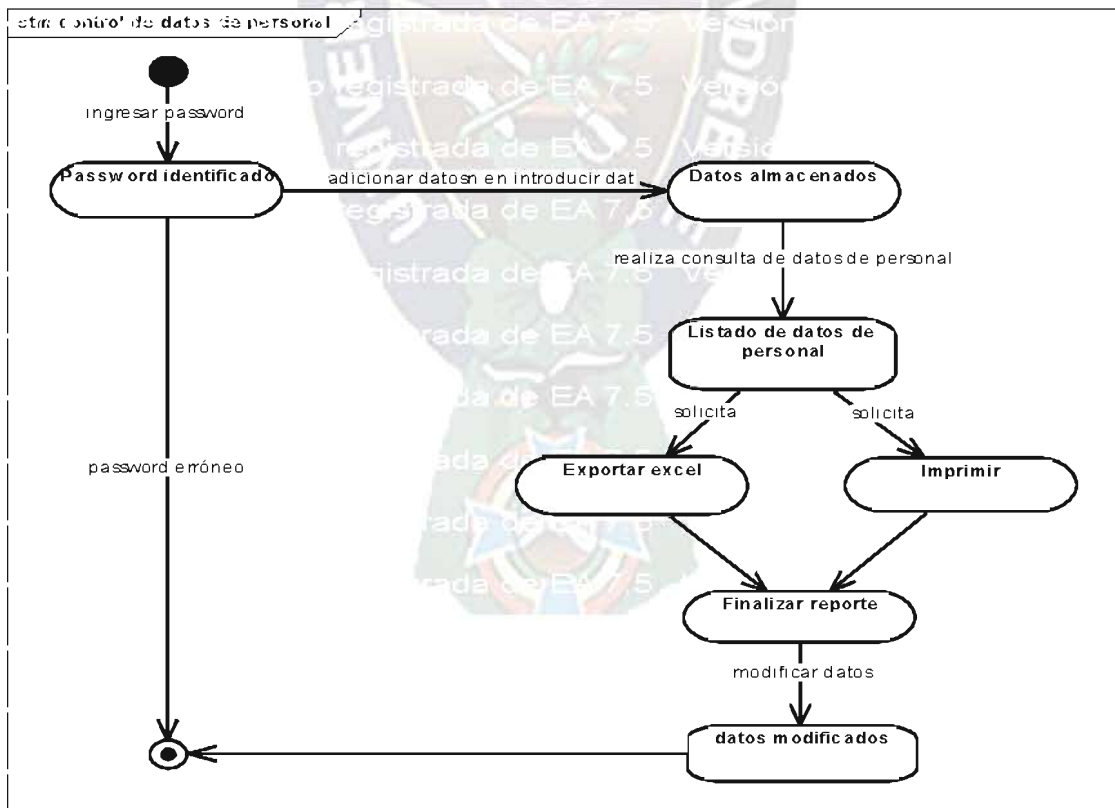


Figura 3.34: Diagrama de Estados: Control de datos de personal
Fuente: Elaboración Propia

3.2.6 Diagrama de Clases

El diagrama de clases describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones que existen. En la Figura 3.35 siguiente se muestra el diseño de las clases correspondiente al presente proyecto.

class otro

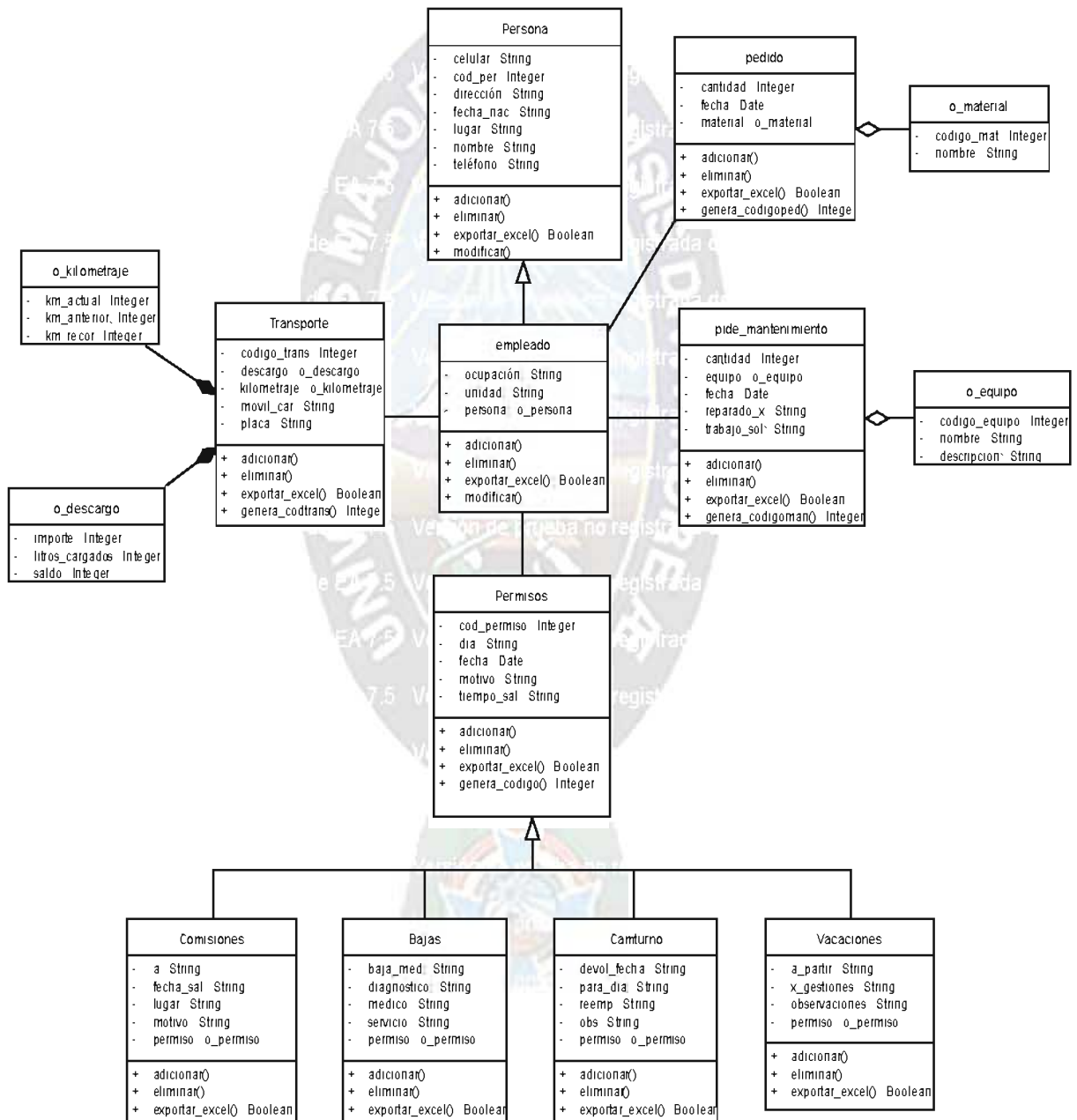


Figura 3.35: Diagrama de Clases
Fuente: Elaboración Propia

3.3 FASE DE CONSTRUCCIÓN

En esta fase se explica el diseño e implementación del proyecto tomando en cuenta la especificación de los componentes, de los nodos y de la interfaz. Para esto se describe el diagrama de componentes.

3.3.1 Diagrama de Componentes

Este diagrama muestra sus principales componentes y sus interfaces para la comunicación del sistema, como también la interacción de los usuarios con estos. En la Figura 3.36 se muestra el diagrama de componentes.

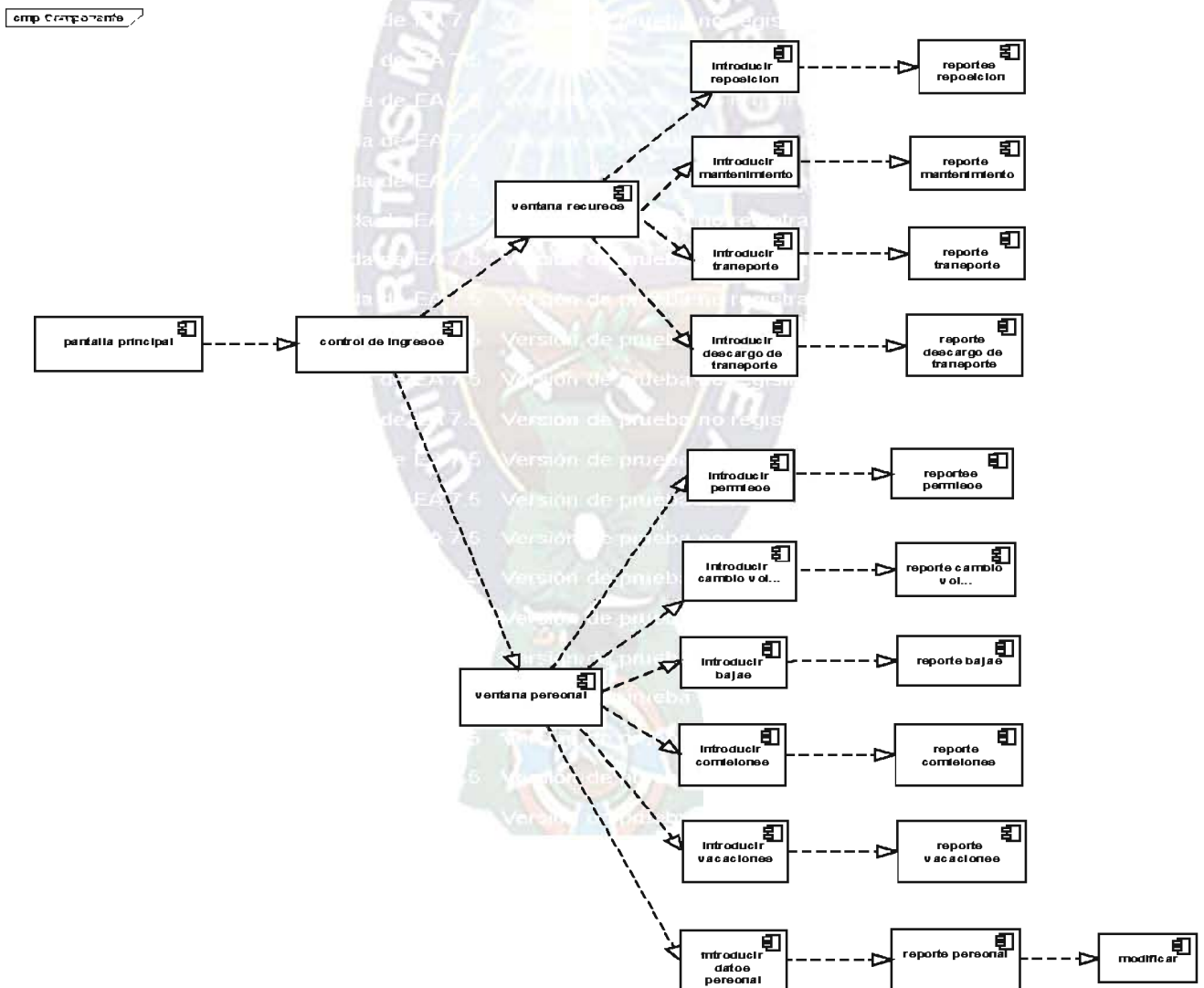


Figura 3.36: Diagrama de Componentes
Fuente: Elaboración Propia

3.3.2 Implementación de la Base de Datos

En este punto se explica la implementación de la base de datos objeto relacional tomado como base el diagrama de clases desarrollado en la fase de elaboración.

En forma inicial se realiza la creación de los tipos objetos:

```
o_kilometraje = class
private
public
km_actual :integer;
km_aterior:integer;
km_rer:integer;
```

La creación de las tablas tipo objetos se muestra en la figura 3.37 :

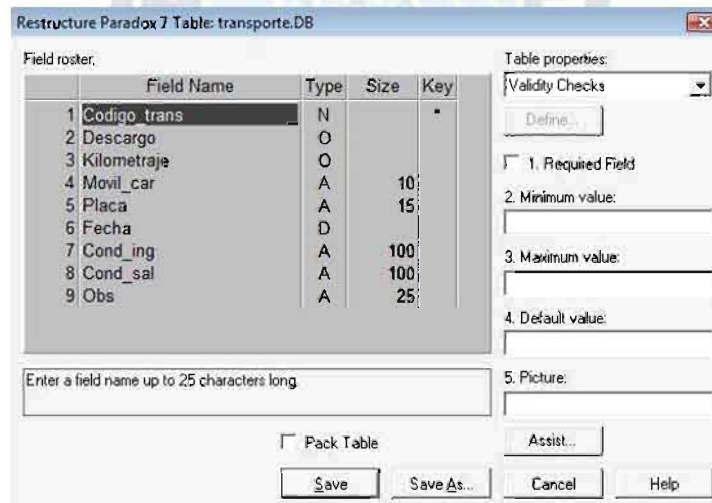


Figura 3.37: Tabla objeto
Fuente: Elaboración Propia

Donde los campos descargo y kilometraje son de tipo objeto.

Para trabajar con la inserción de datos se presenta el siguiente código:

```
o_kilometraje.create();
o_kilometraje.GetObjectContext(km_anterior:edit45.text,km_actual:edit44.text,km_recor:edit44.text);
```

La manera de realizar consultas en las bases de datos objeto relacionales es la siguiente:

```
Select transporte.codigo_trans, transporte.fecha,
```

```
transporte.GetPChar(o_descargoASObject), transporte.GetPChar(o_kilometraje
```

```
ASObject),transporte.movil_car,transporte.placa,transporte.cond_ing,transporte.cond
```

```
_sal,transporte.obs from \ccps\db\transporte
```

3.3.3 Interfaz de Usuario

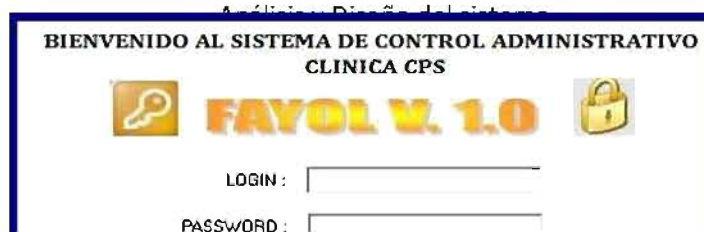
Para el sistema tenemos la siguiente secuencia de pantallas básicas.

En la Figura 3.38 se muestra la pantalla de inicio del sistema donde se encuentran el menú principal.



Figura 3.38: Pantalla Principal
Fuente: Elaboración Propia

En la Figura 3.39 se muestra la ventana de bienvenida, que pide al usuario *login* y *password* para ingresar, las distintas opciones del sistema se habilitan o deshabilitan



dependiendo del nivel de accesibilidad que tenga cada usuario, esto quiere decir que algunas opciones pueden no estar disponibles para algunos usuarios.

Figura 3.39: Control de ingresos
Fuente: Elaboración Propia

En la Figura 3.40 se presenta la pantalla del módulo control de recursos.



Figura 3.40: Pantalla control de recursos
Fuente: Elaboración Propia

En la Figura 3.41 Se muestra la ventana de Introducir reposición de materiales, donde se ingresa los datos de reposición.

INTRODUCIR REPOSICION

DATOS DE REPOSICION

CODIGO FORMULARIO: FECHA: 29/06/2009

ENCARGADO

NOMBRE:

UNIDAD:

COSTOS

TIPO DE PRODUCTO: MATERIALES

DESCRIPCION DE PRODUCTO:

CANTIDAD DE PRODUCTO:

GUARDAR CANCELAR

REPOSICIONES DE UNIDAD

cod_int	codigo	fecha	unidad	cod_enc	desc_prod
5	345	27/05/2009	COSTOS	1	SELLO
6	456	14/06/2009	SECRETARIA	5	SELLO

Figura 3.41: Ventana Introducir Reposición
Fuente: Elaboración Propia

En la figura 3.42 se muestra la ventana de reportes de reposición de materiales, donde se introducen los criterios de búsqueda.

REPORTES REPOSICIÓN

CRITERIOS DE BÚSQUEDA

Fecha Ini: Fecha Fin:

Tipo de productor:

Producto:

Unidad:

TODAS LAS FECHAS

Encargado:

Listar por:

TODOS LOS CAMPOS ENCARGADO, PRODUCTO, CANTIDAD

CÓDIGO, FECHA, PRODUCTO FECHA, ENCARGADO, PRODUCTO

fecha	unidad	nombre
27/05/2009	COSTOS	ROSMERY FORONDA
14/06/2009	SECRETARIA	ARIEL VELA

Figura 3.42: Ventana Reportes Reposición
Fuente: Elaboración Propia

En la figura 3.43 se presenta la ventana imprimir reportes de reposición de materiales, donde se detalla las fechas del reporte.

Print Preview

Control Administrativo Clínica C.P.S.

REPORTE DE REPOSICIÓN DE MATERIALES

Fecha de emisión: TODAS LAS FECHAS

Nº	Código	Fecha	Encargado	Unidad	Descripción de producto	Cant
----	--------	-------	-----------	--------	-------------------------	------

Figura 3.43: Ventana imprimir
Fuente: Elaboración Propia

En la figura 3.44 se muestra una pequeña ventana de exportar a Excel, donde se introduce el nombre con el que se guardará el archivo.

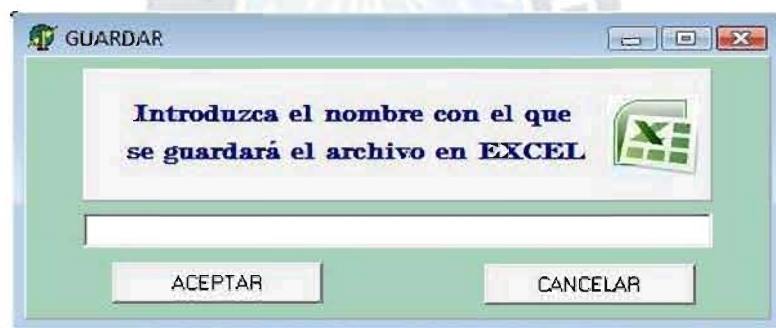
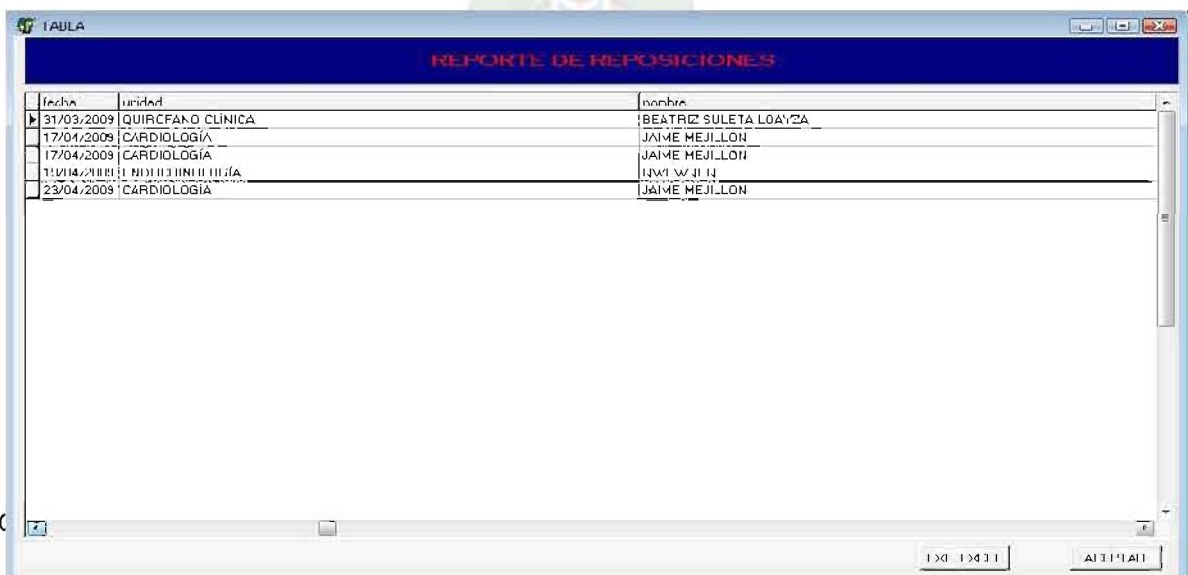


Figura 3.44: Ventana exportar excel
Fuente: Elaboración Propia

En la figura 3.45 se muestra la ventana maximizar tabla que permite tener un listado de una manera más amplia.



Fecha	Unidad	Nombre
31/03/2009	QUIRUFANO CLINICA	BEATRIZ SULETA LOAYZA
17/04/2009	CARDIOLOGÍA	JAIME MEJILLON
17/04/2009	CARDIOLOGÍA	JAIME MEJILLON
17/04/2009	CARDIOLOGÍA	JAIME MEJILLON
23/04/2009	CARDIOLOGÍA	JAIME MEJILLON

Figura 3.45: Ventana maximizar tabla
Fuente: Elaboración Propia

En la figura 3.46 se presenta la ventana eliminar registro.



Figura 3.46: Ventana eliminar registro
Fuente: Elaboración Propia

En la figura 3.47 se muestra la pantalla del módulo control de personal.



Figura 3.47: Ventana control de personal
Fuente: Elaboración Propia

3.4 Fase de Transición

Se ha procedido a la instalación del producto software en la institución, realizando además un curso de capacitación de 2 semanas para que los usuarios del sistema se habitúen con este.



CALIDAD DE SOFTWARE

4.1 INTRODUCCIÓN

Cuando se habla de calidad de software esto quiere decir que miramos el valor técnico del producto, medido normalmente con expresiones matemáticas que incluyen defectos, las fallas y el tiempo, los mismos que permiten evaluar la calidad del producto en este caso software. En este capítulo se desarrolla algunas métricas para tener una visión de la calidad del sistema obtenido en el presente proyecto de grado.

4.2 FUNCIONALIDAD

La funcionalidad de un sistema se mide según la complejidad del mismo, cuanto más complejo sea un sistema, es menos funcional y viceversa. Estas consideraciones se toman desde el punto de vista del usuario.

La funcionalidad de un sistema no puede ser medido directamente, entonces corresponde derivar, mediante otras medidas directas como es el Punto Función, para esto se tiene la siguiente relación:

$$PF = \text{Cuenta_Total} \times ([0,65+0,01 \times \Sigma[Fi]])$$

Donde:

PF: Medida de funcionalidad entregada a la aplicación como valor de normalización.

Cuenta_Total: Es la suma de todas las entradas obtenidas.

([0,65+0,01 x Σ[Fi]): Ajuste a la complejidad según las medidas contables (directas) del dominio de información del software, en donde Σ[Fi] (i = 1 a 14) son los valores de ajuste de la complejidad.

En la tabla 4.1 se muestra las características del dominio de información del sistema

Tabla 4.1: Parámetro de medición
Fuente: Elaboración Propia

Parámetros de medición	
Entrada de Usuario	<ol style="list-style-type: none"> 1. Pantalla de contraseña de ingreso 2. Pantalla control de recursos 3. Pantalla para registrar datos de reposición de materiales 4. Pantalla para registrar datos de mantenimiento de equipos 5. Pantalla control de personal 6. Registro de permisos 7. Registro de cambio voluntario de turno 8. Registro de bajas 9. Registro de comisiones 10. Registro de vacaciones 11. Pantalla de adición a personal 12. Pantalla de control de insumos de transporte 13. Pantalla de control de usuarios 14. Pantalla Acerca de.. <div style="text-align: right; border: 1px solid black; padding: 2px; display: inline-block;">Total 14</div>
Salida de Usuario	<p>22 pantallas de reporte 14 pantallas de confirmación</p> <div style="text-align: right; border: 1px solid black; padding: 2px; display: inline-block;">Total 34</div>
Peticiones de Usuario	<ol style="list-style-type: none"> 1. Consulta de reposición de materiales 2. Consulta de mantenimiento de equipos 3. Consulta de insumos de transporte 4. Listado de personal 5. Consulta de permisos 6. Consulta de bajas 7. Consulta de cambios voluntarios de turno 8. Consulta de comisiones 9. Consulta de vacaciones

	10. Consulta de descargo de transporte 11. Consulta de usuarios <div style="text-align: right; border: 1px solid black; padding: 2px;">Total 11</div>
Número de Archivos	Archivos de la base de datos <div style="text-align: right; border: 1px solid black; padding: 2px;">Total 12</div>
Interfaces Externas	Archivos de datos en cintas o discos <div style="text-align: right; border: 1px solid black; padding: 2px;">Total 2</div>

Los puntos de función se calculan contemplando la tabla 4.2 se determinan cinco características de dominios de información (parámetros de medición) y se proporcionan las cuentas en la posición apropiada de la tabla, a la cuenta se asocia un valor de complejidad.

Tabla 4.2: Cálculo de Conteo Total
Fuente: Elaboración Propia

Factor de ponderación							
Parámetros de medición	Cuenta		Simple	Medio	Complejo		Total
Número de entradas de usuario	14	*	3	4	6	=	56
Número de salidas de usuario	34	*	4	5	7	=	175
Número de peticiones de usuario	11	*	3	4	6	=	66
Número de archivos	12	*	7	10	15	=	120
Numero de interfaces externas	2	*	5	7	10	=	10
Cuenta Total	→						427

Este resultado se asocia con el valor de complejidad a cada cuenta. Las organizaciones que utilizan métodos de punto función desarrollan criterios para

determinar si una entrada es denominada simple, media o compleja. No obstante la determinación de la complejidad es algo subjetivo.

Los valores de ajuste de la complejidad se calculan en base a las siguientes preguntas que se muestran en la tabla 4.3.

Tabla 4.3: Ajuste de complejidad del punto de función
Fuente: Elaboración Propia

Nro.	FACTOR	Valor
1.	¿Requiere el sistema de copias de seguridad y de recuperación fiables?	3
2.	¿Se requiere de comunicación de datos?	5
3.	¿Existen funciones de procesamiento distribuido?	1
4.	¿Es crítico el rendimiento?	2
5.	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	5
6.	¿Requiere el sistema entrada de datos interactiva?	5
7.	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	3
8.	¿Se actualiza los archivos maestros de forma interactiva?	4
9.	¿Son complejas las entradas, las salidas, los archivos ó las peticiones?	4
10.	¿Es complejo el procesamiento interno?	3
11.	¿Se ha diseñado el código para ser reutilizable?	4
12.	¿Están incluidas en el diseño la conversión y la instalación?	5
13.	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	2
14.	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
	$\Sigma[F_i] = \text{Total}$	51

Tabla 4.4: Escala de ajuste
Fuente: Elaboración Propia

DESCRIPCIÓN	ESCALA
No influencia	0
Incidencia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Se debe tomar en cuenta que los puntos de características y los puntos de función representan lo mismo. “Funcionalidad o utilidad” en forma de software.

Para calcular los puntos de función (PF), se utiliza la siguiente relación:

$$PF = \text{Cuenta_Total} \times ([0,65+0,01 \times \Sigma[F_i]])$$

$$PF = 427 * [0.65 + 0.01 * 51]$$

$$PF = 427 * 1.16$$

$$PF = 495.32$$

Si asignamos como valor máximo (5) a cada una de las preguntas, entonces $\Sigma[F_i]$ varia en el intervalo de [0 - 70] y $([0,65+0,01 \times \Sigma[F_i]])$ varia en el intervalo de [0.65 – 1.35]. Relacionando el valor obtenido para el sistema y su valor máximo, que puede alcanzar PF, obtendremos la siguiente relación:

$$PF \text{ Mximo} = 427 * [0.65 + 0.01 * 7]$$

$$PF \text{ Mximo} = 427 * 1.35$$

$$PF \text{ Mximo} = 576.45$$

$$FUNCIONALIDAD = (PF / PF \text{ Mximo}) * 100$$

$$FUNCIONALIDAD = \frac{496.32}{576.45}$$

FUNCIONALIDAD = 85.92

La funcionalidad del sistema es de 86 % tomando en cuenta el punto de función máximo.

4.3 FACILIDAD DE MANTENIMIENTO

La mantenibilidad del sistema está asociada a la detección y corrección de fallas, como también a los cambios que son requeridos por el usuario. El tiempo de vida es indefinido, pero está sujeto a las necesidades del usuario, renovación, expansión, que desee realizar posteriormente.

La métrica que se usa para medir la mantenibilidad es el índice de madurez del software (IMS) el cual proporciona un indicador de la estabilidad del sistema. Se considera lo siguiente:

M_T = Número de módulos en la versión actual

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

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

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

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = \frac{M_T - (F_a + F_c + F_d)}{M_T}$$

A medida que el IMS se aproxime al valor de 1, el producto se empieza a estabilizar.

Para el sistema de Sistema de Control de Administrativo Clínica Caja Petrolera de Salud se han realizado unas pruebas a los prototipos, que fueron sujetos a cambios y modificaciones en sus diferentes versiones, esta relación se puede observar de la siguiente manera:

Tabla 4.5: Relación de cambios y modificaciones
Fuente: Elaboración Propia

Versión del sistema	MT	Fc	Fa	Fd	IMS
VER 1.0	8	2	0	0	$IMS = \frac{8 - (0 + 2 + 0)}{8}$ IMS= 0.75

Se observa que a medida que el software está siendo probado, se identifican las diferencias y sus correcciones, alcanzando así una estabilidad deseable.

4.4 PORTABILIDAD

Un sistema se considera portable, si el costo de transportar y adaptar un ambiente es menor al costo de rediseñar el sistema para el mismo ambiente.

La portabilidad en el caso del actual sistema presenta las siguientes características:

Hardware:

- Microprocesador PENTIUM III o superior.
- Tarjeta de Video Con resolución de 800 x 600 pixeles o superior y capacidad de 16 bit o superior de manejo de colores.
- 70Mb. de espacio de disco duro en la unidad C: del sistema.
- 64 Mb de memoria RAM para Windows 95,98 / 128Mb para XP / 1Gb para Windows VISTA.

Software: El código puede ser ejecutado en la plataforma LINUX previa compilación en Kylix (versión para Linux de Delphi).

4.5 FACILIDAD DE USO

Es un factor que se mide indirectamente y esta relacionado con el intento de cuantificar lo amigable que puede ser el software con el usuario, para ello se hizo un cuestionario que podemos ver en la tabla 4.6 para facilitar la evaluación de la facilidad de uso del sistema, basándose en una escala de evaluación.

Tabla 4.6: Evaluación de facilidad de uso
Fuente: Elaboración Propia

Nro.	FACTOR	Valor
1.	¿Se ha satisfecho todos los requerimientos establecidos por la empresa?	5
2.	¿Es sencillo acceder a los datos de recursos y personal?	4
3.	¿Presenta la suficiente ayuda durante el tiempo el tiempo que accede al sistema?	5
4.	¿Los reportes son suficientemente representativos?	4
5.	¿El sistema tiene la seguridad necesaria?	4
6.	¿Esta de acuerdo con el funcionamiento del sistema?	4
7.	¿El sistema facilita el trabajo que realiza?	4
	Total	30

Tabla 4.7: Escala de evaluación
Fuente: Elaboración Propia

DESCRIPCIÓN	ESCALA
Pésimo	0
Malo	1
Regular	2
Bueno	3
Muy bueno	4

En base al cuestionario, se calcula la facilidad mediante la siguiente fórmula :

$$\text{Facilidad de uso} = [(\sum \text{Valor} / n) * 100] / 5$$

Entonces:

$$\text{Facilidad de uso} = [(30 / 7) * 100] / 5$$

$$\text{Facilidad de uso} = 85 \%$$

Por lo tanto la facilidad de uso es buena.



CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

A la culminación del proyecto de grado se ha concluido que la utilización del Sistema de Control Administrativo implementado en la Clínica de la Caja Petrolera de Salud contribuye y facilita los procesos de control al departamento de administración de la Clínica por los siguientes aspectos:

- La implementación del Sistema de Control Clínica Caja Petrolera de Salud (CPS) logra integrar los módulos de Control de recursos y Control de personal que mejora el manejo de la información generada por la institución.
- Se reducen los errores producidos al manejar la información de forma manual en el control de recursos, con respecto a reposición de materiales, mantenimiento de equipos y control de insumos de transporte.
- Debido al diseño, el sistema trabaja tranquilamente en una red LAN dando la facilidad a la institución en el manejo de la información integrada y centralizada.
- Los reportes generados por el sistema son un apoyo para la toma de decisiones de la parte ejecutiva de la institución.
- La posibilidad de las Bases de Datos Objeto Relacionales de implementar diferentes tipos de datos y relaciones complejas, permitió modelar e implementar la base de datos de un modo más natural y directo.

- Las metodologías de análisis y diseño de sistemas apropiadas, pueden ser mejor aprovechadas cuando son utilizadas en conjunto, ya que cada una cuenta con herramientas que permiten identificar requerimientos, analizar el problema y diseñar la solución, de acuerdo al conocimiento a priori del desarrollador. Aspecto que se muestra en este trabajo y además que puede servir de guía para el desarrollo e implementación de futuros proyectos.

5.2 RECOMENDACIONES

- El sistema toma en cuenta los módulos de control de recursos y control de personal de la empresa pero no el área contable, se recomienda diseñar el módulo contable faltante.
- Debido a las características del sistema se recomienda llevar a cabo una capacitación pormenorizada, con respecto a la manipulación del sistema al personal encargado para lograr un correcto funcionamiento del mismo.

REFERENCIAS BIBLIOGRÁFICAS

[JGB, 2000]

IVAR JACOBSON, GRADY BOOCH, JAMES RUMBAUGH (Año 2000): El Proceso Unificado de desarrollo de software. 1ra. Edición en español: Pearson Education S.

[JAC, 2000]

JACOBSON, I. (Año 2000): El Lenguaje Unificado de Modelado, España: Edit. Addison Wesley-Grady Booch, James Rumbaugh, Ivar Jacobson.

[RAN, 2000]

RANDALL, A. (2000): Proyecto monográfico – Análisis y diseño de sistemas con Lenguaje de Modelaje Unificado (UML).

[LAR, 1999]

LARMAN, C. (1999): UML y Patrones: Introducción al Diseño Orientado a Objetos, México: Primera Edición Edic. Prentice hall.

[PRESS, 05]

PRESSMAN, R. (2005): Ingeniería de Software. España: 6ta edición. McGraw- Hill

[WEB, 01]

Introducción al Proceso Unificado de Desarrollo (RUP) – Patricio Letelier Torres, Universidad Politécnica de Valencia, España

<http://www.dsic.upv.es/~letelier/pub/p16.ppt>

[WEB, 02]

El Proceso Unificado de Desarrollo de Software- A.U.S. Gustavo Torossi – 2004

<http://www.chaco.gov.ar/UTN/disenodesistemas/apuntes/oo/ApunteRUP.pdf>

[WEB, 03]

Estructura Básica del Proceso Unificado de Desarrollo de Software – Robin Alberto Castro Gil – Universidad Icesi

http://www.icesi.edu.co/es/publicaciones/publicaciones/contenidos/sistemas_telematica/3/rcastro_estructura-bas-puds.pdf

[WEB, 04]

UML – Lenguaje de Modelado Unificado

<http://usuarios.lycos.es/oopere/uml.html>

[WEB, 05]

Enterprise Architect Herramienta de diseño UML

<http://www.sparxsystems.com.ar>

[WEB, 06]

Tipos de redes

<http://es.kioskea.net/contents/initiation/lan.php3>

[WEB, 07]

Definiciones LAN

<http://www.masadelante.com/faq-LAN.htm>

[WEB, 08]

Diseño de Base de datos

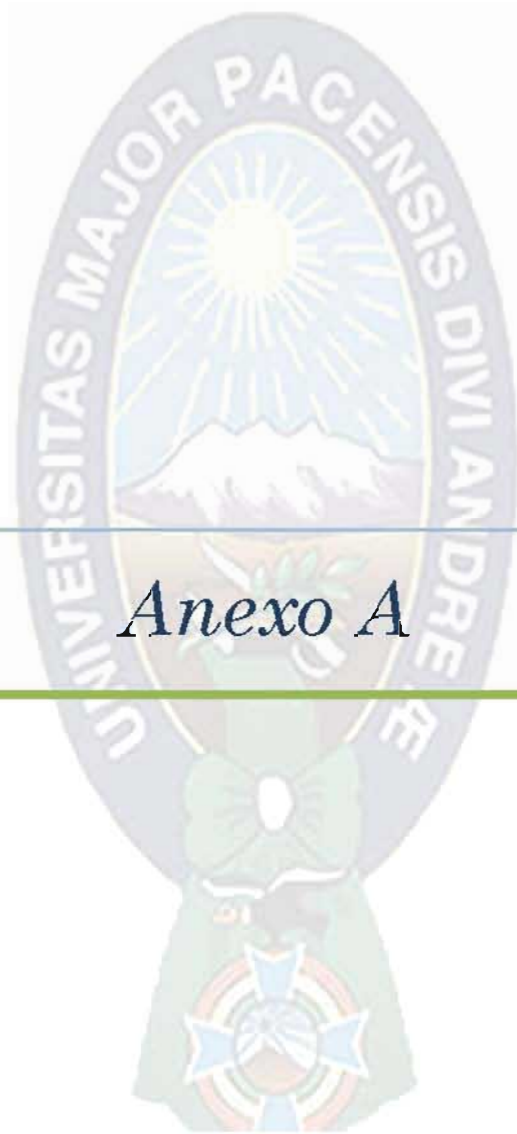
http://utpl.edu.ec/clase_bdor

[WEB, 09]

Ingeniería Inversa

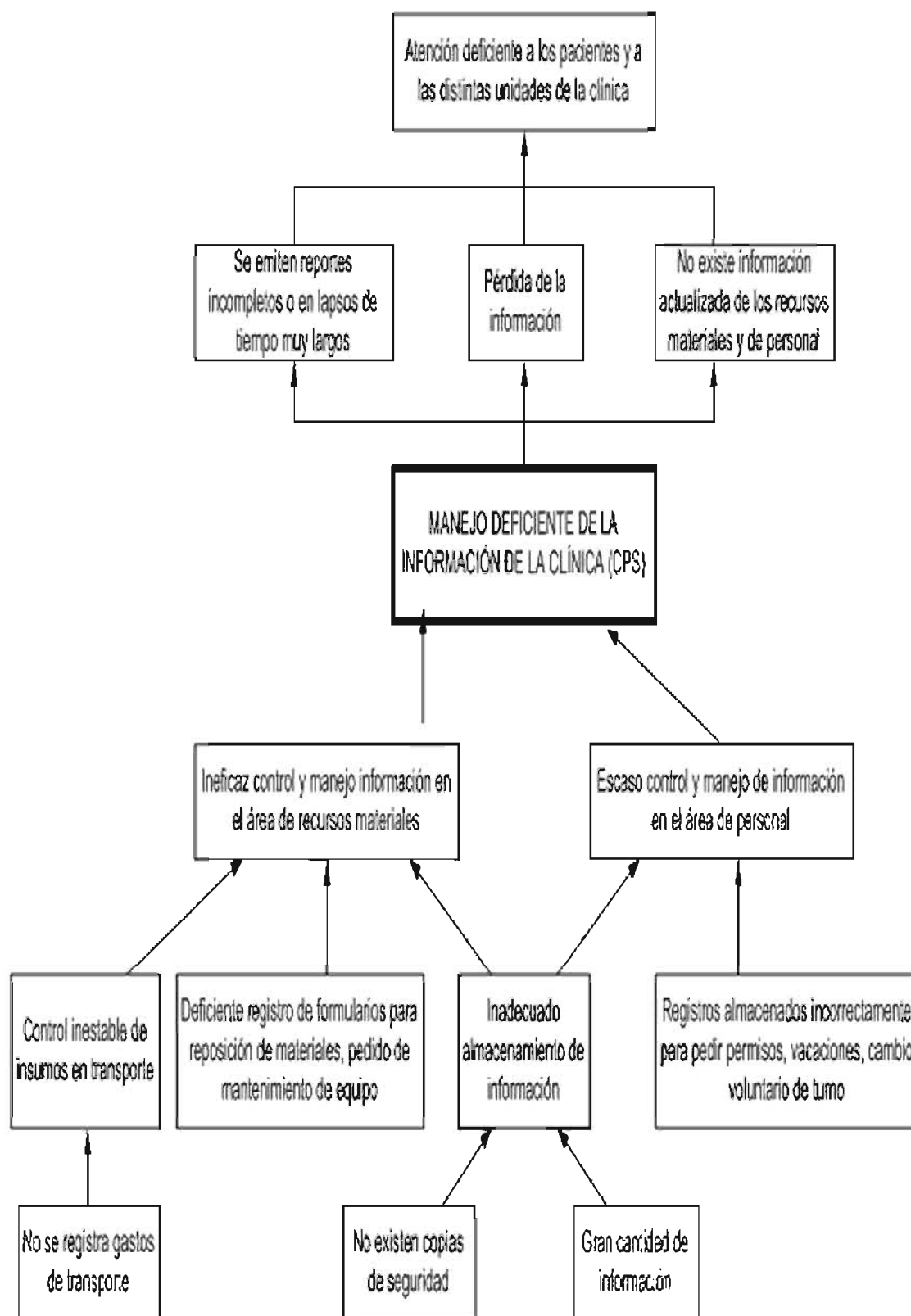
<http://www.acm.uiuc.edu/sigmil/RevEng/>



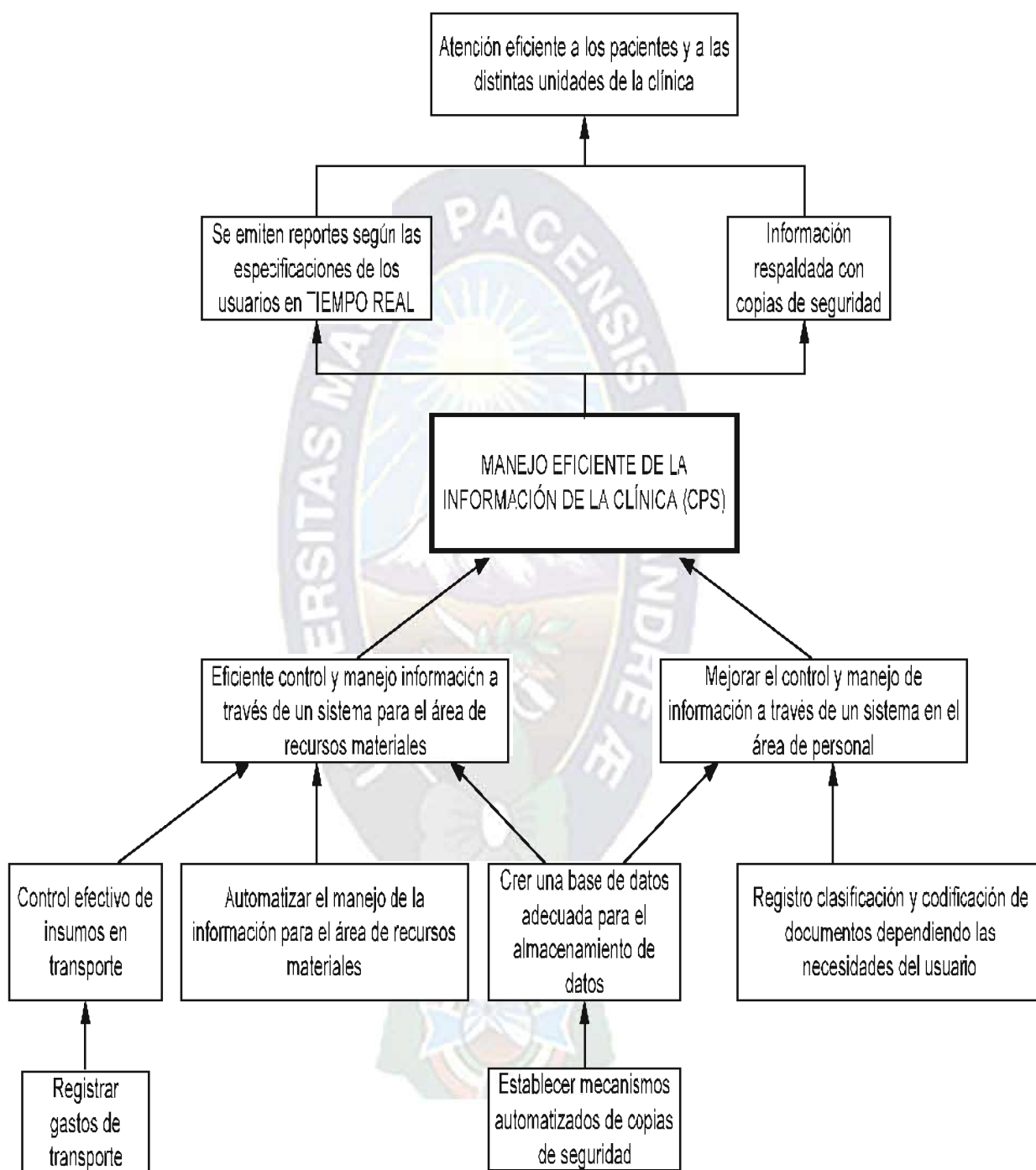


Anexo A

Árbol de Problemas



Árbol de objetivos



Matriz de Marco Lógico

Como producto del estudio anterior se tiene la siguiente Matriz de Marco Lógico

RESUMEN NARRATIVO DE OBJETIVOS	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACION	SUPUESTOS
FIN Manejo efectivo de la información administrativa en la Clínica Caja Petrolera de Salud que permita el seguimiento oportuno de cada una de las actividades que desarrolla la institución, para evitar pérdidas económicas, disconformidad en los distintos departamentos de la institución	Después de la implementación del sistema y su funcionamiento durante una gestión, el control administrativo mejoro en un 70% en comparación a la anterior gestión	Estadísticas realizadas por personal de la clínica	Los funcionarios de la institución consideran a la variable información como recurso que fortalece el desarrollo de la institución
PROPOSITO Manejo eficiente de la información de la institución Clínica Caja Petrolera de Salud	Información administrativa manejada de forma eficiente, creando reportes adecuados para los distintos requerimientos del usuario en tiempo real	Reportes impresos para el control de los distintos módulos	El personal maneja de forma optima el sistema de información implantado en la institución
COMPONENTES ❶ Control automatizado de permisos, bajas, compensaciones y comisiones del personal en la institución ❷ Control automatizado de mantenimiento, reposición de materiales de los recursos y transporte de la institución	Módulo de control instalado en los equipos pertinentes en el departamento de control administrativo Módulo instalado	Observación directa efectuada por encargados de evaluación del proyecto	Todos los funcionarios que intervienen en el proceso de manejo de información de forma directa están dispuestos a capacitarse para el manejo de los módulos correspondientes a cada funcionario
ACTIVIDADES ❶ 1.1 Recolección de información acerca del control de personal de la institución 1.4 Diseño y desarrollo del modulo CONTROL DE PERSONAL 1.5 Pruebas del modulo CONTROL DE PERSONAL 1.6 Capacitación al personal sobre el manejo de este modulo ❷ 2.1 Recolección de información acerca del control de recursos en la institución 2.2 Diseño y Desarrollo del modulo CONTROL DE RECURSOS 2.3 Pruebas del modulo CONTROL DE RECURSOS 2.4 Capacitación al personal sobre el manejo de este modulo	Control del personal } Presupuesto 1600 - (Bolivianos) Personal capacitado, Control de recursos } Presupuesto 1200 - (Bolivianos) Personal capacitado	Manual de funciones Manual de funciones	Personal dispuesto a proporcionar la información pertinente en cada uno de los departamentos de la institución Personal dispuesto a proporcionar la información pertinente en cada uno de los departamentos de la institución