

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



**PROYECTO DE GRADO**

**CONTROL FINANCIERO VÍA WEB PARA EL  
SERVICIO EXTERIOR – MINISTERIO DE RELACIONES  
EXTERIORES Y CULTOS**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENSIÓN : INGENIERIA DE SISTEMAS INFORMÁTICOS**

**AUTOR: HERNÁN ELVIO YUJRA CHIPANA**  
**TUTOR: LIC. EFRAÍN SILVA SANCHEZ**  
**REVISORA: LIC. MENFY MORALES RIOS**

**LA PAZ – BOLIVIA**  
**2006**

*Este trabajo está dedicado...*

*A mis padres El Señor Esteban Yujra  
Saire y la Señora Flora Chipana Ramos,  
por que siempre me demostraron su amor,  
cariño y comprensión.*

*El Autor*

---

## *Agradecimientos*

*Deseo expresar mis agradecimientos a aquellas personas que durante todo este tiempo han sido una fuente de apoyo, motivación y aprendizaje*

- ü Agradecer a Dios por darme la oportunidad de vivir.*
- ü Agradecer a mis padres por la paciencia que siempre tuvieron en cada uno de mis días, espero no haberles fallado.*
- ü Agradecer a mis hermanas Beatriz, Virginia y Susy por su comprensión y afecto.*
- ü Agradecer a mi Docente Revisor Lic. Menfy Morales Rios, por toda la paciencia, por su conocimiento y apoyo incondicional.*
- ü Agradecer a mi Docente Tutor Lic. Efraín Silva Sánchez, por su apoyo y sus consejos.*
- ü Agradecer al Ing. Edwin V. Mamani Ch. y el personal de la institución por su colaboración.*
- ü Agradecer a todos mis amigos (as) de la Universidad, que con su apoyo y aliento me enseñaron lo que significa la amistad, muchas gracias a todos (as).*
- ü Agradecer a todos mis amigos que compartieron momentos especiales en los años pasados, siempre los recuerdos pese al tiempo y la distancia.*

*Todos ellos me ayudaron en su momento, y solo me resta decirles ¡Muchas Gracias!*

*El Autor*

---

## RESUMEN

Las tecnologías relacionadas con Internet en los últimos años han avanzado exponencialmente, es imprescindible que las instituciones avancen juntamente con estas para poner en manos del personal las nuevas características ofrecidas por dichas tecnologías. La tendencia actual y futura inmediata nos marca un camino en el cual se enlazan todo tipo de aplicaciones mediante redes, tanto intranets como intranets, y donde el usuario pretende realizar sus gestiones desde el lugar donde se encuentre.

La Unidad de sistemas Informáticos del Ministerio de Relaciones Exteriores y Cultos es la encargada de hacer uso de las nuevas tecnologías existentes en el mercado para colaborar en las tareas que tienen los funcionarios que se encuentran dentro del mismo ministerio como aquellos que se encuentran fuera de las fronteras Bolivianas.

Uno de los aspectos fundamentales a controlar en las instituciones y más aún en la públicas es el manejo de dineros y saber cual el uso de se le dio. Es así que el presente trabajo brinda los beneficios de las tecnologías existen para dicho fin. Adopta como metodología de desarrollo el RUP (Rational Unified Process), cuyos flujos de trabajo fundamentales son: Captura de Requerimientos, Análisis, Diseño, pruebas e Implementación, apoyados en el lenguaje de modelados UML (Unified Modeling Language).

Para la implementación se hace uso de Microsoft Visual Studio el cual nos proporciona varios lenguajes de programación y una IDE de desarrollo adecuado para aplicaciones Web y como gestor de base de datos SQL Server del cual se aprovecha los procedimientos almacenados para separar el acceso a la base de datos.

---

## ÍNDICE GENERAL

	<u>Pag.</u>
<b>Capítulo 1. Marco preliminar.</b>	
1. Introducción.....	1
1.1 Antecedentes.....	3
1.2 Planteamiento del problema.....	5
1.3 Objetivos.....	6
1.3.1 Objetivo General.....	6
1.3.2 Objetivos Específicos.....	6
1.4 Justificación.....	7
1.4.1 Justificación Social.....	7
1.4.2 Justificación Técnica.....	7
1.4.3 Justificación Económica.....	7
1.5 Alcances y Aportes.....	8
1.6 Metodología.....	9
1.7 Técnicas y herramientas.....	10
<b>Capítulo 2. Marco Teórico.</b>	
2.1 El Proceso Unificado (RUP).....	11
2.1.1 Características.....	11
2.1.1.1 Proceso dirigido por casos de uso.....	12
2.1.1.2 Proceso centrado en la arquitectura.....	12
2.1.1.3 Proceso iterativo incremental.....	15
2.1.2 Estructura del proceso.....	18
2.1.2.1 Eje horizontal.....	18
2.1.2.1 Eje vertical.....	19
2.1.3 Estructura dinámica del proceso, fases e iteraciones.....	19
2.1.3.1 Fase de inicio.....	20
2.1.3.2 Elaboración.....	21
2.1.3.3 Construcción.....	21
2.1.3.4 Transición.....	22
2.1.4 Flujo de trabajo fundamentales.....	23
2.1.4.1 Requisitos.....	23
2.1.4.2 Análisis.....	26
2.1.4.2.1 Clases con estereotipos.....	27
2.1.4.2.2 Identificación de clases.....	29
2.1.4.2.3 Diagramas de interacción.....	29
2.1.4.3 Diseño.....	30
2.1.4.3.1 Estrategias de diseño.....	31

2.1.4.3.2 Diagramas de secuencias del diseño.....	34
2.1.4.3.3 Diagramas de clases.....	35
2.1.4.3.4 Identificar operaciones.....	35
2.1.4.4 Implementación.....	36
2.1.4.5 Pruebas.....	37
2.1.5 Correspondencia entre el modelo de objetos y las tablas.....	37
2.1.6 Seguridad.....	39
2.1.7 Arquitectura.....	41
2.1.8 Calidad de la aplicación.....	42
2.1.9 Calidad de la aplicación en el entorno Web.....	44

### **Capítulo 3. Marco Aplicativo.**

3.1 Análisis de requisitos.....	47
3.1.1 Descripción del problema.....	47
3.1.2 Requisitos candidatos.....	48
3.1.3 Modelado del dominio.....	49
3.1.4 Modelado del negocio.....	50
3.1.5 Requisitos adicionales.....	52
3.1.6 Modelo de casos de uso.....	53
3.1.6.1 Actores encontrados.....	55
3.1.6.2 Casos de uso encontrados.....	57
3.2 Análisis.....	65
3.2.1 Diagramas de interacción.....	65
3.3 Diseño.....	71
3.3.1 Diagramas de secuencia.....	71
3.3.2 Diagrama de clases - Control Financiero vía Web.....	77
3.3.3 Diagrama relacional del sistema.....	78
3.4 Implementación.....	79
3.4.1 Modelo Físico de la Base de datos.....	79
3.4.2 Seguridad.....	80
3.4.3 Arquitectura.....	81
3.5 Pruebas.....	82
3.5.1 Factores de Calidad ISO 9126.....	82
3.5.2 Calidad de la aplicación en el entorno Web.....	86

### **Capítulo 4. Conclusiones y Recomendaciones.**

4.1 Conclusiones.....	88
4.2 Recomendaciones.....	89

### **Bibliografía**

### **Anexos**

### **Documentación**

---

## ÍNDICE DE FIGURAS

	<u>Pag.</u>
Figura 2.1 Los Modelos se completan, la arquitectura no cambia drásticamente...	14
Figura 2.2 Iteración con los cinco flujos de trabajo fundamentales.....	16
Figura 2.3 Esfuerzo en actividades según fase del proyecto.....	17
Figura 2.4 Fases e hitos en RUP.....	19
Figura 2.5 Distribución típica de recursos humanos.....	20
Figura 2.6 Clase de Entidad.....	27
Figura 2.7 Clase de Interfaz.....	28
Figura 2.8 Clase de Control.....	28
Figura 2.9 Escala de preferencia.....	46
Figura 3.1 Diagrama de clases del modelado del dominio.....	49
Figura 3.2 Diagrama de casos de uso del modelado del negocio.....	50
Figura 3.3 Representación del modela de caso de uso.....	50
Figura 3.4 Casos de uso del Sistema rendición de cuentas.....	51
Figura 3.5 Casos de uso del Sistema (registro e ingreso de usuarios).....	53
Figura 3.6 Casos de uso del Sistema (transacciones de cuentas).....	54

---

## ÍNDICE DE TABLAS

	<u>Pag.</u>
Tabla 2.1 Distribución típica de esfuerzo y tiempo.....	20
Tabla 2.2 Descripción de actores.....	26
Tabla 2.3 Descripción de casos de uso.....	26
Tabla 2.4 Amenazas más comunes a considerar para aplicaciones de tipo Web...	39
Tabla 2.5 Modelo de amenazas.....	40
Tabla 2.6 Descripción del mecanismo DREAD.....	40
Tabla 2.7 Descripción de las ponderaciones para aplicar DREAD.....	41
Tabla 3.1 Requisitos candidatos.....	48
Tabla 3.2 Descripción del actor Usuario.....	55
Tabla 3.3 Descripción del actor Responsable Contable.....	55
Tabla 3.4 Descripción del actor Jefe de Misión.....	56
Tabla 3.5 Descripción del actor Administrador del Sistema.....	56
Tabla 3.6 Descripción del actor Base de Datos Usuario.....	56
Tabla 3.7 Descripción del actor Director General de Asuntos Administrativos....	56
Tabla 3.8 Descripción del actor Director Administrativo Financiero.....	56
Tabla 3.9 Descripción del actor Control Financiero.....	57
Tabla 3.10 Descripción del caso de uso Solicita registro.....	57
Tabla 3.11 Descripción del caso de uso Confirma registro.....	58
Tabla 3.12 Descripción del caso de uso Registrar usuario.....	58
Tabla 3.13 Descripción del caso de uso Guardar datos.....	58
Tabla 3.14 Descripción del caso de uso Ingresar al sistema.....	59
Tabla 3.15 Descripción del caso de uso Loguea.....	59
Tabla 3.16 Descripción del caso de uso Validar usuario.....	59
Tabla 3.17 Descripción del caso de uso Remite programa trimestral.....	60
Tabla 3.18 Descripción del caso de uso Llena formulario.....	60
Tabla 3.19 Descripción del caso de uso Programación trimestral.....	61
Tabla 3.20 Descripción del caso de uso Desembolsar fondos.....	61
Tabla 3.21 Descripción del caso de uso Analizar pedido.....	62
Tabla 3.22 Descripción del caso de uso Rinde cuentas.....	62
Tabla 3.23 Descripción del caso de uso Prepara informe.....	62
Tabla 3.24 Descripción del caso de uso Comprobante de ingreso.....	63
Tabla 3.25 Descripción del caso de uso Comprobante de egreso.....	63
Tabla 3.26 Descripción del caso de uso Caja chica.....	64
Tabla 3.27 Descripción del caso de uso Planilla de sueldos.....	64

---

## Capítulo 1

### 1. Introducción

El manejo adecuado del dinero en toda institución es fundamental y más aun en las Instituciones Publicas llámese estos Ministerios, Prefecturas, Alcaldías, y otros, Bolivia es un país donde más se escucha denuncias de corrupción, de apropiaciones indebidas de bienes pero principalmente del mal uso y/o abuso de los dineros asignados por el estado Boliviano desembolsados por el TGN<sup>1</sup>, esto se evidencia al hacer una auditoria contable y se establece que las cuentas no son claras.

Toda institución pública para llevar a cabo la misión asignada se circunscribe en un conjunto de Leyes, Decretos, Resoluciones, Reglamentos internos, y otros, los cuales marcan las atribuciones que le competen a cada elemento que forma parte de la institución. El Ministerio de Relaciones Exteriores y Culto como parte del aparato gubernamental según la LOPE<sup>2</sup> tiene como una de las labores el de coordinar y ejecutar la política exterior del Estado, Administrar las relaciones exteriores con los sujetos de Derecho Internacional Público.

El Ministerio es el órgano central del Servicio de Relaciones Exteriores el cual se ejerce mediante el Ministerio del ramo, de las Misiones Diplomáticas, Representaciones Permanentes ante Organismos Internacionales y los

---

<sup>1</sup> Tesoro General de la Nación.

<sup>2</sup> Ley de Organización del Poder Ejecutivo, Ley No. 2446.

---

Consulados a las que denominaremos simplemente Misión<sup>3</sup> a excepción del Ministerio. El Tesoro General de la Nación, asigna fondos por concepto de gastos de funcionamiento y gastos de representación a favor de las Misiones en el exterior, los mismos por disposiciones legales deben rendir cuentas documentadas del uso de dichos fondos a través de procedimientos relacionados con el registro y descargo de cuentas, puesto que las Misiones se encuentran en el exterior del país se hace dificultosa hacer el seguimiento de los fondos destinados, además que se hace imperioso uniformar los informes presentados por las distintas Misiones en el exterior.

En esta perspectiva y apoyados en los puntos anteriores citados, el Ministerio de Relaciones Exteriores y Culto tiene la necesidad de contar con la rendición de cuentas de cada una de las Misiones acreditadas en el exterior de la manera más oportuna posible para llevar a cabo un adecuado seguimiento de los dineros asignados en las partidas<sup>4</sup> correspondientes.

La Unidad de Sistemas Informáticos pretende desarrollar un sistema el cual cumpla con las expectativas del personal del Ministerio el cual considera los puntos que más dificultan para realizar las actividades cotidianas, por ejemplo cada Misión tendrá que rendir cuentas en la moneda local del país donde reside, considerar las diferencias de horarios entre países, y otros. Dentro del sistema se considera la ejecución presupuestaria, la rendición de cuentas, y la caja chica de cada una de las Misiones en el exterior.

Para la rendición de cuentas se tiene los siguientes formularios:

*(Ver anexo C – Formularios)*

- Programación Trimestral para la Ejecución Presupuestaria.

---

<sup>3</sup> En el presente documento, los términos de Embajada, Representación Permanente, Consulado, se denominarán simplemente “Misión”.

<sup>4</sup> Clasificador contable.

- 
- Comprobante de Ingreso.
  - Comprobante de Egreso.
  - Registro Diario de Ingresos y Egresos.
  - Informe Financiero Mensual.
  - Conciliación Bancaria.
  - Informe Financiero Anual.
  - Caja Chica.
  - Registro Diario de Ingresos por Gastos de Representación.
  - Registro Diario de Egresos por Gastos de Representación.
  - Planilla de Sueldos Servicio Exterior.

### **1.1 Antecedentes**

Para un adecuado funcionamiento, el Poder Ejecutivo establece una estructura orgánica de las distintas Instituciones Públicas Nacionales sobre las que tiene tuición la Administración Central, esta organización se puntualiza en la Ley No. 2446 Ley de Organización del Poder Ejecutivo más conocida como LOPE, dicha ley tiene por objeto establecer el número y atribuciones de los Ministros de Estado y otras normas relacionadas con la organización del Poder Ejecutivo.

El Ministerio de Relaciones Exteriores y Culto es el órgano central del Servicio de Relaciones Exteriores, que bajo la conducción constitucional del Presidente de la Republica está encargada de analizar, planificar, coordinar, desarrollar, centralizar, ejecutar y evaluar la Política Exterior del estado Boliviano, además de tener a su cargo los asuntos relativos al culto. Para llevar a cabo estas tareas el Ministerio es una institución que se puede decir descentralizada, pues esta constituida por un conjunto de embajadas, consulados, representaciones en el exterior. El servicio de Relaciones Exteriores está constituido por las Misiones Diplomáticas, Representaciones Permanentes ante Organismos Internacionales, Misiones Especiales y Consulados que se rigen bajo los reglamentos, manuales

---

e instrucciones del Ministerio de Relaciones Exteriores y Culto, uno de los reglamentos y en el que se basa el presente trabajo es el “Reglamento para la rendición de cuentas en el servicio exterior”, el cual se creó para efectuar un adecuado seguimiento y control de los fondos asignados a las diferentes Misiones.

El año 2003 por Resolución Ministerial 511/2003 se aprobó el Reglamento para Rendiciones de Cuentas con la finalidad de persuadir a la Misiones a mantener e informar de los fondos recibidos, el reglamento describe un conjunto de reglas, normas y procedimientos relacionados con el registro y descargo de cuentas que regulan la aplicación de los recursos destinados a gastos de funcionamiento y representación de las embajadas, representaciones permanentes y consulares de Bolivia acreditados en el exterior a las que se asignen recursos. Esta resolución se considera como un primer paso para estandarizar la información.

En los últimos años se desarrollaron diferentes proyectos similares, como ser:

- “Sistema Administrativo Financiero para el Gobierno Municipal de Chulumani”, realizado por la postulante Martha Patricia Aguilar Salomón [AGU98], el cual hace uso el modelo Orientado a Objetos tomando la notación de COAD Peter & YOURDON Edward.
- “Análisis y Diseño del Sistema Gerencial Financiero Contable Integrado de Lotería Nacional”, realizado por el postulante Milton Ramírez Linares [RAM01], el cual hace uso de las técnicas del modelo Estructurado, proponiendo implementar en diferentes estaciones de trabajo con una arquitectura Cliente-Servidor.
- “Sistema de Control Económico Financiero MB” realizado por el postulante Hugo Jacob Calle Álvarez [CAL02], el cual hace uso de las técnicas del modelo Estructurado de Análisis y Diseño, se encuentra orientado al

---

control y registro de las transacciones realizadas por la empresa mediante un sistema contable que emite estados financieros, además de contemplar inventarios.

- “Sistema Integrado de Información Administrativo – Financiero AADAPAL” realizado por el postulante Alex Adalid Ticona Gutiérrez [TIC03], el cual hace uso del modelo Esencial con dos componentes, que son el modelo Ambiental y el modelo del Comportamiento, bajo el enfoque Cliente-Servidor adoptando la topología estrella.

Si bien es cierto que existen diversos sistemas con características similares al propuesto, también es cierto que las técnicas utilizadas en los trabajos anteriormente mencionados son diversos, cada uno aplicado de manera particular al rubro al que pertenece la institución, es así que los modelos se ajustan a las necesidades de cada empresa, la principal diferencia con el presente proyecto radica que éste hace uso de Internet como vía de comunicación.

## **1.2 Planteamiento del problema**

El Ministerio de Relaciones Exteriores y Culto viendo las necesidades observadas para el mejor cumplimiento de las tareas asignadas pretende implementar un sistema informático denominado SIGASE<sup>5</sup> que permita la gestión y control administrativo de las oficinas del ministerio en el exterior, además el manejo centralizado de los datos en forma eficiente con la seguridad y disponibilidad adecuada. Es en este sentido el problema central a resolver es el de centralizar los datos que se envíen desde el exterior por parte de las Misiones en lo que respecta a la rendición de cuentas de cada una de las partidas asignadas, para

---

<sup>5</sup> Sistema Integrado de Gestión Administrativa del Servicio Exterior.

---

poder realizar un adecuado seguimiento y control de los fondos asignados (*Ver anexo A1 – Árbol de problemas*).

Problemas secundarios:

- Algunas Misiones presentan documentación que contienen muchas observaciones en la rendición de cuentas.
- Los funcionarios encargados hacen mal uso del reglamento de rendición de cuentas del servicio exterior.
- Se cuenta con pocos funcionarios para revisar la documentación presentada por las diferentes Misiones.
- Estandarizar formularios para la rendición de cuentas.
- Los informes presentados por las Misiones no tienen la consistencia deseada.

### **1.3 Objetivo**

#### **1.3.1 Objetivo General**

Desarrollar e implementar el subsistema de Control Financiero vía Web para el servicio exterior como parte del sistema SIGASE, el cual permita llevar un adecuado seguimiento y control de las diferentes partidas asignadas a las Misiones (*Ver anexo A2 – Árbol de objetivos*).

#### **1.3.2 Objetivos Específicos**

- ✓ Definir y crear el conjunto de controles de usuario<sup>6</sup> los cuales permitan la construcción del sistema informático.

---

<sup>6</sup> Controles personalizados que ofrecen una forma sencilla de dividir y reutilizar funciones de interfaz de usuario comunes entre las aplicaciones Web ASP.NET.

- 
- ✓ Establecer una interfaz<sup>7</sup> de comunicación con los demás subsistemas del SIGASE.
  - ✓ Establecer una arquitectura<sup>8</sup> de información en red.
  - ✓ Implementar una Base de Datos centralizados.
  - ✓ Tomar en cuenta los aspectos mas importantes para establecer la seguridad deseada para el sistema.

## **1.4 Justificación**

### **1.4.1 Justificación Social**

La población como parte del estado Boliviano resultará beneficiada con el presente proyecto dado que se podrá efectuar un adecuado control de los recursos del estado desembolsados por el TGN.

### **1.4.2 Justificación Técnica**

El sistema de información permitirá tener los registros de las partidas de manera clara y oportuna para ser manipulada en cualquier momento dado que se cuenta con todos los elementos necesarios para su implementación como servidores, conexión a Internet, VPN<sup>9</sup>, herramientas de desarrollo y otros.

### **1.4.3 Justificación Económica**

El sistema de información reducirá los errores más comunes tanto en el llenado como en la elaboración de informes, de esta manera se disminuirá los gastos

---

<sup>7</sup> Conexión física y/o funcional entre dos aparatos o sistemas independientes.

<sup>8</sup> Estructura lógica y/o física de los componentes que constituyen dicha estructura.

<sup>9</sup> Virtual Private Network (Redes Privadas Virtuales) es una red privada que se extiende, mediante un proceso de encapsulación o en algún caso de encriptación.

---

destinados a los materiales de escritorio así como el tiempo que se emplea para su realización.

### **1.5 Alcances y Aportes**

El sistema se enmarca dentro de las normas y procesos internos de la institución, el proyecto principalmente se basa en el “Reglamento para la rendición de cuentas en el servicio exterior”, el reglamento mencionado contempla la ejecución presupuestaria, comprobantes de ingreso, comprobantes de egreso, informes financieros, conciliación bancaria, caja chica y planilla de sueldos. El proyecto toma como alcance a todas las Misiones en el exterior.

El sistema se desarrollará mediante un conjunto de controles de usuario (User Control) esta es una técnica que brinda el Visual Studio .NET, esta nueva tecnología cuenta con un grupo de controles más robustos y la posibilidad de agruparlos mediante la utilización de los controles de usuario. En estos controles es posible utilizar controles conocidos como el Textbox, el DropDownList, Validation Controls, entre otros. Este tipo de controles (tienen extensión .ascx) se pueden reutilizar por toda la aplicación Web, incrustándolos en cualquier pagina asp.net, evitando tener que reescribir código. Los controles de usuario ofrecen una forma sencilla de dividir y reutilizar funciones de interfaz de usuario comunes entre las aplicaciones Web ASP.NET. Asimismo, como ocurre con las páginas de formularios, los controles de usuario se compilan cuando se solicita por primera vez y se almacenan en la memoria del servidor para reducir el tiempo de respuesta de las solicitudes siguientes.

Los controles de usuario ofrecen mayor flexibilidad que los archivos de inclusión del servidor SSI<sup>10</sup> gracias a la compatibilidad con los modelos de objetos que proporciona ASP.NET. En lugar de incluir simplemente las funciones

---

<sup>10</sup> Conjunto de comandos que se ejecutan en el servidor.

---

proporcionadas por otro archivo, puede programar las propiedades que declare en el control, al igual que con cualquier otro control de servidor ASP.NET.

## **Planificación del Proyecto**

Cuando uno trabaja para desarrollar un sistema, es importante seguir una serie de pasos que facilite obtener el resultado buscado [PRE03].

### **1.6 Metodología**

El sistema de marco lógico es una de las herramientas más conocidas por las organizaciones. Este ofrece herramientas para la conceptualización, el diseño, la ejecución, el seguimiento del desempeño y la evaluación de proyectos.

El sistema de marco lógico principalmente está compuesto por el árbol de problemas, árbol de objetivos y la matriz de marco lógico el cual debe ser mejorado a lo largo de las etapas de preparación y ejecución del proyecto (*Ver anexo A3 – Marco lógico*).

En las últimas décadas se desarrollaron varias metodologías al igual que notaciones como el UML [RJB99] [LAR99], el proceso unificado actúa como un modelo. En el presente trabajo se desarrolla con la metodología el Proceso de desarrollo propuesto por “Rational Software Corporation” resultado del esfuerzo de las tres últimas décadas en desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh [JBR99], dado las características del sistema a desarrollar en su complejidad y la imposibilidad de definir completamente el problema. RUP<sup>11</sup> es una de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software.

Las características del proceso unificado de modelado son:

---

<sup>11</sup> Rational Unified Process (Proceso Unificado de Rational), Metodología de desarrollo de software.

---

**Centrado en los Modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.

**Guiado por los casos de uso:** Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

**Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

**Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

Un proyecto realizado siguiendo RUP se divide en cuatro fases

1. Inicio (puesta en marcha)
2. Elaboración (definición, análisis, diseño)
3. Construcción (Implementación)
4. Transición (fin del proyecto y puesta en producción)

RUP define nueve actividades básicas a realizar en cada fase del proyecto

1. Modelado del negocio.
2. Análisis de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Pruebas.

## 1.7 Técnicas y herramientas

Dado las características y requerimientos de la institución en el presente proyecto emplea como servidor Windows Server 2003, como sistema operativo se usará Windows XP para los clientes estos deben contar con el navegador Internet Explorer 6 o superiores, el sistemas se lo hará en lenguaje de desarrollo Visual

---

Studio NET, usando como Gestor de Base de Datos SQL-Server 2000, además de contar con VPNs para proporcionar la seguridad deseada.



---

## CAPITULO 2

*En el presente capítulo se define los fundamentos teóricos bajo los cuales se realiza el proyecto, se describe tanto la metodología aplicada para su desarrollo, así como las técnicas y herramientas usadas para su implementación.*

### 2.1 El proceso unificado (RUP)

El Proceso Unificado de Rational (RUP, Rational Unified Process) es el producto final del esfuerzo realizado por un gran número de investigadores y empresas dedicadas al desarrollo de software en las últimas décadas, este un concepto de metodologías antes separadas. Entre las metodologías que más aportaron y sirvieron de base para el RUP se puede citar a Ericsson Approach<sup>12</sup>, Objectory<sup>13</sup> (abreviación de Object Factory), Rational Approach<sup>14</sup> y Rational Objectory Process (ROP)<sup>15</sup>.

Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza Rational Unified Process (RUP).

#### 2.1.1 Características

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

---

<sup>12</sup> Método de desarrollo propuesto por Ivar Jacobson, una aproximación de desarrollo basada en componentes e introdujo el concepto de casos de uso.

<sup>13</sup> Método de desarrollo propuesto por Jacobson, basado principalmente en el modelo de casos de uso.

<sup>14</sup> Método de desarrollo resultante de la unificación de conceptos desarrollados por Kruchten, Booch y Roice del cual se destacan el proceso iterativo y el desarrollo centrado en la arquitectura del programa.

<sup>15</sup> Método de desarrollo por Rational a partir de Objectory 3.8 y el enfoque Rational Approach adoptando UML como lenguaje de modelado.

---

### 2.1.1.1 Proceso dirigido por Casos De Uso

La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

Según [KRU00], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo.

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad<sup>16</sup> o rastreabilidad entre los artefactos<sup>17</sup> que son generados en las diferentes actividades del proceso de desarrollo.

### 2.1.1.2 Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados

---

<sup>16</sup> Capacidad de seguir una representación del diseño o un componente real del programa hasta los requisitos.

<sup>17</sup> Producto, Trozo de información que es producido, modificado o usado durante el proceso de desarrollo de Software.

---

(desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [KRU00].

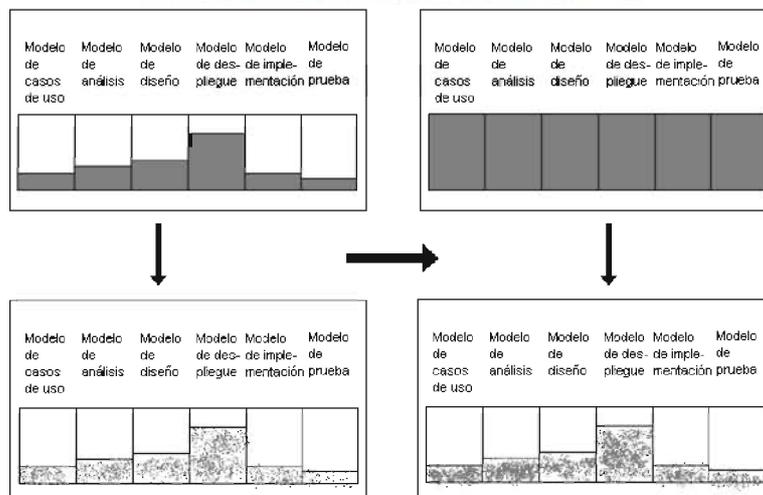
La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos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

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

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

La evolución de la arquitectura es manifiesta en las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de baselines<sup>18</sup> (línea base) y se va modificando dependiendo de las necesidades del proyecto.

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura [KRU00], el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.



**Figura 2.1** Los Modelos se completan, la arquitectura no cambia drásticamente.  
**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

Al final de la fase de elaboración se obtiene una baseline de la arquitectura donde fueron seleccionados una serie de Casos de Uso arquitectónicamente relevantes

<sup>18</sup> Instantánea del estado de todos los artefactos del proyecto, registradas para el efecto de gestión de configuración y control de cambios.

---

(aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas)

Como se observa en la Figura 2.1, durante la construcción los diversos modelos van desarrollándose hasta completarse (según se muestra con las formas rellenas en la esquina superior derecha). La descripción de la arquitectura sin embargo, no debería cambiar significativamente (abajo a la derecha) debido a que la mayor parte de la arquitectura se decidió durante la elaboración. Se incorporan pocos cambios a la arquitectura (indicados con mayor densidad de puntos en la figura inferior derecha) [JBR99].

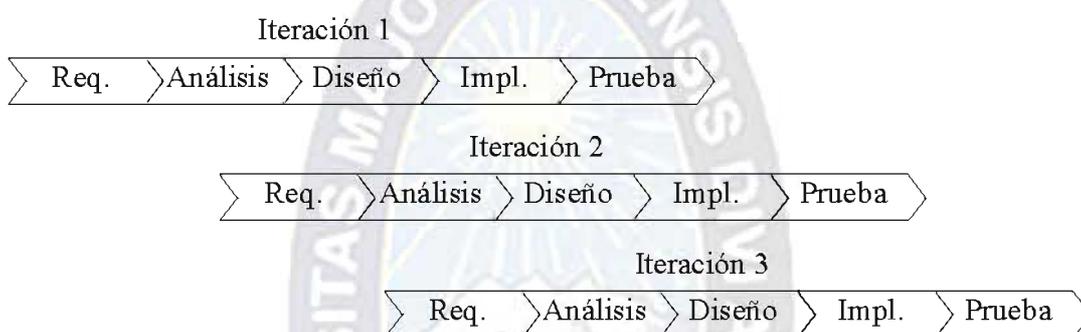
### **2.1.1.3 Proceso iterativo incremental**

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto [JBR99].

---

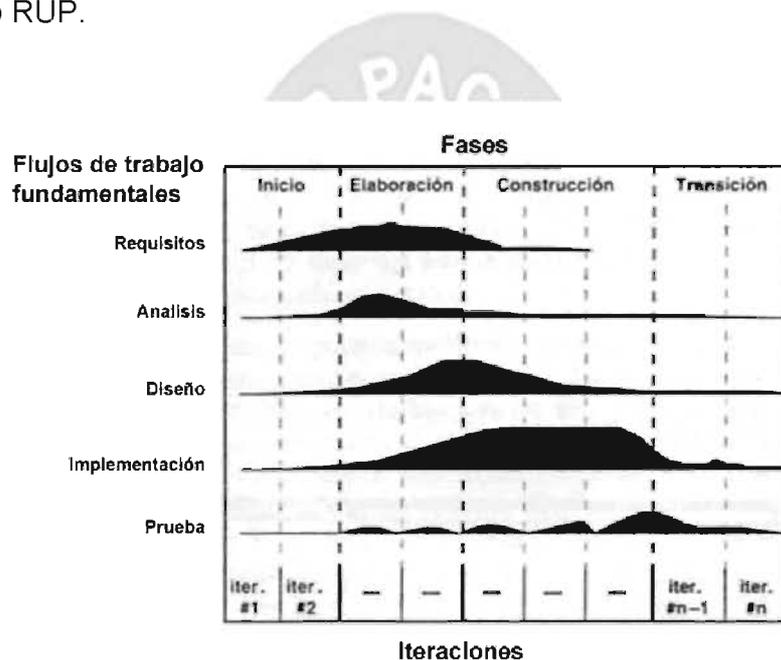
Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 2.2. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la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



**Figura 2.2** Cada iteración constituye una pasada a través de los cinco flujos de trabajo fundamentales.  
**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En la Figura 2.3 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



**Figura 2.3** Esfuerzo en actividades según fase del proyecto.  
**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline de la arquitectura [WEI02].

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

---

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

## **2.1.2 Estructura del proceso**

El proceso puede ser descrito en dos dimensiones o ejes [RSC98], estos son:

### **2.1.2.1 Eje Horizontal**

Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la Figura 2.3 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.

---

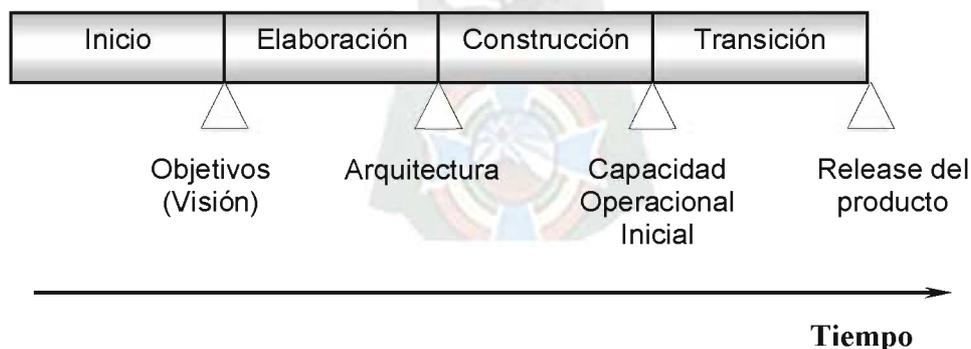
### 2.1.2.1 Eje Vertical

Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

### 2.1.3 Estructura dinámica del proceso, fases e iteraciones

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio, Elaboración, Construcción, Transición. Las fases y sus respectivos hitos se ilustran en la Figura 2.4.

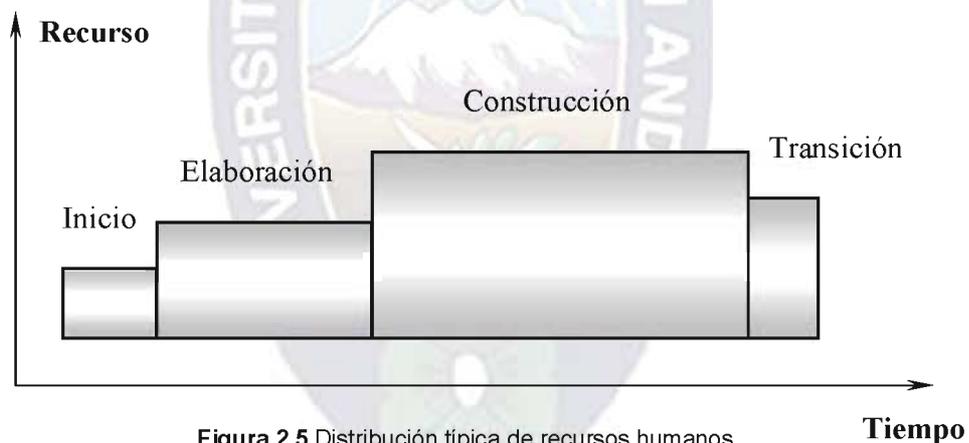


**Figura 2.4** Fases e hitos en RUP.  
**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, la Figura 2.5 ilustra porcentajes frecuentes al respecto. Consecuente con el esfuerzo señalado, la Figura 2.6 ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5%	20%	65%	10%
Tiempo Dedicado	10%	30%	50%	10%

**Tabla 2.1** Distribución típica de esfuerzo y tiempo.  
Fuente: Proceso Unificado de Desarrollo de Software, 1999.



**Figura 2.5** Distribución típica de recursos humanos.  
Fuente: Proceso Unificado de Desarrollo de Software, 1999.

### 2.1.3.1 Fase de Inicio

Su meta principal es lograr el consenso de todos los involucrados acerca de los objetivos del ciclo de vida del proyecto. Es muy importante especialmente en proyectos nuevos en que existen riesgos significativos en el negocio o la implementación de los requisitos, y deben ser solucionados para que el proyecto proceda. Para los proyectos que se enfocan en mejorar sistemas existentes, esta

---

fase es más breve, pero aún así centrada en asegurar que el proyecto vale la pena y se puede realizar. Además establece los objetivos para el ciclo del vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

**HITO.** Establecer el ámbito del producto, la identificación de los principales riesgos y la viabilidad del proyecto.

### **2.1.3.2 Elaboración**

El propósito de la etapa de Elaboración es crear la línea base de la arquitectura para el ciclo de vida del producto para así disponer de unos cimientos sólidos sobre los que se basará el grueso del esfuerzo de diseño e implementación durante la fase de Construcción. La arquitectura evoluciona de los requisitos más significativos considerados (aquellos que tienen un fuerte impacto en la arquitectura del sistema) y la evaluación de riesgos. La estabilidad de la arquitectura se evalúa mediante el uso de prototipos de arquitectura. 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.

**HITO.** Obtener una línea base de la arquitectura del sistema, capturar la mayoría de los requisitos y reducir los riesgos principales así como permitir la escalabilidad del equipo del proyecto durante la fase de construcción.

### **2.1.3.3 Construcción**

En la fase de Construcción se deben aclarar los requisitos restantes y completar el desarrollo del sistema basándose en la arquitectura que se ha sido añadida a la línea base. Puede ser vista como un proceso de fabricación donde se hace

---

énfasis en la administración de los recursos y el control de operaciones para optimizar costes, planificaciones y calidad. En este sentido la administración experimenta una transición del desarrollo de propiedad intelectual durante las fases de Inicio y Elaboración al desarrollo de productos instalables durante la Construcción y Transición.

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 versión beta.

**HITO.** Desarrollo del sistema con calidad de producción y prepararse para la entrega al equipo de transición. Toda la funcionalidad debe haber sido implementada y las pruebas para el estado beta de la aplicación completadas. Si el proyecto no logra alcanzar este hito, entonces la transición deberá posponerse una iteración.

#### **2.1.3.4 Transición**

La atención se enfoca en asegurar que el software está disponible para los usuarios finales, realizar la entrega del producto. Incluye las pruebas del producto como parte de su preparación para ser entregado, y la realización de ajustes menores en respuesta a la retroalimentación recibida de los usuarios. En este punto del ciclo de vida la retroalimentación de los usuarios debe enfocarse fundamentalmente en ajustes específicos y de corto alcance al producto junto a otros temas como configuración, instalación, y usabilidad. Referencias a otros ajustes estructurales mayores debieron haber sido solucionados anteriormente en el ciclo de vida y deberán documentarse para futuras generaciones del software. Se concluye una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones requeridas.

---

**HITO.** Consiste en decidir si los objetivos se cumplieron y si debe comenzarse otro ciclo de desarrollo. Es el resultado de la revisión y aceptación por parte del cliente de los artefactos que le han sido entregados.

#### **2.1.4 Flujo de trabajo fundamentales**

Los flujos fundamentales comprenden:

- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

Es importante mencionar antes de pasar a describir cada flujo que estos se repiten para cada iteración, vale decir si hacemos cinco iteraciones sobre las cuatro fases se debe llevar a cabo cinco veces todo el proceso, pero esto no siempre es así, y mas bien se deja al criterio de los desarrolladores los flujos que deben tomar para cada fase, por ejemplo en la fase inicial se debe dar mayor importancia a la recolección de requisitos y no así a la implementación o las pruebas que bien pueden ser nulas.

##### **2.1.4.1 Requisitos**

El modelo de requisitos tiene como objetivo delimitar el sistema y capturar la funcionalidad que debe ofrecer desde la perspectiva del usuario. Este modelo puede funcionar como un contrato entre el desarrollador y el cliente o usuario del sistema, y por lo tanto proyecta lo que el cliente desea según la percepción del desarrollador. Por lo tanto, es esencial que los clientes puedan comprender este modelo [JBR99].

---

El propósito del modelo de requisitos es comprender completamente el problema y sus implicaciones. Todos los modelos no solamente se verifican contra el modelo de requisitos, sino que también se desarrollan directamente de él. El modelo de requisitos sirve también como base para el desarrollo de las instrucciones operacionales y los manuales ya que todo lo que el sistema deba hacer se describe aquí desde la perspectiva del usuario. El modelo de requisitos no es un proceso mecánico, el analista debe interactuar constantemente con el cliente para completar la información faltante, y así clarificar ambigüedades e inconsistencias. El analista debe separar entre los requisitos verdaderos y las decisiones relacionadas con el diseño e implementación. Se debe indicar cuales aspectos son obligatorios y cuales son opcionales para evitar restringir la flexibilidad de la implementación.

#### **a) Descripción del Problema**

Una primera actividad que es aconsejable cumplir es llevar una pequeña descripción del problema, esta es una descripción muy preliminar de necesidades que sirve únicamente como punto de inicio para comprender los requisitos del sistema. Esta descripción es a menudo preparada por un cliente la cual debe evolucionar por medio del modelo de requisitos para lograr la especificación final del sistema a desarrollarse. La descripción del problema debe ser una descripción de necesidades y no una propuesta para una solución. La descripción inicial puede ser incompleta e informal. No hay razón para esperar que la descripción inicial del problema, preparada sin un análisis completo, sea correcta.

#### **b) Enumerar los requisitos**

Durante la vida del sistema, los clientes, usuarios, analistas y desarrolladores aparecen con muchas buenas ideas que podrían convertirse en verdaderos requisitos [JBR99].

Cada una de estas características lleva un nombre corto y una breve explicación o definición y un conjunto de valores tales como estado, coste, prioridad, nivel de riesgo.

---

### **c) Modelo del dominio**

El modelado del dominio captura los conceptos más importantes del contexto del sistema, nos ayuda a comprender y describir las clases más significativas.

### **d) Modelado del negocio**

El modelado de negocio es una técnica para comprender los procesos de negocio de la organización, y nos ayuda a identificar los casos de uso más relevantes que el sistema debe soportar, es modelo es muy importante pues nos ayuda a comprender el contexto del sistema y es representado median un diagrama de caso de uso.

### **e) Requisitos adicionales**

Los requisitos adicionales son fundamentalmente requisitos no funcionales que no pueden asociarse a ningún casos de uso en concreto, en cambio cada uno de estos requisitos tiene impacto en varios casos de uso o en ninguno [JBR99].

### **f) Modelo de casos de uso**

El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario.

Para modelar el sistema mediante casos de uso primero es necesario identificar a los actores del sistema, luego se identifica a los casos de usos en los que intervienen dichos actores.

Parte fundamental del modelo de casos de uso es una descripción textual detallada de cada uno de los actores y casos de uso identificados. Estos documentos son sumamente críticos ya que a partir de ellos se desarrollará el sistema completo. Para documentar los actores encontrados, como los casos de uso se usará el siguiente formato:

<b>Actor:</b>	Nombre del actor.
<b>Casos de uso:</b>	Nombre de los casos de uso en los que participa.
<b>Tipo:</b>	Primario o Secundario.
<b>Descripción:</b>	Breve descripción del actor.

**Tabla 2.2** Descripción de actores.  
**Fuente:** Ingeniería de software orientada a Objetos, 2002.

<b>Casos de uso:</b>	Nombre del caso de uso.
<b>Actores:</b>	Actores primarios y secundarios que interactúan en los casos de uso.
<b>Tipo:</b>	Tipo de flujo: Básico, Inclusión, Extensión, Generalización, o algún otro.
<b>Propósito:</b>	Razón de ser del caso de uso.
<b>Descripción:</b>	Breve descripción del caso de uso.
<b>Precondiciones:</b>	Condiciones que deben satisfacerse para poder ejecutar el caso de uso.
<b>Flujo principal:</b>	El flujo de eventos más importante del caso de uso, donde dependiendo de las acciones de los actores se continuará con alguno de los subflujos.
<b>Subflujos:</b>	Los flujos secundarios del caso de uso.
<b>Excepciones:</b>	Excepciones que pueden ocurrir durante el caso de uso.
<b>Post Condiciones:</b>	Condición cumplida luego de ejecutar el caso de uso.

**Tabla 2.3** Descripción de casos de uso.  
**Fuente:** Ingeniería de software orientada a Objetos, 2002.

Finalmente es importante mencionar que se pueden usar cualquier otro diagrama UML el cual nos pueda ayudar a la captura de requisitos, esto también es válido para cualquier otro flujo de trabajo.

#### 2.1.4.2 Análisis

Cuando ya se ha desarrollado y aceptado el modelo de requisitos se comienza el desarrollo del modelo de análisis. El objetivo del modelo de análisis es comprender y generar una arquitectura de objetos para el sistema en base a lo especificado en el modelo de requisitos. Durante esta etapa no se considera el ambiente de implementación, lo cual incluye al lenguaje de programación,

---

manejador de base de datos, distribución o configuración de hardware, etc. El análisis pretende modelar el sistema bajo condiciones ideales, garantizando que la arquitectura de software resultante se suficientemente robusta y extensible para servir de base a la estructura lógica de la aplicación pero sin consideraciones relativas al entorno de implementación que es posible que cambien incluso radicalmente.

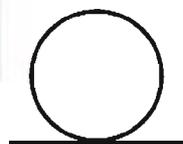
Es importante enfatizar que el modelo de análisis no es una reflexión del dominio del problema sino una representación de ésta adaptada a la aplicación particular. El modelo de análisis genera una representación conceptual del sistema, consistiendo de clases de objetos. Cada una de las clases de objetos contribuye de manera especial para lograr una arquitectura robusta.

#### **2.1.4.2.1 Clases con estereotipos**

Para los sistemas de información la arquitectura del sistema según nuestro modelo de análisis se basa en tres estereotipos básicos de objetos:

##### **a) Clase Entidad**

El estereotipo entidad (“entity” en inglés) para objetos que guarden información sobre el estado interno del sistema, a corto y largo plazo, correspondiente al dominio del problema. Todo comportamiento naturalmente acoplado con esta información también se incluye en los objeto entidad.



**Figura 2.6** Clase de Entidad.  
**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

---

## b) Clase Interfaz

El estereotipo interface o borde (“boundary” en inglés) para objetos que implementen la presentación o vista correspondiente a las bordes del sistema hacia el mundo externo, para todo tipo de actores, no sólo usuarios humanos.



**Figura 2.7** Clase de Interfaz.

**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

## c) Clase Control

El estereotipo control (“control” en inglés) para objetos que implementen el comportamiento o control especificando cuando y como el sistema cambia de estado, correspondiente a los casos de uso. Los objetos control modelan funcionalidad que no se liga naturalmente con ningún otro tipo de objeto, como el comportamiento que opera en varios objetos entidad a la vez, por ejemplo, hacer algún proceso y luego devolver el resultado a un objeto borde.



**Figura 2.8** Clase de Control.

**Fuente:** Proceso Unificado de Desarrollo de Software, 1999.

---

#### **2.1.4.2.2 Identificación de Clases**

Para llevar a cabo la transición del modelo de requisitos al modelo de análisis se deben identificar los objetos necesarios para implementar todos los casos de uso. La arquitectura de objetos debe considerar los tres tipos de estereotipos. Para lograr esto se debe identificar primero las clases interfaz, luego las clases entidad y finalmente las de control [JBR99].

Por otro lado, los objetos entidad e interfaz deben contener funcionalidad más bien local limitando su efecto en los demás objetos. El trabajo del analista consiste en distribuir lo mejor posible el comportamiento especificado en el modelo de requisitos en los diferentes tipos de objetos de la arquitectura de análisis.

Finalmente una vez identificados las clases de entidad, interfaz y control se construye los diagramas de colaboración.

#### **2.1.4.2.3 Diagramas de Interacción**

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si consideramos el “interior” del sistema, un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje a algún otro objeto, y de esta forma los objetos implicados interactuarán para llevar a cabo el caso de uso. En el análisis preferimos mostrar esto con diagramas de colaboración ya que nuestro objetivo es identificar requisitos y responsabilidades sobre objetos, y no identificar secuencias de interacción detallada y ordenadas cronológicamente (en ese caso, utilizaremos los diagramas de secuencia).

---

En los diagramas de colaboración, mostramos las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debería denotar el propósito del objeto invocante en la interacción con el objeto invocado.

### **2.1.4.3 Diseño**

El modelo de diseño es un refinamiento y formalización adicional del modelo de análisis donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado del modelo de diseño son especificaciones muy detalladas de todos los objetos, incluyendo sus operaciones y atributos [JBR99].

Se requiere un modelo de diseño ya que el modelo de análisis no es lo suficientemente formal para poder llegar al código fuente. Por tal motivo se debe refinar los objetos, incluyendo las operaciones que se deben ofrecer, la comunicación entre los diferentes objetos, los eventos que los objetos envían entre si, etc. El sistema real debe adaptarse al ambiente de implementación. En el análisis se asume un mundo ideal para el sistema, en la realidad se debe adaptar el sistema al ambiente de implementación, algo que puede cambiar durante el ciclo de vida del sistema. Se busca además aspectos como, los requisitos de rendimiento, necesidades de tiempo real, concurrencia, el lenguaje de programación, el sistema de manejo de base de datos, etc. Se desea también validar los resultados del análisis. Según el sistema crece y se formaliza, se verá qué tan bien los modelos de requisitos y análisis describen al sistema.

Durante el diseño, se puede ver si los resultados del análisis son apropiados para su implementación. Si se descubre aspectos que no están claros en alguno de los modelos anteriores, estos deben ser clarificados, regresando a etapas anteriores.

---

Aunque esto pudiera verse como deficiencias del resultado de las fases anteriores que deben ser clarificadas aquí, esto sería una visión incorrecta de las diferentes etapas del desarrollo, ya que el propósito de los modelos de requisitos y análisis es comprender el sistema y darle una buena estructura. También es importante comprender que las consideraciones tomadas en cuenta durante el diseño deben influir en la estructura del sistema lo menos posible. Es la propia aplicación la que controla la estructura, no las circunstancias de su implementación.

#### **2.1.4.3.1 Estrategias de Diseño**

Antes de poder resolver el diseño es necesario tomar decisiones generales sobre las estrategias de diseño a seguir. Algunas de las decisiones a tomar se presentan a continuación y se relacionan con aspectos que incluyen la arquitectura, robustez, reuso y extensibilidad del sistema [WEI02].

##### **a) Arquitectura**

El término arquitectura se refiere, en nuestro caso, a la organización de las clases dentro del sistema. Durante el modelo de análisis se generó una arquitectura de clases para el sistema y se definió la funcionalidad “conceptual” ofrecida por las distintas clases dentro de la arquitectura. Durante el diseño esta arquitectura debe detallarse, pudiéndose cambiar los aspectos considerados inicialmente, como fue la funcionalidad inicialmente asignada a cada clase, e incluso las propias clases, como hemos mencionado al inicio del capítulo.

El conocimiento y funcionalidad asignada a cada clase puede ser vista como la “inteligencia” de cada clase dentro del sistema. En otras palabras, algunas clases pueden ser vistas como más inteligentes que otras según el conocimiento y control que tengan sobre las demás clases. Por ejemplo, colecciones de objetos tales como listas o arreglos, no se consideran como particularmente inteligentes ya que pueden manipular y obtener información sobre las clases que almacenan,

---

pero tienen relativamente poco impacto sobre estas u otras clases dentro del sistema. Por otro lado, un manejador de interface de usuario requiere mayor inteligencia, ya que debe poder administrar la interacción con el usuario, incluyendo manejo de eventos y manipulaciones sobre las pantallas. Una clase aún más inteligente es el controlador o manejador de la lógica completa de la aplicación, ya que es responsable de administrar a los propios manejadores de interface de usuario y relacionar su funcionalidad con el resto del sistema.

## **b) Robustez**

La robustez de un sistema debe ser uno de los objetivos principales del diseño. Jamás debe agregarse funcionalidad o simplificar código a expensas de la robustez. El sistema debe estar protegido contra errores y debe al menos ofrecer diagnósticos para las fallas que aún pudiesen ocurrir, en particular aquellas que son fatales. Durante el desarrollo es a veces bueno insertar instrucciones internas en el código para descubrir fallas, aunque luego sean removidas durante la producción. En general se debe escoger lenguajes de programación que apoyen estos aspectos, como son el manejo de excepciones. Las principales consideraciones relacionadas con la robustez de un sistema son las siguientes:

- El sistema debe estar protegido contra parámetros incorrectos proporcionados por el usuario.
- El sistema no debe optimizarse hasta que este funcione de manera correcta.
- El sistema debe incluir estructuras de datos que no tengan límites predefinidos. Durante el diseño es difícil predecir la capacidad máxima esperada para la estructura de datos en la aplicación. Por lo tanto, se debe escoger estructuras de datos como las listas, a diferencia de los arreglos.
- El sistema debe instrumentar un monitoreo de rendimiento y búsqueda de errores.
- El encapsulamiento juega un papel fundamental para la robustez del sistema. Ocultar la información interna, atributos e implementación de métodos, a una clase permite que ésta pueda ser cambiada sin afectar al resto del sistema.

---

### **c) Reuso**

El reuso es un aspecto fundamental del diseño. Cuanto más se pueda reutilizar el código mejor será la robustez del sistema. Las siguientes son algunas estrategias para mejorar las posibilidades de reuso del diseño:

- A través de la herencia se puede incrementar el reuso de código. Se toman los aspectos comunes a clases similares utilizando superclases comunes. Este enfoque es efectivo cuando las diferencias entre las clases son pequeñas y las similitudes son grandes.
- El uso impropio de herencia puede hacer que los programas sean difíciles de mantener y extender. Como alternativa, la delegación provee un mecanismo para lograr el reuso de código pero sin utilizar herencia. Esto se basa en el uso de agregación a través de clases intermediarias que ocultan la funcionalidad de las clases a las cuales se delega.
- El encapsulamiento es muy efectivo para lograr el reuso, pudiéndose aplicar tanto al nivel de los objetos como de componentes desarrollados en otras aplicaciones. Estos componentes pueden ser reutilizables como fueron diseñados a simplemente agregando nuevas interfaces.

### **d) Extensibilidad**

La mayoría de los sistemas son extendidos en manera no prevista por el diseño original. Por lo tanto, los componentes reutilizables mejorarán también la extensibilidad. Las siguientes son algunas de las perspectivas de extensibilidad:

- Nuevamente, se debe encapsular clases, ocultando su estructura interna a las otras clases. Sólo los métodos de la clase deben acceder sus atributos.
- No se debe exportar estructuras de datos desde un método. Las estructuras de datos internas son específicas al algoritmo del método. Si se exporta las estructuras se limita la flexibilidad para poder cambiar el algoritmo más tarde.
- Una clase debe tener un conocimiento limitado de la arquitectura de clases del sistema. Este conocimiento debe abarcar únicamente las asociaciones

---

entre ella y sus vecinos directos. Para interactuar con un vecino indirecto, se debe llamar una operación del objeto vecino para atravesar la siguiente relación. Si la red de asociaciones cambia, el método de la clase puede ser modificado sin cambiar la llamada.

- Se debe evitar expresiones de casos sobre tipos de objetos. Para ello, se debe usar métodos (polimorfismo) para seleccionar el comportamiento a ejecutarse basado en el tipo del objeto en lugar de expresiones de casos. El polimorfismo evita muchas de estas comparaciones de tipos.
- Se debe distinguir entre operaciones privadas y públicas. Cuando una operación pública es usada por otras clases, se vuelve costoso cambiar la interface, por lo cual las operaciones públicas deben ser definidas con cuidado. Las operaciones privadas son internas a la clase y sirven únicamente de ayuda para implementar operaciones públicas. Las operaciones privadas pueden ser removidas o su interface cambiada para modificar la implementación de la clase, teniendo un impacto limitado en los demás métodos de la clase.

#### **2.1.4.3.2 Diagramas de secuencias del diseño**

Una vez completado tanto el diseño de objetos como el del sistema, es posible describir los casos de uso del análisis en base a los protocolos de clases anteriormente definidos. Esto es muy importante ya que permitirá revisar que el diseño esté lógicamente completo. Para ello se describen los casos de uso mediante diagramas de secuencia, los cuales se pueden referir directamente a las clases, o incluso a partir de la interacción entre subsistemas. Esto es a menudo una técnica muy útil ya que se puede diseñar un subsistema mostrando sólo las interfaces de los subsistemas relacionados. A un nivel más detallado se pueden mostrar las interacciones internas del subsistema. Normalmente, los eventos en el diagrama corresponden a los protocolos anteriormente diseñados y se especifican exactamente como se verían en el código final.

---

#### **2.1.4.3.3 Diagramas de Clases**

Este tipo de diagramas nos muestra la estructura estática de conceptos, tipos y clases. Los conceptos indican cómo piensan los usuarios acerca del mundo; los tipos muestran las interfaces de los componentes de software; las clases muestran la implementación de los componentes de software [FOW99].

Una clase de diseño y sus objetos, y de ese modo también los subsistemas que contienen las clases de diseño, a menudo participan en varias realizaciones de casos de uso. También puede darse el caso de algunas operaciones, atributos y asociaciones sobre una clase específica que son relevantes para sólo una realización de caso de uso. Esto es importante para coordinar todos los requisitos que diferentes realizaciones de casos de uso imponen a una clase, a sus objetos y a los subsistemas que contiene. Para manejar todo esto, utilizamos diagramas de clases conectados a una realización de caso de uso, mostrando sus clases participantes, subsistemas y sus relaciones. De esta forma podemos guardar la pista de los elementos participantes en una realización del caso de uso.

#### **2.1.4.3.4 Identificar operaciones**

En esta etapa identificamos las operaciones que las clases de diseño van a necesitar y describimos esas operaciones utilizando la sintaxis de los lenguajes de programación. Esto incluye especificar la visibilidad de cada operación. Algunas entradas importantes en esta etapa son:

- Las responsabilidades de cualquier clase del análisis que tenga una traza con la clase del diseño. Una responsabilidad a menudo implica una o varias operaciones. Es más, si las entradas y salidas se describen para las responsabilidades, pueden ser utilizadas como un primer bosquejo de parámetros formales y valores de retomo de las operaciones.

- 
- Los requisitos especiales de cualquier clase de análisis que tenga una traza con la clase del diseño. Recuérdese que aquellos requisitos a menudo necesitan ser manejados por el modelo de diseño, posiblemente incorporando algún mecanismo o tecnología de diseño genérico o tecnología de base de datos.
  - Las interfaces que la clase de diseño necesita proporcionar. Las operaciones de las interfaces también necesitan ser proporcionadas por la clase de diseño.
  - Las realizaciones de caso de uso-diseño en las que la clase participa

#### **2.1.4.4 Implementación**

El modelo de implementación toma el resultado del modelo de diseño para generar el código final. Esta traducción debe ser relativamente sencilla y directa, ya que las decisiones mayores han sido tomadas durante las etapas previas.

Durante el modelo de implementación se hace una adaptación al lenguaje de programación y/o la base de datos de acuerdo a la especificación del diseño y según las propiedades del lenguaje de implementación y base de datos.

Aunque el diseño de objetos es bastante independiente del lenguaje actual, todos los lenguajes tendrán sus particularidades, las cuales deberán adecuarse durante la implementación final. La elección del lenguaje influye en el diseño, pero el diseño no debe depender de los detalles del lenguaje. Si se cambia de lenguaje de programación no debe requerirse el rediseño del sistema.

En general, no se debe comenzar prematuramente a programar, es importante primero completar el proceso de planeación del sistema final desarrollado durante el diseño. Se debe usar guías de programación existentes en la organización. Si no existen, el equipo de software debe crear sus propias guías para decidir aspectos, como formatos para la asignación de nombres a las variables, estilo de programación, métodos de documentación, y documentación en línea. Vale la pena resaltar que aunque existe cierta automatización en el proceso de generación del código final, en su gran mayoría los programadores hacen de manera “manual” la transición final a código fuente.

---

#### **2.1.4.5 Pruebas**

Probar un producto es relativamente independiente de la metodología de desarrollo utilizada para construirlo. Existen diversos tipos de pruebas aplicados durante las diferentes actividades del proceso de desarrollo. Estas pruebas requieren de tiempo y presupuesto adicional, pudiendo llegar a significar entre un 30% y un 50% del costo total de desarrollo. Por tal motivo, el modelo de pruebas debe ser planificado con anticipación y de manera integral junto con el propio desarrollo del sistema. Es un error pensar que las pruebas son la última actividad del desarrollo ya que no se puede lograr software de alta calidad sólo mediante pruebas finales y depuraciones. Las pruebas deben hacerse en paralelo al desarrollo del sistema, teniendo pruebas finales únicamente como certificación final de la calidad del producto y no como la oportunidad para encontrar errores. Encontrar errores al final del desarrollo es bastante problemático dado que requerirá regresar a etapas anteriores para resolverlos. Se considera que "evitar defectos" es más poderoso que "remover defectos" [WEI02].

#### **2.1.5 Correspondencia entre el modelo de objetos y las tablas**

##### **a) Correspondencia entre clase de objetos y tablas**

- Cada clase se corresponde con una o más tablas. (Similarmente, una tabla puede corresponder a más de una tabla si están conectadas mediante una asociación uno a uno o bien uno a muchos) [RUM98].

##### **b) Correspondencia entre asociaciones y tablas**

- Cada asociación uno a muchos se corresponde con una tabla diferente, o bien se puede incluir en forma de clave externa dentro de la tabla para la clase "muchos".

- 
- Cada asociación uno a uno se corresponde con una tabla distinta, o bien puede ser incluida en forma de clave externa dentro de la tabla de cualquiera de las clases.
  - Para las asociaciones uno a muchos y uno a uno, si no hay ciclos, se tiene la opción adicional de almacenar la asociación y ambos objetos relacionados dentro de una misma tabla. Hay que tener en cuenta que esto puede introducir redundancias y violar las formas normales.
  - Los nombres de rol se incorporan como parte del nombre de atributo de las claves externas.
  - Las asociaciones n-arias ( $n > 2$ ) se corresponden con distintas tablas. En algunas ocasiones puede servir de ayuda promocionar una asociación n-aria a una clase.
  - Las asociaciones cualificadas se corresponden con una tabla independiente con no menos de tres atributos, la clave primaria de cada una de las clases relacionadas, y el cualificador.
  - Las agregaciones siguen las mismas reglas que las asociaciones.

**c) Correspondencia entre generalizaciones de herencia simple y tablas:**

- La superclase y todas las subclases se corresponden con una tabla.
- No hay tabla de superclase, los atributos de la superclase se duplican para cada subclase.
- No hay tablas de subclases, se llevan todos los atributos de las subclases al nivel de la superclase.

**d) Correspondencia entre herencia múltiple disjunta y tablas:**

- La superclase y todas las subclases se corresponden con una tabla.

**e) Correspondencia entre herencia múltiple solapada y tablas:**

- La superclase y todas las subclases se corresponden con una tabla; la relación de generalización también se corresponde con una tabla.

---

## 2.1.6 Seguridad

La seguridad de las aplicaciones no es un tema que comience en el momento de escribir el código, armar una aplicación segura implica pensar en esta en durante todo el proceso de desarrollo, podemos comenzar definiendo que es lo que hace la aplicación y como va ser utilizado [SEI06].

Establecer un diagrama de esa aplicación, donde quede bien claro los subsistemas que esta aplicación va a tener, cuales son los flujos de datos que la aplicación va a manipular, finalmente hacer un listado de afirmaciones de las cosas reales que la aplicación hace.

Hacer consideraciones de seguridad significa de alguna manera ponerse y pensar como el agresor, tratar de encontrar en el diagrama las vulnerabilidades y buscar mecanismos que solucionen estas vulnerabilidades encontradas.

Amenaza	Ejemplos
Inyección de SQL	Incluyendo comandos de SQL en el texto ingresado
Script entre sitios	Usando script del lado del cliente malicioso
Modificación del ingreso	Cambiando valores de campos ocultos
Robo de claves	Usando un investigador de paquetes ( <i>sniffer</i> ) para obtener claves de acceso o cookies no encriptadas
Reemplazo de sesión	Uso de Cookie de sesión "robado", para acceder a sitios
Reemplazo de identidad	Uso de Cookie de Autenticación por Formulario, para hacerse pasar por otro
Revelado de información	Mostrar al cliente el seguimiento de la pila, cuando sucede un error

**Tabla 2.4** Amenazas más comunes a considerar para aplicaciones de tipo Web.

Existen dos mecanismos básicos para identificar amenazas, el primero de ellos es armar una lista de las mismas comenzando siempre con una lista base la cual contiene las amenazas más comunes, dada esta lista identificar las que comprometen a nuestra aplicación y determinar puntualmente en que punto o puntos de la aplicación puede ser afectado. El segundo método conocido como STRIDE es una lista con categorías de los distintos tipos de amenazas, los identifica de acuerdo a ciertos valores que permite darles mayor importancia aquellos que puedan resultar más agresivos, en algunas oportunidades se puede dibujar árboles donde los nodos raíz representan las metas del atacante y de esa manera dibujar en dicho árbol cuales son las distintas condiciones que podrían facilitar a un atacante encontrar un espacio por el cual realizar su acción de ataque [AEA06].

Modelando amenazas	
1	Identificar elementos
2	Documentar Arquitectura
3	Descomponer la Aplicación
4	Identificar Amenazas
5	Documentar Amenazas
6	Cuantificar Amenazas

**Tabla 2.5** Modelo de amenazas.

Una manera de cuantificar una amenaza es usando el método DREAD cuya sigla deriva de las distintas palabras que compone el método.

<b>D</b>	Damage	<i>¿Cuales son las consecuencias?</i>
<b>R</b>	Reproducibility	<i>¿Se puede reproducir bajo ciertas circunstancias?</i>
<b>E</b>	Exploitability	<i>¿Cuan fácil es realizarla?</i>
<b>A</b>	Affected users	<i>¿Cuantos usuarios pueden verse afectados?</i>
<b>D</b>	Discoverability	<i>¿Es fácil de descubrir?</i>

**Tabla 2.6** Descripción del mecanismo DREAD.

Lo que hacemos con el método de medición DREAD es cuantificar los distintos elementos de este método en tres valor alto medio y bajo, la siguiente tabla muestra el significado de cada elemento que compone el método DREAD.

	<b>Alto (3)</b>	<b>Medio (2)</b>	<b>Bajo (1)</b>
Daño Potencial	El agresor puede obtener datos muy sensibles, dañar servidores etc.	Puede obtener datos sensibles, pero casi nada más	Puede acceder a datos poco importantes
Reproductibilidad	Siempre es posible	Sucedé si se realiza en un corto tiempo	Raramente se puede hacer
Explotabilidad	puede hacerlo	Se deben tener ciertos conocimientos	Tal vez alguno
Usuarios afectados	La mayoría	Algunos	Pocos, si es que hay alguno
Encubrimiento	Fácil de ver	Más costoso de ver	Muy difícil de encontrar

**Tabla 2.7** Descripción de las ponderaciones para aplicar DREAD.

### 2.1.7 Arquitectura

La arquitectura es un término general que se aplica a la estructura de un sistema informático o de una parte del mismo. El término se aplica también al diseño del software de sistema, por ejemplo, el sistema operativo, y a la combinación de hardware y software básico que comunica los aparatos de una red informática. La arquitectura de ordenadores se refiere a toda una estructura y a los detalles necesarios para que sea funcional, es decir, cubre sistemas informáticos, microprocesadores, circuitos y programas del sistema [ARQ06].

La arquitectura en aplicaciones Web se divide en funcional y estructural.

---

### **a) Arquitectura Funcional**

La arquitectura funcional describe como la aplicación esta montado físicamente, es decir describe las redes, computadoras, servidores de paginas, servidores de archivos, servidores de base de datos, etc.

### **b) Arquitectura Estructural**

La arquitectura estructural describe como la aplicación será diseñada a nivel lógico, es decir la correspondencia que existe entre directorios y archivos que contenga la aplicación.

### **2.1.8 Calidad de la aplicación**

Un elemento clave para cualquier proceso es la medición. Empleamos medidas para entender mejor los atributos de los modelos que creamos pero fundamentalmente empleamos las medidas para valorar la calidad de los productos de ingeniera o de los sistemas que construimos [PRESS, 2005].

### **Factores de Calidad ISO 9126.**

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos claves de calidad para el software. El estándar identifica seis atributos claves de calidad: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad.

Los factores ISO9126 no necesariamente son utilizados para medidas directas, en cualquier caso facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calida de un sistema.

### **a) Confiabilidad**

La confiabilidad de un producto se define como la probabilidad de operación libre de fallos del sistema en un entorno determinado y durante un tiempo específico. La confiabilidad del sistema se calcula a partir de los subsistemas o módulos.

---

Se considera dos casos:

**Caso 1:** Si n componentes que funcionan independientemente están conectados en serie, y si el i-ésimo componente tiene la confiabilidad  $R_i(t)$ , la confiabilidad  $R(t)$  del sistema completo esta dada por:

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

**Caso 2:** si n componentes que funcionan independientemente están conectados en paralelo, y si el i-ésimo componente tiene la confiabilidad  $R_i(t)$ , la confiabilidad  $R(t)$  del sistema completo esta dada por:

$$R(t) = 1 - \{1 - R_1(t)\} \{1 - R_2(t)\} \{1 - R_3(t)\} \dots \{1 - R_n(t)\}$$

Como  $R(t)$  se halla en un intervalo de tiempo, se puede aplicar la distribución exponencial para su calculo como se muestra a continuación:

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

Donde:

- R (t): confiabilidad del subsistema.
- $\lambda$ : Error o tasa constante de fallas.
- t: Tiempo de operación del sistema.

### b) Usabilidad

El estándar ISO9126 define la usabilidad como “la capacidad de un producto de software de facilitar a usuarios específicos alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción en un contexto específico de uso”. Añade que “calidad en uso es la visión de calidad de los usuarios de un ambiente conteniendo software, y es medida sobre los resultados de usar el software en el ambiente, antes que sobre las propiedades del software en sí mismo”.

### c) Portabilidad

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

---

#### **d) Mantenibilidad.**

La Mantenibilidad esta asociado a la detección y corrección de fallas, a los cambios debido a los requerimientos del usuario, a las adaptaciones requeridas a medida que evoluciona el software.

El estándar IEEE 982.1-1999 [IEEE 1994] sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software. Se determina así.

Mt = Numero de Módulos en la versión actual.

Fc = Numero de Módulos en la versión actual que se han cambiado.

Fa = Numero de Módulos en la versión actual que se han añadido.

Fd = Numero de Módulos en la versión anterior que se han borrado en la versión actual.

#### **2.1.9 Calidad de la aplicación en el entorno Web**

Adicionalmente se ve por conveniente aplicar al sistema la metodología de inspección Cuantitativa como la WebQEM, ampliamente reconocida para la evaluación de WebApps, en el tipo de Métodos de Inspección por Características en relación al atributo de confiabilidad del sistema, evaluado a los enlaces y paginas.

Para determinar la calidad del entorno Web se hará uso de la metodología Web Site QEM [OLS06].

A continuación se presenta la selección y representación de las características y atributos para este trabajo.

- 
- 1. Usabilidad**
    - 1.1 Compresibilidad Global del sistema
      - 1.1.1 Esquema de Organización Global
      - 1.1.2 Calidad en el Sistema de Etiquetado
    - 1.2 Mecanismos de ayuda y retroalimentación en línea
      - 1.2.1 Calidad de la ayuda
      - 1.2.2 Indicador de Última Actualización
      - 1.2.3 Retroalimentación
        - 1.2.3.1 Cuestionario
        - 1.2.3.2 Comentarios y ayuda
    - 1.3 Aspectos de Interfaces y Estéticos.
      - 1.3.1 Permanencia y estabilidad en la presentación de los controles Principales
      - 1.3.2 Aspectos de Estilos
      - 1.3.3 Preferencia Estética.
    - 1.4 Misceláneas
      - 1.4.1 Soporte a Lenguaje extranjero
      - 1.4.2 Indicador de Resolución de Pantalla
  - 2. Funcionalidad**
    - 2.1 Aspectos de Recuperación y Búsquedas
      - 2.1.1 Mecanismo de Búsqueda en el sistema
        - 2.1.1.1 Búsqueda Restringida
      - 2.1.2 Mecanismos de Recuperación
        - 2.1.2.1 Nivel de Personalización
    - 2.2 Aspectos de Navegación y Exploración
      - 2.2.1 Navegabilidad
      - 2.2.2 Orientación
    - 2.3 Aspectos de Dominio Orientado al Visitante
      - 2.3.1 Relevancia de Contenido
        - 2.3.1.1 Información de Unidades
        - 2.3.1.2 Informaciones de Representaciones en el Exterior
  - 3. Confiabilidad**
    - 3.1 No Deficiencia
      - 3.1.1 Errores de enlaces
        - 3.1.1.1 Enlaces rotos
        - 3.1.1.2 Enlaces inválidos
        - 3.1.1.3 Enlaces no implementados
      - 3.1.2 Errores o Deficiencias Varias
        - 3.1.2.1 Deficiencias o cualidades ausentes debido a diferentes navegadores(Browser)
        - 3.1.2.2 Deficiencias o Resultados Inesperados Independientes de Browsers (Por ej. Errores de búsqueda imprevistos)
  - 4. Eficiencia**
    - 4.1 Performance
      - 4.1.1 Páginas de Acceso Rápido
    - 4.2 Accesibilidad
      - 4.2.1 Accesibilidad de la Información
        - 4.2.1.1 Imagen con título
        - 4.2.1.2 Legibilidad Global
      - 4.2.2 Accesibilidad de ventanas
        - 4.2.2.1 Versión de Macros

---

### a) Tipo de Criterio Elemental

CVN:  $IE = (X/Y) * 100$  con  $X = \Sigma$  Puntaje Mximo  $Y = \Sigma$  Puntaje Obtenido

CB:  $IE = 0$  si no existe                       $IE = 1$  Si Existe

CPD: Sujeto a la Objetividad del Observador

CMN:  $IE = 0 \approx 0$  Ausente     $IE = 1 \approx 60$  Presencia Parcial             $IE = 2 \approx 100$  Presente

donde:

CVN = Criterio de variable Normalizada

CB = Criterio Binario

CPD = Criterio de Preferencia Directa

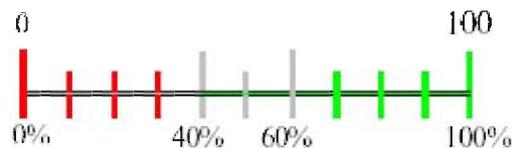
CMN = Criterio de Multi-Nivel Definido como Subconjunto

### b) Escala de Medicin de Aceptabilidad

Insatisfactorio (0 – 40) %

Aceptabilidad Marginal (40-60) %

Satisfactorio (60 - 100) %



**Figura 2.9** Escala de preferencia.

**Fuente:** Metodologa Cuantitativa para la Evaluacin y Comparacin de la Calidad de Sitios Web, 1999.

Por medio de la evaluacin de la calidad de artefactos Web, podemos comprender el grado de cumplimiento de un conjunto de caractersticas, subcaractersticas y atributos con respecto a los requerimientos de calidad establecidos.

### 3.1 Análisis de requisitos

#### 3.1.1 Descripción del problema

El modelo de requisitos comienza con la descripción del problema como se explica en la sección 2.1.4.1-a

Control Financiero del Servicio Exterior es el departamento encargado de revisar la rendición de cuentas efectuadas por cada una de las Misiones establecidas en el exterior, dado que se cuenta con un número considerable de Misiones los cuales supera en número de cincuenta, al departamento se le hace difícil realizar las tareas de revisar, evaluar, verificar los saldos y las cuentas correspondiente a cada una de las Misiones. Para facilitar dicho proceso la Dirección de Sistemas informáticos del Ministerio de Relaciones Exteriores y Cultos resolvió implementar un sistema el cual permita al responsable contable de la Misión llenar los diferentes formularios que se cuentan para registrar los ingresos y egresos que se efectúan diariamente vía Internet para contar con estos datos de manera oportuna en el Ministerio. Además el Jefe de Misión como responsable ante el Ministerio debe tener el control de las transacciones a realizar.

Cada una de las transacciones efectuadas por los responsables contables debe verse reflejada para el personal del Departamento de Control Financiero en forma de reportes de esta manera el personal de Control Financiero podrá llevar a cabo las tareas encomendadas de mejor manera

### 3.1.2 Requisitos candidatos

Pasamos a describir los requisitos candidatos encontrados (ver 2.1.4.1-b).

Nombre	Definición	Estado	Costo	Prioridad	Riesgo
Base de datos centralizada	Datos ubicados en un mismo lugar.	A	M	C	S
Bitácora	Registrar el ingreso de los usuarios.	A	M	I	S
Cuenta	Depósito de dinero en una entidad financiera.	I	B	I	O
Estructura en capas	Organización lógica del sistema.	A	A	C	C
Facilitar reportes	Elaborar reportes en corto tiempo.	I	M	I	O
Foros para usuarios	Ayuda extra brindada luego de la implementación del sistema.	P	A	I	O
Información oportuna	Información accesible en cualquier momento.	I	M	C	S
Interrelación entre módulos	Comunicación entre módulos.	A	A	C	C
Log	Registrar las operaciones realizadas por los usuarios.	A	M	I	S
Mantenimiento	Conjunto de operaciones y/o cuidados necesarios para un adecuado funcionamiento.	I	B	I	O
Misión	Embajadas, Consulados, representaciones permanentes o diplomáticas.	I	B	S	O
Notificar	Hacer saber a una persona algo de su interés.	I	B	S	O
País	Territorio que forma una unidad geográfica o política.	I	B	S	O
Partidas	Clasificador contable.	I	B	I	S
Transacción	Convenio comercial; particularmente de compra y venta.	I	B	I	O
Uniformar	Hacer uniforme dos o más cosas.	A	M	I	O
Usuario	Persona que usa el sistema.	I	B	I	O

**Estado-** Propuesto (P), Aprobado (A), Incluido (I).

**Costo-** Alto(A), Medio (M), Bajo (B).

**Prioridad-** Crítico (C), Importante (I), Secundario (S).

**Riesgo-** Crítico (C), Significativo (S), Ordinario (O).

**Tabla 3.1** Requisitos candidatos.

### 3.1.3 Modelado del dominio

El siguiente diagrama de clases (ver fig.3.1) nos muestra las entidades más importantes del contexto del sistema, entre los cuales podemos destacar la clase Misión, Responsable contable y transacción, este diagrama se puede considerar como el diagrama de clases muy preliminar (ver 2.1.4.1-c).

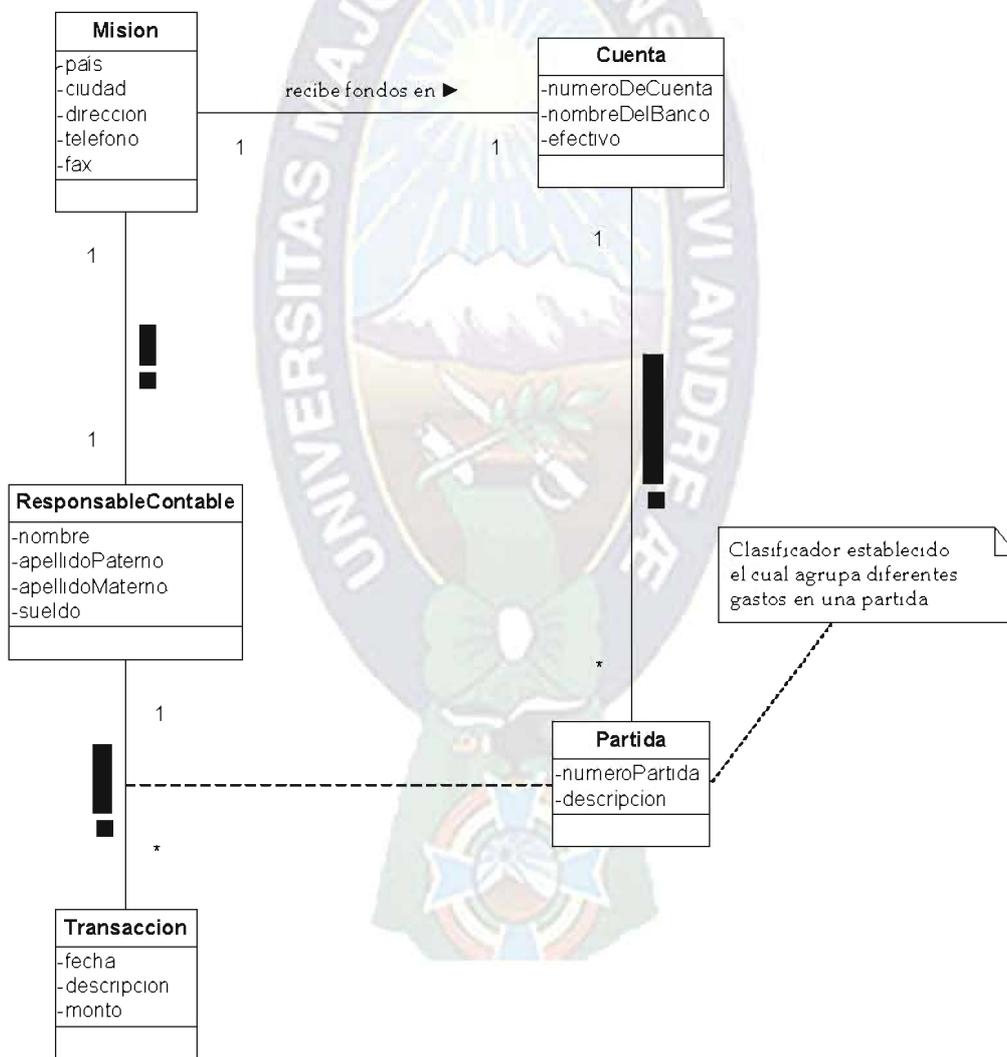


Figura 3.1 Diagrama de clases del modelado del dominio.

### 3.1.4 Modelado del negocio

La figura 3.2 y la figura 3.3 nos muestra los procesos de negocio referente al control financiero (ver 2.1.41-d).

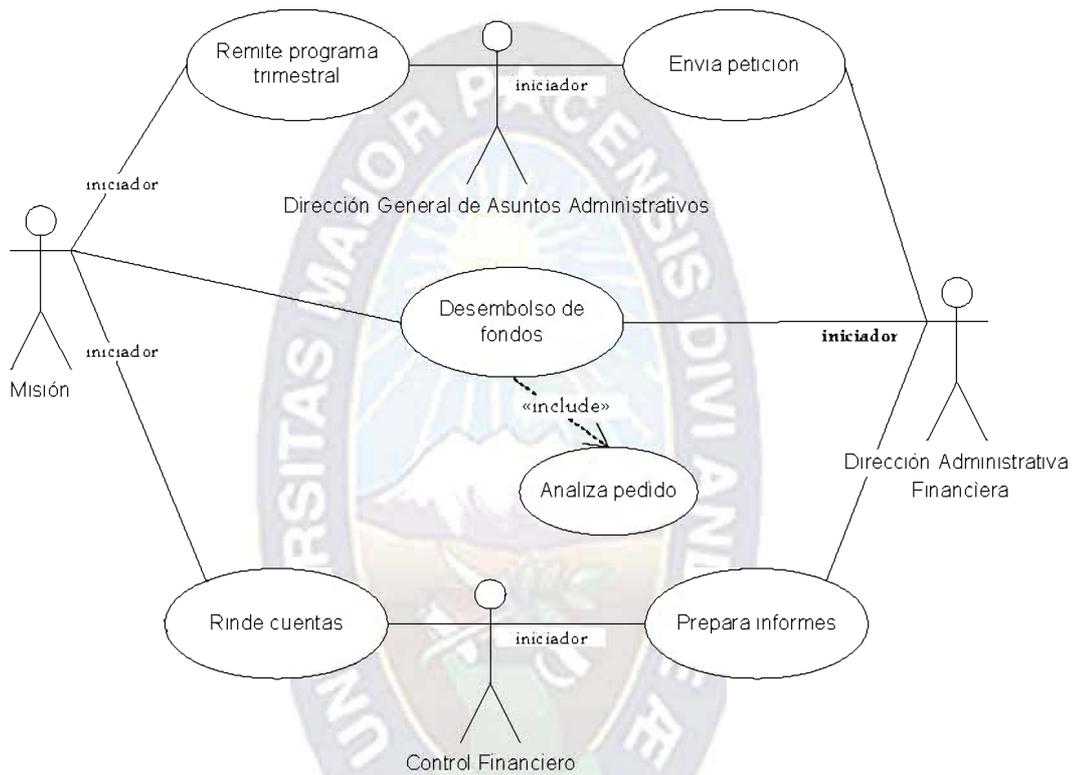


Figura 3.2 Diagrama de casos de uso del modelado del negocio.

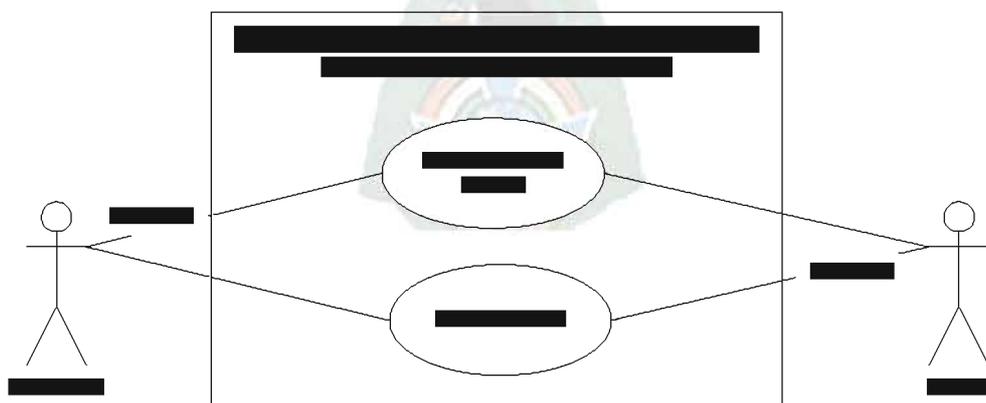
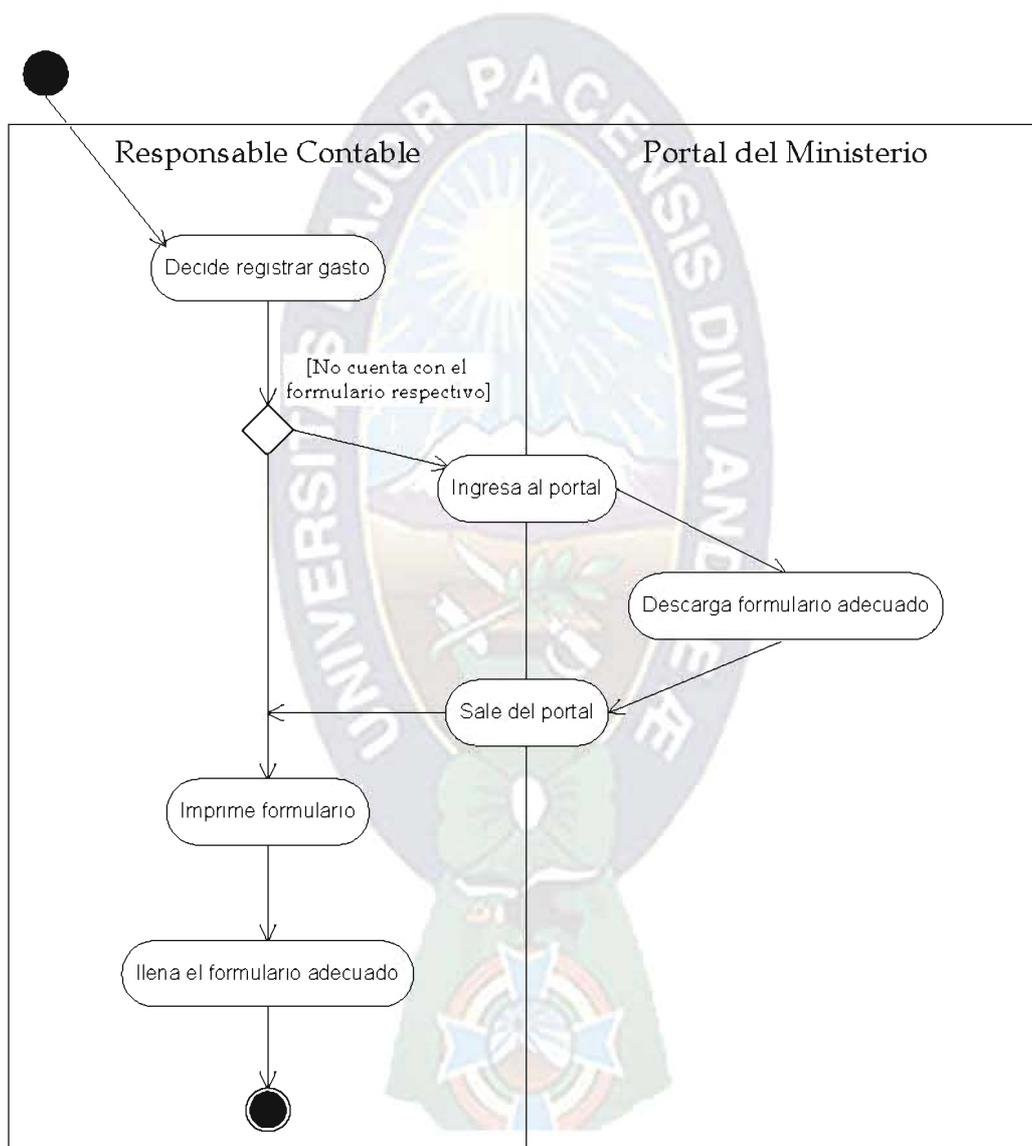


Figura 3.3 Representación del modelo de caso de uso.

Diagrama de actividades que describe como el responsable contable, el cual es un funcionario de alguna Misión en el exterior interactúa con el portal del Ministerio.



**Figura 3.4** Casos de uso del Sistema de información para la rendición de cuentas.

---

### 3.1.5 Requisitos adicionales. (ver 2.1.4.1-e)

**a) Requisitos de interfaz.**- El sistema provee la comunicación, pues este cuenta con la capacidad de interactuar con los otros módulos.

#### **b) Requisitos físicos**

**Servidor** – compatible con Windows Server 2003.

**Cliente** – PC compatible.

#### **c) Restricción de diseño**

El sistema debe ser diseñado por capas, separando principalmente las capas de acceso a datos y la lógica del negocio, además que se debe de contemplar el uso de controles de usuario.

#### **d) Restricciones de implementación**

##### **Software del sistema**

Sistema Operativo de servidor: Windows Server 2003

Sistema Operativo de cliente: Windows XP

##### **Software Para Internet**

Microsoft Internet Explorer 6.0

#### **e) Seguridad**

La transmisión debe ser segura, entendiéndose por esto que solo las personas autorizadas pueden tener acceso al sistema. Las personas autorizadas son los responsables contables de las Misiones y los administradores del sistema.

#### **f) Disponibilidad**

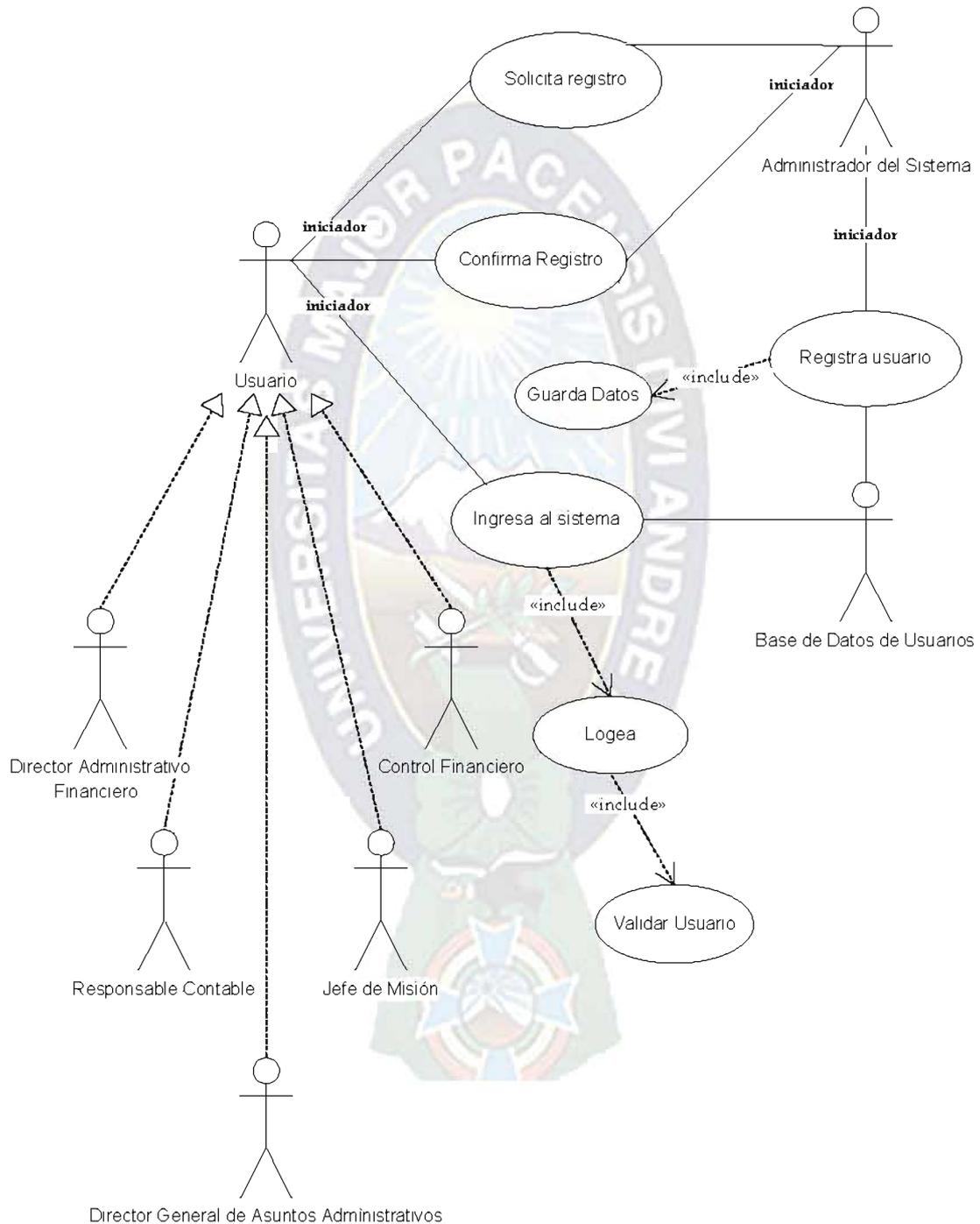
El sistema debe de estar en lo posible siempre disponible pues los usuarios se encuentran en diferentes países con horarios variados.

#### **g) Facilidad de aprendizaje**

El sistema debe ser comprensible pues los usuarios se encuentran en el exterior del país a los cuales será difícil el llegar a capacitarlos personalmente.

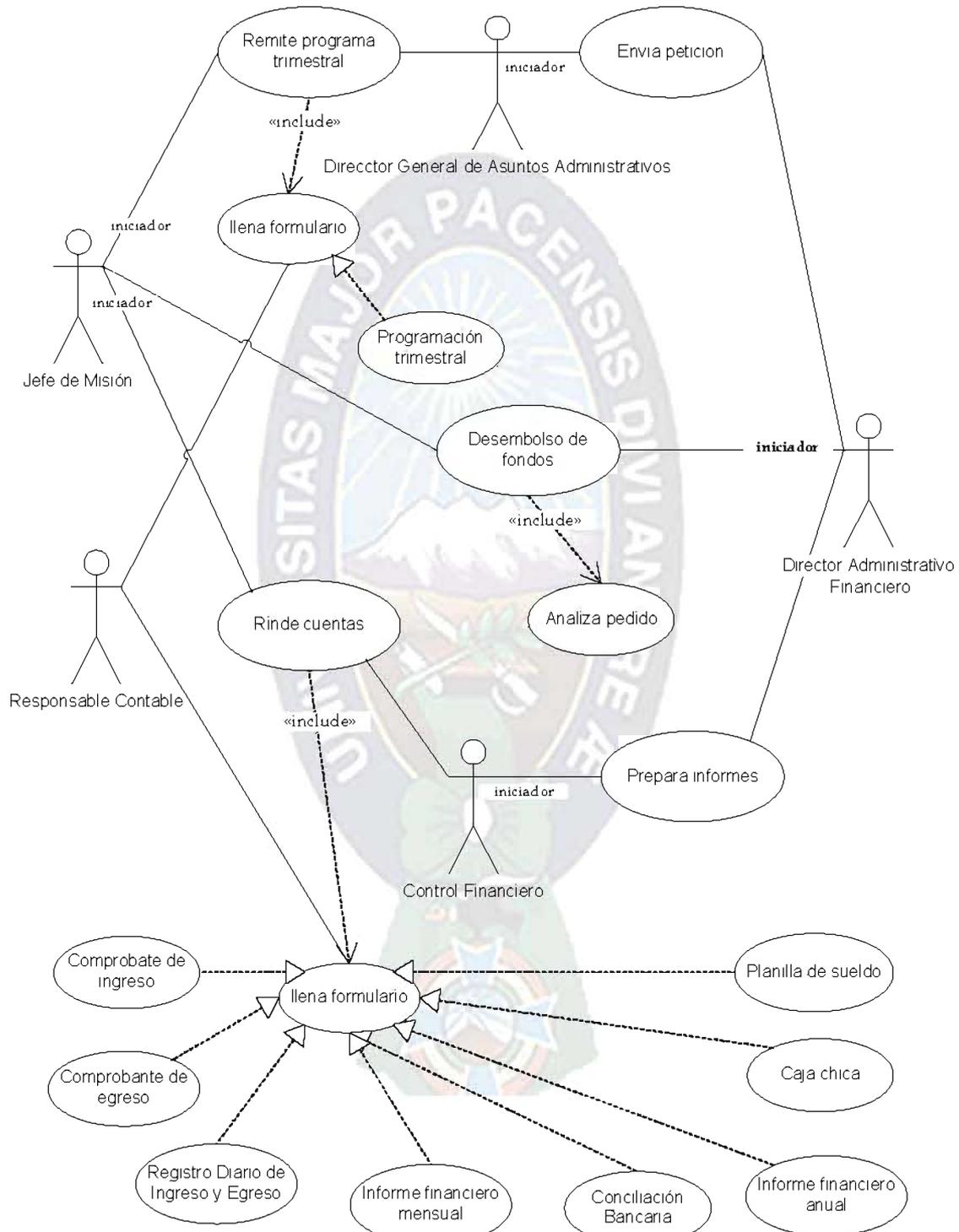
### 3.1.6 Modelo de casos de uso

Como se explicó en la sección 2.1.1.1 RUP es un proceso dirigido por casos de uso, las figuras 3.5 y 3.6 nos muestra los casos de uso encontrados.



**Figura 3.5** Casos de uso del Sistema (registro e ingreso de usuarios).

## Casos de uso del sistema



**Figura 3.6** Casos de uso del Sistema (transacciones de cuentas).

### 3.1.6.1 Actores encontrados

Los actores identificados en los casos de uso de las figuras 3.5 y 3.6 son: Usuario, Responsable Contable, Jefe de Misión, Administrador del Sistema, Base de Datos de Usuarios, Director General de Asuntos Administrativos, Director Administrativo Financiero y Control Financiero, los cuales pasamos a describirlos según la tabla 2.2 (ver modelo de casos de uso 2.1.4.1-f).

Las siguientes tablas (tabla 3.2 hasta la tabla 3.9) describen los actores encontrados.

<b>Actor:</b>	Usuario.
<b>Casos de uso:</b>	Solicita Registro, Ingresar al sistema, Logea, Remite programa trimestral, Llena formulario, Rinde cuentas.
<b>Tipo:</b>	Principal.
<b>Descripción:</b>	Un usuario es en general toda persona que interactúa con el sistema. Tenemos como usuarios a Responsable Contable, Jefe de Misión, etc.

**Tabla 3.2** Descripción del actor Usuario

<b>Actor:</b>	Responsable Contable.
<b>Casos de uso:</b>	Solicita registro, Ingresa al sistema, Llena formulario.
<b>Tipo:</b>	Principal.
<b>Descripción:</b>	Es la persona responsable de realizar el registro de las transacciones realizadas por la Misión. El Responsable Contable puede ser una persona designada por el Jefe de Misión. Tareas asignadas: <ul style="list-style-type: none"><li>• Registrar los fondos de las cuentas bancarias de la misión.</li><li>• Registrar las operaciones en los formularios respectivos.</li><li>• Remitir documentación de descargo.</li><li>• Elaborar cheques y recibos.</li><li>• Realizar conciliaciones bancarias.</li><li>• Elaborar inventarios.</li><li>• Elevar informes de gestión.</li><li>• Elaborar cuadros comparativos.</li><li>• Remitir mensualmente las rendiciones de cuentas.</li><li>• Informar al Jefe de Misión en la elaboración y ejecución presupuestaria, velando por la correcta aplicación y apropiación de las partidas.</li></ul>

**Tabla 3.3** Descripción del actor Responsable Contable.

<b>Actor:</b>	Jefe de Misión
<b>Casos de uso:</b>	Solicita registro, Ingresa al sistema, Remite programa trimestral.
<b>Tipo:</b>	Principal.
<b>Descripción:</b>	Es la persona responsable de la Misión.

**Tabla 3.4** Descripción del actor Jefe de Misión

<b>Actor:</b>	Administrador del Sistema.
<b>Casos de uso:</b>	Registra usuario, Confirma registro.
<b>Tipo:</b>	Principal.
<b>Descripción:</b>	Es la persona encargada del adecuado funcionamiento del sistema

**Tabla 3.5** Descripción del actor Administrador del Sistema.

<b>Actor:</b>	Base de Datos de Usuarios.
<b>Casos de uso:</b>	Valida usuario, Guarda datos.
<b>Tipo:</b>	Secundario.
<b>Descripción:</b>	Es un actor secundario y representa a la base de datos donde se guarda toda la información relacionada con los usuarios registrados para el ingreso al sistema.

**Tabla 3.6** Descripción del actor Base de Datos Usuario.

<b>Actor:</b>	Director General de Asuntos Administrativos.
<b>Casos de uso:</b>	Envía petición
<b>Tipo:</b>	Secundario.
<b>Descripción:</b>	El Director de Asuntos Administrativos se encarga de recibir la solicitud presupuestaria hecha por las Misiones para luego poner a consideración del Director Administrativo Financiero.

**Tabla 3.7** Descripción del actor Director General de Asuntos Administrativos

<b>Actor:</b>	Director Administrativo Financiero.
<b>Casos de uso:</b>	Desembolso de fondos.
<b>Tipo:</b>	Secundario.
<b>Descripción:</b>	El Director Administrativo Financiero es la persona encargada de analizar y dar curso a la petición hecha por las Misiones, dicha petición puede ser aprobada o denegada en función del techo presupuestario estimado, luego se informa al Director General de Asuntos Administrativos.

**Tabla 3.8** Descripción del actor Director Administrativo Financiero.

<b>Actor:</b>	Control Financiero.
<b>Casos de uso:</b>	Principal.
<b>Tipo:</b>	Prepara Informes.
<b>Descripción:</b>	Control Financiero es el departamento encargado de verificar las cuentas enviadas por las misiones para luego enviar un informe de conformidad u observación al Director Administrativo Financiero.

**Tabla 3.9** Descripción del actor Control Financiero.

### 3.1.6.2 Casos de uso encontrados

Sin duda que los casos de uso al igual que los actores deben documentarse, como se mencionó en la sección 2.1.4.1-f es así que pasamos a describir los casos de usos según la tabla 2.3 que hace referencia a las figuras anteriores (figura 3.5 y figura 3.6)

Las siguientes tablas (tabla 3.10 hasta la tabla 3.27) describen los casos de uso encontrados.

<b>Casos de uso:</b>	Solicita registro.
<b>Actores:</b>	Usuario (Responsable contable, Jefe de Misión , Director general de asuntos administrativos, Director administrativo financiero), Administrador del sistema
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Solicita ser registrado en la base de datos de usuarios.
<b>Descripción:</b>	El usuario solicita ser registrado en la base de datos, requisito indispensable para ingresar al sistema.
<b>Precondiciones:</b>	Ingresar al portal.
<b>Flujo principal:</b>	1.- El usuario ingresa al portal. 2.- El usuario selecciona la opción "Registro". 3.- El usuario llena el formulario con sus datos. 4.- Finalmente envía los datos introducidos.
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Solicitud efectuada.

**Tabla 3.10** Descripción del caso de uso Solicita registro.

<b>Casos de uso:</b>	Confirma registro.
<b>Actores:</b>	Administrador del sistema, Usuario.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Dar respuesta a la solicitud presentada.
<b>Descripción:</b>	El Administrador del sistema revisa la solicitud de registro y

	responde a la solicitud.
<b>Precondiciones:</b>	Contar con una solicitud pendiente.
<b>Flujo principal:</b>	1.- El Administrador del sistema revisa las solicitudes pendientes. 2.- Consulta dicha solicitud con el departamento de recursos humanos. 3.- Responde a la solicitud.
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	En el paso 2 puede que no se trate de un funcionario aceptado en el departamento de recursos humanos, en dicho caso se le informa al usuario que su solicitud ha sido rechazada.
<b>Post Condiciones:</b>	Usuario registrado en la base de datos.

**Tabla 3.11** Descripción del caso de uso Confirma registro.

<b>Casos de uso:</b>	Registrar usuario.
<b>Actores:</b>	Administrador del sistema, Base de datos de usuarios.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Registrar a un usuario dentro la base de datos.
<b>Descripción:</b>	Graba en la Base de datos los datos del usuario que pidió ser registro.
<b>Precondiciones:</b>	Aceptación por parte de recursos humanos.
<b>Flujo principal:</b>	1.- El Administrador del sistema ingresa al sistema.. 2.- Graba los datos en la Base de datos de usuarios.
<b>Subflujos:</b>	Guardar datos.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Registro de un nuevo usuario.

**Tabla 3.12** Descripción del caso de uso Registrar usuario.

<b>Casos de uso:</b>	Guarda datos.
<b>Actores:</b>	Administrador del sistema.
<b>Tipo:</b>	Inclusión.
<b>Propósito:</b>	Registrar datos.
<b>Descripción:</b>	Guardar en la base de datos.
<b>Precondiciones:</b>	Se requiere haber ejecutado anteriormente el caso de uso Registrar usuario.
<b>Flujo principal:</b>	1.- Recibe los datos enviados. 2.- Base de datos de usuarios registra datos.
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Datos grabados.

**Tabla 3.13** Descripción del caso de uso Guardar datos.

<b>Casos de uso:</b>	Ingresar al sistema.
<b>Actores:</b>	Usuario, Base de datos de usuarios.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Ingresar al sistema para poder realizar tareas.
<b>Descripción:</b>	El usuario ingresa sus datos para poder logearse en el sistema.
<b>Precondiciones:</b>	Logearse
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla dos campos para ser llenado con los datos correspondientes. 2.- La información llega a la Base de datos de usuarios vía Internet. 3.- Logear usuario. 4.- Se presenta al usuario en pantalla las opciones.
<b>Subflujos:</b>	Logea.
<b>Excepciones:</b>	En el paso 4 datos incorrectos ocasionan que no se le presente ninguna opción, sino que vuelve a la pantalla inicial enviando un mensaje que los datos introducidos no son correctos.
<b>Post Condiciones:</b>	Usuarios dentro del sistema.

**Tabla 3.14** Descripción del caso de uso Ingresar al sistema.

<b>Casos de uso:</b>	Logea.
<b>Actores:</b>	Usuario, Base de datos de usuarios.
<b>Tipo:</b>	Inclusión.
<b>Propósito:</b>	Logear al usuario.
<b>Descripción:</b>	Identificar al usuario mediante los datos ingresados.
<b>Precondiciones:</b>	Se requiere haber ejecutado anteriormente el caso de uso ingresar al sistema.
<b>Flujo principal:</b>	1.- En la Base de datos de usuarios valida los datos enviados.
<b>Subflujos:</b>	Validar.
<b>Excepciones:</b>	Ninguna.
<b>Post Condiciones:</b>	Usuario logeado.

**Tabla 3.15** Descripción del caso de uso Logea.

<b>Casos de uso:</b>	Validar usuario.
<b>Actores:</b>	Base de datos de usuarios.
<b>Tipo:</b>	Inclusión.
<b>Propósito:</b>	Validar los datos enviados por el usuario.
<b>Descripción:</b>	Verificar los datos del usuario y comprobar realmente quien dice ser.
<b>Precondiciones:</b>	Se requiere haber ejecutado anteriormente el caso de uso logea.
<b>Flujo principal:</b>	1.- Comparar los datos enviados con la registrada en la Base de datos de usuarios
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Usuario identificado.

**Tabla 3.16** Descripción del caso de uso Validar usuario.

<b>Casos de uso:</b>	Remite programa trimestral.
<b>Actores:</b>	Responsable contable, Jefe de Misión, Dirección general de asuntos administrativos.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Remitir el programa trimestral para gastos de funcionamiento y/o representación de la Misión, información indispensable sin la cual no procederá el desembolso.
<b>Descripción:</b>	El jefe de Misión remite el formulario a la Dirección general de asuntos administrativos el cual posteriormente envía dicha solicitud a la Dirección administrativa financiera para dar curso a dicha petición.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Jefe de Misión. Se requiere haber llenado el formulario correspondiente.
<b>Flujo principal:</b>	1.- Se presenta al Jefe de Misión en pantalla el formulario llenado previamente por el Responsable contable. 2.- Se aceptan los montos estimados para luego ser registrados en la base de datos. 3.- Se graba los datos en la base de datos. 4.- El Director general de asuntos administrativos recibe la programación trimestral.
<b>Subflujos:</b>	Llena formulario.
<b>Excepciones:</b>	Ninguna.
<b>Post Condiciones:</b>	Información presupuestaria grabada en la base de datos.

**Tabla 3.17** Descripción del caso de uso Remite programa trimestral.

<b>Casos de uso:</b>	Llena formulario.
<b>Actores:</b>	Responsable contable.
<b>Tipo:</b>	Inclusión.
<b>Propósito:</b>	Llenar correctamente los diferentes formularios presentados.
<b>Descripción:</b>	El Responsable contable de la Misión llena los diferentes formularios establecidos que el caso así lo amerite.
<b>Precondiciones:</b>	Se requiere haber ejecutado anteriormente el caso de uso Remite programa trimestral. Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere que el usuario seleccione el formulario a usar.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla los diferentes formularios. 2.- El Responsable contable elige el formulario correspondiente para registrar la transacción. 3.- Ingresa los datos necesarios para el llenado. 4.- Graba los datos.
<b>Subflujos:</b>	Programación Trimestral.
<b>Excepciones:</b>	Ninguna
<b>Post Condiciones:</b>	Formulario llenado correctamente.

**Tabla 3.18** Descripción del caso de uso Llena formulario.

<b>Casos de uso:</b>	Programación trimestral.
<b>Actores:</b>	Responsable contable o Jefe de Misión.
<b>Tipo:</b>	Generalización.
<b>Propósito:</b>	Programar los gastos previstos para el funcionamiento de la Misión.
<b>Descripción:</b>	El Responsable contable o el Jefe de Misión realizan una programación de los gastos necesarios para cumplir con el objetivo asignado.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere haber elegido el formulario programación trimestral para ejecución presupuestaria.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla el formulario programación trimestral para ejecución presupuestaria. 2.- El Responsable contable debe ingresar los meses a los que corresponde la programación trimestral. 3.- Se debe llenar los montos por partida para cada mes. 4.- Se debe grabar los datos introducidos.
<b>Subflujos:</b>	Ninguno
<b>Excepciones:</b>	Ninguno
<b>Post Condiciones:</b>	Formulario de programación trimestral llenado correctamente.

**Tabla 3.19** Descripción del caso de uso Programación trimestral.

<b>Casos de uso:</b>	Desembolsar fondos.
<b>Actores:</b>	Director administrativo financiero, Jefe de Misión.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Desembolsar fondos en la cuenta bancaria de la Misión.
<b>Descripción:</b>	La Dirección Administrativa Financiera hecho el estudio de la petición, luego de su aprobación hace el desembolso de los fondos en la cuenta bancaria perteneciente a la Misión
<b>Precondiciones:</b>	Se requiere contar con el dinero presupuestado. Se requiere logearse como Director administrativo financiero.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla la solicitud presupuestaria enviada por la Misión. 2.- Se desembolsa los montos solicitados. 3.- Se registra la respuesta.
<b>Subflujos:</b>	Analiza pedido.
<b>Excepciones:</b>	En el paso 2 puede que no se acepte los montos solicitados, esto sucede cuando el monto solicitado sobrepasa el techo máximo fijado para la Misión.
<b>Post Condiciones:</b>	Transferencia bancaria aprobada.

**Tabla 3.20** Descripción del caso de uso Desembolsar fondos.

<b>Casos de uso:</b>	Analizar pedido
<b>Actores:</b>	Director administrativo financiero.
<b>Tipo:</b>	Inclusión.
<b>Propósito:</b>	Estudiar la programación trimestral solicitada.
<b>Descripción:</b>	El Director administrativo financiero estudia la solicitud echa por la Misión, si se encuentra dentro del los montos fijados previamente viendo los techos que sé estimó para cada Misión.

<b>Precondiciones:</b>	Se requiere haber ejecutado anteriormente el caso de uso Desembolsar fondos.
<b>Flujo principal:</b>	1.- Se presenta al Director administrativo financiero en pantalla el formulario de solicitud hecha por la Misión. 2.- El Director administrativo financiero compara los montos solicitados con los previstos. 3.- Concluye si es factible.
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Respuesta a la solicitud.

**Tabla 3.21** Descripción del caso de uso Analizar pedido.

<b>Casos de uso:</b>	Rinde cuentas.
<b>Actores:</b>	Jefe de Misión, Control financiero
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Rendir cuentas de las transacciones efectuadas por la Misión.
<b>Descripción:</b>	La Misión rinde cuentas de los gastos de funcionamiento y/o representación de manera mensual al departamento de Control financiero, estos evalúan los descargos presentados por la Misión e informan a la Dirección de asuntos administrativos.
<b>Precondiciones:</b>	Fin de mes.
<b>Flujo principal:</b>	1.- Preparar documentación de descargo. 2.- Enviar documentación de descargo.
<b>Subflujos:</b>	Llena formulario.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Cuentas claras.

**Tabla 3.22** Descripción del caso de uso Rinde cuentas.

<b>Casos de uso:</b>	Prepara informe.
<b>Actores:</b>	Control financiero, Director administrativo financiero.
<b>Tipo:</b>	Básico.
<b>Propósito:</b>	Informar al Director administrativo financiero de la rendición de cuentas hecha por la Misión.
<b>Descripción:</b>	El Departamento de Control financiero elabora un informe para hacer conocer al Director administrativo financiero si se produjo alguna anomalía en la rendición de cuentas.
<b>Precondiciones:</b>	Revisión de los descargos presentados.
<b>Flujo principal:</b>	1.- El Departamento de Control financiero recibe los descargos. 2.- Revisa toda la documentación. 3.- elabora un informe de la revisión. 4.- Envía el informe al Director administrativo financiero.
<b>Subflujos:</b>	Ninguno.
<b>Excepciones:</b>	Ninguno.
<b>Post Condiciones:</b>	Informe de la rendición de cuentas

**Tabla 3.23** Descripción del caso de uso Prepara informe.

<b>Casos de uso:</b>	Comprobante de ingreso.
<b>Actores:</b>	Responsable contable o Jefe de Misión.
<b>Tipo:</b>	Generalización.
<b>Propósito:</b>	Registrar el comprobante de ingreso.
<b>Descripción:</b>	El Responsable contable o el Jefe de Misión realizan el llenado del comprobante de ingreso.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere haber elegido el formulario comprobante de ingreso.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla el formulario comprobante de ingreso. 2.- El Responsable contable debe ingresar la fecha en la que se produjo la transacción. 3.- Se debe llenar origen (TGN, otros ingresos), la partida, una descripción, el importe y número del cheque. 4.- Se debe grabar los datos introducidos.
<b>Subflujos:</b>	Ninguno
<b>Excepciones:</b>	Ninguno
<b>Post Condiciones:</b>	Formulario comprobante de ingreso llenado correctamente.

**Tabla 3.24** Descripción del caso de uso Comprobante de ingreso.

<b>Casos de uso:</b>	Comprobante de egreso.
<b>Actores:</b>	Responsable contable o Jefe de Misión.
<b>Tipo:</b>	Generalización.
<b>Propósito:</b>	Registrar el comprobante de ingreso.
<b>Descripción:</b>	El Responsable contable o el Jefe de Misión realizan el llenado del comprobante de egreso.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere haber elegido el formulario comprobante de egreso.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla el formulario comprobante de egreso. 2.- El Responsable contable debe ingresar la fecha en la que se produjo la transacción. 3.- Se debe llenar la partida, una descripción, el importe y número del cheque. 4.- Se debe grabar los datos introducidos.
<b>Subflujos:</b>	Ninguno
<b>Excepciones:</b>	Ninguno
<b>Post Condiciones:</b>	Formulario comprobante de egreso llenado correctamente.

**Tabla 3.25** Descripción del caso de uso Comprobante de egreso.

<b>Casos de uso:</b>	Caja chica.
<b>Actores:</b>	Responsable contable o Jefe de Misión.
<b>Tipo:</b>	Generalización.
<b>Propósito:</b>	Registrar caja chica.
<b>Descripción:</b>	El Responsable contable o el Jefe de Misión realizan el llenado de caja chica.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere haber elegido el formulario registro de caja chica.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla el formulario registro de caja chica. 2.- El Responsable contable debe ingresar la fecha, la partida, el número de factura, el beneficiario, el concepto y el importe. 3.- Se debe grabar los datos introducidos.
<b>Subflujos:</b>	Ninguno
<b>Excepciones:</b>	Ninguno
<b>Post Condiciones:</b>	Formulario registro de caja chica llenado correctamente.

**Tabla 3.26** Descripción del caso de uso Caja chica.

<b>Casos de uso:</b>	Planilla de sueldo.
<b>Actores:</b>	Responsable contable o Jefe de Misión.
<b>Tipo:</b>	Generalización.
<b>Propósito:</b>	Registrar la planilla de sueldos.
<b>Descripción:</b>	El Responsable contable o el Jefe de Misión realizan el llenado de la planilla de sueldos.
<b>Precondiciones:</b>	Se requiere haber ingresado al sistema. Se requiere logearse como Responsable contable o Jefe de Misión Se requiere haber elegido el formulario planilla de sueldos.
<b>Flujo principal:</b>	1.- Se presenta al usuario en pantalla el formulario planilla de sueldos. 2.- El Responsable contable debe ingresar el mes, nombre y apellido(s), de los funcionarios, rango o cargo, número de cheque y líquido pagable. 3.- Se debe grabar los datos introducidos.
<b>Subflujos:</b>	Ninguno
<b>Excepciones:</b>	Ninguno
<b>Post Condiciones:</b>	Formulario planilla de sueldos llenado correctamente.

**Tabla 3.27** Descripción del caso de uso Planilla de sueldos.

## 3.2 Análisis

### 3.2.1 Diagramas de interacción

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. En el análisis preferimos mostrar esto con *diagramas de colaboración* los cuales describimos a continuación (ver 2.1.4.2.3).

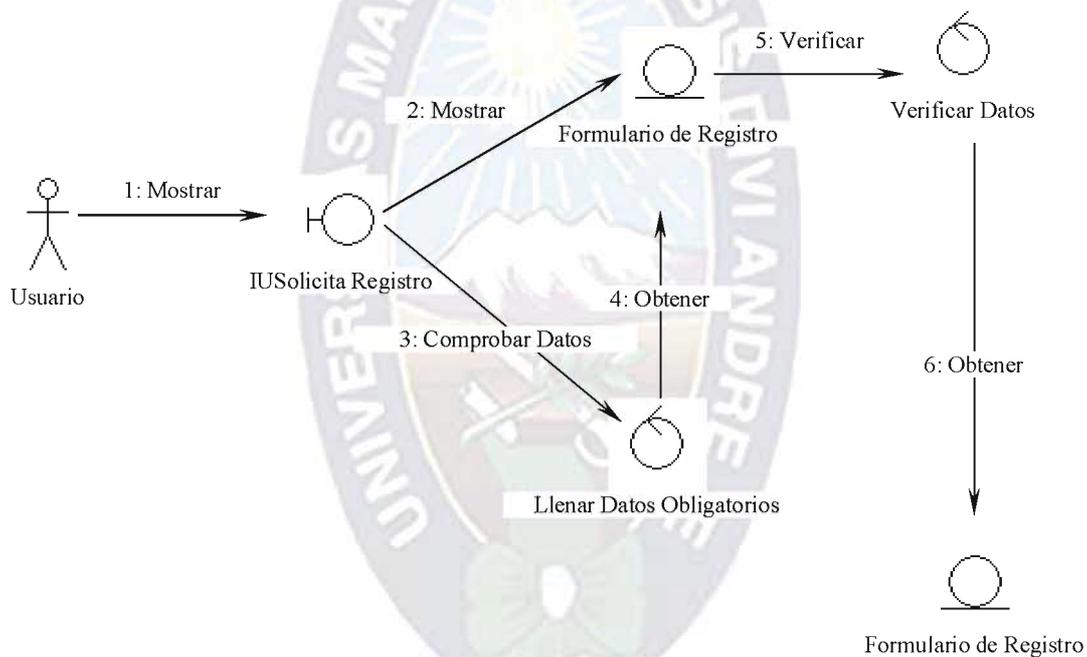
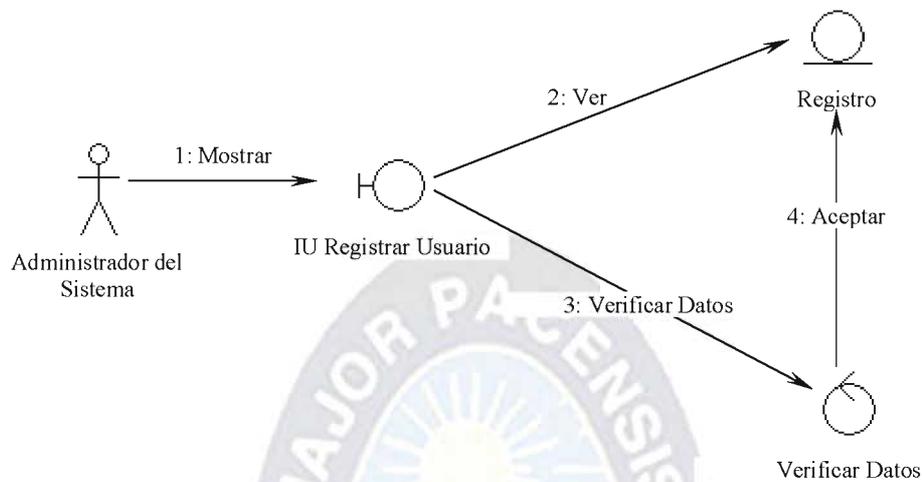


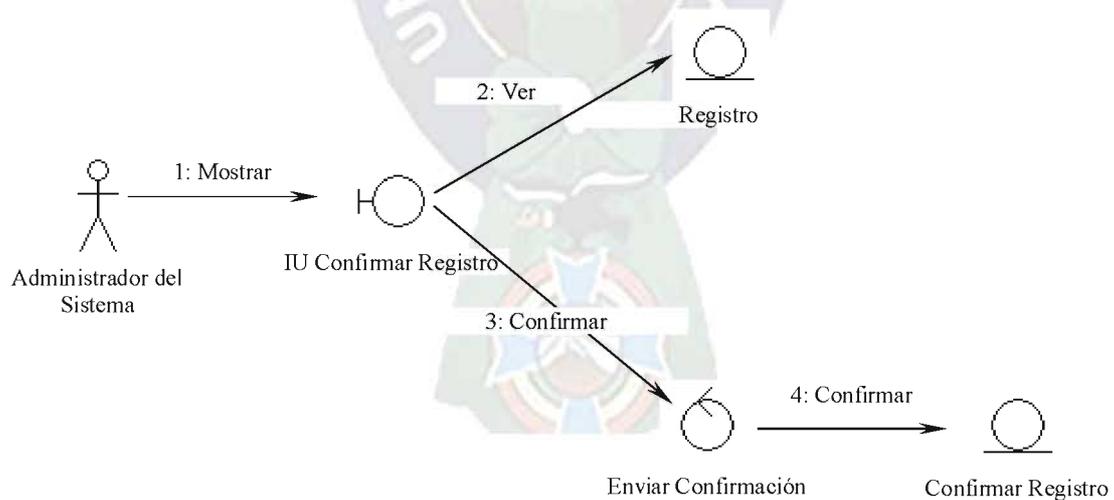
Figura 3.7 Diagrama de colaboración (Solicita Registro).

Un usuario cualquiera que visita el portal, pretende ser registrado para tener acceso a las diferentes opciones que puede brindar el sistema. El usuario llena un formulario en el cual se le solicita llenar sus datos personales de los cuales algunos son obligatorios (1, 2, 3), antes de registrar los datos se verifica si estos son correctos (4, 5). Finalmente los datos se registran para luego evaluar su solicitud (ver fig. 3.7).



**Figura 3.8** Diagrama de colaboración (Registra Usuario).

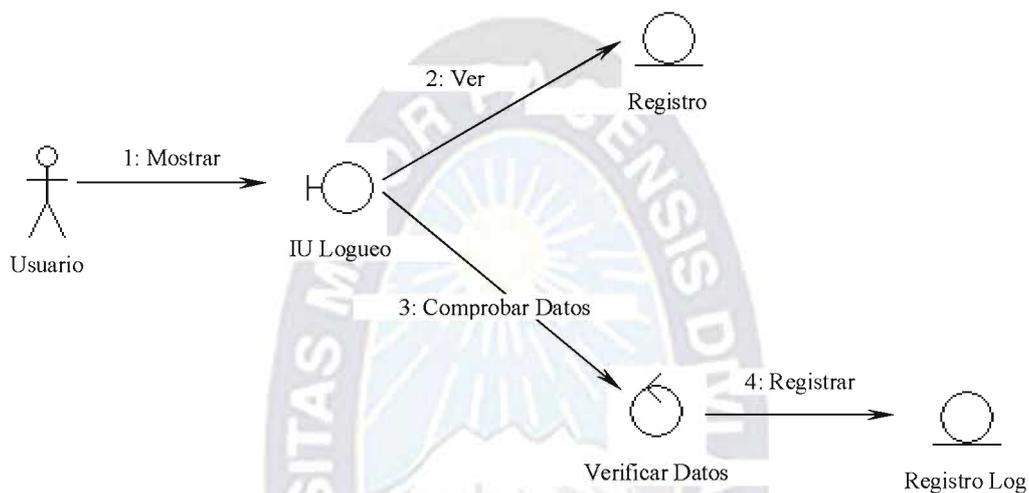
El Administrador del Sistema mediante una interfaz puede ver todos los registros enviados por las personas que desean ser registrados como usuarios del sistema (1, 2, 3), seguidamente el Administrador verifica si estos pueden ser registrados como usuarios (ver fig. 3.8).



**Figura 3.9** Diagrama de colaboración (Confirmar Registro).

---

El Administrador del Sistema una vez aceptada la solicitud para ser registrado en el sistema como usuario debe enviar una confirmación a quienes fueron aceptadas para que estos sean considerados como usuarios (ver fig. 3.9).

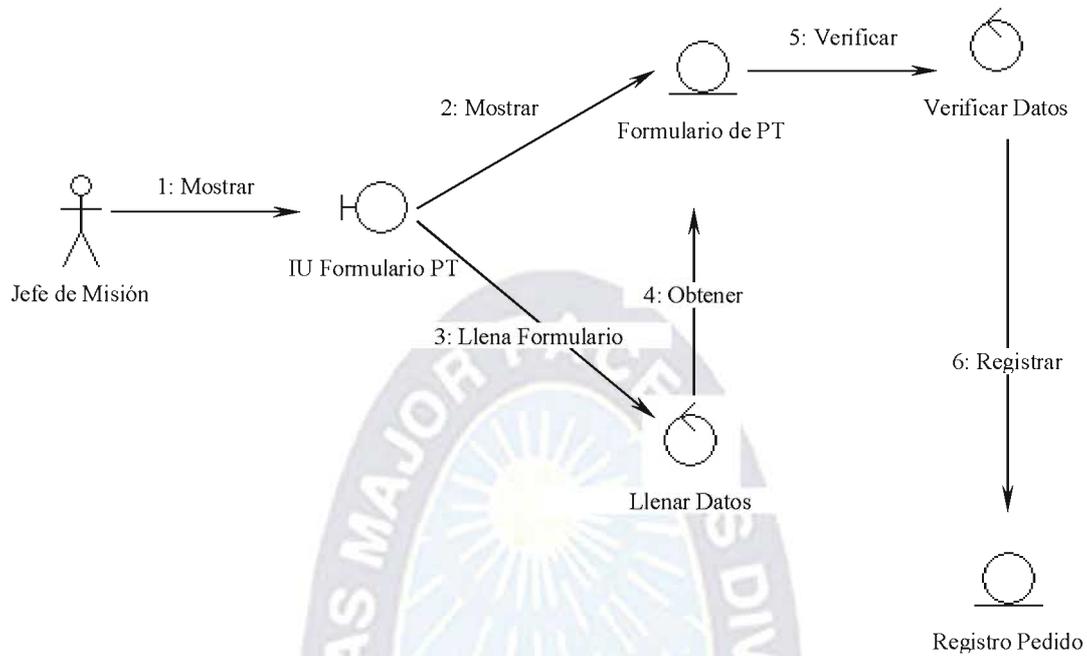


**Figura 3.10** Diagrama de colaboración (Ingresar al Sistema).

El Usuario ingresa dos datos a través de la interfaz de logueo (1), estos datos son buscados y verificados con el registro de los usuarios (2, 3), finalmente se crea un registro log el cual nos sirve para llevar un control de los usuarios ingresados en el tiempo.

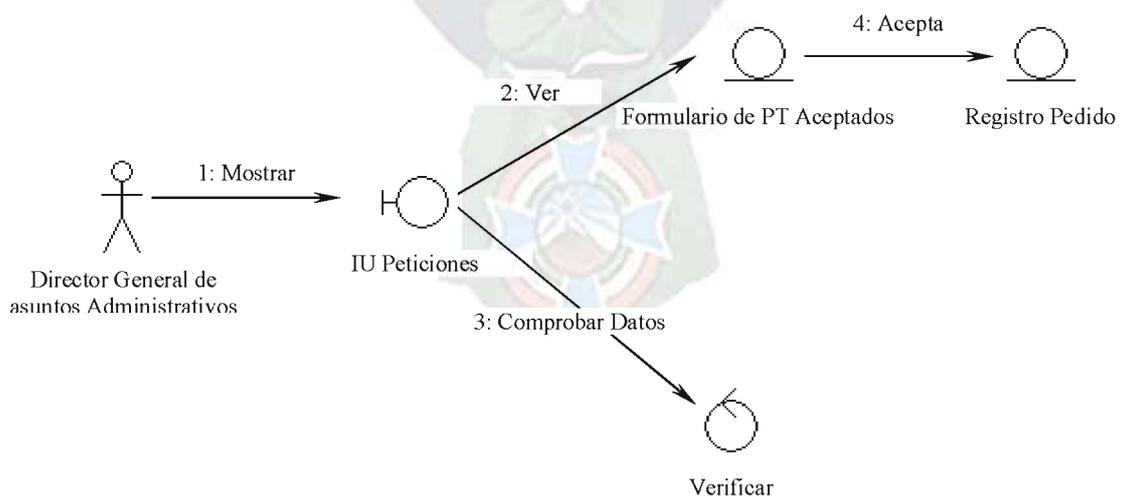
Si el usuario no pasa el proceso de verificación se le regresa a la interfaz en la cual se le pide nuevamente ingrese sus datos.

El resultado de buscar y verificar los datos del usuario nos devuelve los privilegios que tiene ese usuario, estos pueden ser como usuario administrador del sistema el cual tiene acceso total al sistema y quien fija datos macro con los cuales debe trabajar el sistema, usuario editor quien se encarga del llenado de los formularios, el último tipo de usuario es aquel que solo puede ver los datos sin la posibilidad de modificarlos (ver fig. 3.10).



**Figura 3.11** Diagrama de colaboración (Remite programa trimestral).

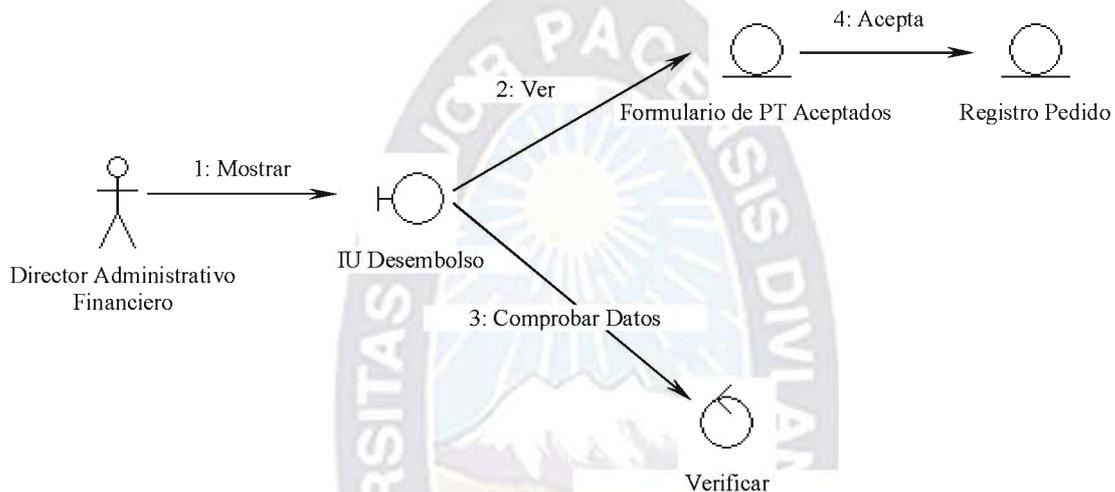
El Jefe de Misión llena los datos solicitados en el formulario (Programación Trimestral) desplegados en la interfaz (1, 2, 3, 4), una ingresados todos los datos solicitados estos son verificados para luego registrarse (5, 6) (ver fig. 3.11).



**Figura 3.12** Diagrama de colaboración (Envía Petición).

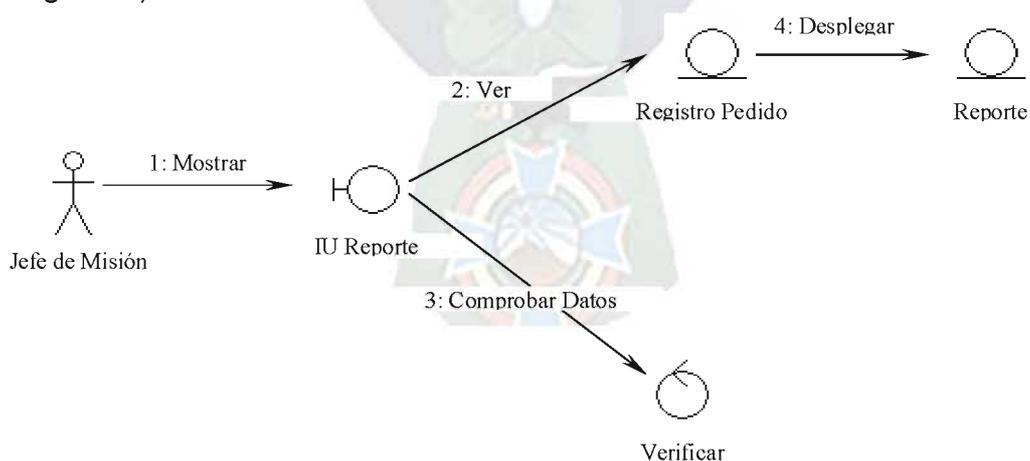
El Director General de Asuntos Administrativos examina las peticiones hechas por los diferentes usuarios (1, 2, 3), el cual luego de analizar estos son aceptados (4).

No todas las peticiones son aceptadas, en el caso de que ocurra esto simplemente se cancela la petición o se solicita que realice una nueva petición (ver fig. 3.12).



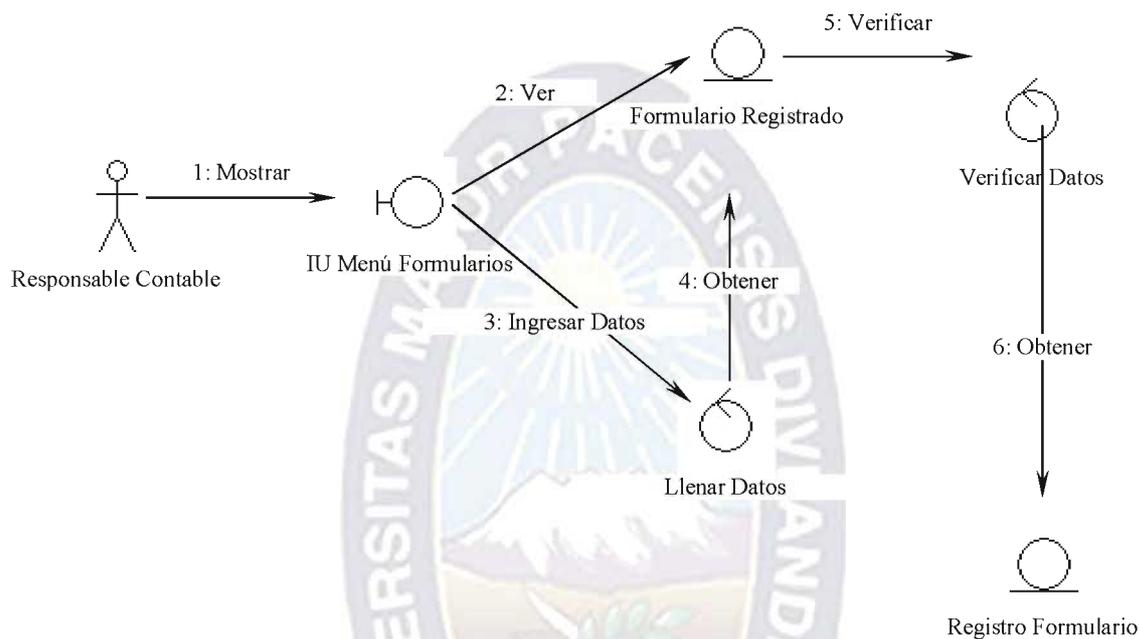
**Figura 3.13** Diagrama de colaboración (Desembolso de fondos).

El Director Administrativo Financiero examina las peticiones verificadas por el Director General de Asuntos Administrativos (1, 2, 3), para luego registrarlos (4) (ver fig. 3.13).



**Figura 3.14** Diagrama de colaboración (Rinde cuentas).

El Jefe de Misión se encarga de elaborar los reportes para ser presentados como informes de todas las actividades realizadas en los cuales se hizo uso de los fondos asignados (ver fig. 3.14).



**Figura 3.15** Diagrama de colaboración (Llena formulario).

Este es uno de los casos de uso más frecuentes en el cual el Responsable Contable o la persona encargada del llenado de los diferentes formularios interactúa con el menú de formularios el cual despliega el listado de los mismos (1), estos de acuerdo al formulario seleccionado es llenado y posteriormente verificado si los datos ingresados son adecuados para el tipo de formulario (2, 3, 4, 5), como ultimo paso se procede al registro del formulario.

Es importante mencionar que no todos los formularios serán llenados pues algunos vienen a ser un resumen de otros, los formularios a llenar son los de Comprobante de Ingreso, Comprobante de Egreso, Caja Chica y Planilla de Sueldos (ver fig. 3.15).

### 3.3 Diseño

#### 3.3.1 Diagramas de secuencia

Los diagramas correspondientes muestran la interacción entre los objetos participantes a nivel de eventos que se envían entre sí, excluyendo cualquier detalle interno de ellos (ver 2.1.4.3.2).

##### a) Solicita Registro

El diagrama de secuencia para Solicitar Registro de usuarios se muestra en la Figura 3.16. Este describe como un usuario hace una petición para ser tomado en cuenta y cuyos datos sean registrados en la Base de datos, para luego ser dado de alto en dicha petición.

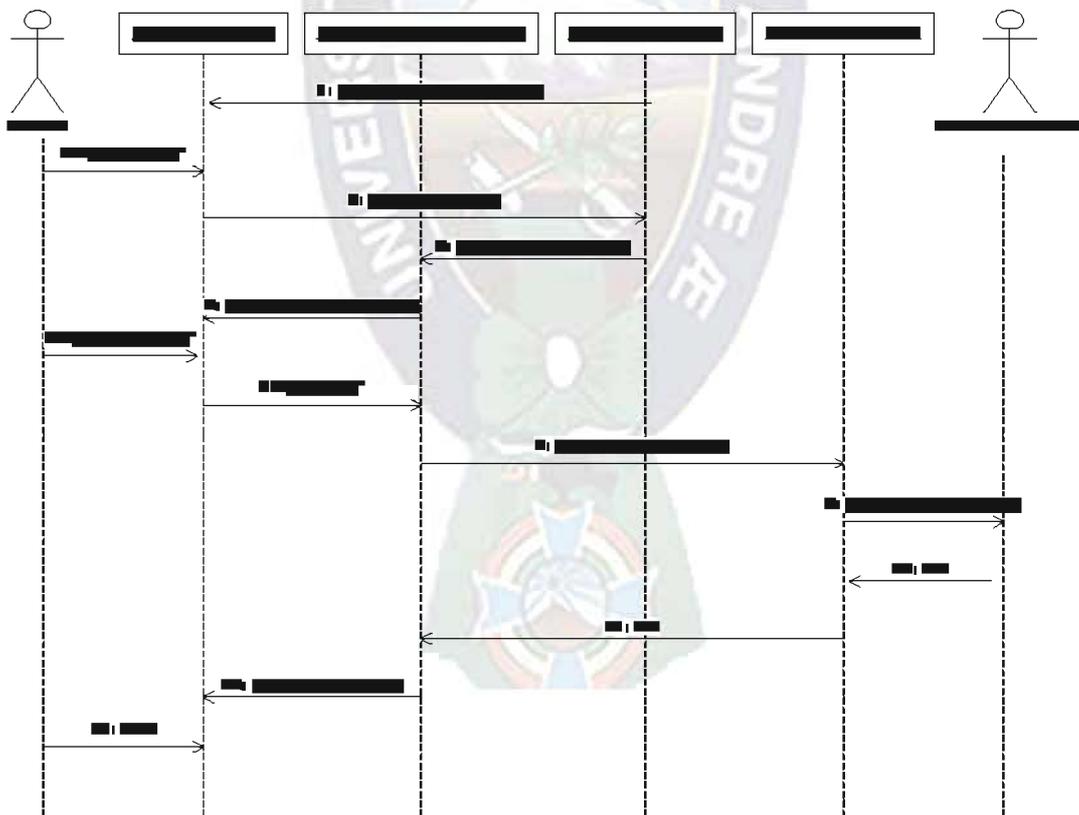


Figura 3.16 Diagrama de secuencia (Solicita Registro).

## b) Registro de Usuario

El diagrama de secuencia para Registro de usuarios se muestra en la Figura 3.17. Este describe como un usuario es dado de alta en la Base de datos, para que pueda ingresar como usuario normal a las opciones asignadas dado su rol.

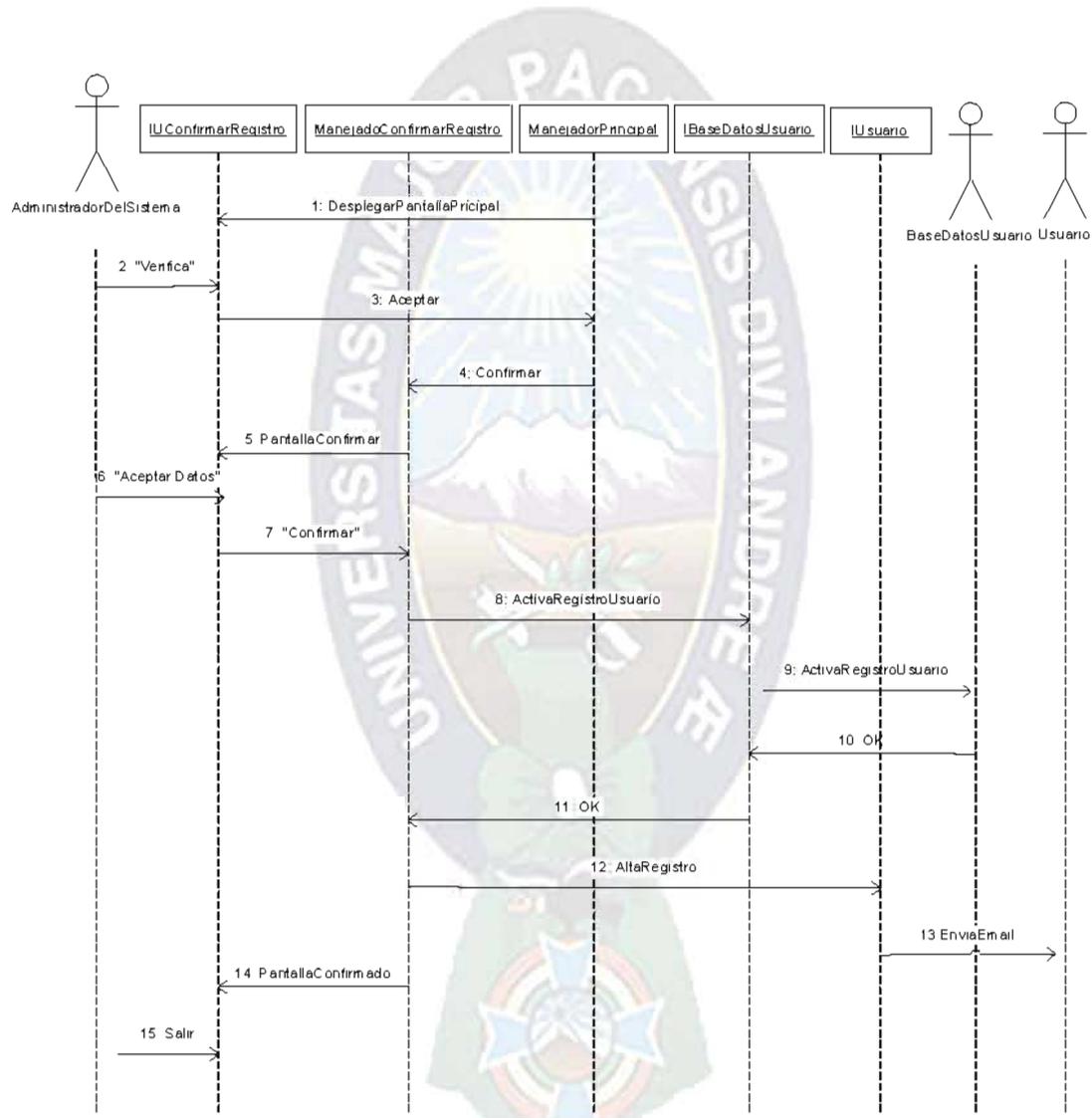
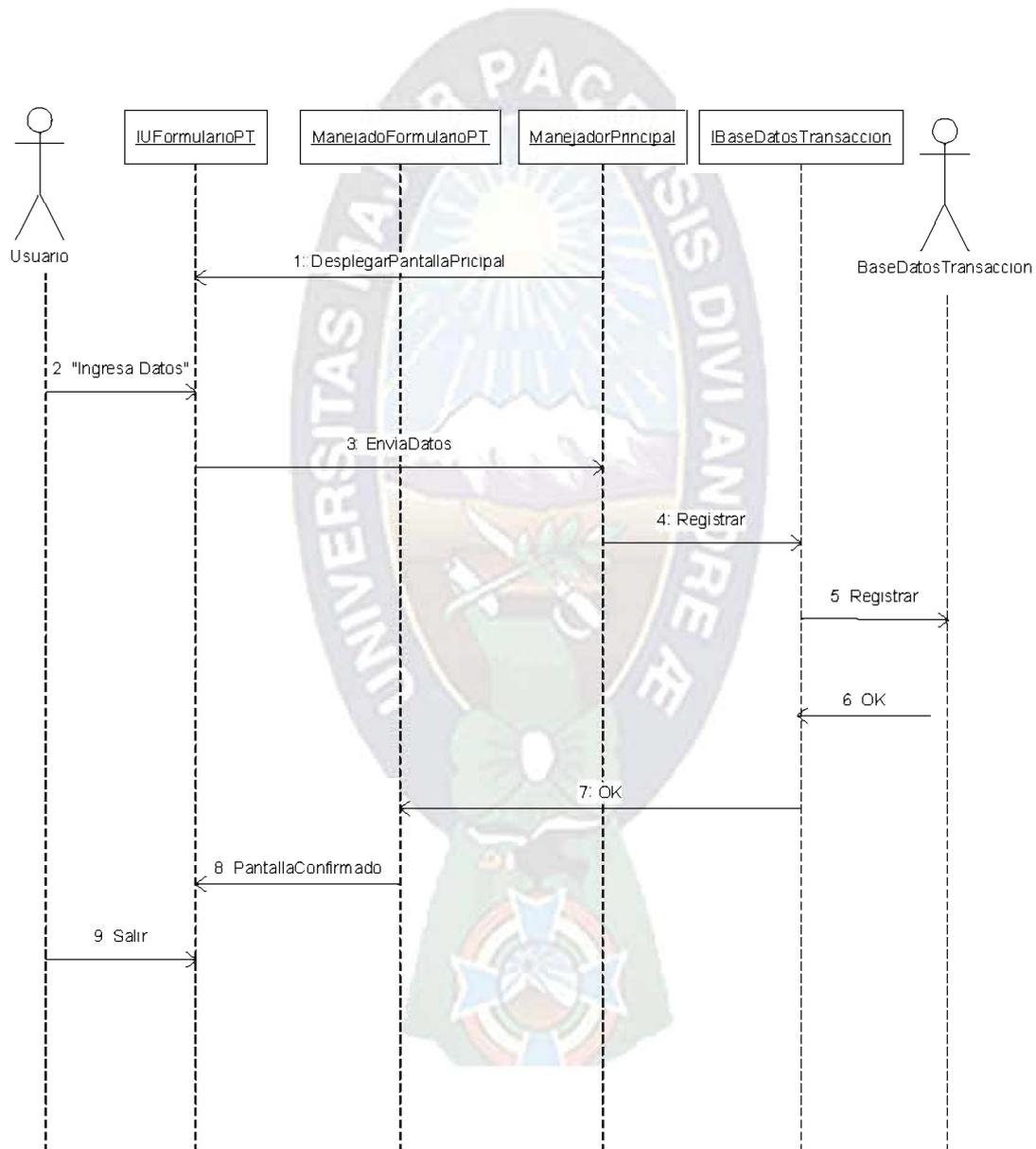


Figura 3.17 Diagrama de secuencia (Registro de Usuario).



#### d) Remitir Programa

El diagrama de secuencia para remitir programa se muestra en la Figura 3.19. Este describe como un usuario ingresa datos al sistema para luego ser registrados en la Base de datos y desplegar la pantalla de confirmación.



**Figura 3.19** Diagrama de secuencia (Remitir Programa).



### f) Llenar Formulario

El diagrama de secuencia para el llenado de formulario se muestra en la Figura 3.21. Este describe como un usuario llena todos los datos solicitados para ser registrados en la base de datos.

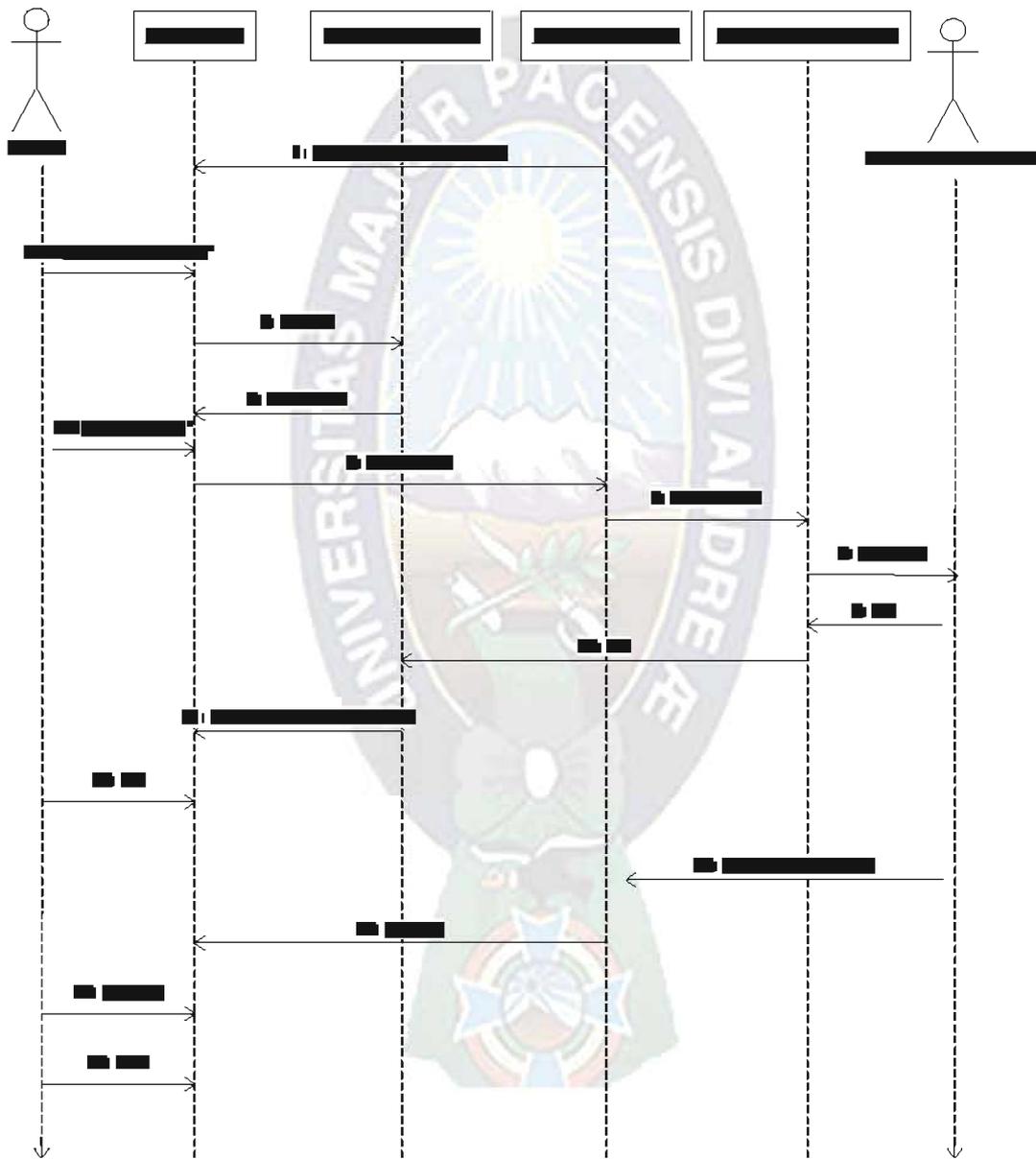


Figura 3.21 Diagrama de secuencia (Llenar Formulario).

### 3.3.2 Diagrama de clases - Control Financiero vía Web

La figura 3.22 muestra la representación de las clases encontradas y sus relaciones (ver 2.1.4.3.3).

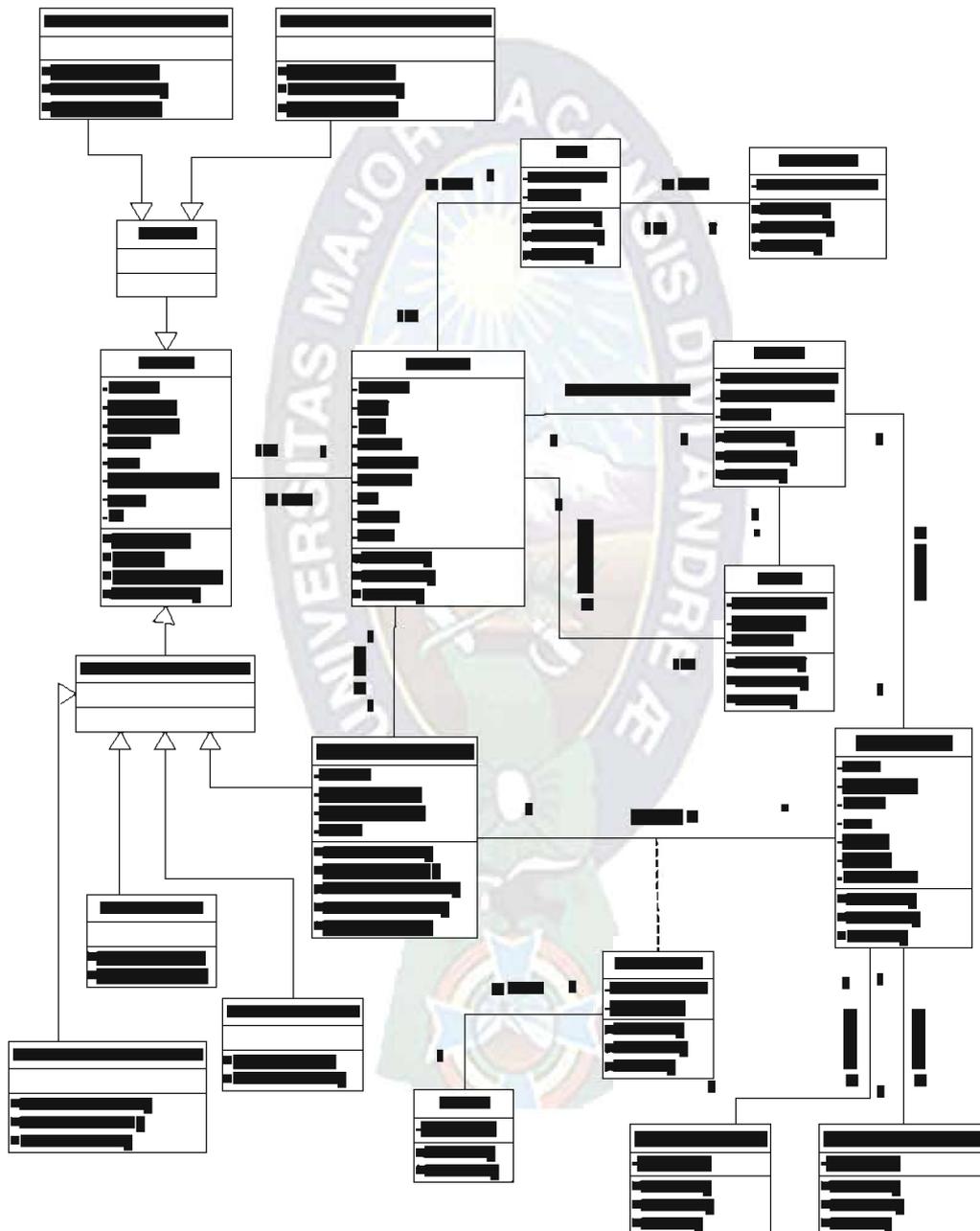


Figura 3.22 Diagrama de clases (Control Financiero vía Web).

### 3.3.3 Diagrama relacional del sistema

Para hacer la correspondencia entre el diagrama de clases con el de las tablas se tomo las consideraciones propuestas de Craig Larman (ver 2.1.5).

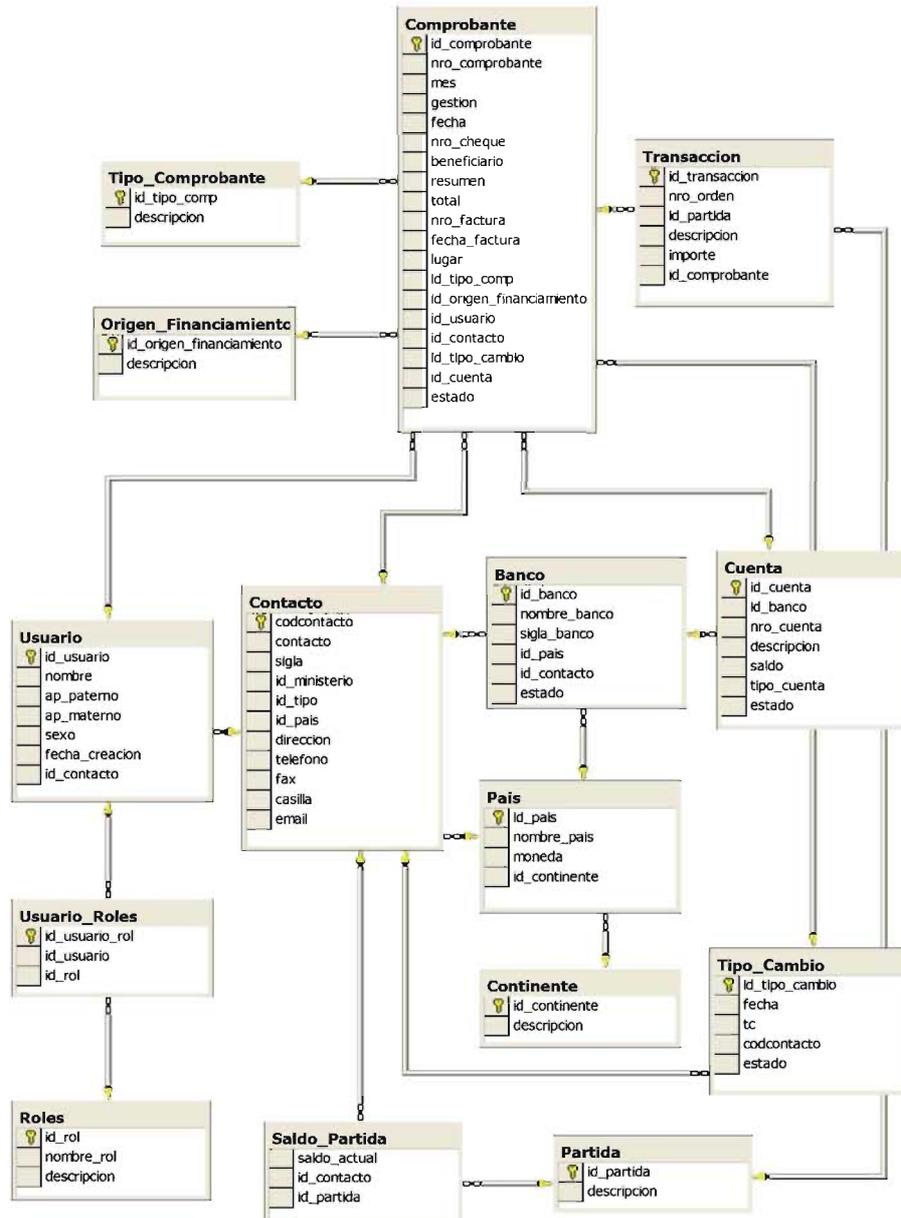


Figura 3.23 Diagrama Relacional (Control Financiero via Web).

## 3.4 Implementación

### 3.4.1 Modelo Físico de la Base de datos

El modelo físico de la base de datos descritos en las tablas 3.28 a la 3.46 nos muestra la estructura de cada una de las tablas de nuestro modelo.

<b>Sistema:</b> CONTROL FINANCIERO VÍA WEB PARA EL SERVICIO EXTERIOR					
<b>Nombre de la tabla:</b> Banco					
<b>Tipo llave</b>	<b>Nombre de la columna</b>	<b>Tipo de dato</b>	<b>Longitud</b>	<b>Permitir nulos</b>	<b>Restricciones</b>
PK	id_banco	int	4	no	>0
	nombre_banco	varchar	200	no	
	sigla_banco	varchar	10	no	
FK	id_pais	char	10	no	>0
FK	id_contacto	int	4	no	>0
	estado	char	2	no	[AC,AN]

**Tabla 3.28** Tabla Banco.

<b>Sistema:</b> CONTROL FINANCIERO VÍA WEB PARA EL SERVICIO EXTERIOR						
<b>Nombre de la tabla:</b> Comprobante						
<b>Tipo llave</b>	<b>Nombre de la columna</b>	<b>Tipo de dato</b>	<b>Longitud</b>	<b>Permitir nulos</b>	<b>Restricciones</b>	
PK	id_comprobante	int	4	no	>0	
	nro_comprobante	char	10	no		
	mes	smallint	2	no		
	gestion	smallint	2	no	>0	
	fecha	smalldate	4	no		
	nro_cheque	vachar	20	si		
	beneficiario	vachar	20	si		
	resumen	vachar	200	no		
	total	money	8	no		
	nro_factura	varchar	20	si		
	fecha_factura	smalldate	4	si		
	lugar	varchar	20	no		
	FK	id_tipo_comp	smallint	2	no	>0
	FK	id_origen_financiamiento	smallint	2	no	>0
FK	id_usuario	int	4	no	>0	
FK	id_contacto	int	4	no	>0	
FK	id_tipo_cambio	int	4	no	>0	
FK	id_cuenta	int	4	no	>0	
	estado	char	2	no	[AC,AN]	

**Tabla 3.29** Tabla Comprobante.

### 3.4.2 Seguridad

Pasamos a evaluar las amenazas de las que en algún momento podemos pasar a sufrir (ver 2.1.6), la columna “Sum” es el valor que podría tener esta amenaza, a mayor valor en esta columna significa que es la amenaza que se debe tomar en cuenta en la aplicación.

Nro.	AMENAZA	D	R	E	A	D	Sum
1	Inyección de SQL	3	3	2	3	3	14
2	Script entre sitios	3	3	2	3	3	14
3	Modificación del ingreso	2	3	3	2	3	13
4	Robo de claves	2	2	2	2	2	10
5	Reemplazo de sesión	3	2	1	2	2	10
6	Reemplazo de identidad	2	2	1	1	1	7
7	Revelado de información	3	2	2	2	2	11

**Tabla 3.43** Evaluación de las vulnerabilidades según DREAD.

En la tabla 3.43 observamos que las filas que arrojan mayor número son las de “Inyección de SQL” y el “Script entre sitios”, valores que nos muestran que se debería priorizar estos aspectos en el desarrollo.

Para eliminar la “inyección de SQL” se hace uso de procedimientos almacenados, de esta manera se elimina la inserción de instrucciones SQL que no requiere la aplicación.

Para eliminar el “Script entre sitios” se hace uso de validaciones en las entradas al sistema mediante Java Script, Expresiones Regulares, haciendo uso del Encoding y limitando los caracteres de entrada de la aplicación.

### 3.4.3 Arquitectura

En la figura 3.24 se muestra la implementación de la arquitectura funcional de la aplicación (ver 2.1.7 a)

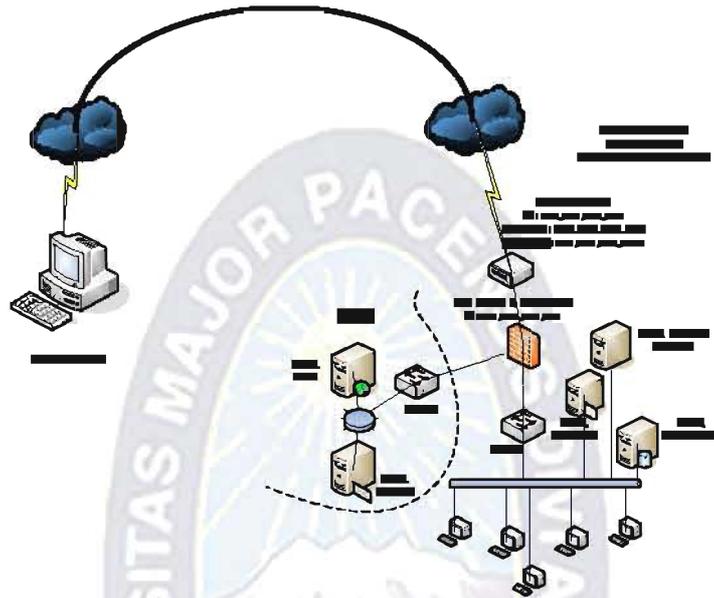


Figura 3.24 Arquitectura funcional del sistema.

En la figura 3.25 se muestra la implementación de la arquitectura estructural de la aplicación (ver 2.1.7 b)



Figura 3.25 Arquitectura Estructural del sistema.

---

## 3.5 Pruebas

### 3.5.1 Factores de Calidad ISO 9126.

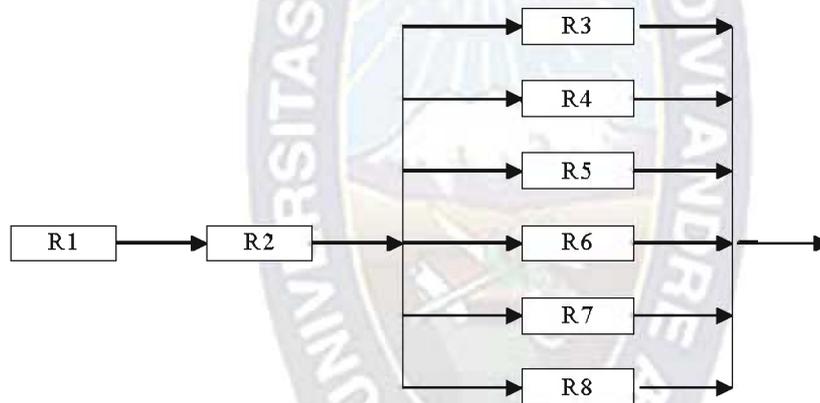
#### a) Confiabilidad

En la figura 3.26 se muestra el análisis del modelo planteado en el punto 2.1.8–a donde se muestran las combinaciones.

Entonces se tienen las siguientes ecuaciones:

$$R9 = 1 - (1 - R3) * (1 - R4) * (1 - R5) * (1 - R6) * (1 - R7) * (1 - R8)$$

$$R10 = R1 * R2 * R9$$



**Figura 3.26** Modelo de Sistema Propuesto.

Dentro de sistema existen 8 módulos, de los cuales 2 funcionan independientemente y están conectados en serie R1 y R2, y los restantes 6 módulos también funcionan independientemente pero están conectados en paralelo tal como se ve en la figura 3.26.

Ahora pasamos a calcular la confiabilidad de cada modulo tomando y considerando para ello que el tiempo para fallar es una variable aleatoria exponencial, se tiene  $R_i$  confiabilidad del subsistema (con  $i=1, 2, 3, \dots, 8$ ) y un error o tasa constante de fallo de  $\lambda=0.05$  en un tiempo de operación del sistema

de 1 día  $P(t)=1$ . Para una mayor comprensión de lo descrito se observa en la siguiente tabla 3.44.

Ri	$\lambda$	P(t)	$e^{-\lambda P(t)}$
R1	0.05	1	0.95
R2	0.05	1	0.95
R3	0.05	1	0.95
R4	0.05	1	0.95
R5	0.05	1	0.95
R6	0.05	1	0.95
R7	0.05	1	0.95
R8	0.05	1	0.95

**Tabla 3.43** Confiabilidad del Sistema.

Aplicando la distribución exponencial a Ri se obtiene que cada modulo presenta o tiene un 95% de confiabilidad en 1 día.

$$R9 = 1 - (1 - 0.95) * (1 - 0.95) * \dots * (1 - 0.95) = 1 - (1 - 0.95)^6 = 0.99 \quad (\text{en paralelo})$$

Con este resultado expresamos que la confiabilidad del funcionamiento de los módulos o subsistemas conectados en paralelo tiene un 99% de confiabilidad o dicho de otra forma, la confiabilidad de que no se presenten fallas en los módulos conectados en paralelo en 1 día es del 99%.

$$R10 = 0.95 * 0.95 = 0.90 \quad (\text{en serie})$$

Con este resultado expresamos que la confiabilidad del funcionamiento de los módulos o subsistemas conectados en serie tiene un 90% de confiabilidad o dicho de otra forma, la confiabilidad de que no se presenten fallas en los módulos conectados en serie en 1 día es del 90%.

---

Entonces uniendo tanto los módulos que se encuentran en paralelo como en serie se tiene:

$$R_{11} = R_9 * R_{10}$$

$$R_{11} = 0.90 * 0.99 = 0.90 \quad (\text{total})$$

Por tanto la confiabilidad del sistema completo esta dada por la funcionalidad de los módulos conectados tanto en serie como en paralelo, dando lugar que la confiabilidad de que no se presenten fallas del sistema durante un día es del 90%

Observando el % de la confiabilidad (90%) del sistema completo se pueden identificar la existencia de un margen de error debido a que pueda existir fallas en la conexión al sistema SIGASE los cuales infieren a los módulos R1 y R2, y también debido a que pueda presentar errores en la entrada de datos por parte del usuario interfiriendo en los módulos R3, R4, R5, R6, R7 y R8. las fallas por cortes de energía, virus, etc. no están incluidas.

### **b) Usabilidad**

Según el puneto 2.1.8-b el sistema en cuanto a la usabilidad es comprensible y dispone de la facilidad de uso para los usuarios destinados al llenado de comprobantes del Servicio Exterior del Ministerio de Relaciones Exteriores y Cultos, esto debido a que durante la entrega de software se han venido haciendo las modificaciones necesarias a las peticiones del usuario en cuanto a la interfaz, al manejo del sistema, esto para facilitarle al usuario a alcanzar sus objetivos y metas específicas.

### **c) Portabilidad**

Para ver la portabilidad (ver 2.1.8-c) del sistema dividiremos este en 2 secciones: la portabilidad del servidor y la portabilidad del lado cliente.

---

➤ Portabilidad del lado del Servidor

**Software:** Se necesita como sistema operativo Windows 2000 Server con servidor IIS para adelante en la familia de servidores con NetFrameWork 1.1 o posterior. Además para la Base de Datos se necesita Microsoft SQL Server 2000.

**Hardware:** Se necesita Un ordenador con Procesador 1.7 GHz., Memoria RAM 512, Memoria en video 64 MB.

➤ Portabilidad del lado del Cliente

**Software:** El ordenador del cliente tiene que tener un navegador de Internet, preferiblemente con soporte JavaScript. Acrobat Reader, debe tener el software necesario para conexión a Internet.

**Hardware:** Procesador de 1 GHz, memoria RAM de 256 MB, memoria de video 16 MB.

**d) Mantenibilidad.**

El cálculo del Índice de Madurez de Software (IMS) que se realiza (ver 2.1.8-d) es de la siguiente manera:

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

A medida que el IMS se aproxima a 1 el producto se empieza a estabilizar.

$$IMS = [6 - (1 + 0 + 0)] / 6 = 0.83$$

Por tanto el sistema es maduro en un 83 % y es estable. Lo que nos indica la facilidad del mantenimiento con la que se puede corregir el sistema si es que se puede encontrar un error, o se puede adaptar si su entorno cambia o mejor si el cliente necesita un cambio en el sistema.

### 3.5.2 Calidad de la aplicación en el entorno Web

En la tabla 3.44 se muestra el desarrollo de la metodología con los pesos de agregación a cada punto (ver 2.1.9).

Código	Nombre	Criterio	Preferencia Elemental AF
<b>1</b>	<b>Usabilidad</b>	<b>CVN</b>	<b>77.50</b>
<b>1.1</b>	<b>Comprensibilidad Global del Sistema.</b>	<b>CVN</b>	<b>90</b>
	Esquema de la Organización Global	CVN	90
	Calidad del Sistema Etiquetado	CPD	75
<b>1.2</b>	<b>Mecanismos de ayuda y retroalimentación en línea.</b>	<b>CVN</b>	<b>85</b>
	Calidad de la ayuda	CVN	90
	Indicador de Última Actualización	CMN	0
	Retroalimentación	CVN	80
	Cuestionario	CPD	0
	Comentarios y Ayuda	CPD	40
<b>1.3</b>	<b>Aspectos de Interfaces y Estéticos.</b>	<b>CVN</b>	<b>95</b>
	Permanencia y estabilidad en la presentación de los controles Principales	CVN	95
	Aspectos de Estilos	CMN	2= 100
	Preferencia Estética	CPD	90
<b>1.4</b>	<b>Misceláneas</b>	<b>CVN</b>	<b>40</b>
	Soporte a Lenguaje extranjero	CB	0=0
	Indicador de Resolución de Pantalla	CB	0=0
<b>2</b>	<b>Funcionabilidad</b>	<b>CVN</b>	<b>85.55</b>
<b>2.1</b>	<b>Aspectos de Recuperación y Búsquedas</b>	<b>CVN</b>	<b>86.66</b>
	Mecanismo de Búsqueda en el sistema	CVN	80
	Búsqueda Restringida	CVN	100
	Mecanismos de Recuperación	CVN	80
	Nivel de Personalización	CMN	1=60
<b>2.2</b>	<b>Aspectos de Navegación y Exploración</b>	<b>CVN</b>	<b>80</b>
	Navegabilidad	CVN	60
	Orientación	CVN	100
<b>2.3</b>	<b>Aspectos de Dominio Orientado al Visitante</b>	<b>CVN</b>	<b>90</b>
	Relevancia de Contenido	CVN	90
	Información de Unidades Informaciones de Representaciones en el Exterior	CVN	90
<b>3</b>	<b>Confiabilidad</b>	<b>CVN</b>	<b>80</b>
<b>3.1</b>	<b>No Deficiencia</b>	<b>CVN</b>	<b>80</b>
	Errores de enlaces	CVN	100
	Enlaces rotos	CMN	2= 100
	Enlaces inválidos	CMN	2= 100
	Enlaces no implementados	CMN	2= 100
	Errores o Deficiencias Varías	CVN	60
	Deficiencias o cualidades ausentes debido a diferentes navegadores(Browser)	CMN	1=60
	Deficiencias o Resultados Inesperados Independientes de Browsers (Por ej. Errores de búsqueda imprevistos)	CMN	1=60
<b>4</b>	<b>Eficiencia</b>	<b>CVN</b>	<b>86.5</b>
<b>4.1</b>	<b>Performancia</b>	<b>CVN</b>	<b>80</b>
	Paginas de Acceso Rápido	CPD	80
<b>4.2</b>	<b>Accesibilidad</b>	<b>CVN</b>	<b>92.50</b>
	Accesibilidad de la Información	CVN	95
	Imagen con título	CB	1=100
	Legibilidad Global	CB	1=100
	Accesibilidad de ventanas	CVN	90
	Versión de Macros	CMN	0=0

**Tabla 3.44** Desarrollo de la metodología con los peso de agregación.

---

La tabla 3.45 muestra un resumen de los datos obtenidos a partir de la tabla anterior (tabla 3.44)

NOMBRE	PREFERENCIA ELEMENTAL DE ACTIVOS FIJOS
Usabilidad	77.50 %
Funcionabilidad	85.55 %
Confiabilidad	80.00 %
Eficiencia	86.00 %
Calidad Global	82.26 %

**Tabla 3.45** Resumen de los resultados obtenidos.

En este contexto observamos y concluimos que el sistema de “Control financiero vía Web para el servicio exterior” en el entorno Web, los valores (Usabilidad, Funcionabilidad, Confiabilidad, Eficiencia y la Calidad Global) obtenidos se encuentran en el rango de 60 a 100 %. La cual nos indica que tiene un nivel de aceptabilidad **Satisfactorio**, de los requerimientos de los diferentes perfiles de usuario del Ministerio (ver 2.1.9 b).

---

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

Con la implementación del Sistema de Información “Control Financiero vía Web para el servicio exterior – Ministerio de Relaciones Exteriores y Cultos” en la oficina central del Ministerio de Relaciones Exteriores y Cultos se logro cumplir con el objetivo general que es el de “Desarrollar e implementar el subsistema de control financiero como parte del sistema SIGASE para tener un adecuado control de los dineros desembolsados a las diferentes Misiones”.

Con la ayuda de los controles de usuario se pudo construir tanto la interfaz de usuario como funcionalidad que requería dicho control para que pueda ser usado dentro el sistema.

Se logró definir una nomenclatura tanto en el diseño de las bases de datos como en los controles de usuario para que otros sistemas que forman parte del SIGASE puedan interactuar.

El sistema contribuye a cumplir aspectos de la ley 1178 (SAFCO), como ser:

- Lograr la administración eficaz y eficiente de los recursos públicos.
- Generar información que revele con transparencia la gestión de cada misión.

---

## 4.2 Recomendaciones

Se recomienda desarrollar un sistema en Windows Forms (Aplicación de Escritorio) como un añadido del sistema con sus respectivas Base de Datos distribuidas, independiente de la Web, pero dependiente de la Base de Datos Centralizada del Ministerio y que se ejecute en cada máquina cliente del Servicio Exterior, esto para subsanar la posible falla de conexión con el SIGASE y mejorar la performance del servidor o bien implementar los denominados granjas de servidores.

Se recomienda realizar la capacitación al usuario encargado de la verificación del ingreso de los comprobantes de ingreso y egreso, esto para el buen manejo de la información y del propio sistema. Además es recomendable realizar una evaluación constante con el fin de garantizar la fiabilidad de la misma, realizar un mantenimiento del sistema con el fin de mejoras funcionales y posibles ataques.

---

## Bibliografía

- [PRE03] PRESSMAN, Roger S. "INGENIERÍA DE SOFTWARE Un enfoque práctico", 2003, Quinta Edición, Editorial. McGraw Hill / Interamericana de España., S.A.U.
- [JBR99] JACOBSON Ivar, BOOCH Grady, RUMBAUGH James. "EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE", 1999, Primera Edición, Editorial. Pearson Educación S.A./Madrid 2000.
- [RJB99] RUMBAUGH James, JACOBSON Ivar, BOOCH Grady, "EL LENGUAJE UNIFICADO DE MODELADO. MANUAL DE REFERENCIA", 1999, Primera Edición, Editorial. Pearson Educación S.A./Madrid 2000.
- [LAR99] LARMAN Graig. "UML y Patrones", 1999 Primera Edición, Editorial. Prentice Hall, México 1999.
- [RUM98] RUMBAUGH James: "Modelado y Diseño Orientado a Objetos – Metodología OMT", 1998, Prentice Hall Internacional, UK Ltda.
- [FOW99] FOWLER Martin: "UML gota a gota", Addison Wesley Longman de Mexico, S.A. de C.V, Mexico 1999.
- [WEI02] WEITZEFELD Alfredo, Ingeniería de Software Orientada a Objetos, Teoría y Práctica con UML y Java, México, Octubre 2002
- [KRU00] KRUCHTEN P., The Rational Unified Process: An Introduction, 2000 Addison Wesley.
- [RSC98] Rational Software Corporation, Rational Unified Process. Best Practices for Software Development Teams, 1998

## Leyes, Decretos y Reglamentos.

La constitución política del estado.

Ley No. 2446 Ley de Organización del Poder Ejecutivo (LOPE).

Ley No. 1444 Ley del Servicio de Relaciones Exteriores.

Ley No. 1178 Ley de Administración y Control Gubernamental (SAFCO).

Ley No. 2042 Ley de Administración Presupuestaria.

Resolución ministerial No. 511\2003

Decreto Supremo No. 25964 Normas Básicas del Sistema de Administración de Bienes y Reglamento para la rendición de cuentas en el Servicio Exterior. R. S. 511/2003

---

## Referencias Web

**[MRE06]** Portal del Ministerio de Relaciones Exteriores y Cultos, del 30-09-06  
<http://portal.rree.gov.bo>

**[MSUML06]** Modelado de Sistemas con UML, del 03-10-06.  
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>

**[AEA06]** Amenazas en una aplicación, del 16-10-06  
[http://msdn.microsoft.com/library/en\\_us/dnnetsec/html/THCMCh10.asp?frame=true#c10618429\\_004](http://msdn.microsoft.com/library/en_us/dnnetsec/html/THCMCh10.asp?frame=true#c10618429_004)

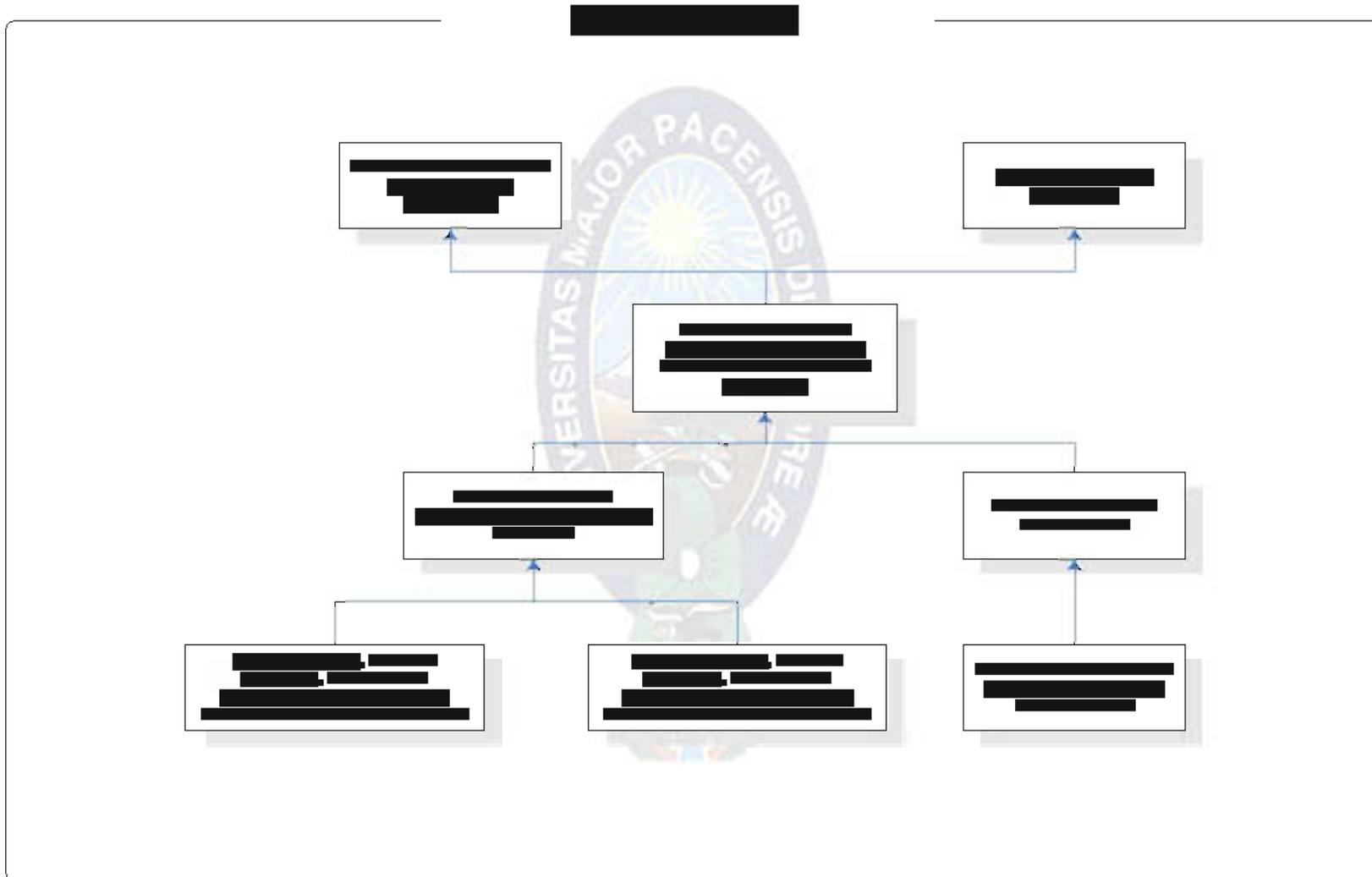
**[ARQ06]** Arquitectura, del 16-10-06  
<http://www.microsoft.com/spanish/msdn/arquitectura/default.asp>

**[SEI06]** Security Engineering Index, del 16-10-06  
[http://msdn.microsoft.com/practices/default.aspx?pull=/library/en\\_us/dnpag2/html/securityengindex.asp](http://msdn.microsoft.com/practices/default.aspx?pull=/library/en_us/dnpag2/html/securityengindex.asp)

**[OLS06]** OLSINA, Luís Antonio "Web Site QEM" Tesis Doctoral: Metodología cuantitativa para la evaluación y comparación de sitios Web, del 03-10-06.  
<http://www.di.uniovi.es/~cueva/investigacion/tesis/WebsiteQEM.pdf>



ANEXO A2



## ANEXO A3 - Marco Lógico

Resumen narrativo	Indicadores objetivamente verificables	Medios de verificación	Supuestos
<b>Fin</b> Información confiable y actualizada	Reducción del tiempo en un 80% a partir del inicio de operación del proyecto.	<ul style="list-style-type: none"> <li>- Cronograma de actividades</li> <li>- Informes del desarrollo del proyecto</li> <li>- Implementación de prototipos del sistema</li> </ul>	<b>Sostenibilidad</b> El seguimiento y posterior control de varias representaciones en el exterior se hace cada vez más difícil
<b>Propósito</b> Diseñar un sistema de información vía Web el cual lleve las cuentas de las misiones en una base de datos centralizada	<ul style="list-style-type: none"> <li>- Se facilita el trabajo en el llenado de formularios</li> <li>- Se reducen los posibles errores en el llenado de formularios</li> <li>- Se tiene información en línea</li> <li>- Se realiza controles más adecuados</li> <li>- Se reduce las cuentas observadas</li> </ul>	<ul style="list-style-type: none"> <li>- Informes emitidos por las direcciones del Ministerio en función al cronograma de actividades</li> <li>- Informes detallados de procesos</li> <li>- Informes de inspección para cada una de las actividades preestablecidas según el cronograma de actividades</li> </ul>	<b>Propósito a Fin (es)</b> El sistema de información está implementado y es una herramienta que es usado por las representaciones en el exterior
<b>Componentes</b> <ul style="list-style-type: none"> <li>- Documentos de análisis, diseño e implementación del sistema</li> <li>- Manual de usuario</li> <li>- Software implementado en el Ministerio de relaciones exteriores y culto</li> <li>- Personal capacitado en el manejo del software</li> <li>- Pruebas realizadas</li> </ul>	<ul style="list-style-type: none"> <li>- Los controles de usuario (user control) están funcionando correctamente para Julio</li> <li>- El personal prueba el sistema y está satisfecho con el, mostrando su aceptación con los resultados.</li> <li>- El conjunto de reportes solicitados están siendo utilizados sin observaciones para Julio</li> </ul>	<ul style="list-style-type: none"> <li>- Informes de evaluación satisfactoria del funcionamiento del sistema por parte del Ministerio.</li> <li>- Reportes generados por el sistema</li> <li>- Informes de las direcciones respaldando la veracidad de los datos</li> </ul>	<b>Componentes a Propósito</b> Se tiene el apoyo y aprobación del Ministerio al igual que las direcciones interesadas
<b>Actividades</b> <ul style="list-style-type: none"> <li>- Análisis del sistema</li> <li>- Especificación de requisitos</li> <li>- Diseño de la arquitectura del sistema</li> <li>- Diseño lógico de procesos</li> <li>- Diseño lógico de datos</li> <li>- Pruebas de implementación</li> <li>- Elaboración de manuales</li> <li>- Capacitación al personal</li> </ul>	Para cada actividad se tiene un presupuesto, que incluye <ul style="list-style-type: none"> <li>- Trabajo 5 Bs / hrs de transcripción</li> <li>- Trabajo 15 Bs / hrs. de análisis del sistema</li> <li>- Trabajo 10 Bs / hrs de programación</li> <li>- Costos en equipos computacionales</li> <li>- Material usado en la elaboración de manuales respectivos y en su capacitación</li> </ul>	<ul style="list-style-type: none"> <li>- Documentos de respaldo en el desarrollo de los módulos del sistema</li> <li>- Facturas por la compra de materiales</li> </ul>	<b>Actividades a Componentes.</b> Se dispone de los recursos financieros para cada una de las actividades, además de contar con el personal necesario y la colaboración constante de las direcciones para la recopilación de información, mejoramiento de los procesos y tareas que realiza el sistema

## ANEXO B - Cronograma de actividades

Código	Actividad	Inicio	Fin	Duración	Cronograma de actividades																		
					1	2	3	4	5	6	7	8	9	10	11	12							
I	Actividad 1	1	2	2	█																		
II	Actividad 2	1	3	3		█	█	█															
III	Actividad 3	2	4	3					█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IV	Actividad 4	3	5	3																		█	█

**Fase de Inicio.-** Se desarrolla los requisitos del producto desde la perspectiva del usuario. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto.

**Fase de Elaboración.-** Se analizan los requisitos y se desarrolla un prototipo de arquitectura. Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera fase de construcción deben estar analizados y diseñados.

**Fase de Construcción.-** Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. Se comienza la elaboración de material de apoyo al usuario.

**Fase de Transición.-** Se prepara para la distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios.

---

## **ANEXO C – Formularios**

Formularios para la rendición de cuentas estipulados en el reglamento para la rendición de cuentas en el servicio exterior – Ministerio de relaciones exteriores y culto, resolución ministerial No. 511/2003.

FORMULARIO-01 Programación Trimestral para la Ejecución Presupuestaria

FORMULARIO-02 Comprobante de Ingreso

FORMULARIO-03 Comprobante de Egreso

FORMULARIO-04 Registro Diario de Ingresos y Egresos

FORMULARIO-05 Informe Financiero Mensual

FORMULARIO-06 Conciliación Bancaria

FORMULARIO-07 Informe Financiero Anual

FORMULARIO-08 Caja Chica

FORMULARIO-09 Registro Diario de Ingresos y Egresos por Gastos de Representación

FORMULARIO-10 Planilla de Sueldos Servicio Exterior

## ANEXO D – Interfaces de la aplicación

En esta parte pasamos a describir las pantallas más importantes con las que cuenta el sistema.

La figura c1, nos muestra la pantalla inicial en la cual el usuario debe ingresar el nombre de usuario y su password para ingresar al sistema.

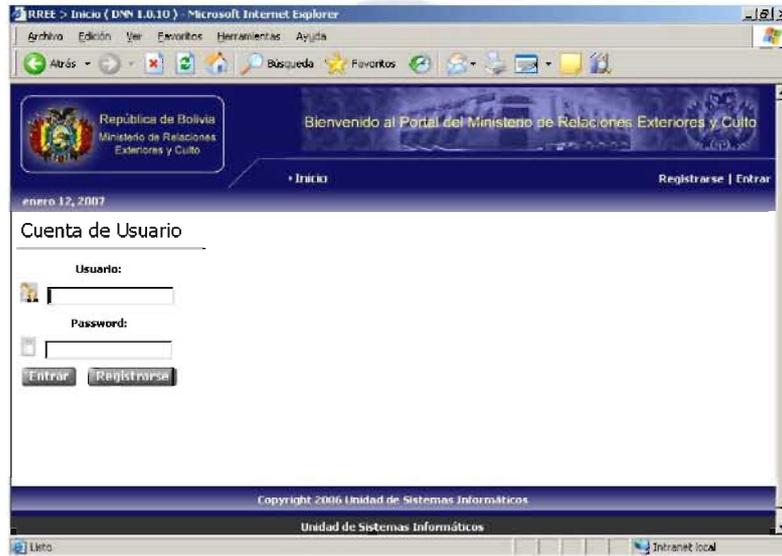


Figura c1 Pantalla inicial de logeo.

La figura c2, nos muestra las opciones principales con las que cuenta el administrador local de la Misión.

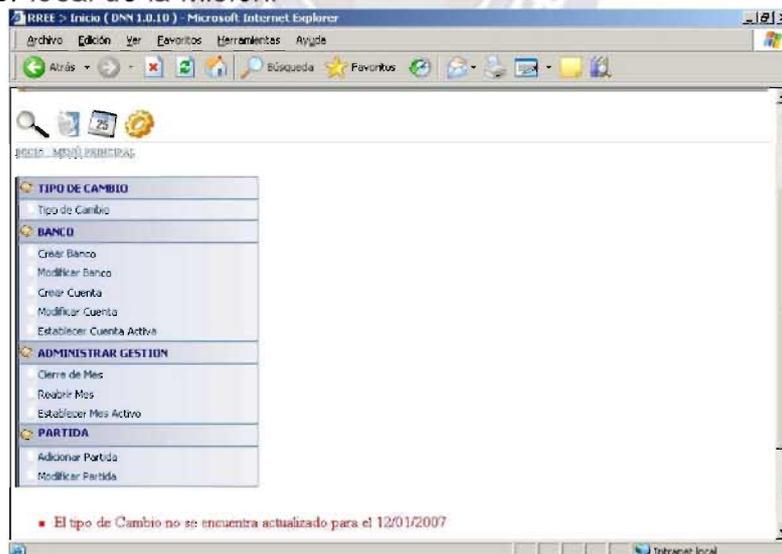


Figura c2 Opciones principales del administrador local de la Misión.

La figura c3, nos muestra la confirmación para el tipo de cambio con la cual se trabajará los comprobantes.

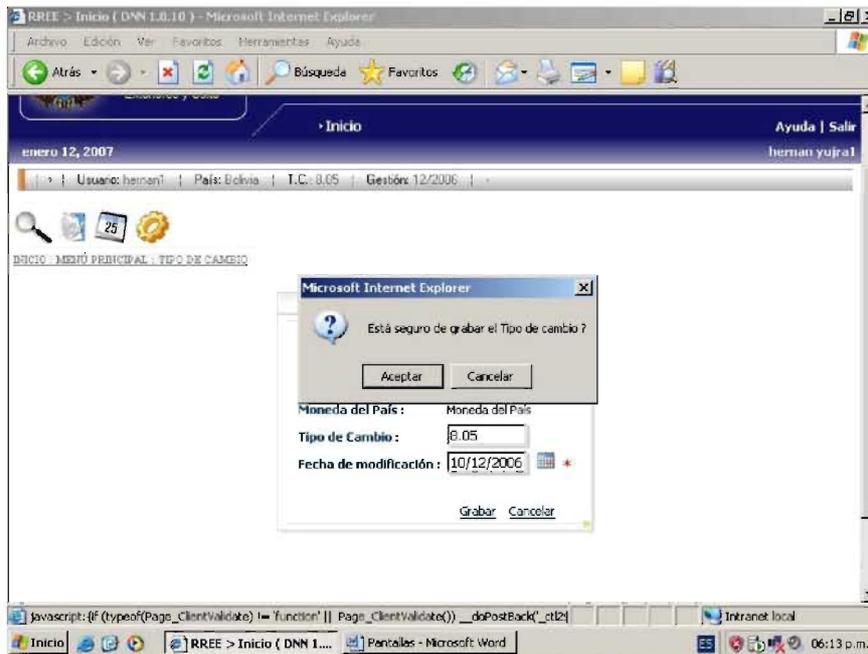


Figura c3 Opción tipo de cambio.

La figura c4, nos muestra la confirmación para el ingreso de un comprobante de ingreso.

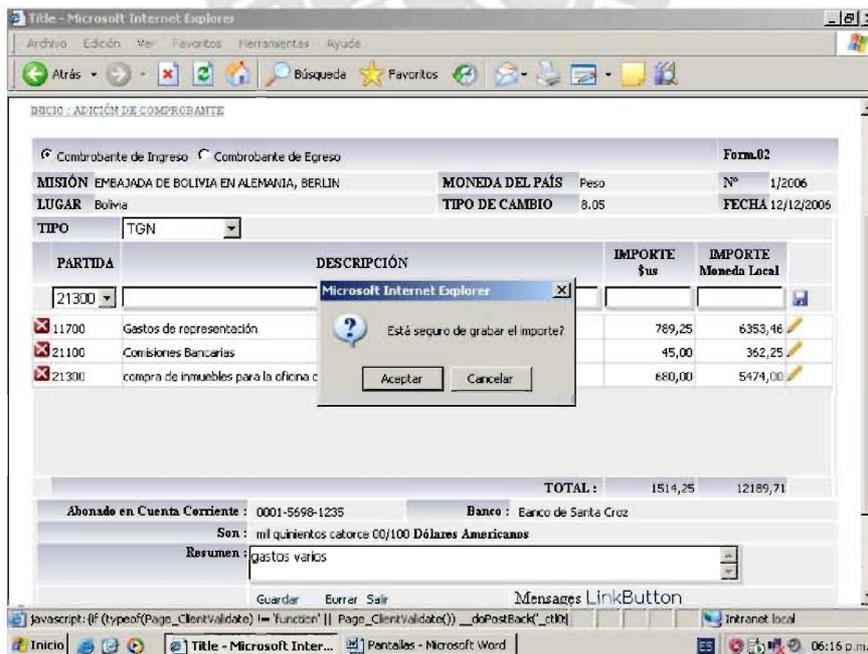


Figura c4 Confirmación para el ingreso de un comprobante.

La figura c5, nos muestra el reporte del comprobante que se ingresó anteriormente

The screenshot shows a PDF document titled 'COMPROBANTE DE INGRESO' (Receipt Report) from 'Form. 02' dated '1/2006'. The mission is 'EMBAJADA DE BOLIVIA EN ALEMANIA, BERLIN' and the location is 'Bolivia'. The date is '12/12/2006'. The currency is 'Moneda del país : 1' and the exchange rate is 'Tipo de cambio : 8.05'. The report includes a table of items with columns for 'Partida', 'Descripción', 'Importe en \$us', and 'Importe en moneda local'. The total amount is \$1514.25, which is equivalent to 12189.71 local currency units. The account is 'Cuenta Abonado en Corriente : 0001-5698-1235' at 'Banco : Banco de Santa Cruz'. The amount is written as 'Son : mil quinientos catorce 25/100 Dolares Americanos'. The concept is 'Resumen Concepto : gastos varios'.

Partida	Descripción	Importe en \$us	Importe en moneda local
11700	Gastos de representación	789,25	6353,46
21100	Comisiones Bancarias	45,00	362,25
21300	compra de inmuebles para la oficina central	680,00	5474,00
<b>Total :</b>		<b>1514,25</b>	<b>12189,71</b>

Figura c5 Reporte para el ingreso de un comprobante.

La figura c6, nos muestra el buscador con que cuenta el sistema, un comprobante puede buscarse de diferentes formas como muestra la figura.

The screenshot shows a web browser window with a search interface titled 'BUSQUEDA'. It has two radio buttons: 'Comprobante de Ingreso' (selected) and 'Comprobante de Egreso'. The search criteria include:
 

- Nº Comprobante: [ ] (16/2006)
- Origen Financiamiento: [ -- Cualquiera -- ]
- Banco: [ Banco Bolivia Americano ]
- Nro. Cuenta: [ 0125-5698-0158 ]
- Partida: [ -- Cualquiera -- ] (dropdown menu is open showing options: 11700, 21100, 21300, 21400, 23100, 24100, 25900, 32100, 34100)
- Entre fechas: [ ] [ 13/12/2006 ]

Figura c6 Buscador de comprobante

La figura c7, nos muestra los resultados arrojados luego de efectuar una búsqueda

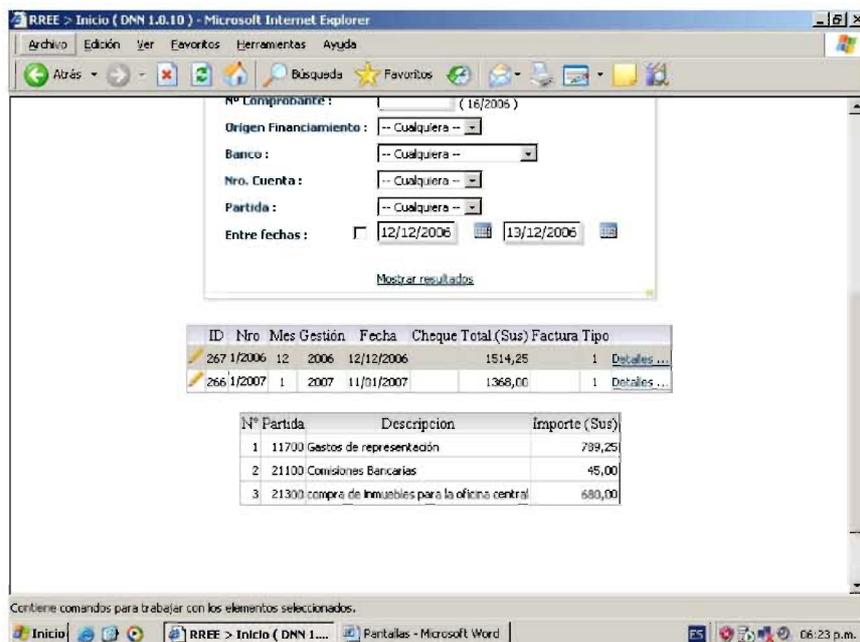


Figura c7 Resultados arrojados por el buscador.

La figura c8, nos muestra el modo de edición del comprobante,

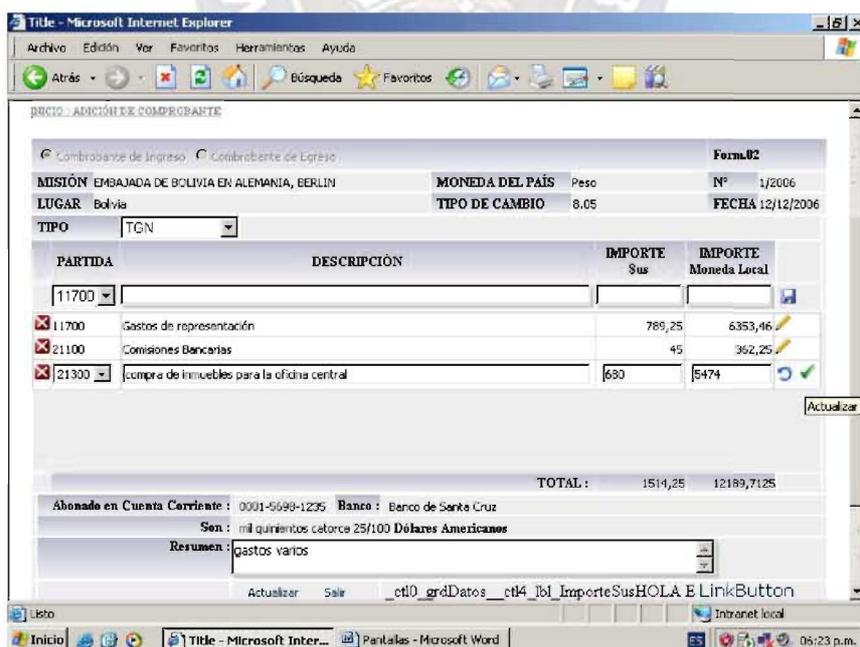


Figura c8 Edición de un comprobante

La figura c9, nos muestra los registros ingresados en un mes determinado.

Form 64  
N° .....

**REGISTRO DIARIO DE INGRESOS Y EGRESOS**  
 MISION : EMBAJADA DE BOLIVIA EN ALEMANIA, BERLIN  
 MES : Diciembre

FECHA	Nro. COMPR.	PARTIDA PPTO.	DESCRIPCION	CHEQUE Nro.	MONEDA LOCAL			TIPO DE CAMBIO	MONEDA LOCAL		
					INGRESO	EGRESO	SALDO		INGRESO	EGRESO	SALDO
12/12/2006	1/2006	11700	Gastos de representacion		96,04			8,05	769,25		
12/12/2006	1/2006	21100	Comisiones Bancarias		5,39			8,05	43,00		
12/12/2006	1/2006	21300	compra de inmuebles para la oficina central		84,47			8,05	680,00		
<b>TOTAL:</b>									1314,25	8,00	1134,33

Figura c9 Reporte de registro diario