



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

“SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”

APLICATIVO:

“DESARROLLO E IMPLEMENTACIÓN DE LOS MÓDULOS DE ADMINISTRACIÓN, AHORROS Y CAPTACIONES PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”

AUTOR: HENRY ESTEBAN GUDIÑO BARAHONA

DIRECTORA: DRA. CPA. MARÍA DE LA PORTILLA VERA MBA

**IBARRA-ECUADOR
2012**

CERTIFICACIÓN

Certifico que la Tesis **“SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”** con el aplicativo **“DESARROLLO E IMPLEMENTACIÓN DE LOS MÓDULOS DE ADMINISTRACIÓN, AHORROS Y CAPTACIONES PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”** ha sido realizada en su totalidad por el señor: Henry Esteban Gudiño Barahona portador de la cédula de identidad número: 1002966404.

Dra. CPA María De La Portilla Vera MBA

DIRECTORA DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
INVESTIGACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, HENRY ESTEBAN GUDIÑO BARAHONA, con cédula de identidad Nro. 1002966404-4, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”** con el aplicativo **“DESARROLLO E IMPLEMENTACIÓN DE LOS MÓDULOS DE ADMINISTRACIÓN, AHORROS Y CAPTACIONES PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

HENRY ESTEBAN GUDIÑO BARAHONA

C.I: 1002966404-4

Ibarra a los 17 días del mes de Julio del 2012



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

| DATOS DE CONTACTO | |
|----------------------------|--|
| CEDULA DE IDENTIDAD | 100296640-4 |
| APELLIDOS Y NOMBRES | GUDIÑO BARAHONA HENRY ESTEBAN |
| DIRECCIÓN | El Olivo alto |
| EMAIL | estebannnn666@hotmail.com |
| TELÉFONO FIJO | 062605103 |
| TELÉFONO MÓVIL | 097122133 |

| DATOS DE LA OBRA | |
|--------------------------|---|
| TÍTULO | “SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA-MÓDULOS DE CAPTACIONES, AHORROS Y ADMINISTRACIÓN” |
| AUTOR | GUDIÑO BARAHONA HENRY ESTEBAN |
| FECHA | 17 DE JULIO DEL 2012 |
| PROGRAMA | PREGRADO |
| TÍTULO POR EL QUE | INGENIERÍA EN SISTEMAS COMPUTACIONALES |
| DIRECTOR | DRA. C.P.A. MARÍA DE LA PORTILLA VERA M.B.A. |

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, HENRY ESTEBAN GUDIÑO BARAHONA, con cedula de identidad Nro. 100296640-4, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.

HENRY ESTEBAN GUDIÑO BARAHONA

C.I. 100296640-4

Ibarra a los 17 días del mes de Julio del 2012

DEDICATORIA

A Dios, por haberme dado la vida, y la oportunidad de seguir creciendo mentalmente, y poner siempre a las personas indicadas en el transcurrir de mi vida.

A mis padres Nelson y Elena, quienes con su ejemplo de lucha y de vida han reflejado en mí sus buenos valores, por haberme guiado por el camino correcto, por la fuerza que me han dado para seguir adelante y sobre todo por darme la oportunidad de ser un profesional.

A mis hermanos Andrés, Ricardo, Katherine y a mi sobrino Alejandro por el apoyo moral y sentimental que me han dado durante el desarrollo de este proyecto.

A mis amigos/as y compañeros/as, quienes estuvieron en la buenas y en las malas durante mi carrera universitaria compartiendo conocimientos, valores y alegrías.

A mis profesores, quienes me brindaron sus conocimientos y encaminaron por el camino del bien.

Esteban...

AGRADECIMIENTOS

A la Dra. María de la Portilla por su orientación y conocimientos otorgados para el desarrollo del presente proyecto.

Al Sr. Flandes Ibarra Gerente de la Cooperativa de Ahorros y Crédito Unión Cochapamba por ser el facilitador para el desarrollo del presente proyecto.

A mi familia, porque siempre me han apoyado, aconsejado y brindado todo el cariño que ha sido fundamental en mi vida, familia este logro es de todos.

A mis profesores, amigos y compañeros por todos los buenos momentos que vivimos y compartimos durante nuestra vida universitaria.

A todas aquellas personas que fueron parte y pieza fundamental para la culminación de este proyecto.

Esteban...

TABLA DE CONTENIDOS

| | |
|---|----|
| <i>CERTIFICACIÓN</i> | i |
| <i>DEDICATORIA</i> | v |
| <i>AGRADECIMIENTOS</i> | vi |
| ÍNDICE DE FIGURAS..... | 6 |
| ÍNDICE DE TABLAS..... | 10 |
| SUMARY | 13 |
| 1. Introducción..... | 19 |
| 1.1. Base legal de la institución..... | 19 |
| 1.1.1. Constitución y domicilio..... | 19 |
| 1.1.2. Objetivos de la cooperativa..... | 19 |
| 1.1.3. Principios de la cooperativa | 20 |
| 1.1.4. Socios fundadores..... | 20 |
| 1.2. Situación actual diagnóstico..... | 21 |
| 1.2.1. Misión | 21 |
| 1.2.2. Visión | 21 |
| 1.2.3. Valores corporativos | 21 |
| 1.2.4. FODA..... | 22 |
| 1.2.5. Situación actual | 22 |
| 1.2.6. Diagnostico 6M | 23 |
| 1.2.7. Objetivos operativos y actividades del plan anual operativo..... | 23 |
| 1.3. Estructura | 25 |
| 1.3.1. Organigrama estructural..... | 25 |
| 1.4. Información general..... | 26 |
| 1.4.1. Clientes, segmentado por valores | 26 |
| 1.4.2. Préstamos | 27 |
| 2. Marco teórico..... | 29 |
| 2.1. Normas legales | 29 |
| 2.1.1. Ley general de instituciones del sistema financiero | 29 |
| 2.1.2. Ley de cooperativas | 29 |
| 2.1.3. Normas de basilea..... | 30 |
| 2.2. Herramientas y estándares de código libre | 30 |
| 2.2.1. WORLD WIDE WEB..... | 30 |
| 2.2.2. HTML (Hyper Text Markup Language) | 31 |
| 2.2.2.1. Elementos de HTML..... | 31 |
| 2.2.2.2. Estructura de un documento HTML | 32 |
| 2.2.3. CSS (Cascading Style Sheets)..... | 32 |
| 2.2.3.1. Funcionamiento de CSS | 33 |
| 2.3. Seguridad en sistemas web | 34 |

| | | |
|----------|---|----|
| 2.3.1. | Introducción | 35 |
| 2.3.2. | Seguridad en el cliente | 36 |
| 2.3.3. | Seguridad en el servidor..... | 36 |
| 2.3.4. | Seguridad en la aplicación | 37 |
| 2.3.5. | Seguridad en la comunicación..... | 38 |
| 2.4. | Sistemas operativos(SO)..... | 39 |
| 2.4.1. | Definición..... | 39 |
| 2.4.2. | Características y funciones de los sistemas operativos..... | 40 |
| 2.4.3. | Ejemplos de sistema operativo | 40 |
| 2.4.4. | Sistema operativo Fedora..... | 41 |
| 2.4.4.1. | Características principales | 42 |
| 2.4.5. | Sistema operativo CentOS | 42 |
| 2.4.5.1. | Características..... | 43 |
| 2.5. | Servidor de aplicaciones | 43 |
| 2.5.1. | Introducción a Apache Tomcat | 44 |
| 2.5.2. | Estructura del servidor | 45 |
| 2.5.3. | Funcionamiento del servidor de aplicaciones | 46 |
| 2.5.4. | Características de Tomcat | 47 |
| 2.6. | Bases de datos | 47 |
| 2.6.1. | PostgreSQL | 48 |
| 2.6.1.1. | Arquitectura de Postgres | 49 |
| 2.6.1.2. | Características..... | 50 |
| 2.7. | Plataformas y frameworks | 52 |
| 2.7.1. | NetBeans IDE..... | 52 |
| 2.7.1.1. | Características..... | 52 |
| 2.7.2. | Java | 53 |
| 2.7.2.1. | Definición | 53 |
| 2.7.2.2. | Arquitectura..... | 53 |
| 2.7.2.3. | Características..... | 54 |
| 2.7.3. | Java EE (Java Platform, Enterprise Edition)..... | 55 |
| 2.7.3.1. | Definición..... | 56 |
| 2.7.3.2. | Arquitectura..... | 57 |
| 2.7.3.3. | Características..... | 58 |
| 2.7.3.4. | Beneficios | 59 |
| 2.7.4. | Hibernate | 59 |
| 2.7.4.1. | Definición..... | 59 |
| 2.7.4.2. | Características de Hibernate | 60 |
| 2.7.5. | JSF (Java Server Faces) | 61 |
| 2.7.5.1. | Características..... | 62 |
| 2.7.5.2. | JSF con Netbeans..... | 63 |
| 2.7.6. | RichFaces como framework RIA..... | 64 |
| 2.7.6.1. | Definición..... | 64 |
| 2.7.6.2. | Componentes de RichFaces..... | 65 |
| 2.7.6.3. | Características..... | 66 |
| 2.7.6.4. | Requerimientos Técnicos. | 66 |

| | | |
|----------|---|-----|
| 2.7.6.5. | Librerías del Framework RichFaces..... | 67 |
| 2.7.7. | Herramienta de Reportes JasperReports..... | 67 |
| 2.7.7.1. | ¿Qué es JasperReports?..... | 67 |
| 2.7.7.2. | Requerimientos de JasperReports..... | 68 |
| 2.7.7.3. | Funcionamiento de JasperReports..... | 69 |
| 3. | DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MODULOS..... | 70 |
| 3.1. | Funcionamiento del sistema..... | 71 |
| 3.1.1. | Vista general..... | 71 |
| 3.1.2. | Características del sistema..... | 71 |
| 3.1.3. | Dependencias para la implementación..... | 72 |
| 3.1.3.1. | Costos..... | 73 |
| 3.1.3.2. | Licenciamiento..... | 73 |
| 3.1.3.3. | Instalación..... | 73 |
| 3.1.4. | Funcionamiento..... | 74 |
| 3.1.4.1. | Arquitectura del sistema..... | 75 |
| 3.1.4.2. | Diagrama General de Casos de Uso..... | 75 |
| 3.1.4.3. | Diagrama entidad relación..... | 76 |
| 3.2. | Descripción y funcionamiento de los módulos..... | 76 |
| 3.2.1. | Descripción del módulo de Administración y Seguridad..... | 76 |
| 3.2.2. | Funcionamiento del módulo de Administración y Seguridad..... | 76 |
| 3.2.3. | Descripción de los casos de uso del módulo de administración..... | 79 |
| 3.2.3.1. | Especificación de Caso de Uso: Autenticación de Usuarios..... | 79 |
| 3.2.3.2. | Especificación de Caso de Uso: Administración de Usuarios..... | 82 |
| 3.2.3.3. | Especificación de Caso de Uso: Administrar Parámetros Generales..... | 85 |
| 3.2.4. | Descripción del módulo de Captaciones (Caja)..... | 96 |
| 3.2.5. | Funcionamiento del módulo de Captaciones (Caja)..... | 96 |
| 3.2.6. | Descripción de los casos de uso del módulo de Captaciones..... | 99 |
| 3.2.6.1. | Especificación de Caso de Uso: Apertura de Caja..... | 99 |
| 3.2.6.2. | Especificación de Caso de Uso: Apertura de Cuenta..... | 102 |
| 3.2.6.3. | Especificación de Caso de Uso: Depósitos de Ahorros..... | 105 |
| 3.2.6.4. | Especificación de Caso de Uso: Retiros en Efectivo..... | 107 |
| 3.2.6.5. | Especificación de Caso de Uso: Pago de Préstamo..... | 109 |
| 3.2.7. | Descripción del módulo de Ahorros..... | 112 |
| 3.2.8. | Funcionamiento del módulo de Ahorros..... | 112 |
| 3.2.9. | Descripción de los casos de uso del módulo de Ahorros..... | 114 |
| 3.2.9.1. | Especificación de Caso de Uso: Cuentas de Ahorro..... | 115 |
| 3.2.9.2. | Especificación de Caso de Uso: Débitos y Créditos..... | 118 |
| 3.2.9.3. | Especificación de Caso de Uso: Notas de Crédito por Depósito en Bancos | 121 |
| 3.2.9.4. | Especificación de Caso de Uso: Transferencia de Cuenta a Préstamos..... | 123 |
| 3.2.9.5. | Especificación de Caso de Uso: Transferencia de Cuenta a Cuenta..... | 126 |
| 3.2.9.6. | Especificación de Caso de Uso: Depósitos a Plazo Fijo..... | 128 |
| 3.2.9.7. | Especificación de Caso de Uso: Consultas y Reportes..... | 131 |
| 4. | DISEÑO Y DESARROLLO DEL APLICATIVO..... | 135 |

| | | |
|----------|--|-----|
| 4.1. | Introducción | 135 |
| 4.2. | Fase de Incepción | 136 |
| 4.2.1. | Documento de Visión..... | 136 |
| 4.2.1.1. | Introducción..... | 136 |
| 4.2.1.2. | Posicionamiento..... | 137 |
| 4.2.1.3. | Descripción de los interesados y usuarios | 139 |
| 4.2.1.4. | Vista General del Producto | 147 |
| 4.2.1.5. | Características del producto | 152 |
| 4.2.1.6. | Restricciones..... | 153 |
| 4.2.1.7. | Rangos de calidad | 153 |
| 4.2.1.8. | Precedencia y Prioridad..... | 153 |
| 4.2.1.9. | Otros requerimientos del producto..... | 154 |
| 4.2.2. | Plan de Desarrollo de Software | 154 |
| 4.2.2.1. | Introducción..... | 154 |
| 4.2.2.2. | Vista General del Proyecto | 155 |
| 4.2.2.3. | Organización del Proyecto | 160 |
| 4.2.2.4. | Gestión del Proceso..... | 161 |
| 4.2.2.5. | Referencias | 167 |
| 4.2.3. | Actas de Trabajo | 168 |
| 4.2.3.1. | Acta de Trabajo N° 1 requerimientos módulos del sistema..... | 168 |
| 4.2.3.2. | Acta de Trabajo N° 2 requerimientos cuentas de usuario..... | 169 |
| 4.2.3.3. | Acta de Trabajo N° 9 requerimientos cuentas de ahorro..... | 170 |
| 4.3. | Fase de Elaboración..... | 171 |
| 4.3.1. | Arquitectura..... | 171 |
| 4.3.1.1. | Introducción..... | 171 |
| 4.3.1.2. | Representación de la Arquitectura | 171 |
| 4.3.1.3. | Objetivos y Restricciones de la Arquitectura | 171 |
| 4.3.1.4. | Vista de Casos de Uso | 172 |
| 4.3.1.5. | Vista de Restricciones | 182 |
| 4.3.1.6. | Vistas Lógicas..... | 185 |
| 4.3.1.7. | Vista de Despliegue..... | 187 |
| 4.3.2. | Especificación de Casos de Uso | 188 |
| 4.3.2.1. | Documentos | 188 |
| 4.4. | Fase de Construcción..... | 188 |
| 4.4.1. | Modelo de Datos..... | 189 |
| 4.4.1.1. | Introducción..... | 189 |
| 4.4.1.2. | Modelos de Datos | 189 |
| 4.4.2. | Plan de Pruebas..... | 193 |
| 4.4.2.1. | Introducción..... | 193 |
| 4.4.2.2. | Estrategia de Pruebas | 193 |
| 4.4.2.3. | Herramientas..... | 198 |
| 4.4.2.4. | Recursos | 198 |
| 4.4.2.5. | Entregables | 198 |
| 4.4.3. | Lista de Riegos | 199 |
| 4.5. | Fase de Transición | 200 |

| | | |
|---------|--------------------------------------|--------------------------------------|
| 4.5.1 | Manual de Instalación..... | 200 |
| 4.5.1.1 | Introducción..... | 200 |
| 5. | Conclusiones y Recomendaciones | 215 |
| 5.1. | Conclusiones | 215 |
| 5.2. | Recomendaciones | 216 |
| 6. | GLOSARIO | 217 |
| 6.1. | Introducción. | 218 |
| 6.2. | Propósito | 218 |
| 6.3. | Referencias..... | 218 |
| 6.4. | Glosario..... | 218 |
| 7. | Referencias | 223 |
| 7.1. | Leyes para Sistemas Financieros..... | 223 |
| 7.2. | Sistemas Operativos | 223 |
| 7.3. | Seguridad Web | 223 |
| 7.4. | Bases de Datos | 224 |
| 7.5. | Herramientas de Desarrollo | 224 |
| 8. | ANEXOS | ¡Error! Marcador no definido. |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Diagnóstico de la Cooperativa (6 M)..... | 23 |
| Figura 2: Organigrama Estructural | 25 |
| Figura 3: Gráfico clientes cooperativa | 26 |
| Figura 4: Estructura de un documento HTML | 32 |
| Figura 5: Ejemplo CSS..... | 33 |
| Figura 6: Forma de uso de CSS | 33 |
| Figura 7: Seguridad en sistemas web..... | 35 |
| Figura 8: Sistemas de seguridad para servidores..... | 36 |
| Figura 9: Seguridad..... | 37 |
| Figura 10: HTTP sobre SSL..... | 39 |
| Figura 11: Sistemas Operativos..... | 39 |
| Figura 12: Logo de Tomcat | 43 |
| Figura 13: Estructura de Tomcat | 45 |
| Figura 14: Funcionamiento de Apache Tomcat | 46 |
| Figura 15: Logo del Servidor de BDD PostgreSQL..... | 48 |
| Figura 16: Arquitectura de Postgres | 50 |
| Figura 17: Arquitectura de Java..... | 54 |
| Figura 18: Contenedor de JEE..... | 56 |
| Figura 19: Arquitectura Java EE..... | 57 |
| Figura 20: Capa de Persistencia..... | 59 |
| Figura 21: Arquitectura base de Hibernate | 60 |
| Figura 22: Diagrama de una aplicación JSF..... | 62 |
| Figura 23: Componentes de JSF..... | 62 |
| Figura 24: Componentes de RichFaces..... | 65 |
| Figura 25: Funcionamiento de JasperReports | 69 |
| Figura 26: Prospectiva del Sistema..... | 71 |
| Figura 27: Funcionamiento General del Sistema..... | 74 |
| Figura 28: Arquitectura de la Aplicación..... | 75 |
| Figura 29: Modelo de casos de uso del módulo de Administración..... | 79 |
| Figura 30: pantalla de Autenticación | 79 |
| Figura 31: Modelo de la tabla usuarios | 80 |
| Figura 32: Caso de uso del proceso de Autenticación | 81 |
| Figura 33: Caso de uso administración de usuarios..... | 82 |
| Figura 34: Pantalla de ingreso de nuevos usuarios..... | 83 |
| Figura 35: Diseño de la tabla usuarios | 84 |
| Figura 36: Casos de uso del proceso de parámetros generales..... | 86 |
| Figura 37: Pantalla de ingreso de estados de cuenta..... | 86 |
| Figura 38: Pantalla de edición de estados de cuenta..... | 87 |

| | |
|---|-----|
| Figura 39: Diseño de la tabla tipos de estados de cuenta | 88 |
| Figura 40: Pantalla de administración de productos | 89 |
| Figura 41: Diseño de la tabla productos..... | 90 |
| Figura 42: Pantalla de conceptos de Ingresos..... | 91 |
| Figura 43: Diseño de la tabla conceptos de ingresos | 92 |
| Figura 44: Pantalla administración de causas de notas de débito y crédito | 92 |
| Figura 45: Diseño de la tabla de causas de notas de débito y crédito..... | 93 |
| Figura 46: Pantalla de administración de tipos de transacciones | 94 |
| Figura 47: Diseño de la tabla tipos de transacciones | 95 |
| Figura 48: Casos de uso del módulo de Captaciones..... | 99 |
| Figura 49: Pantalla selección de la fecha de trabajo | 99 |
| Figura 50: Pantalla de apertura de Caja | 100 |
| Figura 51: Diseño de la tabla apertura de caja | 100 |
| Figura 52: Caso de uso Apertura de Cuenta..... | 102 |
| Figura 53: Pantalla apertura de cuentas | 102 |
| Figura 54: Diseño de la tabla movimientos..... | 103 |
| Figura 55: Caso de uso depósito de ahorros..... | 105 |
| Figura 56: Pantalla de depósitos de ahorros | 106 |
| Figura 57: Caso de uso retiros de Ahorros | 107 |
| Figura 58: Pantalla de retiros de ahorros..... | 108 |
| Figura 59: Caso de uso pago de préstamos | 109 |
| Figura 60: Pantalla de pagos de Préstamos | 110 |
| Figura 61: Diseño de la tabla amortizaciones para pago de préstamos | 111 |
| Figura 62: Casos de uso del módulo de ahorros..... | 114 |
| Figura 63: Caso de uso Cuentas de Ahorro..... | 115 |
| Figura 64: Pantalla cuentas de ahorros | 115 |
| Figura 65: Pantalla de productos activos del Cliente..... | 116 |
| Figura 66: Pantalla de lista de movimientos de una cuenta de ahorros | 116 |
| Figura 67: Diseño de las tablas para registro de cuentas de ahorros | 117 |
| Figura 68: Cado de uso realizar débitos y créditos..... | 118 |
| Figura 69: Pantalla para realizar Débitos y Créditos | 118 |
| Figura 70: Pantalla de ingreso de información para acreditación | 119 |
| Figura 71: Diseño de la tabla débitos y créditos..... | 119 |
| Figura 72: Caso de uso nota de crédito por depósito en bancos..... | 121 |
| Figura 73: Pantalla de crédito por depósitos en el banco..... | 121 |
| Figura 74: Diseño de la tabla depósitos en bancos | 122 |
| Figura 75: Campos de la tabla depósitos en el banco | 122 |
| Figura 76: Caso de uso Transferencias de ahorros a préstamos | 123 |
| Figura 77: Pantalla de transferencia de cuentas de ahorros a préstamos | 124 |
| Figura 78: Diseño de la tabla transferencias | 125 |
| Figura 79: Caso de uso transferencia de ahorros a ahorros..... | 126 |

| | |
|--|-----|
| Figura 80: Pantalla de transferencias de ahorros a ahorros | 127 |
| Figura 81: Pantalla de depósitos a plazo fijo | 129 |
| Figura 82: Pantalla de registros de depósitos a plazo fijo | 129 |
| Figura 83: Diseño y campos de la tabla depósitos a plazo fijo | 130 |
| Figura 85: Pantalla para reportes de saldos de ahorros | 132 |
| Figura 86: Pantalla para reportes de transacciones de ahorros | 132 |
| Figura 87: Pantalla para reportes de saldos totales de Ahorros | 133 |
| Figura 88: Prospectiva del Producto | 148 |
| Figura 89: Calendario de las principales tareas del proyecto | 164 |
| Figura 90: Procesos del Módulo Administración y Seguridad | 172 |
| Figura 91: Procesos del Módulo Ahorros | 173 |
| Figura 92: Procesos del Módulo Cajero | 173 |
| Figura 93: Actores | 174 |
| Figura 94: Modelos de Casos de Negocio | 175 |
| Figura 96: Casos de Uso del Módulo de Administración y Seguridad | 177 |
| Figura 97: Casos de Uso del Módulo de Ahorros | 178 |
| Figura 98: Casos de Uso del Módulo Cajero | 179 |
| Figura 99: Patrón Arquitectónico MVC | 185 |
| Figura 100: Arquitectura Lógica | 186 |
| Figura 101: Interfaz de Usuario | 186 |
| Figura 102: Vista de Despliegue | 188 |
| Figura 103: Modelo de Datos Módulo de Administración | 190 |
| Figura 104: Modelo de Datos del Módulo de Ahorros | 191 |
| Figura 105: Modelo de Datos Módulo de Captaciones (Caja) | 192 |
| Figura 106: Instalación de la máquina virtual de Java | 201 |
| Figura 107: Configuración de la variable JAVA_HOME | 201 |
| Figura 108: Configuración del PATH de java | 202 |
| Figura 109: Asistente de instalación de Postgres. | 202 |
| Figura 110: Seleccionar el directorio de Instalación de Postgres. | 203 |
| Figura 111: Seleccionar el directorio de datos de Postgres. | 203 |
| Figura 112: Ingresando la contraseña de root de Postgres. | 204 |
| Figura 113: Seleccionar el puerto que utilizará Postgres. | 204 |
| Figura 114: El programa listo para instalar. | 205 |
| Figura 115: Instalación en progreso de Postgres. | 205 |
| Figura 116: Terminar la instalación de Postgres. | 206 |
| Figura 117: Creación de nuevo usuario. | 206 |
| Figura 118: Creación de nuevo usuario. | 207 |
| Figura 119: Creación de nuevo usuario. | 207 |
| Figura 120: Creación de una nueva base de datos. | 208 |
| Figura 121: Subir backup a la base de datos. | 208 |
| Figura 122: Subir backup a la base de datos. | 209 |

| | |
|--|-----|
| Figura 123: Subir backup a la base de datos. | 209 |
| Figura 124: Página de administración de Apache Tomcat..... | 211 |
| Figura 125: Página de inicio del sistema..... | 212 |
| Figura 126: Generación de la clave. | 212 |
| Figura 127: Parámetros de la clave. | 213 |
| Figura 128: Configuración del server.xml | 213 |

ÍNDICE DE TABLAS

| | |
|---|-----|
| Tabla 1: Análisis de la Cooperativa (FODA) | 22 |
| Tabla 2: Tabla de Préstamos..... | 27 |
| Tabla 3: Comparativo de Sistemas Operativos libres y Propietarios..... | 41 |
| Tabla 4: Límites de PostgreSQL..... | 51 |
| Tabla 5: Características del Sistema | 72 |
| Tabla 6: Funcionalidad de Modulo de Administración y Seguridad | 78 |
| Tabla 7: Campos de la tabla usuario | 80 |
| Tabla 8: Campos de la tabla usuarios | 84 |
| Tabla 9: Campos de la tabla estados de cuenta | 88 |
| Tabla 10: Campos de la tabla productos | 90 |
| Tabla 11: Campos de la tabla conceptos de ingresos..... | 92 |
| Tabla 12: Campos de la tabla de causas de notas de débito y crédito | 93 |
| Tabla 13: Campos de la tabla de tipos de transacciones | 95 |
| Tabla 14: Funcionalidad de Modulo de Captaciones..... | 98 |
| Tabla 15: Campos de la tabla apertura de caja | 101 |
| Tabla 16: Campos de la tabla movimientos | 104 |
| Tabla 17: Campos de la tabla amortizaciones | 111 |
| Tabla 18: Funcionalidad de Modulo de Administración y Seguridad | 114 |
| Tabla 19: Campos de la tabla débitos y créditos | 120 |
| Tabla 20: Campos de la tabla transferencias | 125 |
| Tabla 21: Definición del Problema..... | 139 |
| Tabla 22: Resumen de Interesados | 140 |
| Tabla 23: Resumen de los Usuarios..... | 141 |
| Tabla 24: Perfil de Coordinador del Proyecto | 142 |
| Tabla 25: Perfil Responsable del Proyecto | 143 |
| Tabla 26: Perfil responsable Funcional..... | 144 |
| Tabla 27: Perfil Administrador del Sistema | 144 |
| Tabla 28: Perfil Administrador Funcional del Sistema..... | 145 |
| Tabla 29: Perfil del Usuario del Sistema..... | 145 |
| Tabla 30: Necesidades de los interesados y usuarios..... | 147 |
| Tabla 31: Resumen de Capacidades Módulo de Ahorros | 149 |
| Tabla 32: Resumen de Capacidades Módulo de Captaciones | 149 |
| Tabla 33: Resumen de Capacidades Módulo de Administración y Seguridad | 150 |
| Tabla 34: Costos del Proyecto | 152 |
| Tabla 35: Roles y Responsabilidades | 160 |
| Tabla 36: Planificación de fases | 161 |
| Tabla 37: Hitos Finales de cada Fase..... | 162 |
| Tabla 38: Objetivos de las Iteraciones | 163 |
| Tabla 39: Calendario Fase de Inicio | 165 |

| | |
|--|-----|
| Tabla 40: Calendario Fase de Elaboración..... | 166 |
| Tabla 41: Descripción de Clientes. | 175 |
| Tabla 42: Prioridad de Casos de Uso..... | 176 |
| Tabla 43: Tecnologías Empleadas. | 184 |
| Tabla 44: Pruebas de Integridad de Datos..... | 194 |
| Tabla 45: Pruebas del Sistema..... | 194 |
| Tabla 46: Pruebas del Ciclo de Negocio | 195 |
| Tabla 47: Pruebas de Interfaz de Usuario | 196 |
| Tabla 48: Pruebas de Desempeño | 197 |
| Tabla 49: Pruebas de Seguridad y Control de Acceso | 197 |
| Tabla 50: Herramientas | 198 |
| Tabla 51: Recursos..... | 198 |
| Tabla 52: Entregables..... | 198 |
| Tabla 53: Lista de Riesgo..... | 199 |

RESUMEN

El presente proyecto ha sido realizado para solventar las necesidades de la Cooperativa de Ahorro y Crédito Unión Cochapamba mediante la implementación de un Sistema Integrado de Información Financiera para lo cual se desarrollo un Sistema Web que cumple con las especificaciones y necesidades de la Cooperativa.

En la actualidad las aplicaciones de escritorio están siendo remplazadas por sistemas o aplicaciones Web debido a que éstas se destacan principalmente por su usabilidad y portabilidad, razón para inclinarse por esta forma de desarrollo, para lo cual se utilizó herramientas y estándares de código abierto, aplicando los principios y políticas del Estado Ecuatoriano con respecto a las tecnologías de la información de software libre.

Para la implementación y desarrollo del Sistema Web se utilizó como base fundamental a PostgreSQL como servidor de bases de datos y la tecnología Java con sus extensiones y frameworks que son de código libre y que cumplen con las mismas características que las herramientas comerciales.

La metodología aplicada para la elaboración de la documentación respectiva es RUP (PROCESO UNIFICADO DE RATIONAL) la cual permitió mantener un orden mediante el proceso de desarrollo del proyecto.

El desarrollo del presente proyecto se lo realizó gracias a la ayuda e información proporcionada por cada uno de los responsables de los departamentos de la Cooperativa antes mencionada.

SUMMARY

This project has been done to address the needs of the Cooperative Savings and Credit Union Cochapamba by implementing an Integrated Financial Information System for which we developed a Web system that meets the specifications and requirements of the Cooperative.

Currently desktop applications are being replaced by systems or Web applications because they stand out mainly because of its usability and portability, reason to opt for this form of development, which was used for tools and open standards, applying principles and policies of the Ecuadorian State with respect to information technology free software.

For the implementation and development of Web System was used as the foundation to PostgreSQL database server and Java technology with its extensions and frameworks that are open source and meet the same features as commercial tools.

The methodology used to prepare the relevant documentation is RUP (Rational Unified Process) which allowed for an order through the process of project development.

The development of this project was made with the help and information provided by each of the heads of the departments of the above Cooperative.

TEMA

“Sistema Integrado de Información Financiera para la Cooperativa de Ahorro y Crédito Unión Cochapamba”

APLICATIVO

“Desarrollo e Implementación de los Módulos de Administración, Ahorros y Captaciones para la Cooperativa de Ahorro y Crédito Unión Cochapamba”

PROBLEMA

La Cooperativa de Ahorro y Crédito Unión Cochapamba maneja sus procesos de Administración, Ahorros y Captación a través de un sistema informático desactualizado, y cada uno de ellos se manejan independientemente lo que no les permita agilizar su trabajo y atención a sus clientes, es por ello que se ven en la necesidad de rediseñar estos procesos utilizando tecnología moderna y adecuada para este tipo de instituciones.

JUSTIFICACIÓN

El presente estudio de factibilidad se realiza para determinar la viabilidad del proyecto, el cual tiene como finalidad de automatizar los procesos de Administración y seguridades, Captaciones y Ahorros de la Cooperativa de Ahorro y Crédito Unión Cochapamba, esto con el objetivo de agilizar los trámites y papeleos que en él se realizan.

Mediante visitas realizadas a la Cooperativa se observó la falta de un sistema que les permita a ellos como Institución agilizar y automatizar los procesos que hasta ahora se los ha realizado de manera incorrecta repitiendo procesos en todos los departamentos, al implementar este proyecto se pretende reducir el tiempo de respuesta de las diferentes instancias y departamentos de la Cooperativa.

Además no cuenta con un profesional en el área de informática, ni con políticas de seguridad, dado que en la Cooperativa no existe separación de funciones, es decir no hay un profesional asignado a cada área, esto hace que la información sea vulnerable a manipulaciones por parte de cualquier persona y con ello propensa a fraudes de todo tipo.

Para el desarrollo e implementación del sistema se utilizará herramientas con licencias GPL ya que la institución no cuenta con las licencias de algún software en especial.

OBJETIVO GENERAL

Analizar, Diseñar e Implementar los Módulo de Administración, Ahorros y Captaciones para el Sistema Integrado de Información Financiera para la Cooperativa de Ahorro y Crédito Unión Cochapamba mediante el uso de herramientas libres y estándares abiertos, para automatizar y rediseñar los procesos de Administración, Ahorros y Captación que se realizan en la cooperativa.

OBJETIVOS ESPECÍFICOS

- 1.** Realizar un módulo que permita registrar en un sistema informático todos los procesos de: captaciones de la institución, de Administraciones y seguridades, los movimientos de cuentas y procesos de Ahorros de la institución,
- 2.** Investigar y establecer los procesos financieros que realiza la Cooperativa de Ahorro y Crédito Unión Cochapamba.
- 3.** Documentar la información y los procesos generados al desarrollar los módulos de Administración, Ahorros y Captaciones.
- 4.** Investigar y estudiar el uso de herramientas gratuitas apropiadas para utilizar en el correcto desarrollo del proyecto.
- 5.** Analizar las ventajas y desventajas que se obtiene al implementar sistemas web utilizando herramientas libres y estándares abiertos.
- 6.** Establecer las debidas seguridades del sistema para mantener protegida la información.

istema Integrado de Información Financiera FINANSYS



Introducción

- 1) Base legal de la Institución.
- 2) Situación actual diagnóstico.
- 3) Estructura
- 4) Información General

CAPÍTULO 1: INTRODUCCIÓN

1. Introducción

En este capítulo se describen datos importantes de la Cooperativa de ahorro y Crédito Unión Cochapamba (en adelante CUC), lo que permitirá conocer las principales actividades y funciones de la CUC.

1.1. Base legal de la institución.

A continuación se presenta las bases legales de la CUC.

1.1.1. Constitución y domicilio.

La “CUC” con domicilio en la parroquia de Ambuquí, Comunidad de Chaupi Guaranguí, Cantón Ibarra de la Provincia de Imbabura, se constituye como una cooperativa sin fines de lucro y se regirá por la ley de Cooperativas, su Reglamento General, por otras leyes que fueren aplicables y por un estatuto único aprobado por los directivos pertinentes. La responsabilidad de la Cooperativa ante terceros está limitada a su capital social y al capital que hubieren suscrito personal en la entidad.

La Cooperativa tendrá duración indefinida pero sin embargo podrá disolverse y liquidarse por las causales y en forma establecida en la ley de Cooperativas, su Reglamento General y las normas señaladas en el estatuto.

1.1.2. Objetivos de la cooperativa

Los objetivos de la Cooperativa son los siguientes:

- a) Promover la cooperación económica entre sus socios, para cuyo cumplimiento recibirá los ahorros, certificados de aportación y depósitos que estos realicen, efectuará cobros y pagos así como todas aquellas operaciones necesarias para el fortalecimiento de la cooperativa crediticia dentro del marco legal permitido por las cooperativas.
- b) Otorgar préstamos a sus miembros de conformidad al reglamento que para el efecto se establezca.

- c) Proporcionar a sus asociados mayor capacidad en lo económico y social, mediante una adecuada educación cooperativista.
- d) Establecer nexos dentro y fuera del país con entidades similares en beneficio de la cooperativa.
- e) Obtener fuentes de financiamiento interno y externo para el desarrollo de la institución.
- f) Establecer otros servicios y otras actividades que estén encuadradas en la ley y Reglamento de Cooperativas y otras leyes que sean aplicables, que conlleven al mejoramiento social y económico de sus miembros.
- g) Efectuar actividades inherentes a buscar financiamiento con la finalidad de capitalizar y otorgar más beneficios a los asociados.

1.1.3. Principios de la cooperativa

La cooperativa regulará sus actividades de conformidad y de acuerdo con los siguientes principios:

- a) Igualdad de obligaciones y derechos de los socios que llenen los requisitos establecidos en la Ley de Cooperativas, su Reglamento General y el Estatuto.
- b) Control democrático, un socio es un voto.
- c) Distribución de los excedentes económicos entre los socios en proporción a las operaciones o al trabajo efectuado por los socios de la cooperativa.
- d) Interés limitado sobre los certificados de aportación que en ningún caso será mayor al señalado por la Ley o por el organismo estatal competente.
- e) Neutralidad política y religiosa.
- f) Fomento de la educación cooperativa.
- g) Integración cooperativista.

1.1.4. Socios fundadores

La Cooperativa, inicio sus actividades con la colaboración de 42 socios fundadores, todos ellos domiciliados en la comunidad de Chaupi Guaranguí, Parroquia de Ambuquí, del cantón Ibarra.

1.2. Situación actual diagnóstico.

La situación actual de la cooperativa se puede ver reflejada a través de los ítems que se presenta a continuación.

1.2.1. Misión

“Ser una Cooperativa rural altamente eficiente y de amplia cobertura, con una permanente innovación de productos y servicios de ahorro y crédito con una cultura de calidad en todo el recurso humano”.

1.2.2. Visión

“La Cooperativa de Ahorro y Crédito “Unión Cochapamba” hasta el año 2015, desarrollará e implementará estrategias administrativas y financieras que nos permitan tener sucursales grandes y competitivas, que generen microempresas sin descuidar el medio ambiente, de tal manera que sus socios interactúen como una familia en armonía con Dios y la naturaleza”.

1.2.3. Valores corporativos

- a) **Servicio:** En la atención a nuestros clientes siempre demostramos entusiasmo, cordialidad, respeto y amabilidad porque sentimos la satisfacción por la asistencia a los demás.
- b) **Cumplimiento:** Mediante nuestra labor diaria demostramos el compromiso en la entrega de soluciones a problemas financieros del sector rural, este valor se refleja en la responsabilidad y organización con que tratamos a nuestros clientes.
- c) **Compromiso:** Demostrada en la convicción de nuestro grupo de trabajo en torno a los beneficios que trae el desempeño responsable y organizado.
- d) **Honestidad y transparencia:** Base de la relación con nuestros usuarios garantizando la integridad, el respaldo y seguridad.
- e) **Responsabilidad social:** Es la contribución activa y voluntaria al mejoramiento social, económico y ambiental con los sectores vulnerables del sector rural.

1.2.4. FODA

La Cooperativa es una institución que tiene un buen prestigio financiero y es bien recibida por las comunidades rurales de la provincia de Imbabura, pero existen algunos aspectos que alteran el desempeño normal de la institución, como son el tecnológico, hacen que instituciones de estas características tengan que buscar mejores oportunidad esas innovaciones constantes. A continuación el diagnóstico a través de un análisis FODA.

| FORTALEZAS | OPORTUNIDADES |
|---|---|
| <ol style="list-style-type: none"> 1. Prestigio institucional. 2. Directivos y personal comprometidos. 3. Alianzas Estratégicas permiten brindar mayores servicios. 4. Morosidad decreciente. 5. Solvencia patrimonial. 6. Alto nivel de rentabilidad. 7. Servicio Social de la Comisión Educación. 8. Ayuda social mediante actividades del mismo orden. | <ol style="list-style-type: none"> 1. Desarrollo de micro y pequeña empresa. 2. Posibilidad de utilizar nuevas fuentes de ingresos para aumentar los fondos de la CUC. 3. Utilización de línea de Corporación Financiera Nacional (CFN) para financiamiento de largo plazo. 4. Apoyo de entidades del exterior como el proyecto Fondo Ítalo Ecuatoriano (FIE). 5. Apertura de Sucursales de la CUC. 6. Creciente desarrollo cooperativo en el país. |
| DEBILIDADES | AMENAZAS |
| <ol style="list-style-type: none"> 1. Servicio segmentado a comunidades de las zonas rurales 2. Falta de sucursales 3. Bajo nivel de control interno. 4. No contar con un sistema de información integrado 5. Eficiencia financiera inferior al promedio 6. Demora en el trámite de créditos. 7. Datos erróneos previos a la presentación de informes financieros. 8. Corrección de errores manual. 9. No existe plan estratégico. | <ol style="list-style-type: none"> 1. Fortalecimiento del Sistema Financiero de la Cooperativa. 2. Contexto socio-político del país. 3. Situación económica financiera del país. 4. Mercado competitivo. 5. Política estatal sobre el sistema cooperativo. 6. Creación de nuevas instancias reguladoras. |

Tabla 1: Análisis de la Cooperativa (FODA)

Fuente: Propia

1.2.5. Situación actual

La Cooperativa de Ahorro y Crédito Unión Cochapamba realiza una serie de procesos distribuidos en diferentes áreas específicas, dichos procesos se los realiza mediante el uso de un Sistema programado en C++, el cual muestra limitaciones debido a que no controla y

administra todos los procesos de forma transparente y unificada, por lo que se emplea mucho tiempo en procesar la información. Los datos no se guardan en ninguna base de datos, se manejan y almacenan de acuerdo a cada módulo en archivos visibles para el usuario, lo que hace que este sistema sea vulnerable y susceptible a cualquier tipo de fraude o estafa. Todos los módulos funcionan de forma separada e independiente, lo que conlleva a que los usuarios repitan información importante en cada módulo y proceso, ocasionando pérdida de tiempo e inconsistencias en la información.

El Sistema actual no le permite a la Cooperativa acoplarse a las nuevas ordenanzas y manejo de cuentas emitidas por los respectivos órganos reguladores de entidades financieras, ya que no es flexible, modificable o adaptable e integrado.

1.2.6. Diagnóstico 6M



Figura 1: Diagnóstico de la Cooperativa (6 M)

Fuente: Propia

1.2.7. Objetivos operativos y actividades del plan anual operativo.

1. Ser una institución financiera rural posicionada en su área de intervención, con socios que están empoderados, con capacidad de dirigir y administrar los componentes socio-organizativos y financieros de la Cooperativa, en alianza con entidades afines. (FODA).

2. Capacitar a los nuevos directivos en finanzas populares.
3. Mantener alianzas con CODESARROLLO y FODEMI, para acceder a créditos.
 - a. Rotar al menos 200 créditos.
 - b. Realizar tres asambleas generales, dos en el mes de enero, la una extraordinaria, para las elecciones de los nuevos directivos.
 - c. Realizar cuatro reuniones de consejo administrativo.
 - d. Realizar cuatro reuniones de consejo de vigilancia.
4. Preservar el medio ambiente como factor de desarrollo para las comunidades a través de la reforestación y forestación anual de por lo menos 20.000 plantas.
 - a. Fomentar la capacitación en el Manejo de los Recursos Naturales a cada beneficiario.
 - b. Producción de 8 .000 mil plantas forestales para las parcelas campesinas de los socios de la CUC.
5. Fomentar la cultura del ahorro y fortalecer el patrimonio de la Cooperativa vía incremento Certificados de Aportación por parte de los socios con un aporte de \$20 anuales.
 - a. Promover el ahorro y los depósitos a plazo fijo.
 - b. Reformar el Reglamento Interno en lo que corresponde a los Certificados de Aportación.
6. Promover la creación de micro emprendimientos productivos como un nuevo modelo de desarrollo comunitario, mejorando el nivel de vida de los socios.
 - a. Establecer 2 alianzas estratégicas para el fomento de micro emprendimientos.
 - b. Apoyar en lo administrativo, a dos micro emprendimientos ya existentes (quesera y asociación 20 de diciembre).
 - c. Apoyar 4 micro-emprendimientos en la zona.
7. Fortalecer el talento humano y administrativo de la Cooperativa.
 - a. Fortalecer el equipo técnico - administrativo de la CUC con enfoque de género a través de capacitación, acuerdos y convenios.
 - b. Cumplir con las normas de seguridad y control interno de los bienes y recursos de la CUC (Seguros, Garantías).

1.3. Estructura

La estructura de la CUC se puede representar de varias formas una de ellas es la que se muestra a continuación en la Figura 2.

1.3.1. Organigrama estructural

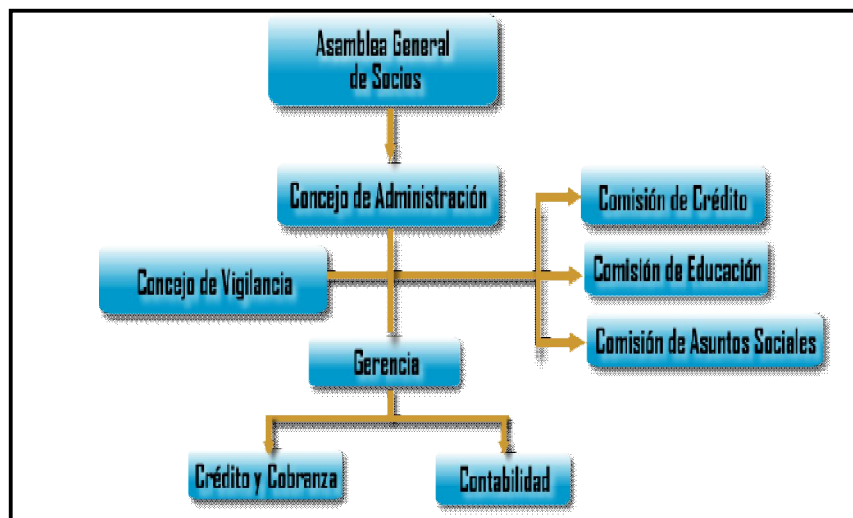


Figura 2: Organigrama Estructural
Fuente: CUC

La dirección, administración y control interno de la Cooperativa se ejercerá por medio de los siguientes organismos:

a) Asambleas generales de socios.

Es la máxima autoridad de la Cooperativa y está constituida por todos los socios que registrados respectivamente y que estuvieren en uso y goce de sus derechos.

b) Concejo de administración.

Es el organismo directo de la Cooperativa quien se atribuye la elección de Presidente, Gerente, Secretario y vocales pertinentes de acuerdo con la ley.

c) Concejo de vigilancia.

Está compuesto por un número de miembros elegidos por la Asamblea General y cuyo objeto es vigilar la elección del presidente y secretario.

d) Gerencia.

Administración General de la cooperativa.

e) Comisiones.

1. **Comisión de crédito.-** Estará compuesta por miembros seleccionados por la Asamblea General o el Consejo de Administración, decidirá todo lo relacionado con las solicitudes de préstamo de los socios.
2. **Comisión de educación.-** Nombrada por el Consejo de Administración elaborará, organizará, promoverá y presentará planes de trabajo educacional.
3. **Comisión de asuntos sociales.-** Nombrada por el Consejo de Administración tiene por finalidad estudiar, analizar y solucionar los problemas sociales, así como programar actividades de orden social.

1.4. Información general

La siguiente tabla muestra datos importantes de ahorros de los clientes de la CUC.

1.4.1. Clientes, segmentado por valores

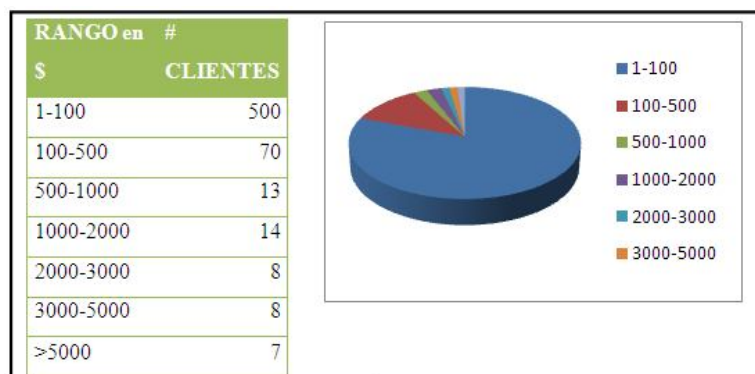


Figura 3: Gráfico clientes cooperativa
Fuente: CUC

1.4.2. Préstamos

| Plazos y Montos de Préstamos | |
|-------------------------------------|---|
| Crédito | Descripción |
| Pecuarios | Se prestará un máximo de \$1500 USD, plazo máximo 2 años, pagos mensuales, trimestrales y máximo semestrales. |
| Agrícolas | Se prestará un máximo de \$1500 USD, plazo dependiendo del cultivo máximo 18 meses y pagos serán mensuales, trimestrales y máximo semestrales. |
| Salud | Serán solo en casos de enfermedades del socio o de los familiares, plazo máximo de 1 año, pagos trimestrales y se prestará máximo hasta \$1000 USD. |
| Viajes de Trabajo | Se entregará a personas que requieran fondos para viajes de trabajo en el exterior, se presta un monto máximo de \$1000 USD. |
| Viviendas | Plazo máximo 4 años, pagos mensuales, trimestrales, semestrales, monto máximo \$5000 USD. |
| Compra de Terrenos | Plazo máximo será de 4 años pagaderos mensuales y trimestrales y un monto máximo de \$5000 USD. |
| Educación | Plazo hasta 18 meses, monto máximo hasta de \$1500 USD. |
| Crédito forestal | Plazo hasta 3 años, monto hasta \$2000 USD, pagos mensuales y trimestrales. |
| Crediempresas | Plazos hasta 4 años, monto máximo \$5000 USD, pagos mensuales y trimestrales. |

Tabla 2: Tabla de Préstamos
Fuente: CUC

Sistema Integrado de Información Financiera

FINANSYS



Marco Teórico

- 1) Normas Legales
- 2) Herramientas y estándares de Código libre
- 3) Seguridad en Sistemas Web
- 4) Sistemas Operativos
- 5) Servidor de Aplicaciones
- 6) Bases de Datos
- 7) Plataformas y Frameworks
 - Netbeans IDE
 - Java
 - Java EE
 - Hibernate
 - JSF (Java Server Faces)
 - RichFaces como framework RIA
- 8) Herramienta de Reportes

Capítulo 2: Marco Teórico

2. Marco teórico

En este capítulo se describen las características, funcionamiento y definiciones de cada una de las herramientas utilizadas para el desarrollo e implementación del presente proyecto.

2.1. Normas legales

A continuación se presenta una definición de las leyes y normas que rigen y controlan a las Cooperativas.

2.1.1. Ley general de instituciones del sistema financiero

“Esta ley regula la creación, organización, actividades, funcionamiento y extinción de las instituciones del sistema financiero privado, así como la organización y funciones de la Superintendencia de Bancos y Seguros, en la órbita de su competencia, entidad encargada de la supervisión y control del sistema financiero, en todo lo cual se tiene presente la protección de los intereses del público”. [LA COMISIÓN DE LEGISLACIÓN, 2012]

2.1.2. Ley de cooperativas

“La Ley de Cooperativas rige la constitución, organización, funcionamiento y liquidación de las cooperativas de ahorro y crédito que realizan intermediación financiera con el público, así como de las cooperativas de segundo piso, sujetas al control de la Superintendencia de Bancos y Seguros.

Las cooperativas se rigen por los valores y principios universales del cooperativismo, los mismos que deberán ser aplicados en las resoluciones de carácter general emitidas para el sector cooperativo. Los principios del cooperativismo son:

- a) Adhesión abierta y voluntaria.
- b) Control democrático de los socios.
- c) Participación económica de los socios.
- d) Autonomía e independencia.
- e) Educación, capacitación e información.

f) Cooperación entre cooperativas.

g) Compromiso con la comunidad.”[**Super Intendencia de Bancos del Ecuador, 2009**]

2.1.3. Normas de Basilea

Las normas de Basilea son una guía para conformar políticas y procedimientos que faciliten la coordinación en el sistema de control eficaz de la banca, teniendo como objetivo el asegurar que las instituciones financieras midan, vigilen y controlen adecuadamente los riesgos de liquidez y de mercado, como una garantía de solvencia y de confianza para sus clientes y asociados en el caso de Cooperativas de Ahorro y Crédito.

“El contenido de las Normas de Basilea son el resultado de las decisiones del grupo de los 10 países más industrializados del mundo, quienes representados por los bancos centrales y organismos de supervisión de cada país miembro, han elaborado normas de regulación prudencial para el sistema bancario de cada nación, éste Comité periódicamente revisa y ajusta dicha normativa a los requerimientos del momento, actualmente ya existe Basilea III la cual está enfocada a prevenir crisis financieras como las ocurridas últimamente en EEUU”.
[**Normas de Basilea, 2004**]

2.2. Herramientas y estándares de código libre

Una herramienta al ser de código libre significa que se puede usar, modificar, compartir sin tener la necesidad de pagar licencia alguna por su uso, a continuación se describen herramientas y estándares que se usan para el diseño de aplicaciones web.

2.2.1. WORLD WIDE WEB

Es el nombre que se le da a las siglas WWW conocida también como la Red informática mundial o simplemente la web a la cuál se le considera como un sistema de distribución de información basado en hipertexto o hipermedia vinculados y accesibles a través de Internet por medio de un navegador web como Internet Explorer, Mozilla Firefox, Chrome, etc., los mismos que permiten al usuario visualizar sitios web compuestos de páginas web donde se

puede encontrar varios elementos como texto, imágenes, vídeos, archivos y demás aplicaciones multimedia, y navega a través de ellas usando hiperenlaces conocidos también como hyperlinks o simplemente links.

2.2.2. HTML (Hyper Text Markup Language)

Es un lenguaje sencillo que nos sirve para la creación de páginas web o escribir hipertexto, es decir texto presentado de forma estructurada y fácil de entender. Una página web se puede entender como el documento que aparece en el intérprete de este lenguaje como son los navegadores o browsers.

Una de las características más importantes de este lenguaje para un programador es que no es necesario ningún programa especial para crear una página Web ya que se ha conseguido que se puedan crear páginas con cualquier ordenador y sistema operativo utilizando cualquier editor de texto como el que acompañan a todos los sistemas operativos como block de notas en Windows, gedit Linux, etc.

Una página escrita con el lenguaje HTML no es más que texto normal, escrito con cualquier editor, y al que cuando se le quiere dar algún aspecto especial o formato, como por ejemplo el tamaño de la letra, color, etc., habrá que acompañar de ciertos códigos para indicar el formato deseado. A estos códigos se les llama elementos del lenguaje.

2.2.2.1. Elementos de HTML

A las instrucciones que forman el lenguaje HTML se les llama elementos o también se les conoce como etiquetas de HTML.

La notación de los elementos consiste en los símbolos < menor y > mayor que encierran dentro de ellos una instrucción como se muestra a continuación:

<CENTER>Hola mundo</CENTER>

El código anterior significa que el texto se mostrara en el centro del documento.

Las etiquetas o elementos de HTML deben estar bien estructuradas indicando cuando se abre y cuando se cierra un elemento, para ello los elementos dividen en dos grupos llenos y vacíos.

1. Elementos llenos

Se forman escribiendo la instrucción correspondiente seguida del texto al que se quiere aplicar la instrucción y se termina repitiendo la instrucción pero con una barra inclinada inmediatamente antes de la misma.

Por ejemplo, el elemento H1 que sirve para dar el máximo tamaño al texto, se escribirá de la siguiente manera:

<H1> Texto de prueba </H1>.

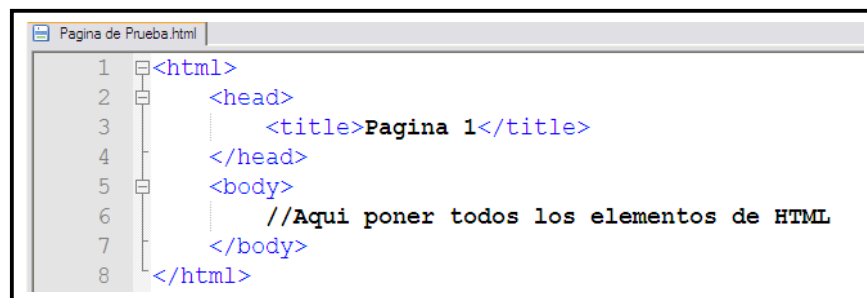
Si no se pone el cierre del elemento </H1> todo el resto de la página tendrá el mismo tamaño grande.

2. Elementos vacíos

Los elementos vacíos se escriben como los llenos, pero no es necesario poner la instrucción de cierre de elemento. Esto se debe a que estos elementos no producen un efecto sobre el texto si no sobre la posición del documento donde se añade la etiqueta, generalmente se utilizan para separar bloques de texto. Por ejemplo, el elemento <HR> que sirve para dibujar una línea horizontal.

2.2.2.2. Estructura de un documento HTML

Los documentos escritos en HTML están estructurados en dos partes diferenciadas: la HEAD que es la cabecera del documento y el BODY es el cuerpo del documento.



```
1 <html>
2   <head>
3     <title>Pagina 1</title>
4   </head>
5   <body>
6     //Aquí poner todos los elementos de HTML
7   </body>
8 </html>
```

Figura 4: Estructura de un documento HTML
Fuente: Propia

2.2.3. CSS (Cascading Style Sheets)

Es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos definidos anteriormente como HTML o XHTML, y se presenta como la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas. Este lenguaje tiene una sintaxis muy sencilla ya que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos como se muestra a continuación.

```

10
11  /* basic elements */
12  html {
13      margin: 0;
14      padding: 0;
15  }
16  body {
17      font: 75% georgia, sans-serif;
18      line-height: 1.88889;
19      color: #555753;
20      background: #fff url(blossoms.jpg) no-repeat bottom right;
21      margin: 0;
22      padding: 0;
23  }

```

Figura 5: Ejemplo CSS
Fuente: Propia

Una hoja de estilos se forma en una serie de reglas, cada regla se forma de uno o más selectores y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor, como se mostró en la imagen anterior.

2.2.3.1. Funcionamiento de CSS



Figura 6: Forma de uso de CSS

Fuente: http://www.scourdesign.com/articulos/tutoriales/css/css14_2.php

Para dar formato a un documento HTML y poner en funcionamiento lo que hemos escrito en el documento CSS, se lo realiza de las siguientes formas:

1. **Un estilo en los elementos de HTML.**-Es una de las formas menos utilizadas para dar formato ya que esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento en concreto. Ejemplo:

```
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
```

2. **Incluir CSS en el mismo documento HTML.**- Es una hoja de estilo que está incrustada dentro de un documento HTML, ubicándole dentro la cabecera del documento específicamente dentro del elemento <head>. De esta manera se emplea solamente cuando se quiere dar formato o estilosa un número pequeño de elementos de una determinada pagina. Ejemplo:

```
<style type="text/css">
  p { color: black; font-family: Verdana; }
</style>
```

3. **Definir una hoja de estilo externa.**-Es una hoja con extensión CSS donde se pueden incluir todo tipo de estilos para los elementos de HTML, a los cuales se enlazan mediante la etiqueta <link> dentro de la cabecera de la página web.

Se pueden crear todos los archivos CSS que sean necesarios y cada documento HTML puede enlazar tantos archivos CSS como necesite. Ejemplo:

```
<head>
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen" />
</head>
```

2.3. Seguridad en sistemas web

Para la implementación de un sistema web se debe tener en cuenta el riesgo que implica mantenerla en producción a través de la web, para ello existen varias seguridades que son indispensables para un buen funcionamiento de la aplicación.

2.3.1. Introducción

Un efecto secundario del enorme crecimiento que ha tenido el Internet es mantener la integridad y privacidad de información tanto personal como profesional ya que en el Internet se encuentran funcionando tiendas online, negocios que mueven grandes cantidades de dinero como bancos, redes sociales las cuales contienen grandes cantidades de información personal y delicada de la vida privada de sus miembros.



Figura 7:Seguridad en sistemas web
Fuente: Basada en <http://www.websense.com>

“Desde muchos puntos de vista, podemos creer sin dudar que el punto más crítico de la seguridad del Internet, lo tienen las piezas que intervienen de forma directa con las masas de usuarios, los servidores web. Respecto a los servidores web, es común escuchar sobre fallas en los sistemas de protección de los servidores más frecuentemente utilizados (Apache, IIS, etc.), o en los lenguajes de programación en los que son escritas las aplicaciones que son ejecutadas por estos servidores. Pero es un hecho, que la mayoría de los problemas detectados en servicios web no son provocados por fallas intrínsecas de ninguna de estas partes, ya que una gran cantidad de los problemas se generan por malos usos por parte de los programadores.”[Wikilearning, 2005]

La mayoría de los problemas de seguridad en los sitios web se encuentran a nivel aplicación ya que son el resultado de la errónea escritura de código por parte de los programadores,

debemos entender que programar aplicaciones web seguras no es una tarea fácil, ya que requiere por parte del programador, no únicamente mostrar atención en cumplir con el objetivo funcional básico de la aplicación, sino una concepción general de los riesgos que puede correr la información contenida, solicitada y recibida por el sistema. Las aplicaciones cliente/servidor que utilizan el protocolo HTTP para interactuar con los usuarios u otros sistemas están siempre propensos a cualquier tipo de ataques ya que el cliente comúnmente utilizado por los usuarios es un navegador.

2.3.2. Seguridad en el cliente

Los usuarios deben contar con navegadores y plataformas seguras libres de virus y vulnerabilidades que garantiesen la privacidad de los datos del usuario.



Protocolo seguro en navegadores

Fuente: <http://onlinezatealultima.blogspot.com/>

2.3.3. Seguridad en el servidor

El desarrollo de una aplicación web requiere de un sinnúmero de herramientas para su funcionalidad como servidores web, servidores de aplicaciones, servidores de bases de datos, lenguajes de servidor, etc.

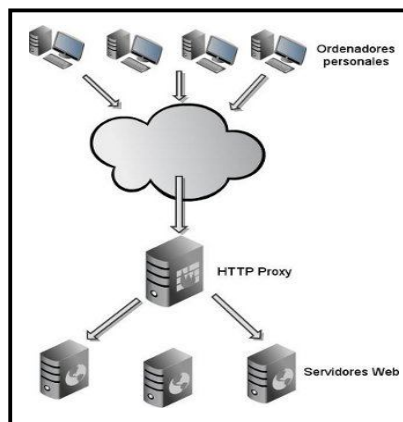


Figura 8: Sistemas de seguridad para servidores

Fuente: <http://rm-rf.es/sobre/apxs/>

Estas herramientas pueden plantear problemas que comprometan a la aplicación como:

- a) Vulnerabilidades debidas al uso de versiones no actualizadas
- b) Configuraciones por defecto inadecuadas
- c) Activación de cuentas por defecto

Para evitar este tipo de problemas las herramientas deben estar actualizadas y bien configuradas para impedir ataques a la aplicación.

- I. **El servidor web.-** Proporciona muchos servicios y es muy probable que algunos de ellos no sean necesarios para el funcionamiento de la aplicación web de tal manera que se recomienda deshabilitar los servicios que no se utiliza para el funcionamiento de la aplicación web.
- II. **Servidor de aplicaciones.-** Es muy conveniente revisar periódicamente los ficheros de log (access_log y error_log en el servidor de aplicaciones Apache) para detectar posibles ataques al servidor.
- III. **Servidor de bases de datos.-** Se debe vigilar la configuración por defecto del servidor y se debe tener en cuenta las siguientes recomendaciones:
 - a) No ejecutar el servidor como root
 - b) No dar a ningún usuario salvo al root permiso de acceso a la tabla de usuarios
 - c) Asegurarse de que el root tiene un password
 - d) Restringir el acceso remoto al servidor
 - e) No dar a un usuario más permisos que los estrictamente necesarios
 - f) Almacenar los datos sensibles de forma encriptadas

2.3.4. Seguridad en la aplicación



Figura 9: Seguridad

Fuente: <http://www.nexica.com/es/firewall-aplicaciones>

Para la implementación de la seguridad en las aplicaciones web se debe considerar las siguientes alternativas:

1. La propia aplicación puede implementar un mecanismo de autenticación que implica la presentación de un formulario para que el usuario introduzca sus credenciales.
2. Encriptación de contraseñas y campos sensibles a través de algoritmos seguros como md5, SHA1, etc.
3. Una vez que el usuario se ha autenticado introduciendo su nombre de usuario y su clave, es preciso mantener esta autenticación en cada conexión subsiguiente realizando la apertura de sesiones.
4. Es el acto de comprobar si un usuario tiene el permiso adecuado para acceder a un cierto menú o información y realizar una determinada acción, una vez que ha sido autenticado.
5. El problema más frecuente que presentan las aplicaciones web es no validar correctamente los datos de entrada.
6. Siempre que se envíen datos sensibles hay que acompañarlos de un identificador de sesión y comprobar que el usuario asociado a la sesión tiene acceso a la información requerida.
7. Si la aplicación web utiliza las cabeceras recibidas del cliente, hay que tener en cuenta que éstas pueden haber sido manipuladas, por lo que deben ser verificadas.
8. Aceptar únicamente datos válidos conocidos, es la estrategia más adecuada y sólo deben aceptarse aquellos datos que se adaptan a lo esperado. Por ejemplo, si un password debe contener letras de la A a la Z y dígitos del 0 al 9, debe verificarse que la entrada es una cadena, que sólo contiene esos caracteres y que tiene una longitud válida.
9. Nunca debe confiarse en la validación de datos en el cliente ya que pueden manipularse con facilidad, para ello todas las validaciones de datos debe realizarse en el servidor

2.3.5. Seguridad en la comunicación

SSL (Secure Socket Layer) es un protocolo para asegurar el transporte de datos entre el cliente y el servidor web, este protocolo fue “diseñado inicialmente por Netscape, hoy día es soportado por la mayoría de los servidores web, podemos reconocer una conexión HTTP sobre SSL porque aparece el prefijo ‘https’ en lugar de ‘http’ en la URL



Figura 10: HTTP sobre SSL.

Fuente: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SSLenTomcat>

SSL proporciona una comunicación segura entre cliente y servidor permitiendo la autenticación mutua, el uso de firmas digitales y garantizando la privacidad mediante encriptación”.

2.4. Sistemas operativos(SO)

Es el principal elemento que se necesita para que cualquier aplicación funcione siempre y cuando estas sean compatibles con el Sistema Operativo (en adelante SO), pero para la implementación de las aplicaciones web no es una pieza clave, puesto que el FINANSYS es totalmente portable lo que quiere decir que no importa el SO que se use para ejecutar la aplicación ya que es Sistema Web y por ende funciona a través de un navegador o browser que puede estar instalado en cualquier SO.

2.4.1. Definición

Sistema Operativo es un software o programa que controla y administra las funciones y servicios de un computador y además permite la ejecución de otros programas compatibles con éste. El SO es software que permite interactuar con los dispositivos de hardware y software del Computador.

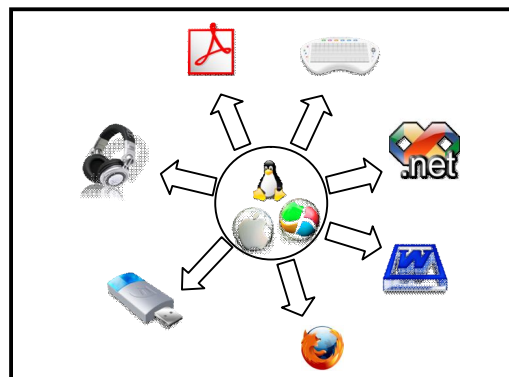


Figura 11: Sistemas Operativos

Fuente: Propia

“Un SO es el software básico de una computadora que provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario.”[Anonimo, 2012]

2.4.2. Características y funciones de los sistemas operativos

Las principales características de los Sistemas Operativos son las siguientes:

1. La instalación de un SO hace más conveniente el uso de la computadora y por ende el uso de programas.
2. Hacen posible que se use de manera eficiente los recursos de la computadora y la correcta utilización de cada uno de ellos.
3. Pueden evolucionar aumentándoles nuevas funciones y servicios, es decir actualizaciones.
4. Administra correctamente cada uno de los dispositivos de hardware del computador, como el mouse, teclado, parlantes, impresoras, etc.
5. Organiza la información y le ordena creando accesos rápidos para poder mostrarla cuando el usuario lo requiera.
6. Facilita el acceso y la comunicación entre usuario, software, y dispositivos de hardware.

Las funciones que realiza un Sistema Operativo al interactuar y responder a las órdenes de los usuarios son las siguientes:

1. Interpreta los comandos que el usuario digita a través del teclado u otros dispositivos de entrada de datos.
2. La información generada por los usuarios se almacena ordenadamente en dispositivos de almacenamiento como discos duros, memoras extraíbles, etc., y además permiten la administración y manipulación de esta información.
3. Sirven como base para la instalación de software adicional siempre que estos sean compatibles con dicho SO.

2.4.3. Ejemplos de sistema operativo

A continuación se muestra un comparativo entre sistemas operativos de código libre vs sistemas operativos propietarios.[Anonimo, 2012]

| Aspecto | GNU/Linux | Windows |
|---|--|--|
| Filosofía | El sistema es libre, cualquiera lo puede usar, modificar y distribuir. | Pertenece a Microsoft, única compañía que lo puede modificar. |
| Precio | Gratis, tantas licencias como se desee. | Según las versiones, cientos de euros, cada licencia. |
| Desarrollo | Miles de voluntarios en todo el mundo, cualquiera puede participar, pertenece a la "comunidad". | Lo desarrolla Microsoft, que vende algunos datos técnicos relevantes y oculta otros. |
| Código fuente | Abierto a todos. | Secreto empresarial. |
| Estabilidad | Muy estable, es difícil que se quede colgado. Los servidores que lo usan pueden funcionar durante meses sin parar. | Poco estable, es común verse obligado a reiniciar el sistema. Los servidores no admiten más allá de un par de semanas sin reiniciar. |
| Seguridad | Extremadamente seguro, tiene varios sistemas de protección. No existen virus para Linux. | Muy poco seguro, existen miles de virus que atacan sistemas Windows. |
| Facilidad de uso | En muchas tareas, poca. Día a día mejora este aspecto. | Cuando funciona, es muy sencillo de manejar. |
| Controladores de hardware | Desarrollados por voluntarios; algunos dispositivos no funcionan en absoluto porque sus fabricantes ocultan los detalles técnicos. | Los fabricantes de dispositivos siempre los venden con controladores para Windows, todos deben funcionar en pocos momentos. |
| Difusión | Poco extendido en hogares y oficinas, muy extendido en servidores. | Copa todo el mercado, salvo el de servidores. |
| Disponibilidad de programas. | Existen programas para casi todas las facetas, pero no hay tanta variedad como los programas para Windows. | Miles y miles de programas de todo tipo que se instalan con facilidad. |
| Precio de los programas. | Existen programas de pago, pero lo más habitual es que sean libres. | La mayor parte de los programas son de pago. |
| Comunicación con otros sistemas operativos. | Lee y escribe en sistemas de archivos de Windows, Macintosh, etc. Por red, se comunica con cualquier otro sistema. | Sólo lee y escribe sus propios sistemas de archivos, y presenta incompatibilidades entre algunas de sus versiones. |

Tabla 3:Comparativo de Sistemas Operativos libres y Proprietarios

Fuente: <http://www.dacostabalboa.com/es/tabla-comparativa-windows-y-linux/1124>

2.4.4. Sistema operativo Fedora

El sistema operativo Fedora en el presente proyecto se utilizó para el desarrollo de la aplicación debido a que mantiene la funcionalidad básica de un servidor y de un SO de escritorio, a continuación se describe sobre este SO.

“Es un sistema operativo basado en Linux y mantiene la estructura de un conjunto de software que hace funcionar al computador, permitiendo utilizar este SO como opción para remplazar otros sistemas operativos, o utilizarlo en conjunto con éstos, por ejemplo mantener Microsoft Windows o Mac OS en una partición diferente.”[Fedora, 2012]

Es una distribución de fácil manejo ya que es orientada para propósitos generales basada en RPM (originalmente llamado Red Hat Package Manager, pero se convirtió en acrónimo recursivo), es por ello que se mantiene gracias a una comunidad internacional de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías, contando con el respaldo y la promoción de Red Hat.

2.4.4.1. Características principales

1. Fedora soporta las arquitecturas x86, x86-64 y PowerPC.
2. Soporta redes instaladas sobre HTTP, FTP y NFS.
3. El entorno de escritorio por defecto es GNOME, e incluye KDE en algunas versiones.
4. El gestor de arranque es GNU GRUB por defecto.
5. Fedora está diseñado para ser fácil de instalar y configurar, incluyendo para esto instaladores y herramientas gráficas.
6. El sistema de archivos por defecto es ext3 sobre LVM.
7. Sólo contiene una pequeña selección de paquetes de software (unos 8 mil), pero existen varios almacenes disponibles con software extra para completar esta distribución, pero muchas no son oficiales de Fedora.
8. Su navegador por defecto es Firefox desde su versión Fedora Core 3 y superior.
9. Incluye el paquete ofimático OpenOffice.org desde su versión 4.

2.4.5. Sistema operativo CentOS

La implementación y funcionamiento de la aplicación del presente proyecto se realiza a través de CentOS que es un sistema operativo que mantiene todas las características de un servidor y lo más importante que es estable y gratuito. A continuación se describe sobre este SO.

“Es una distribución de Linux basada en las fuentes libremente disponibles de Red Hat Enterprise Linux y cada versión es mantenida durante 7 años por medio de actualizaciones de seguridad, las versiones nuevas son liberadas cada 2 años y actualizadas regularmente cada 6 meses para el soporte de hardware nuevo, esto le hace un sistema robusto.”[Centos, 2012]

Este SO se puede usar como servidor Web, servidor de correo, servidor de bases de datos o servidor de aplicaciones ya que incorpora todas las sofisticadas funcionalidades de

compatibilidad con Windows incluidas en Red Hat Enterprise Linux. Este puede ser usado como controlador de dominios primarios o secundarios, o como servidor de clientes Windows basado en Samba y SMB.

“Se puede utilizar como un sistema de escritorio Linux plenamente funcional, debido a que cuenta con la mayoría de las herramientas de escritorio indispensables como Open Office, Mozilla Firefox, Evolution, Gaim, X-chat, Konqueror, Gimp, Gnome, KDE, etc., y de esta manera se puede trabajar como servidor empresarial, servidor para la oficina o incluso como un sistema común de escritorio, ya que es una distribución estable, de alto rendimiento, fácil de administrar y de utilizar”. [CentOS, 2008]

2.4.5.1. Características

1. Es fácil de realizar mantenimientos.
2. Es idóneo para el uso a largo plazo en entornos de producción.
3. Su desarrollo se mantiene en constante actividad (actualizaciones)
4. Es uno de los SO más estables.
5. El soporte de actualizaciones se realiza cada 7 años
6. Mantiene el soporte para varios programas comerciales y libres, que soportan RedHat Enterprise Linux.
7. Tiene varios repositorios que alcanzan más de los 10mil paquetes (entre base, extras, rpmforge, atrpms y el de karan).
8. Yum hace todo el trabajo de instalación o desinstalación. [Oliva, 2009]

2.5. Servidor de aplicaciones

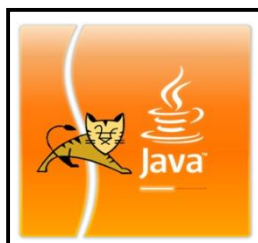


Figura 12: Logo de Tomcat

Fuente: <http://jackfiallos.com/2011/05/17/>

Es la herramienta más importante para la implementación ya que esta permita que la aplicación funcione y se muestre a través de un navegador.

2.5.1. Introducción a Apache Tomcat

“También llamado JakartaTomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Este servidor de aplicaciones implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems”. [Wikipedia, 2012]

Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets ordenados por el usuario. A los contenedores de Servlets podemos dividirlos en:

- Contenedores de Servlets Stand-alone o contenedores de Servlets Independientes que son parte integral del servidor web, este es el caso cuando se usa un servidor web basado en Java. Ejemplo: Java Web Server.
- Cuando los servidores de aplicaciones no están basados en java que son la mayoría de los existentes tenemos otra clasificación de contenedores de Servlets llamados Contenedores de Servlets Dentro de Proceso los que se componen de una combinación de un plugin para el servidor web y una implementación de contenedor Java, donde el plugin abre una JVM (Máquina Virtual del Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él.
- Contenedores de Servlets fuera de proceso, son cuando el contenedor de Servlets es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM pero fuera del servidor web lo contrario del anterior que se ejecutaba dentro del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets a través de TCP/IP).

El servidor de aplicaciones Tomcat también puede utilizarse como un contenedor solitario utilizado de esta forma para desarrollo y depuración o como plugin para un servidor web existente como el Apache, IIS y Netscape.

2.5.2. Estructura del servidor

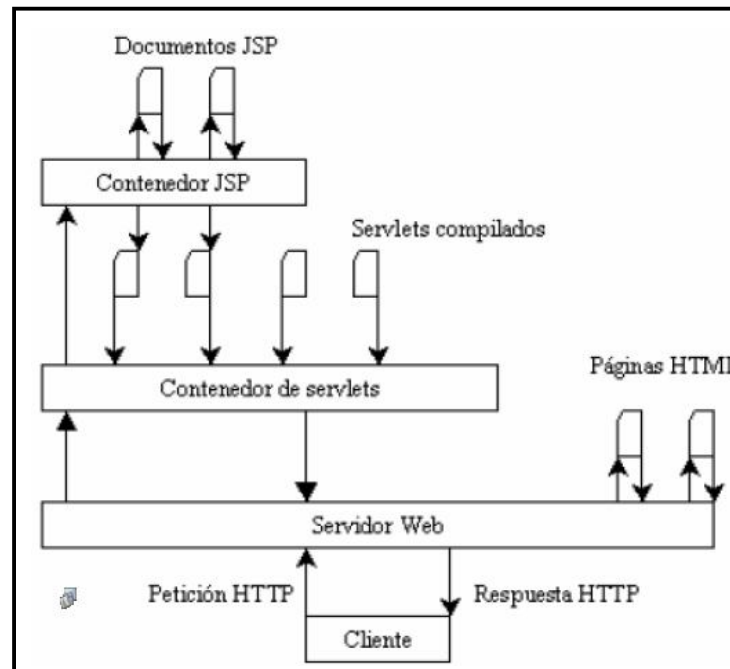


Figura 13: Estructura de Tomcat
Fuente: <http://profesores.elo.utfsm.cl>

Si al descomprimir una de las distribuciones binarias de Tomcat deberíamos tener la siguiente estructura de directorios, teniendo en cuenta que la jerarquía de directorios de instalación de Tomcat incluye lo siguiente:

- bin.-** Contiene los scripts de inicio, parar, reiniciar y otros scripts y ejecutables para el funcionamiento del servidor.
- common.-** Contiene un sinnúmero de clases comunes que puede utilizar el Contenedor de ServletsCatalina y las aplicaciones web que se encuentren alojadas en el servidor web.
- conf.-** Contiene algunos archivos de configuración incluyendo server.xml que es el archivo principal de configuración de Tomcat y el archivo web.xml el cual configura los valores por defecto para las distintas aplicaciones desplegadas en Tomcat como cuando se usa Hibernate, RichFaces, etc. También en este directorio se encuentran los correspondientes DTD para la configuración de Tomcat. La función básica de los DTD es la descripción del formato de datos.

- d. **logs.**- Aquí es donde Tomcat aloja los mensajes cuando se ejecuta, cierra o existen errores al ejecutar las aplicaciones.
- e. **server.**- Existen clases que son utilizadas únicamente por el contenedor de Servlets Catalina.
- f. **shared.**- Ubicación donde se encuentran las clases compartidas por todas las aplicaciones web alojadas en el servidor de aplicaciones Tomcat.
- g. **webapps.**-Es donde se deben copiar o donde deben estar todos los archivos de una aplicación web.
- h. **work.**- Se ubican los archivos que se crean durante la ejecución del servidor es decir el almacenamiento temporal de ficheros y directorios.

2.5.3. Funcionamiento del servidor de aplicaciones

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de una máquina virtual Java, de tal manera es cada vez más utilizado por las empresas en los entornos de producción debido a su comprobada estabilidad. Resulta de gran utilidad para los programadores que deseen usar Tomcat como servidor Web libre, en entornos con alto nivel de tráfico y alta disponibilidad. Constituye además una excelente herramienta para los principiantes ya que se encontrará detalles para usar Tomcat en todas las plataformas principales: Windows, Linux, Mac OS X, Solaris, y FreeBSD, con sus ficheros de configuración específicos, y consejos paso a paso para implementar y correr aplicaciones Web eficazmente.

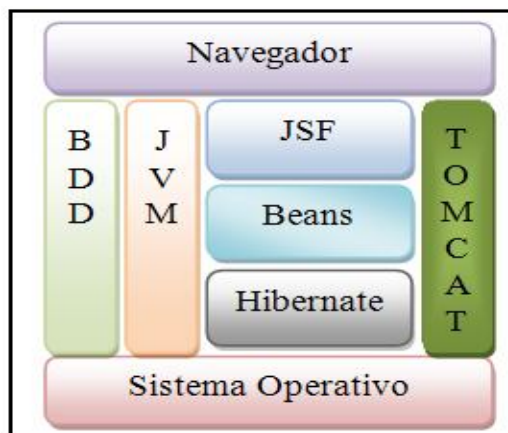


Figura 14: Funcionamiento de Apache Tomcat
Fuente: Propia

2.5.4. Características de Tomcat

A continuación se presentan las principales características del servidor de aplicaciones Tomcat desde la versión 3.0 que es la distribución inicial hasta la versión más actual que es la versión 7.

Tomcat 3.x

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets
- Funciones básicas HTTP

Tomcat 4.x

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina
- Motor JSP rediseñado con Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

Tomcat 5.x

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Análisis rápido JSP

Tomcat 6.x

- Implementado de Servlet 2.5 y JSP 2.1
- Soporte para Unified Expression Language 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores
- Soporte para Comet a través de la interfaz Comet Processor

Tomcat 7.x

- Implementado de Servlet 3.0 JSP 2.2 y EL 2.2
- Mejoras para detectar y prevenir "fugas de memoria" en las aplicaciones web
- Limpieza interna de código
- Soporte para la inclusión de contenidos externos directamente en una aplicación web.

2.6. Bases de datos

“Una **base de datos** o **banco de datos** es un conjunto de datos no redundantes, estructurados, organizados, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Las bases de Datos tienen muchos usos: nos facilitan el almacenamiento de grandes cantidades de información permiten la recuperación rápida y flexible de información, con ellas se puede organizar y reorganizar la información, así como imprimirla o distribuirla en formas diversas”.
[Anonimo B. d., 2009]

Una base de datos debe de contar con independencia lógica y física de los datos, esto se refiere a la capacidad de modificar alguna información específica sin que afecte a los demás registros.

2.6.1. PostgreSQL



Figura 15: Logo del Servidor de BDD PostgreSQL
Fuente: www.postgresql.org

Los sistemas de Bases de Datos relacionales tradicionales conocidas como DBMS's soportan un modelo de datos que fundamentan en una colección de relaciones con nombre, que contienen atributos de un tipo específico, como los sistemas comerciales actuales, los tipos de datos posibles incluyen números de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas, lo que significa que este modelo de datos será inadecuado para futuras aplicaciones de procesamiento de datos.

El servidor de bases de datos Postgres ofrece una potencia adicional notable al incorporar algunos conceptos con los que los usuarios podrán extender fácilmente el sistema. Los conceptos añadidos a Postgres son:

1. Clases
2. Herencia
3. Tipos
4. Funciones

Y otras características que aportan potencia y flexibilidad adicional son las siguientes:

1. Restricciones (Constraints)
2. Disparadores (triggers)
3. Reglas (rules)
4. Integridad transaccional

Por todas las características mencionadas anteriormente colocan a Postgres en la categoría de las bases de datos identificadas como objeto-relacionales, teniendo claramente que éstas son diferentes de las referidas como orientadas a objetos, que en realidad no son bien utilizadas en lenguajes de bases de datos relacionales tradicionales.

Postgres tiene algunas características que son propias y referidas de las bases de datos orientadas a objetos, pero también algunas bases de datos comerciales han incorporado características en las que Postgres fue el primer servidor de base de datos en implementarlas.

2.6.1.1. Arquitectura de Postgres

La arquitectura y procesos de Postgres se muestran en la Figura 10, en la cual se observa que Postgres puede administrar un grupo de bases de datos y cada base de datos tiene su propio conjunto de catálogos del sistema conocido como INFORMATION_SCHEMA y PG_CATALOG.

Todos ellos comparten bases de datos PG_DATABASES como una tabla de sistema común donde cada base de datos es una entidad separada y la colección de bases de datos se llama clúster. Una instancia de PostgreSQL puede ser definida como la gestión de un conjunto de bases de datos, donde el servidor puede ejecutar más de un caso de este tipo de agrupaciones de base de datos.

Una base de datos PostgreSQL se pueden migrar lógicamente a todas las bases de datos compatibles con el tipo de objeto de esquema ya que no se puede acceder a otras bases de datos con una conexión con nombre.

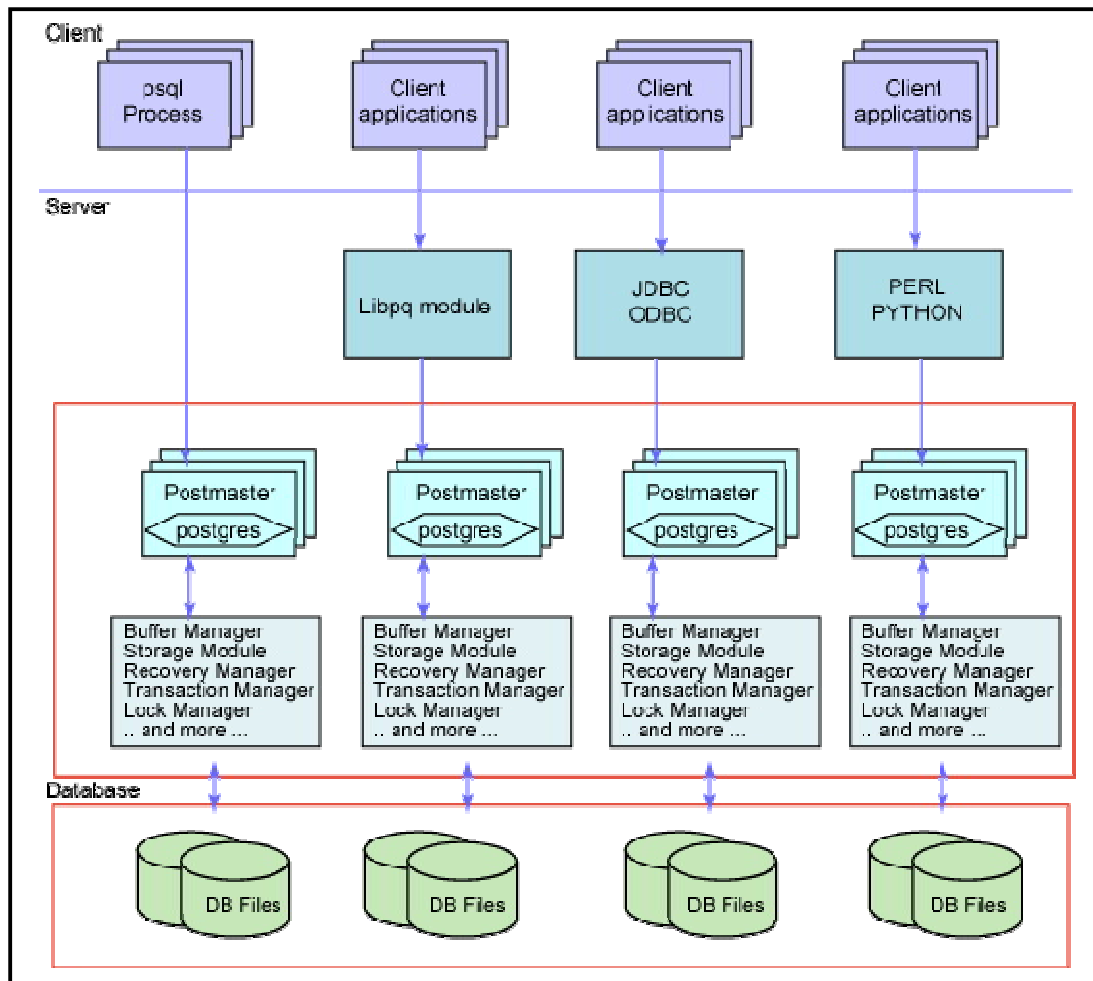


Figura 16: Arquitectura de Postgres

Fuente: <http://www.ibm.com/developerworks/data/library/techarticle/dm-0606khatri/>

2.6.1.2. Características

1. “Implementación del estándar SQL92 y SQL99.
2. Soporte de protocolo de comunicación encriptado por SSL
3. Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.
4. Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
5. Permite la declaración de funciones propias, así como la definición de disparadores y la creación de tipos de datos propios.
6. Soporta el uso de índices, reglas y vistas.

7. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
8. **Portabilidad.-** Funciona en los principales sistemas operativos: Linux, Unix, Mac OS, Windows, etc.
9. **Atomicidad.-** Certifica que la operación se realice por completo o no se realice nada ante fallos o caídas de sistemas ya que una transacción no debe quedarse a medias.
10. **Consistencia.-** Asegura que sólo se empieza aquello que se puede terminar, es decir que solo se ejecutan aquellas operaciones que no van a romper la reglas y normas de integridad de la base de datos.
11. **Aislamiento.-**Asegura que una operación no afecte a otras y genere errores sobre todo cuando dos transacciones trabajan sobre la misma información.
12. **Durabilidad.-**Asegura que una vez realizada la operación, ésta permanecerá y no se podrá deshacer aunque se caiga sistema
13. **Documentación.-** Mantiene una documentación muy bien organizada, pública y disponible para cualquier usuario.
14. Diseñada para entornos con altos volúmenes de tráfico/transacciones.”[PostgreSQL, 2012]

Algunas limitaciones que caracterizan a PostgreSQL son las siguientes:

| Límite | Valor |
|-------------------------------------|---|
| Máximo tamaño base de dato | Ilimitado (Depende de tu sistema de almacenamiento) |
| Máximo tamaño de tabla | 32 TB |
| Máximo tamaño de fila | 1.6 TB |
| Máximo tamaño de campo | 1 GB |
| Máximo número de filas por tabla | Ilimitado |
| Máximo número de columnas por tabla | 250 - 1600 (dependiendo del tipo) |
| Máximo número de índices por tabla | Ilimitado |

Tabla 4: Límites de PostgreSQL

Fuente: www.postgresql.org.pe/articles

2.7. Plataformas y frameworks

Para el desarrollo del presente proyecto se ha utilizado plataformas y Frameworks de código libre pero que tienen una buena acogida a la hora de desarrollar aplicaciones empresariales. A continuación se describen las más importantes:

2.7.1. NetBeans IDE

Es un entorno de desarrollo integrado, libre y está diseñado para soportar e integrar varias tecnologías principalmente java. El entorno de desarrollo es el que facilita el trabajo a los programadores ya agiliza el proceso de programación por medio de ayudas y autocompletados de cada lenguaje que se acople al IDE.

“NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. SunMicroSystems fundó el proyecto de código abierto Netbeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos, aunque hoy en día el dueño de sus derechos es Oracle.” [affiliates, 2012]

La plataforma Netbeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. “Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de Netbeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma Netbeans pueden ser extendidas fácilmente por otros desarrolladores de software.” [Domínguez-Dorado, 2005]

En Netbeans es muy sencillo de añadir tecnologías ya que se integran fácilmente.

2.7.1.1. Características

- 1.** Facilidad de uso.
- 2.** Facilidad de instalación.
- 3.** Característica/funciones extras.

4. Manual de instrucciones.

5. Soporte de fábrica.

2.7.2. Java

Para que un computador pueda entender las líneas de código que escribimos a través de un entorno de desarrollo (IDE), se necesita de un programa especial que pueda interpretar y transformar este código a un lenguaje de bajo nivel o lenguaje máquina, para ello el encargo de realizar esta función es la plataforma de desarrollo en este caso Java.

2.7.2.1. Definición

La tecnología java es un lenguaje de programación y una plataforma de desarrollo originaria de Sun Microsystems y actualmente pertenece a Oracle.

“El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.” [Java, Wikipedia, 2012]

2.7.2.2. Arquitectura

La especificación Java define no sólo un lenguaje de alto nivel, sino también la máquina a bajo nivel y el conjunto de instrucciones que puede ejecutar.

La Arquitectura de Java hace énfasis en sus componentes principales, como el procesador virtual Java, que se encarga de ejecutar los códigos de operación (bytecodes) generados por los compiladores Java, el verificador de código, que junto con el cargador de clases y el administrador de seguridad, se encargan de implementar los mecanismos empleados para proporcionar seguridad a los usuarios de Java.

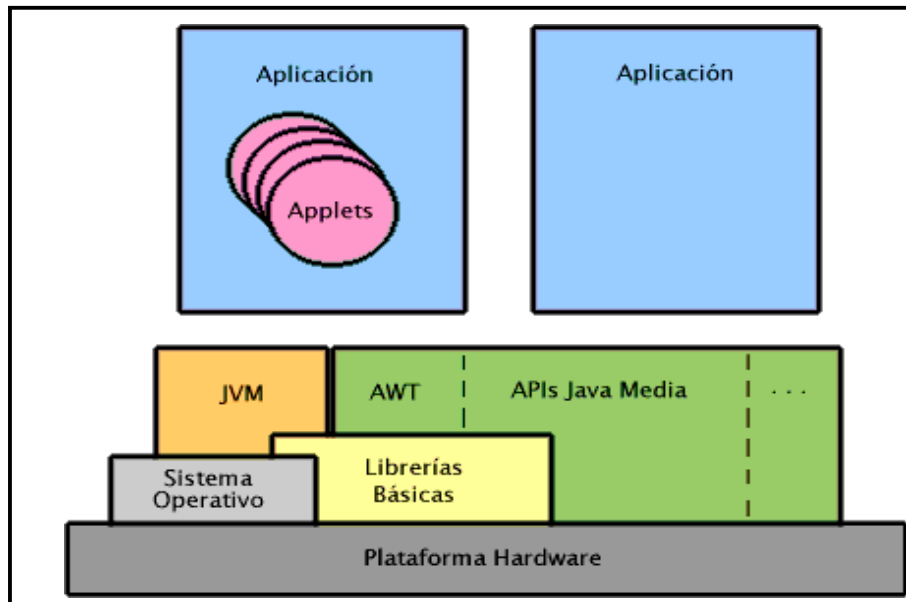


Figura 17: Arquitectura de Java

Fuente: <http://bernabe.atspace.com/froufe/parte2/cap2-5.html>

2.7.2.3. Características

Java proviene de una idea relativamente antigua, pero ha venido a revolucionar la computación de nuestros días, debido a que integra tres características de suma importancia como son la seguridad, la escalabilidad y la portabilidad. A continuación se describen las características más importantes:

1. **Sencillo.-** Elimina características de lenguajes anteriores (C, C++).
2. **Orientado a objetos.-** Java trabaja con sus datos como objetos y con interfaces a esos objetos.
3. **Distribuido.-** Concebido para trabajar en un entorno conectado en red.
4. **Interpretado.-** Traduce código fuente a código intermedio.
5. **Robusto.-** No se interrumpe fácilmente, evita sobrescribir o corromper memoria mediante punteros, maneja excepciones y verificación de ByteCodes.¹
6. **Seguro.-** Aplicaciones seguras ya que no acceden a zonas delicadas de memoria o de sistema.

¹**Bytecode.-** Es un código intermedio más abstracto que el código máquina

7. **Arquitectura neutra.-** Cualquier plataforma donde exista una JVM se puede ejecutar el código objeto independientemente de la máquina en el que ha sido generado.
8. **Portable.-** Corre sobre los principales sistemas operativos existentes.
9. **Altas prestaciones.-** Traslada los bytecodes al código máquina de la plataforma según los va leyendo, realizando cierto grado de optimización.
10. **Multithread o multihilo.-** Ejecución de varias tareas a la vez, permitiendo muchas actividades simultáneas en un programa.
11. **Dinámico.-** Las librerías nuevas o actualizadas no paralizan la ejecución de las aplicaciones siempre que mantengan la API anterior.

2.7.3. Java EE (Java Platform, Enterprise Edition)

Anteriormente era conocida como J2EE hasta la versión 1.4 actualmente existe la versión 6.0 de JEE. Java Enterprise Edition (JEE) es una plataforma que fue creada exclusivamente para creación de aplicaciones empresariales distribuidas, las cuales estarán escritas en el lenguaje de programación Java y que se ejecutan en un servidor de aplicaciones como Apache Tomcat, Jboss, etc.

Una aplicación distribuida es una aplicación con diferentes componentes que se ejecutan en entornos separados es decir en diferentes plataformas pero enlazadas a través de una red.

JEE es un conjunto de especificaciones y permiten soluciones para el desarrollo, despliegue y gestión de aplicaciones de n capas. Conjunto de especificaciones quiere decir que cualquier usuario o fabricante puede coger estas especificaciones y desarrollar un producto final que cumpla con estas especificaciones un ejemplo de ellos es Glassfish ya que permite la ejecución de aplicaciones desarrolladas en la plataforma JEE.

“Esta plataforma de desarrollo incluye varias especificaciones de API², tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc., y define cómo coordinarlos, también configura algunas especificaciones únicas para componentes como Enterprise JavaBeans, Servlets, JavaServerPages y varias tecnologías de servicios web.”[**Wikimedia Project, 2009**]

²API.- Interfaz de Programación de Aplicaciones

La plataforma de desarrollo Java EE se puede dividir en varios contenedores en los cuales se pueden describir varios APIs para cada contenedor tal y como se muestra la figura 12 que indica las relaciones entre los contenedores de la plataforma.

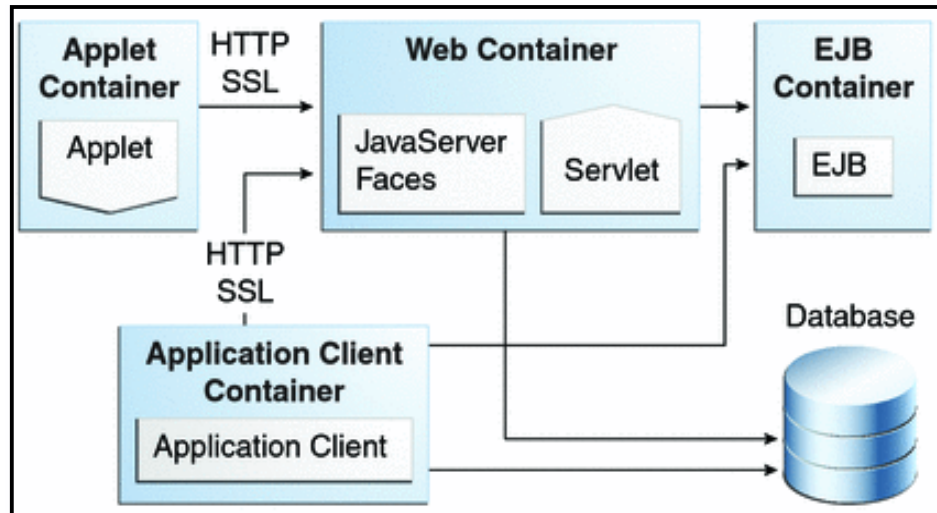


Figura 18: Contenedor de JEE

Fuente:

<http://www.pvv.ntnu.no/~rakhmato/doc/download.oracle.com/javaee/6/tutorial/doc/bnacj.html>

2.7.3.1. Definición

Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones, esto permite al desarrollador crear una aplicación de empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.

“Esta plataforma permite que el servidor de aplicaciones pueda manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.”[**Wikimedia Project, 2009**]

2.7.3.2. Arquitectura

La arquitectura de JEE está basada en nociones de capas, contenedores, componentes y servicios lo que implica un modelo de aplicaciones distribuidas en diversas capas o niveles, es así que las aplicaciones JEE suelen seguir una arquitectura de 3 capas como son la capa de presentación, capa de lógica de negocio y capa de acceso a datos

“La visión de la arquitectura es un esquema lógico, no físico. Cuando hablamos de capas nos referimos sobre todo a servicios diferentes (que pueden estar físicamente dentro de la misma máquina e incluso compartir servidor de aplicaciones y JVM.”[Anónimo, Dos Ideas, 2011]

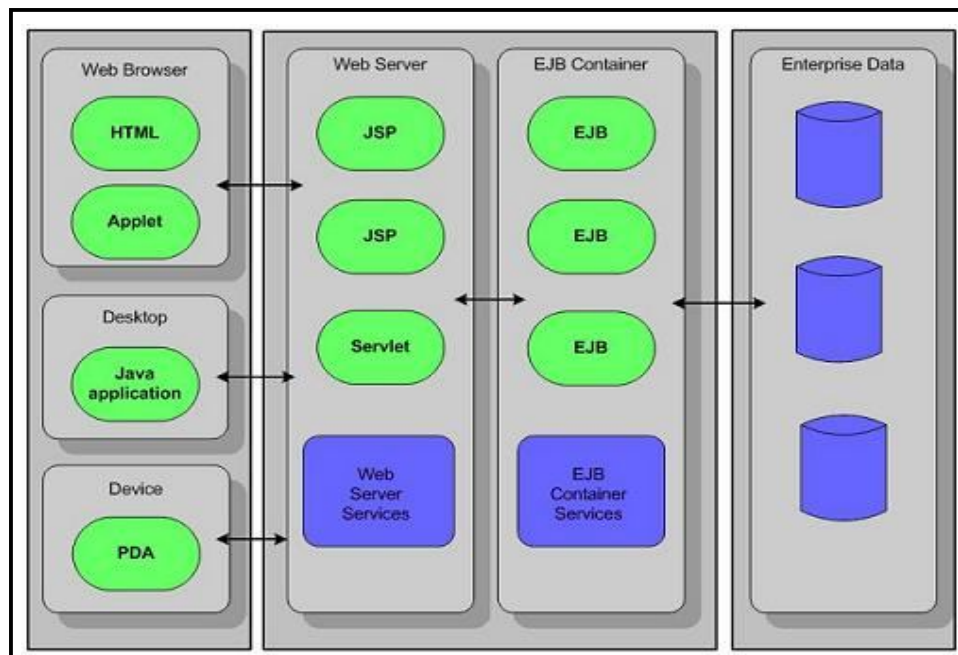


Figura 19: Arquitectura Java EE

Fuente: <http://www.proactiva-calidad.com/java/arquitectura/index.html>

Capa de Presentación.- Esta capa es la que se encuentra visible a los ojos de los usuarios, es decir es la capa que donde interactúa el usuarios con el sistema.

Capa de Lógica del negocio.- En esta capa es donde se realizan todas las transacciones y operaciones con los datos obtenidos de la capa de acceso a datos para mostrarlos en la capa de presentación.

Capa de acceso a datos.- En esta capa es donde se realiza todas las operaciones CRUD de una aplicación, es decir que se realizan los métodos que acceden directamente a nuestro servidor de base de datos.

2.7.3.3. Características

Las principales características de JEE entre las más relevantes e importantes son las siguientes:

a. Portabilidad.- Esta característica es una de las más importantes ya que gracias a esta podemos ejecutar el software en diferentes plataformas. En cuanto mayor sea la portabilidad menor será la dependencia del software para el funcionamiento.

b. Accesibilidad.- Gracias a que JEE se mantiene como proyecto de software libre podemos disfrutar de esta plataforma sin pagar ningún tipo de licencias.

c. Escalabilidad.- Como característica de un sistema es generalmente un poco difícil de definir en cualquier caso, pero podemos decir que la plataforma JEE nos permite realizar aplicaciones totalmente escalables ya que a futuro podemos seguir añadiendo funcionalidades al sistema creado y sin ningún problema debido a que las aplicaciones son fáciles de entender por su separación en capas.

d. Integrable.- Permite que aplicaciones creadas con otras tecnologías formen parte de una aplicación Java, que estas puedan interactuar entre sí.

e. Seguridad.- Si una plataforma que es destinada para la creación de aplicaciones empresariales obviamente va a tener una buena seguridad, lo que nos va a permitir tener un control sobre usuarios y no permitirles acceder a la misma funcionalidad.

f. Concurrencia.- Permite que dos o más usuarios puedan acceder a una aplicación sin alterar su buen funcionamiento.

2.7.3.4. Beneficios

- a. Reduce la complejidad de construir aplicaciones n-capas.
- b. Permite lidiar con fuertes requerimientos no funcionales.
- c. Posibilidad de distribución.
- d. Tolerancia a fallas.
- e. Estandariza una API común entre componentes y servidores de aplicación.
- f. Los servidores de aplicación y contenedores proveen los servicios del framework.

2.7.4. Hibernate

Al trabajar con Java que es una plataforma orientada a objetos y bases de datos relacionales puede hacernos invertir mucho tiempo al momento de desarrollar, para prevenir esto se puede optar por usar Hibernate que es una herramienta que realiza el mapping entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java.

2.7.4.1. Definición

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java, facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

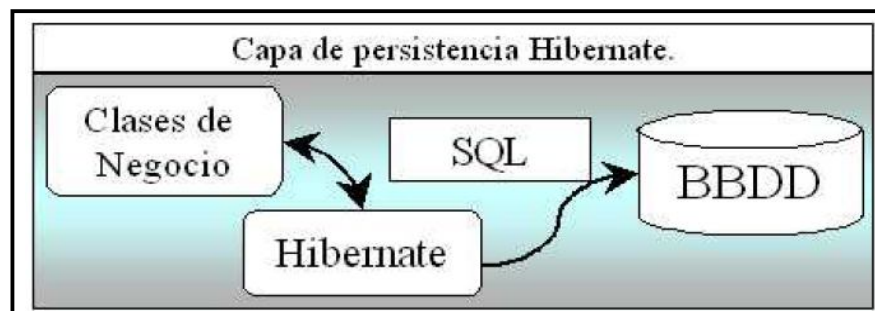


Figura 20: Capa de Persistencia

Fuente: <http://www.javahispano.org/storage/contenidos/ManualHibernate.pdf> - Página 4

Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos, de una manera muy rápida y optimizada podremos generar BDD en cualquiera de los entornos soportados como: Oracle, DB2, MySql, etc., y lo más importante de todo que es un framework open source, lo que quiere decir que no se debe pagar por las licencias para poder utilizar.

El proceso de desarrollo consiste en que una vez que tengamos el diseño de la base de datos realizado, podemos mapear estos datos a ficheros XML siguiendo la estructura (DTD) de mapeo de Hibernate y con estos archivos podremos generar el código de nuestros objetos persistentes en clases Java y también crear Base de Datos independientemente del entorno escogido ya que Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. La configuración básica de hibernate es la siguiente:



Figura 21:Arquitectura base de Hibernate

Fuente: <http://www.javahispano.org/storage/contenidos/ManualHibernate.pdf> - Página 5

2.7.4.2. Características de Hibernate

Todas herramientas ORM's de este tipo, buscan solucionar el problema de la diferencia entre los dos modelos de datos existentes en una aplicación como son el uso de memoria con la programación orientada a objetos y el uso de las bases de datos con modelos relacionales. Hibernate permite a la aplicación administrar los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objetos.

A continuación se detallan algunas características de Hibernate.

1. La manera como trabaja este ORM es convirtiendo los datos entre los tipos utilizados por Java y los definidos por SQL, de tal manera que genera las sentencias SQL y libera al desarrollador del manejo manual de los datos, que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos pero con un ligero incremento en el tiempo de ejecución.
2. Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
3. Tiene la funcionalidad de crear la base de datos a partir de la información disponible.
4. Ofrece un lenguaje de consulta llamado HQL (Hibernate Query Language), el cual permite visualizar datos sin acceder a la base de datos con un administrador.
5. Tiene integrado una API conocida como CRITERIA que sirve para construir consultas programadas.
6. En Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

2.7.5. JSF (Java Server Faces)

El desarrollo de aplicaciones o sistemas web sin el uso de una metodología o un patrón de diseño que permita mantener un orden y una separación de la codificación y demás elementos hace que en muchas ocasiones se mezcle en un mismo archivo JSP la interfaz de usuario, las reglas de validación, el acceso a la base de datos, etc., lo cual impide el mantenimiento de la aplicación y a la vez acorta su tiempo de vida útil, es por ello que se hace necesaria la utilización de herramientas que faciliten el desarrollo y nos permitan trabajar de manera ordenada.

El framework JSF nos ofrece un marco de trabajo que nos permite desarrollar aplicaciones, separando las diferentes capas de una arquitectura: presentación, reglas y entidades de negocio, ya que es un marco de trabajo que simplifica el desarrollo de interfaces de usuario en aplicaciones Web basadas en tecnología Java EE y en el patrón MVC (Modelo Vista Controlador) para ello usa como alternativa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas.

A continuación se muestra el diagrama de una aplicación JSF.

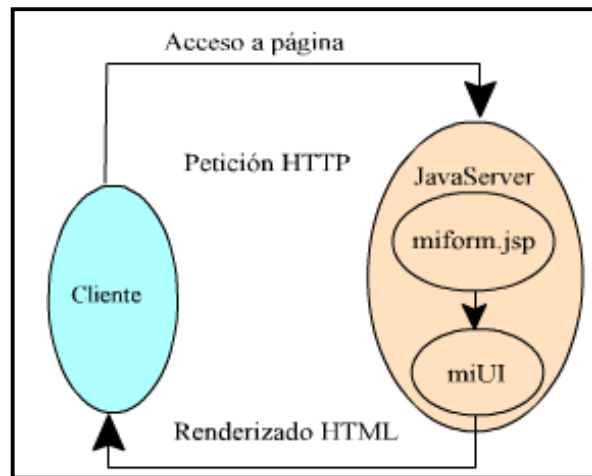


Figura 22: Diagrama de una aplicación JSF
Fuente: <http://www.slideshare.net/ingeniods/manual-jsf>

Este framework pretende facilitar la construcción de estas aplicaciones proporcionando un entorno de trabajo (framework) vía web que gestiona las acciones producidas por el usuario en su página HTML y las traduce a eventos que son enviados al servidor con el objetivo de regenerar la página original y reflejar los cambios pertinentes provocados por dichas acciones.

[Anónimo, Sicuma]

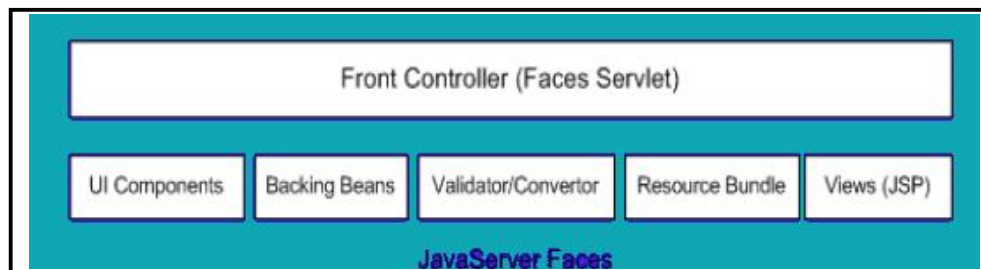


Figura 23: Componentes de JSF

Fuente:

http://www.ndeveloper.com/ndeveloperDocuments/documents/nDeveloper_JavaServerFaces.pdf

2.7.5.1. Características

1. Útil en aplicaciones con tecnología basada en MVC (Modelo, Vista, Controlador).
2. Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios HTML.

3. Asocia a cada vista con formularios un conjunto de objetos java manejados por el controlador (managed beans) que facilitan la recolección, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.
4. Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.
5. Tiene un alto nivel de abstracción que permite una mayor separación entre el diseño y el código permitiendo dividir realmente el desarrollo de una aplicación en diseño y desarrollo.
6. Incluye un conjunto central de componentes y herramientas que facilitan la creación de “widgets”³ estándar como “text boxes” o “radio buttons”.
7. La comunicación entre los componentes, la entrada de datos y la validación forma parte de la tecnología, es decir, no hay que implementar nada de esto como ocurre en otras tecnologías.
8. La visualización de los mensaje de validación y errores de conversión también están estandarizados incluso soportan internacionalización. [Peña, 2011]
9. Es flexible ya que nos permite crear nuestros propios componentes.
10. Permite construir aplicaciones web que introducen realmente una separación entre el comportamiento y la presentación.
11. Proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.

2.7.5.2. JSF con Netbeans

Netbeans es un entorno de desarrollo integrado al cual se describió anteriormente, JSF se integra en Netbeans desde la versión 5.0, mediante bibliotecas que contienen los siguientes archivos jar:

- a. jsf-api.jar
- b. jsf-impl.jar

³**Widget.-** Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets

Estas librerías contienen diferentes clases necesarias para usar los tags⁴ en las páginas jsp, para el uso de este framework en Netbeans solo es necesario crear un proyecto nuevo y especificar el uso del framework JSF, debido a que Netbeans ya viene integrado con las librerías de JSF.

2.7.6. RichFaces como framework RIA.

Normalmente en las aplicaciones web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace, de esta forma se produce un tráfico alto entre el cliente y el servidor llegando muchas veces a recargar la misma página con un cambio mínimo, es por esta razón que utilizamos los Frameworks RIA los cuales nos permiten refrescar partes o elementos de una página web. Existen varios framework RIA pero a continuación se describe solo uno de ellos, RichFaces que es el utilizado en el desarrollo del presente proyecto.

2.7.6.1. Definición

Es un framework open source que añade funcionalidades de Ajax (Asynchronous JavaScript and XML) en aplicaciones JSF sin la necesidad de usar JavaScript.

El framework RichFaces fue creado a partir del framework Ajax4jsf el mismo que posteriormente formaría a ser parte de Exadel que continúa con el proyecto hasta lanzar la primera versión de Ajax4jsf, pero más adelante Exadel decide dividir al framework en Ajax4jsf que se convierte en un proyecto open source y RichFaces que se convierte en una librería de componentes comercial.

Continuamente JBoss y Exadel firman un convenio donde los proyectos Ajax4jsf y RichFaces serán mantenidos por JBoss, convirtiéndole así a RichFaces en un proyecto open source donde después se unifican los proyectos Ajax4jsf y RichFaces manteniéndose hasta hoy con el nombre de RichFaces.

Este framework se integra fácilmente con JSF mediante el uso de dos librerías de componentes, la primera Ajax4jsf la cual integra capacidades Ajax sin usar JavaScript y la segunda es la librería de interfaz de usuario RichFacesUI que fue creada para mejorar y extiende las características de los elementos JSF y gracias a esto también implementa el patrón de diseño MVC (Modelo Vista Controlador), donde la vista corresponde a las páginas JSP, el

⁴**Tags.-** Tag en informática se les conoce como tags a las etiqueta de un determinado lenguaje.

modelo representa a la persistencia de datos y el controlador los Backing Beans que sirven como nexos entre el modelo y la vista.

2.7.6.2. Componentes de RichFaces

El núcleo de este framework se encuentra conformado por un conjunto de componentes fundamentales como se muestra en la Figura 18.

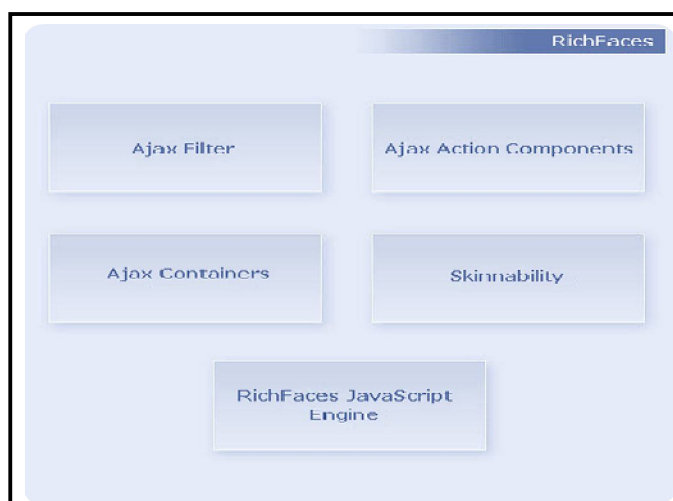


Figura 24: Componentes de RichFaces

Fuente: http://docs.jboss.org/richfaces/latest_3_3_X/en/devguide/html/Introduction.html

Ajax Filter.- Es necesario que se registre un filtro que se utiliza para reconocer los diferentes tipos de request que se realizan desde las vistas, verificando si es de tipo Ajax o de tipo JSF, este registro se realiza en el archivo web.xml de cada proyecto que use RichFaces.

Ajax Action Components.- Se utilizan para enviar request de tipo Ajax desde el cliente.

Ajax Containers.- Son componentes que permiten especificar una o varias áreas de una página JSF que deben ser decodificadas durante un request de tipo Ajax.

JavaScript Engine.- Es un componente que se ejecuta del lado del cliente que se usa para actualizar una determinada área de una página JSF basada en la información que tiene el servidor para la respuesta.

Skinnability.- Es un componente que permite extender los beneficios de CSS para establecer estilos visuales para páginas JSF, ventanas emergentes, colores en cabeceras, etc., lo cual no es posible realizar con CSS normal.

2.7.6.3. Características.

1. “Permite extender las características y beneficios de los elementos de JSF al trabajar con Ajax.
2. Se integra plenamente en el ciclo de vida de JSF, ya que los validadores se invocan del lado del servidor y se convierten durante el ciclo request-response de Ajax.
3. La biblioteca central establece la funcionalidad de Ajax en páginas ya existentes, y con ello no existe la necesidad de escribir ningún código JavaScript para reemplazar los componentes existentes con otras nuevas funciones Ajax.
4. Permite crear vistas rápidas de carácter complejo basándose en los componentes ya que es posible utilizarlas sin problemas con otras bibliotecas de otros fabricantes, así que se tiene más opciones para el desarrollo de aplicaciones.
5. Permite escribir componentes propios y personalizados con una función de soporte para Ajax y también ofrece un soporte avanzado para la gestión de los recursos diferentes: imágenes, código JavaScript y hojas de estilo CSS.
6. Proporciona una función que permite skinnability fácil definir y gestionar los diferentes esquemas de color y otros parámetros de la interfaz de usuario con la ayuda de los parámetros del skin con nombre.
7. Permite probar y crear los componentes, actions, listeners y las páginas a la vez.”
[JBoss.org, 2011]

2.7.6.4. Requerimientos Técnicos.

1. Java
2. JSF
3. Contenedor Web
4. Navegador
5. RichFaces
6. JDK 1.4 o superior
7. Firefox 1.5 o superior
8. Google Chrome
9. Apache Tomcat 5.5 o superior

2.7.6.5. Librerías del Framework RichFaces.

1. commons-beanutils-1.7.0.jar
2. commons-collections-3.2.jar
3. commons-digester-1.8.jar
4. commons-logging-1.0.4.jar
5. jhighlight-1.0.jar

2.7.7. Herramienta de Reportes JasperReports

JasperReports gracias a que proporciona todo lo necesario para diseñar e implementar informes de todo tipo de complejidad y la por la facilidad de aprendizaje es considerada la mejor herramienta de código libre en Java para generar reportes.

Una de las características que le hace popular a esta herramienta de reportes es poder entregar ricas presentaciones o diseños en la pantalla con distintos formatos como PDF, HTML, RTF, XLS, CSV y XML para luego poder imprimirlos.

Uno de los puntos fuertes de jasperreport es su comunidad <http://www.jasperforge.org> y la compañía que está detrás, JasperSoft nacida del éxito alcanzado por las primeras versiones de jasperreport y centrada en el desarrollo de herramientas Business Intelligence. [Roman, 2007]

2.7.7.1. ¿Qué es JasperReports?

JasperReports es una herramienta gratuita y opensource que se compone de un conjunto de librerías java para facilitar la generación de informes en nuestras aplicaciones tanto web como de escritorio.

Los informes se definen en un fichero XML el cual será compilado por las librerías jasperreport y generarán un fichero .jasper que utilizaremos para rellenar y mostrarlo en los informes finales

JasperReports está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE anteriormente y JEE 6.0 actualmente o aplicaciones Web, para generar contenido dinámico.

El Sistema Operativo para JasperReports no es un problema ya que funciona perfectamente tanto en plataformas de Microsoft Windows como en GNU Linux y en ambas plataformas se ha comprobado su buen funcionamiento.

2.7.7.2. Requerimientos de JasperReports

Para el funcionamiento de JasperReports se necesita básicamente 2 cosas principales la primera es JDK 1.4 o una versión superior es decir la máquina virtual de JAVA la cual hace posible el funcionamiento de cualquier aplicación JAVA y como se había explicado anteriormente que JasperReports está completamente escrita en JAVA entonces es requisito principal tener instalado el JDK. Y también se necesita de las siguientes librerías a más de las de JasperReports para empezar a generar reportes.

1. JakartaCommonsDigesterComponent (versión 1.1 o posterior)commons-digester.jar[**Commons Digester, 2011**]
2. JakartaCommonsBeanUtilsComponent (versión 1.1 o posterior)commons-beanutils.jar[**Commons Beanutils, 2010**]
3. JakartaCommonsCollectionsComponent (versión 1.0 o posterior)commons-collections.jar. [**Commons Collections, 2008**]
4. JakartaCommonsLoggingComponent (versión 1.0 o posterior)commons-logging.jar. [**Commons Logging, 2008**]
5. PDF. Librería libre Java-PDFiText por Bruno Lowagie y Paulo Soares(versión 1.01 o posterior)itext-1.02b.jar[**Bruno Lowagie, 2010-2011**]
6. Driver JDBC 2.0 (Usualmente incluido en el SDK y se utiliza de acuerdo al servidor de BDD).
7. JakartaPOI (versión 2.0 o posterior)poi-2.0-final-20040126.jar[**Apache POI Project, 2012**]

Las librerías de JasperReports son las siguientes:

1. jasperreports-javaflow-4.0.2.jar
2. jasperreports-4.0.2.jar
3. jasperreports-applet-4.0.2.jar
4. jasperreports-fonts-4.0.2.jar

Para la generación de Reportes a través de Hibernate es necesaria también la librería que hsqldb.jar

2.7.7.3. Funcionamiento de JasperReports

El funcionamiento de JasperReports es similar a un compilador y a un intérprete, es decir que traduce cada instrucción o sentencia del archivo XML a un lenguaje máquina e inmediatamente se ejecuta y se crea el archivo .jasper que es el resultado de compilar el XML, ver figura 19.

El usuario se encarga de diseñar el reporte codificándolo en XML con las etiquetas y atributos definidos en un archivo llamado jasperreports.dtd el cual es parte de JasperReports. En el archivo XML el usuario define básicamente todos los parámetros que se utilizaran en el reporte, describiendo así donde colocar texto, imágenes, líneas, detalles de tablas, rectángulos, cómo adquirir los datos, como realizar ciertos cálculos para mostrar campos calculados, etc.

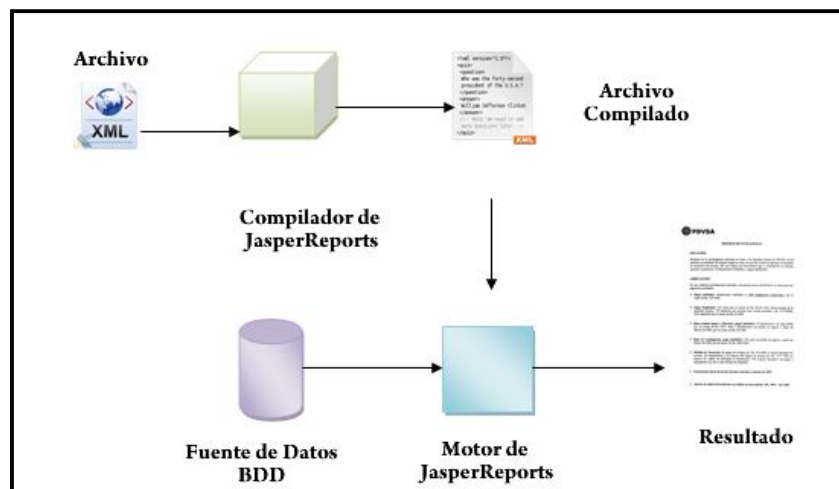


Figura 25:Funcionamiento de JasperReports
Fuente: Propia

Sistema Integrado de Información Financiera

FINANSYS



Descripción y funcionamiento de los módulos

- 1) Funcionamiento del Sistema
- 2) Descripción de los Módulos
 - Módulo de Administración y Seguridad
 - Módulo Captaciones (Caja)
 - Módulo de Ahorros
- 3) Funcionamiento de Módulos
 - Proceso Módulo Administración y Seguridad
 - Proceso Módulo de Captaciones (Caja)
 - Proceso Módulo de Ahorros

Capítulo 3: DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MÓDULOS

3.1. Funcionamiento del sistema

3.1.1. Vista general.

En el presente capítulo se detallan los Modelos Funcionales elaborados como base para el desarrollo informático del Sistema Integrado de Información Financiera. En este documento la Administración Financiera se define como un sistema que integra los subsistemas de Administración y Seguridad, Ahorros, Cartera, Clientes, Cajas, Contabilidad, y Auditoría los cuales se encuentran interrelacionados en cuanto a su normatividad, operatividad e información que generan, manteniendo así las características y necesidades propias de cada uno de estos subsistemas. A partir de la descripción de las funciones propias de los sectores intervinientes en los procesos de cada uno de los departamentos existentes en las Cooperativas y de la información que generan, hemos diseñado al Sistema Integrado de Información Financiera teniendo como módulos principales los siguientes:



Figura 26: Prospectiva del Sistema
Fuente: Propia

3.1.2. Características del sistema

El sistema tiene las características que se describen a continuación:

| Característica | Descripción |
|---|---|
| Administración Agrupada | Se instala y actualiza únicamente del lado del servidor ya que es una aplicación Web y no requiere que se apliquen dichos cambios a los clientes ya que los clientes consumen del servidor. |
| Visualización en navegadores web. | Podrá funcionar en cualquier navegador que tenga soporte para java y java Script ya que la aplicación fue desarrollada con tecnologías de código abierto como JEE (Java Enterprise Edition) pero se recomienda el uso del navegador Firefox. |
| Facilidades en el registro de Información | La interfaz es amigable y de fácil manejo lo que agiliza el proceso de capacitaciones. |
| La Disponibilidad es alta | Los usuarios podrán acceder inmediatamente al sistema desde cualquier punto de la Intranet o en su defecto desde internet a través de cualquier navegador. |
| Eficiencia en la presentación de Información. | Al ser un sistema integrado se puede acceder rápidamente a la información que se encuentran en los diferentes módulos del sistema y tener toda la información necesaria para una tarea en una misma pantalla ya que se puede definir menús y roles personalizados |
| Fácil de Usar | El diseño y la funcionalidad son muy simples ya que consta de un menú intuitivo, fácil navegación entre opciones del sistema y su información es consistente. |
| Facilidad para analizar la información. | A través de los diferentes reportes y opciones de búsquedas y consultas con los que consta el sistema. |
| Sistema Multiplataforma | El acceso al sistema se podrá realizar desde cualquier navegador web. |

Tabla 5: Características del Sistema
Fuente: Propia

3.1.3. Dependencias para la implementación

Para la correcta funcionalidad se deberá implementar primeramente un ambiente de pruebas similar al de producción para evitar posibles errores y conflictos de información.

3.1.3.1. Costos

Para la implementación del sistema se realizará la adquisición de un servidor con características adecuadas para el funcionamiento del sistema. Cabe mencionar que el sistema no aplica costos de licenciamiento ya que se desarrolló a través de una plataforma de código abierto.

3.1.3.2. Licenciamiento

Las tecnologías utilizadas para el desarrollo e implementación del sistema no necesitan de algún pago para la adquisición de licencias ya que el licenciamiento de estas herramientas está regido por las cláusulas de GPL establecido en el software de libre. Para el desarrollo del Sistema Integrado de Información Financiera se utilizará las siguientes herramientas:

- a) Como metodología de Desarrollo RUP (Rational Unified Process).
- b) PostgreSQL como servidor de Base de Datos
- c) Framework Hibernate para mapeo de datos
- d) JEE (Java Enterprise Edition) como plataforma de Desarrollo.
- e) JSF (Java Server Faces) para la interfaz
- f) RichFaces para la integración de funcionalidades Ajax en JSF.
- g) HTML, CSS, Ajax, JavaScript como estándares de código abierto.
- h) JasperReports para presentación de Reportes en formato PDF.

3.1.3.3. Instalación

No existe mayor problema en la instalación ya que esta se la realizará en forma centralizada, es decir que la información estará ubicada en un solo punto llamado servidor, al cual los clientes accederán a través de un navegador web desde cualquier punto de red de la Intranet de la Institución.

3.1.4. Funcionamiento.

El Sistema Integrado de Información Financiera fue creado utilizando la plataforma de Desarrollo JEE⁵ para la cual es recomendable utilizar MVC⁶ como patrón base para diseñar aplicaciones interactivas, la interface gráfica del Sistema utiliza JSF y RichFaces para diseñar la parte que visualizara el usuario, teniendo como template principal el proceso de autenticación de usuario con la página index.jsp.

El flujo de la aplicación inicia por la página de Login o Autenticación, si un usuario intenta acceder directamente a otra página, la aplicación verifica que no se ha autenticado y le re direcciona a la página de autenticación.

Luego de autenticarse el usuario podrá navegar por las opciones que presenta cada módulo de la aplicación dependiendo del rol y el tipo de usuario.

A continuación se muestra el funcionamiento en General de Sistema ya que más adelante se muestra el funcionamiento desglosado de cada Módulo.

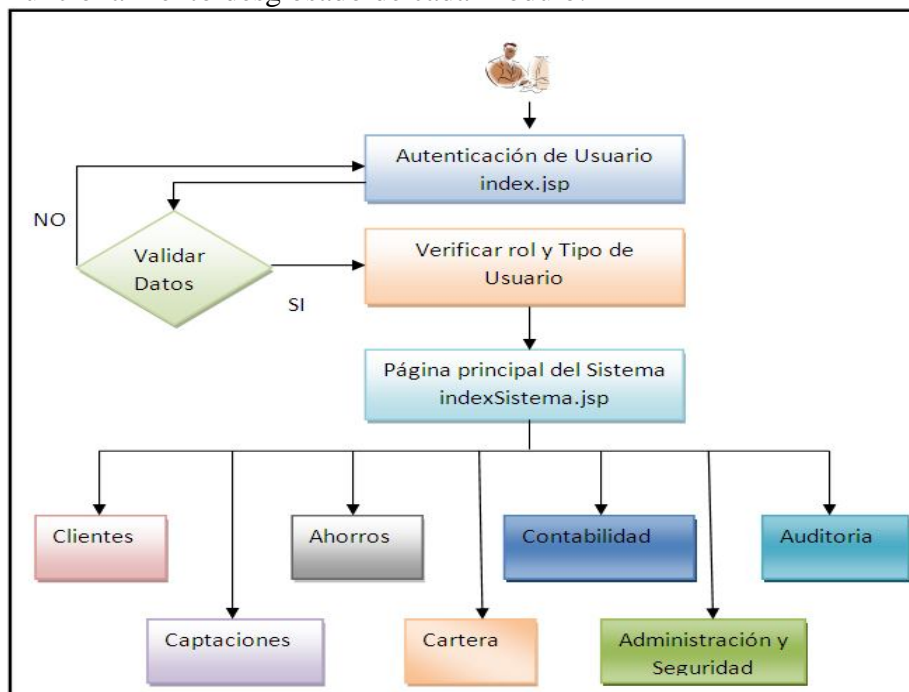


Figura 27: Funcionamiento General del Sistema
Fuente: Propia

⁵ JEE.- Java Enterprise Edition plataforma de desarrollo empresarial

⁶ MVC.- Modelo MVC arquitectura del sistema.

3.1.4.1. Arquitectura del sistema

La arquitectura MVC organiza el diseño de una aplicación interactiva mediante la separación de la presentación de los datos y el comportamiento de la aplicación.

1. El modelo representa la estructura de los datos en la aplicación, es decir que en esta capa se realizan las consultas y manipulación de datos (CRUD), además de los métodos, funciones u operaciones con estos datos.
2. La vista es la capa que presenta los datos al usuario, es decir la interfaz gráfica.
3. El controlador traduce las acciones de los usuarios realizadas en la vista y llama a los métodos de negocio creados en el modelo y seleccionando la vista apropiada.

Un modelo se encarga del estado de la aplicación y su funcionalidad, una vista de la presentación y el controlador del comportamiento en respuesta a una acción del usuario.

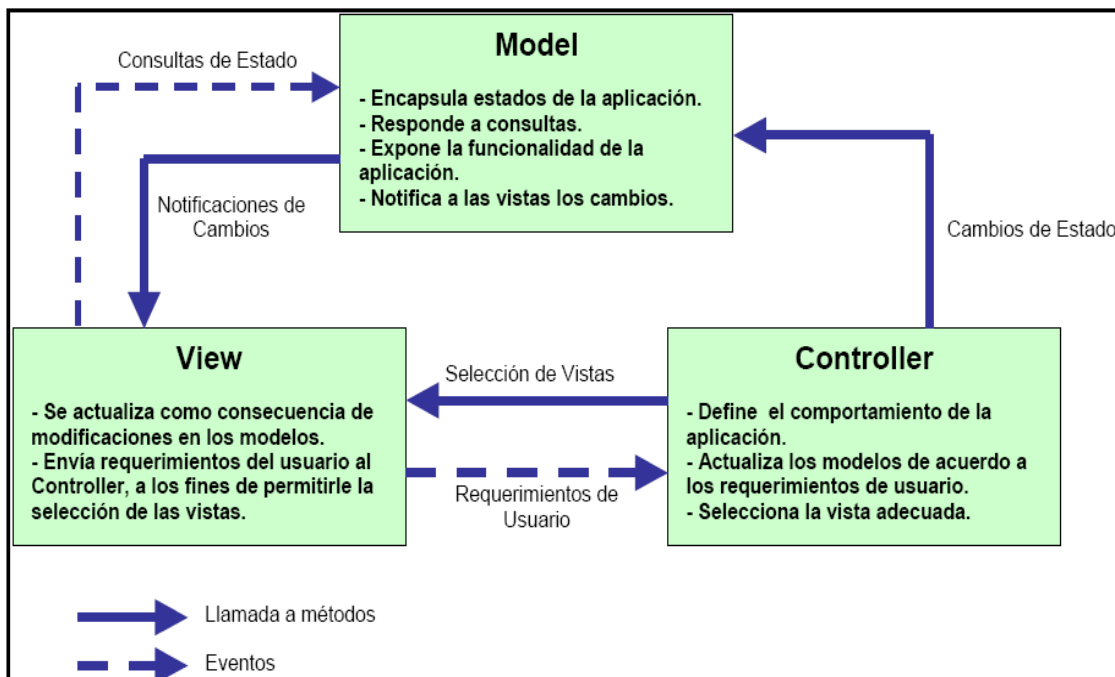


Figura 28: Arquitectura de la Aplicación

Fuente: <http://neurotests.frc.utn.edu.ar/Tests/Docu/Tests.pdf>

Diseño de aplicaciones utilizando la plataforma J2EETM (Ing. Javier Alfredo Voos)

3.1.4.2. Diagrama General de Casos de Uso.

El diagrama general de casos de uso se muestra en capítulo 4 dentro del documento de Arquitectura que genera la metodología RUP, en el apartado 4.3.1.4.4.

3.1.4.3. Diagrama entidad relación

El diagrama Entidad-Relación de la BDD del Sistema Integrado de Información Financiera, se mostrará en la sección de anexos de este documento.

3.2. Descripción y funcionamiento de los módulos

A continuación se indica la descripción y funcionamiento de los módulos de Administración y Seguridad, Captaciones (Caja) y Ahorros de la Cooperativa Unión Cochapamba.

3.2.1. Descripción del módulo de Administración y Seguridad

El módulo de Administración y seguridad se encarga de realizar todas las funcionalidades de carácter administrativo con el objetivo de perfeccionar todo el funcionamiento del mismo.

La seguridad de la aplicación es uno de los requerimientos más importantes que deben ser cumplidos. Está basada en las restricciones más comúnmente usadas en cualquier sistema que la implemente, con el uso de los conceptos de autenticación y autorización, basados en requerir una cuenta de usuario válida y activa, además de un conjunto de roles que le permitirán al usuario realizar solo aquellas operaciones que se le han asignado con el uso de este módulo de administración.

Todo este proceso se realiza al iniciar la aplicación y luego comprueba la autorización a las distintas funcionalidades del sistema.

3.2.2. Funcionamiento del módulo de Administración y Seguridad

Solo un usuario administrador podrá acceder al módulo de administración y Seguridad donde podrá realizar las acciones que se detallan a continuación en la siguiente tabla.

| Paso | Acción |
|------|--|
| 1 | Autenticarse como usuario administrador |
| 2 | Ingresar al menú Administración |
| 3 | Seleccionar una de las siguientes Opciones: 1. Administrar Sucursales |

| | |
|--------------|--|
| | <ol style="list-style-type: none"> 2. Administrar Tipos de Usuarios 3. Administración de Roles 4. Administración de Usuarios 5. Administrar Parámetros Captaciones (Caja) 6. Administrar Parámetros Ahorros 7. Administrar Parámetros Clientes 8. Administrar Parámetros Cartera 9. Administrar Parámetros Auditoría |
| 3.1 | Administrar Sucursales <ol style="list-style-type: none"> 1) Visualizar lista de Sucursales existentes. 2) Registro de Nuevas Sucursales 3) Modificar Sucursales 4) Eliminar Sucursales |
| 3.2 | Administrar Tipos de Usuarios <ol style="list-style-type: none"> 1) Visualizar lista de Tipos de Usuarios 2) Registro de Nuevo Tipo de Usuario 3) Modificar Tipos de Usuarios 4) Eliminar Tipos de Usuarios |
| 3.3 | Administración de Roles <ol style="list-style-type: none"> 1) Visualizar lista de Roles asignados a Tipos de Usuarios 2) Registrar Nuevo Rol, seleccionar un tipo de usuario para asignar rol 3) Seleccionar permisos de acceso a los módulos 4) Modificar roles 5) Eliminar Roles |
| 3.4 | Administración de Usuarios <ol style="list-style-type: none"> 1) Visualizar lista de Usuarios registrados 2) Registro de nuevos usuarios, seleccionar el tipo de usuario ya asignado los roles previamente. 3) Modificar usuarios, para cualquier cambio debe ingresar la contraseña del usuario a modificar. 4) Eliminar usuarios |
| 3.5 | Administrar Parámetros de Captaciones (Caja) |
| 3.5.1 | A. Tipos de Transacciones <ol style="list-style-type: none"> 1) Visualizar lista de Tipos de Transacciones 2) Registro de nuevos Tipos de Transacciones 3) Modificar Tipos de Transacciones 4) Eliminar Tipos de Transacciones |
| 3.5.2 | B. Conceptos de Otros Ingresos <ol style="list-style-type: none"> 1) Registro de nuevos Conceptos de Ingresos 2) Modificar Conceptos de Ingresos 3) Eliminar Conceptos de Ingresos 4) Visualizar lista de Conceptos de Otros Ingresos. |

| | |
|---------------------|---|
| <p>3.5.3</p> | <p><i>C. Causas de Notas de Débito y Crédito (NDC)</i></p> <ol style="list-style-type: none"> 1) Registro de nuevas Causas de NDC 2) Modificar Causas de NDC 3) Eliminar Causas de NDC 4) Visualizar lista de Causas de NDC |
| <p>3.6</p> | <p>Administrar Parámetros de Ahorros</p> |
| <p>3.6.1</p> | <p><i>A. Estados de Cuenta</i></p> <ol style="list-style-type: none"> 1) Visualizar lista de Estados de Cuenta 2) Registro de nuevos Estados de Cuenta (Activa, Inactiva, etc.) 3) Modificar Estados de Cuenta 4) Eliminar Estados de Cuenta |
| <p>3.6.2</p> | <p><i>B. Productos</i></p> <ol style="list-style-type: none"> 1) Visualizar lista de productos 2) Registro de nuevos Productos (Cuentas de Ahorros) 3) Modificar Productos 4) Eliminar Productos |
| <p>3.6.3</p> | <p><i>C. Plazos para Depósitos a Plazo Fijo (DPF)</i></p> <ol style="list-style-type: none"> 1) Visualizar lista de Plazos para DPF 2) Registro de nuevos Plazos para DPF 3) Modificar Plazos para DPF 4) Eliminar Plazos para DPF |
| <p>3.6.4</p> | <p><i>D. Estados de Depósitos a Plazo Fijo (DPF)</i></p> <ol style="list-style-type: none"> 1) Visualizar lista de Estados de DPF 2) Registro de nuevos Estados de DPF 3) Modificar Estados de DPF 4) Eliminar Estados de DPF |

Tabla 6: Funcionalidad de Modulo de Administración y Seguridad
Fuente: Propia

3.2.3. Descripción de los casos de uso del módulo de administración

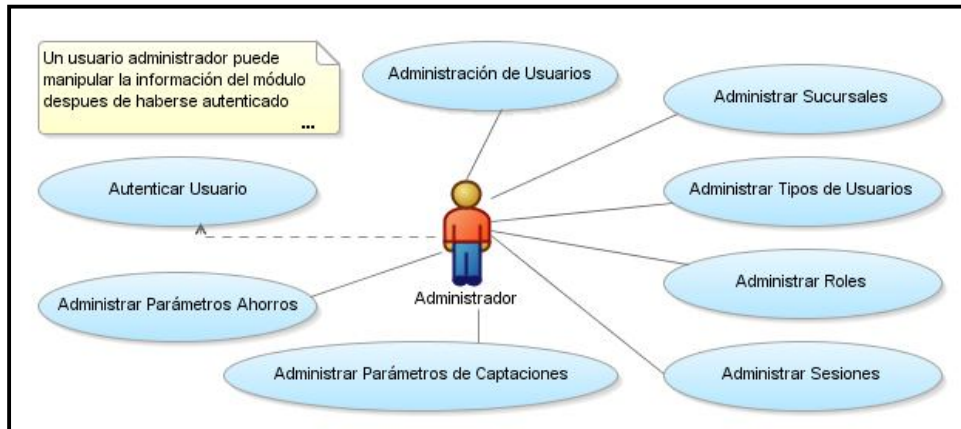


Figura 29: Modelo de casos de uso del módulo de Administración
Fuente Propia

Descripción de los casos de uso más relevantes del módulo de administración.

3.2.3.1. Especificación de Caso de Uso: Autenticación de Usuarios

1. Descripción Breve

El caso de uso describe el proceso de control de acceso al sistema utilizando autenticación de usuario y contraseña, el mismo que se utiliza para poder acceder al sistema siempre y cuando el usuario se encuentre registrado.

2. Flujo Básico de Eventos

- a) El usuario podrá visualizar la interfaz de autenticación para que pueda ingresar sus datos de su cuenta de usuario, como se muestra a continuación.

Figura 30: pantalla de Autenticación
Fuente: Propia

- b) Si el usuario desea ingresar sistema y realizar transacciones tendrá que ingresar los siguientes datos para su autenticación:
- Ingresar nombre de usuario o Login
 - Ingresar Contraseña de Usuario.
 - Seleccionar la Sucursal donde se encuentra el usuario.
- c) Al realizar el proceso de autenticación el sistema validara la información ingresada con la información de la cuenta que se encuentra registrada en la base de datos del sistema. Para ello se realiza una consulta a la tabla tab_administracion_usuarios la misma que consta de la siguiente descripción y estructura.

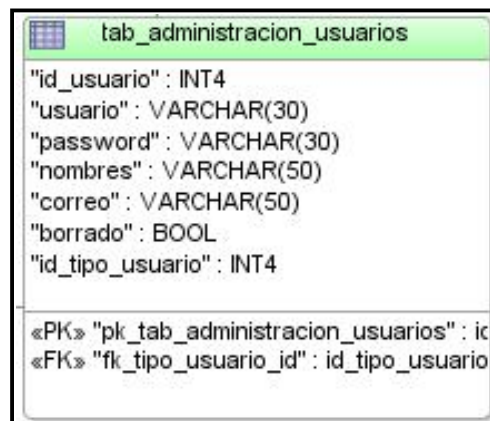


Figura 31: Modelo de la tabla usuarios
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|---------------|---------------------------------|
| ID_USUARIO | INTEGER | Clave primaria de la Tabla |
| USUARIO | VARCHAR2(30) | Nombre de Usuario o Login |
| PASSWORD | VARCHAR2(30) | Contraseña de Usuario |
| NOMBRES | VARCHAR2 (50) | Nombres y Apellidos del usuario |
| CORREO | VARCHAR2 (50) | Correo Electrónico |
| BORRADO | BOOLEAN | Campo para borrar registro |
| ID_TIPO_USUARIO | INTEGER | Identifica el Tipo de Usuario |

Tabla 7: Campos de la tabla usuario
Fuente: Propia

Nota: En la tabla anterior solo se describen los campos utilizados por la autenticación del usuario.

- d) El proceso de autenticación de usuario para el ingreso al sistema se realiza como se muestra en siguiente imagen.

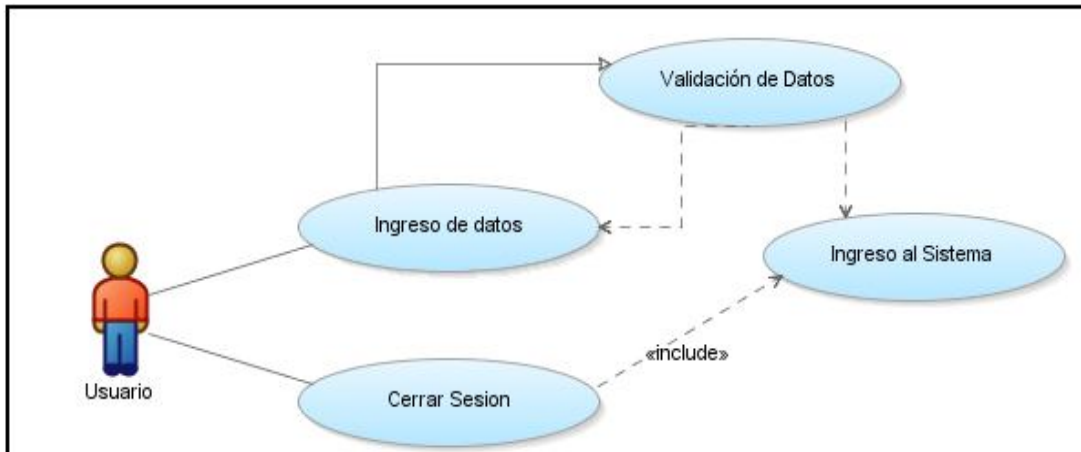


Figura 32: Caso de uso del proceso de Autenticación
Fuente: Propia

3. Flujos Alternativos

- Re direccionar a la página de registro o ingreso si el usuario no ha sido autenticado.
- Limpiar el formulario de ingreso para nuevos ingresos.
- Si los datos ingresados por el usuario no son correctos notificar para que vuelva a ingresar.
- Comunicar a los usuarios sobre errores de acceso.

4. Precondiciones

- Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.

5. Postcondiciones

- a) Si el usuario logro ingresar al sistema tendrá acceso a los diferentes menús dependiendo del rol asignado por el administrador.
- b) Las transacciones realizadas por el usuario se registraran en la base del sistema.
- c) El usuario podrá cambiar su perfil de usuario como cambar contraseñas, usuario, correo, etc., siempre que estas cumplan con las condiciones específicas del sistema.

3.2.3.2. Especificación de Caso de Uso: Administración de Usuarios

1. Descripción Breve

Este caso de uso describe el proceso de gestión de cuentas de usuarios donde se podrá crear, modificar, eliminar y ver la lista de cuentas de usuarios registradas en el sistema. Para poder realizar estas acciones debe ser usuario administrador.

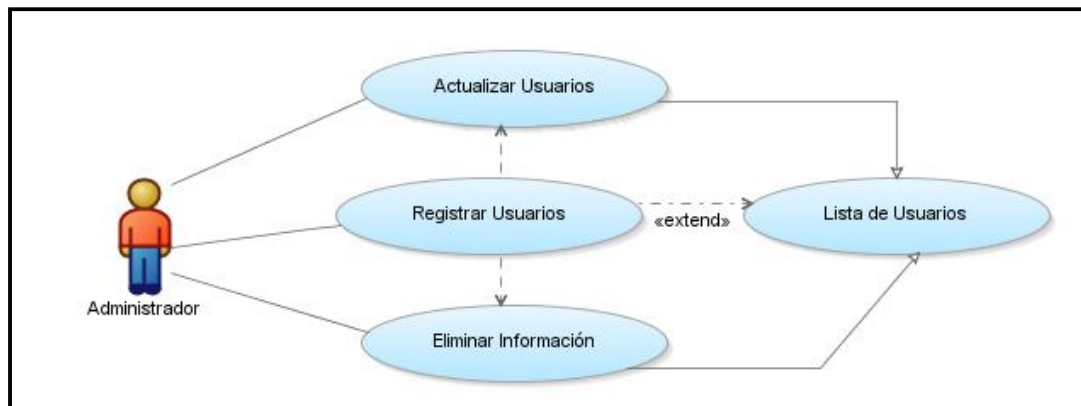


Figura 33: Caso de uso administración de usuarios

Fuente: Propia

2. Flujo Básico de Eventos

- a) El usuario administrador podrá visualizar reportes de usuarios registrados donde podrá modificar o eliminar una cuenta de usuario y también visualizará la interfaz para ingresar una nueva cuenta de usuario, como se muestra a continuación.

COOPERATIVA DE AHORRO Y CRÉDITO "UNIÓN COCHAPAMBA"

 USUARIO: BAIRON

CAJA CLIENTES AHORROS CARTERA CONTABILIDAD ADMINISTRACION AUDITORIA PROCESOS

 Guardar

INGRESO DE NUEVOS USUARIOS

Codigo de Usuario: 19

Nombres del Usuario: (*)

Login o Usuario: (*)

Contraseña de Usuario: (*)

Repetir Contraseña: (*)

Correo Electrónico: (*)

Tipo de Usuario: (*) Administrador

Lista de Usuarios Registrados

| Editar | Login | Contraseña | Nombres | Correo | Tipo Usuario | Borrar |
|--|-----------|------------------------------|------------------|-----------------------|---------------|--|
|  | bairon | v9ixccXrbheEskbG5R91JA.Xho8= | Bairon Gudino | bairongv@gmail.com | Administrador |  |
|  | henryando | +51zSj1Y14KMICZn9bTL3AReE= | Pepe Jose | henry@yahoo.es | Ahorros |  |
|  | manuel | MpdKLzD1Z14mj1QDySQSg235lqw= | esTEBAN bARAHONA | asdsa@hotmail.com | Contador |  |
|  | material | +51zSj1Y14KMICZn9bTL3AReE= | marvin Collucho | material@hoptmail.com | Contador |  |
|  | marlin | +51zSj1Y14KMICZn9bTL3AReE= | Henry Andes | henry@yyy.com | Administrador |  |
|  | marlin | +51zSj1Y14KMICZn9bTL3AReE= | Henry Andes | henry@yyy.com | Administrador |  |
|  | esteban | +51zSj1Y14KMICZn9bTL3AReE= | henry | henry@yahoo.es | Contador |  |
|  | asdasd | xe8EfBRsIWma/cXutjQv919KaU= | asdsad | asdsd@hias.com | Administrador |  |
|  | kathy | 1YPRAsT9pSihQ59dXXSxvAe8HA= | kathy | kathy@gmail.com | Contador |  |
|  | marion | 2rvvzrv++Yvx8UpP3f39w7Nm0= | marionsilver | mavs@yahoo.es | Administrador |  |

Figura 34: Pantalla de ingreso de nuevos usuarios

Fuente: Propia

b) Si el usuarios administrador desea ingresar una nueva cuenta de usuario tendrá que ingresar los siguientes datos:

- 1 Nombre de Usuario o Login
- 2 Contraseña de Usuario para el acceso al sistema.
- 3 Nombres y Apellidos del Usuario
- 4 Correo Electrónico
- 5 Seleccionar el tipo de Usuario.

c) Si el usuario desea modificar información de su cuenta tendrá que ingresar los siguientes datos:

- 1 Nombre de Usuario nuevo
- 2 Contraseña actual
- 3 Contraseña nueva(Si desea cambiar)
- 4 Nombres y Apellidos del Usuario
- 5 Correo electrónico

- d) Si el usuario desea eliminar una cuenta de usuario debe seleccionar el usuario y confirmar su eliminación.
- e) Al realizarse cualquiera de las acciones mencionadas anteriormente se debe trabajar sobre la tabla tab_administracion_usuarios, la misma que se encuentra estructurada de la siguiente forma:

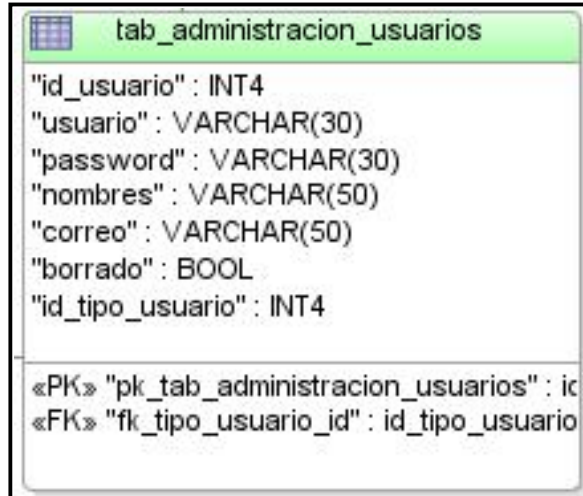


Figura 35: Diseño de la tabla usuarios
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|---------------|---------------------------------|
| ID_USUARIO | INTEGER | Clave primaria de la Tabla |
| USUARIO | VARCHAR2(30) | Nombre de Usuario o Login |
| PASSWORD | VARCHAR2(30) | Contraseña de Usuario |
| NOMBRES | VARCHAR2 (50) | Nombres y Apellidos del usuario |
| CORREO | VARCHAR2 (50) | Correo Electrónico |
| BORRADO | BOOLEAN | Campo para borrar registro |
| ID_TIPO_USUARIO | INTEGER | Identifica el Tipo de Usuario |

Tabla 8: Capos de la tabla usuarios
Fuente: Propia

Nota:

- Si el nombre de usuario registrado ya existe debe volver a ingresa un nuevo nombre de usuario.
- Para poder modificar uno de estos datos debe ingresar la contraseña del usuario.

3. Flujos Alternativos

- a) Se realiza la limpieza de pantalla o formularios para nuevos ingresos de cuentas de usuario.
- b) Se actualiza la lista de usuarios después de ingresar, modificar o eliminar una cuenta de usuario.
- c) Se puede cancelar la modificación o eliminación si no está seguro de hacerlo.
- d) Comunicar al usuario sobre errores de ingreso de datos, validaciones y contraseñas mal ingresadas.
- e) Se debe ingresar información en todos los campos marcados como obligatorios, en estos campos el sistema no aceptara valores nulos.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe cumplir con el proceso de autenticación de usuario para acceder al menú administrar Usuarios.
- d) Para poder ingresar, modificar, eliminar o ver la lista de usuarios debe tener rol de administrador.
- e) Para el ingreso de un nuevo usuario primero deberá ingresar tipos de usuarios y asignarle un rol a ese tipo de usuario.

5. Postcondiciones

- a) El usuario administrador podrá agregar, editar, eliminar información de su cuenta y cuentas de otros usuarios.
- b) Los datos ingresados se registraran en la base del sistema.

3.2.3.3. Especificación de Caso de Uso: Administrar Parámetros Generales

1. Descripción Breve

Este caso de uso describe el registro de parámetros generales del sistema existentes para la Cooperativa. Para poder realizar estas acciones debe ser usuario administrador.

2. Flujo Básico de Eventos

2.1.El usuario administrador podrá seleccionar cualquiera de las siguientes opciones de parámetros Generales: Estados de Cuentas, Productos, Conceptos de Ingresos, Causas Notas de Crédito-Débito y Tipos de Transacciones.

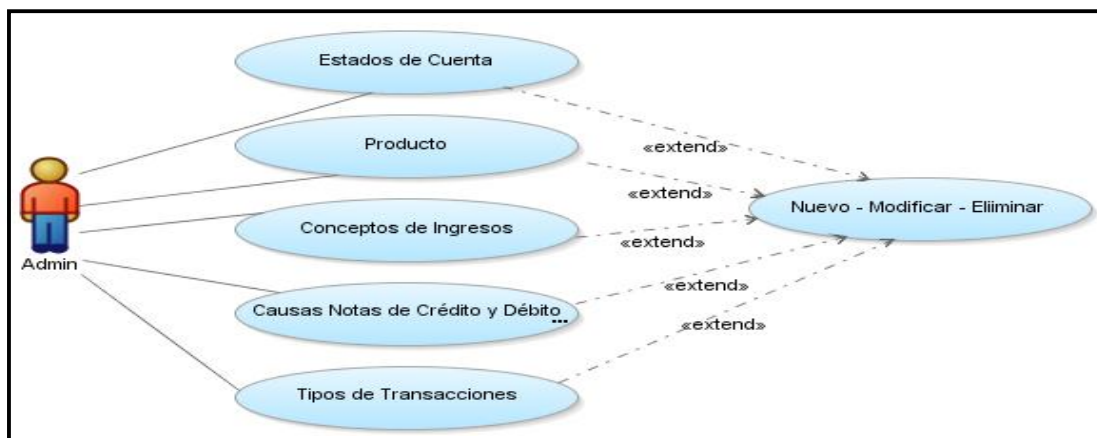


Figura 36: Casos de uso del proceso de parámetros generales
Fuente: Propia

2.2.Administrar Estados de Cuenta

a) Para administrar los Estados de Cuenta el usuario visualizar la lista de estados de cuenta registradas. La interfaz gráfica se muestra a continuación.

| Editar | Código Estado | Estado de Cuenta | Depositos | Retiros | Interes | Capitaliza | Borrar |
|--------|---------------|------------------|-----------|---------|---------|------------|--------|
| | 3 | Bloqueada | s | n | s | n | |
| | 2 | Desactivada | s | n | s | n | |
| | 1 | Activa | s | s | s | s | |

Figura 37: Pantalla de ingreso de estados de cuenta
Fuente: Propia

b) Si el usuarios administrador desea ingresar un nuevo estado de cuenta tendrá que ingresar los siguientes datos:

- 1 Nombre del Estado
- 2 Depósitos
- 3 Retiros
- 4 Interés
- 5 Capitaliza

c) Si el usuario desea modificar información de un estado de cuenta tendrá que ingresar los siguientes datos:

- 1 Nombre del Estado
- 2 Depósitos
- 3 Retiros
- 4 Interés
- 5 Capitaliza

| Edición de Tipos de Estado de Cuenta | |
|--------------------------------------|-------------|
| ✓ Modificar ✕ Cancelar | |
| Click para Modificar | |
| Codigo Estado: | 2 |
| Nombre Estado: | Desactivada |
| Depositos: | SI |
| Retiros: | NO |
| Interes: | SI |
| Capitaliza: | NO |

Figura 38: Pantalla de edición de estados de cuenta
Fuente: Propia

d) Si el usuario desea eliminar un estado de cuenta debe seleccionar el usuario y confirmar su eliminación.

- e) Al realizarse cualquiera de las acciones mencionadas anteriormente para los estados de cuenta se debe trabajar con la tabla `tab_ahorros_tipo_estado_cuenta`, la misma que se encuentra estructurada de la siguiente forma:

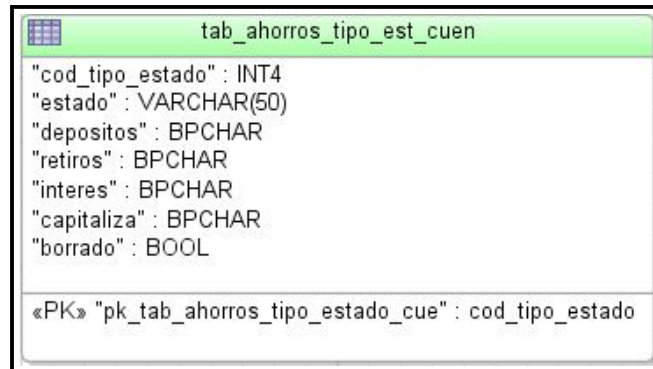


Figura 39: Diseño de la tabla tipos de estados de cuenta
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|-------------|----------------------------|
| COD_TIPO_ESTADO | INTEGER | Clave primaria de la Tabla |
| ESTADO | VARCHAR(50) | Estado de tipo de cuenta |
| DEPOSITOS | CHAR | Seleccionar SI o NO |
| RETIROS | CHAR | Seleccionar SI o NO |
| INTERES | CHAR | Seleccionar SI o NO |
| CAPITALIZA | CHAR | Seleccionar SI o NO |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 9: Campos de la tabla estados de cuenta
Fuente: Propia

2.3.Administrar Productos

- a) Para administrar los Productos de la Cooperativa el usuario visualizar la interfaz para ingresar un nuevo producto y la lista de productos. La interfaz gráfica se muestra a continuación.

INGRESO DE PRODUCTOS

Ingreso de Nuevos Productos

Codigo Producto: 3

Nombre Producto (*)

Tasa de Interés (*) 0.0

Máximo de Retiro (*) 0.0

Mínimo en Libreta (*) 0.0

Cuenta Contable (*)

Cuenta Interés (*)

Lista de Productos

| Editar | Código Producto | Nombre producto | Tasa de Interés | Máximo de Retiro | Mínimo en Libreta | Borrar |
|--------|-----------------|----------------------------|-----------------|------------------|-------------------|--------|
| | 2 | Certificados de Aportacion | 0.0 | 0.0 | 0.0 | |
| | 1 | Cuenta de Ahorros | 1.0 | 5000.0 | 10.0 | |

Figura 40: Pantalla de administración de productos
Fuente: Propia

- b) Si el usuarios administrador desea ingresar un nuevo producto tendrá que ingresar los siguientes datos:
- 1 Nombre del Producto
 - 2 Tasa de Interés
 - 3 Máximo de Retiro
 - 4 Mínimo de Retiro
 - 5 Cuenta Contable
 - 6 Cuenta Interés
- c) Si el usuario desea modificar información de un producto tendrá que modificar los siguientes datos:
- 1 Nombre del Producto
 - 2 Tasa de Interés
 - 3 Máximo de Retiro
 - 4 Mínimo de Retiro
 - 5 Cuenta Contable
 - 6 Cuenta Interés

- d) Si el usuario desea eliminar un producto debe seleccionar el usuario y confirmar su eliminación.
- e) Al realizarse cualquiera de las acciones mencionadas anteriormente para los productos de la cooperativa se debe trabajar sobre la tabla tab_ahorros_productos, la misma que se encuentra estructurada de la siguiente forma:



Figura 41: Diseño de la tabla productos
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|-------------|----------------------------|
| COD_PRODUCTO | INTEGER | Clave primaria de la Tabla |
| PRODUCTO | VARCHAR(50) | Nombre del Producto |
| TASA_INTERES | FLOAT | Ingresar cantidad |
| MAXIMO_RETIRO | FLOAT | Ingresar cantidad |
| MINIMO_LIBRETA | FLOAT | Ingresar cantidad |
| CUENTA_INTERES | VARCHAR(20) | Clave foránea |
| CUENTA_CONTABLE | VARCHAR(20) | Clave foránea |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 10: Campos de la tabla productos
Fuente: Propia

2.4. Administrar Conceptos de Ingresos

- a) Para administrar los Conceptos de Ingresos de la Cooperativa el usuario visualizar la interfaz para ingresar un nuevo concepto de ingreso y la lista de conceptos ingresados. La interfaz gráfica se muestra a continuación.

| INGRESAR CONCEPTOS DE OTROS INGRESOS | | | | | |
|--------------------------------------|--------|-----------------------|------------------|-----------------|--------|
| Codigo de Concepto: 3 | | | | | |
| Descripción de Concepto: (*) | | | | | |
| Selecciones un Tipo: (*) Ingreso | | | | | |
| Cuenta Contable: (*) | | | | | |
| Lista de Otros Ingresos | | | | | |
| Editar | Código | Concepto de Ingresos | Tipo de Concepto | Cuenta Contable | Borrar |
| | 1 | CUOTA DE INGRESO | Ingreso | 5.6.90.03 | |
| | 2 | MOVILIZACION PERSONAL | Egreso | 5.1.02.05 | |

Figura 42: Pantalla de conceptos de Ingresos

Fuente: Propia

- b) Si el usuario administrador desea ingresar un nuevo concepto de ingreso tendrá que ingresar los siguientes datos:
- 1 Descripción de Concepto
 - 2 Seleccionar el tipo (Ingreso o Egreso)
 - 3 Seleccionar Cuenta Contable
- c) Si el usuario desea modificar información de un concepto de ingreso tendrá que modificar los siguientes datos:
- 1 Descripción de Concepto
 - 2 Seleccionar el tipo (Ingreso o Egreso)
 - 3 Seleccionar Cuenta Contable
- d) Si el usuario desea eliminar un concepto de ingreso debe seleccionar el usuario y confirmar su eliminación.

- e) Al realizarse cualquiera de las acciones mencionadas anteriormente para los concepto de ingreso de la cooperativa se debe trabajar sobre la tabla tab_ahorros_conceptos_oi, la misma que se encuentra estructurada de la siguiente forma:

```

TAB_AHORROS_CONCEPTOS_OI
"cod_concepto": INT4
"concepto": VARCHAR(100)
"borrado": BOOL
"tipo": VARCHAR(20)
"cuenta_contable": VARCHAR(20)

«PK» "pk_tab_ahorros_conceptos_otros": cod_concepto
«FK» "fk_tab_aho_concep_o_i": cuenta_contable

```

Figura 43: Diseño de la tabla conceptos de ingresos
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|--------------|----------------------------|
| COD_CONCEPTO | INTEGER | Clave primaria de la Tabla |
| CONCEPTO | VARCHAR(100) | Describir el Concepto |
| TIPO | VARCHAR(20) | Tipo(Ingreso o Egreso) |
| CUENTA_CONTABLE | VARCHAR(20) | Clave foránea |
| BORRADO | BOOLEAN | Campo para borrar |

Tabla 11: Campos de la tabla conceptos de ingresos
Fuente: Propia

2.5.Administrar Causas Débito/Crédito

- a) Para administrar las Causas de Débitos y Créditos deber realizar lo siguiente:



Figura 44: Pantalla administración de causas de notas de débito y crédito
Fuente: Propia

- b) Si el usuarios administrador desea ingresar una nueva Causas de Débitos y Créditos tendrá que ingresar los siguientes datos:
- 1 Descripción de la Causa de Débito o Crédito
 - 2 Seleccionar el tipo (Débito o Crédito)
 - 3 Seleccionar Cuenta Contable
- c) Si el usuario desea modificar información de una Causas de Débitos y Créditos tendrá que modificar los siguientes datos:
- 1 Descripción de la Causa de Débito o Crédito
 - 2 Seleccionar el tipo (Débito o Crédito)
 - 3 Seleccionar Cuenta Contable
- d) Si el usuario desea eliminar una Causas de Débitos y Créditos debe seleccionar el usuario y confirmar su eliminación.
- e) Al realizarse cualquiera de las acciones mencionadas anteriormente para las Causas de Débitos y Créditos de la cooperativa se debe trabajar sobre la tabla tab_ahorros_productos, la misma que se encuentra estructurada de la siguiente forma:

```

TAB_AHORROS_CAUSAS_DEB_C
"cod_causa" : INT4
"descripcion_causa" : VARCHAR(50)
"borrado" : BOOL
"tipo" : VARCHAR(50)
"cuenta_contable" : VARCHAR(20)

«PK» "pk_tab_ahorros_causas_debito_c" : cod_causa
«FK» "fk_tab_ahorros_cau_dc1" : cuenta_contable

```

Figura 45: Diseño de la tabla de causas de notas de débito y crédito
Fuente: Propia

| Campo | Tipo | Observación |
|-----------------|-------------|----------------------------|
| COD_CAUSA | INTEGER | Clave primaria de la Tabla |
| DESCRIPCION | VARCHAR(50) | Describir la causa |
| TIPO | VARCHAR(50) | Tipo(Débito - Crédito) |
| CUENTA_CONTABLE | VARCHAR(20) | Clave foránea |
| BORRADO | BOOLEAN | Campo para borrar |

Tabla 12: Campos de la tabla de causas de notas de débito y crédito
Fuente: Propia

2.6. Administrar Tipos de Transacciones

- a) Para administrar los Tipos de Transacciones de la Cooperativa el usuario visualizar la interfaz para ingresar un nuevo Tipo de Transacción y la lista de estos tipos. La interfaz gráfica se muestra a continuación.

| INGRESAR TIPOS DE TRANSACCIONES | | | | | | | |
|---|-------------------------|---------------------------|-------------|---------|-----------|-----------------|--------|
| Tipo de Transacción: (*) <input type="text"/> | | | | | | | |
| Abreviatura: (*) <input type="text"/> | | | | | | | |
| Sección: (*) Ahorros <input type="text"/> | | | | | | | |
| Operación: (*) Suma <input type="text"/> | | | | | | | |
| Cuenta Contable: (*) <input type="text"/> | | | | | | | |
| Lista de Tipos de Transacciones | | | | | | | |
| Editar | Código Tipo Transacción | Tipo de Transacción | Abreviatura | Sección | Operación | Cuenta Contable | Borrar |
| | 901 | OTROS INGRESO | OTROSI | O | - | 1.1.01.10 | |
| | 211 | PAGO PRESTAMOS CASTIGADOS | PGPCAS | P | - | 1.1.01.10 | |

Figura 46: Pantalla de administración de tipos de transacciones
Fuente: Propia

- b) Si el usuarios administrador desea ingresar un nuevo tipo de transacción tendrá que ingresar los siguientes datos:
- 1 Tipo de Transacción (Descripción)
 - 2 Abreviatura del tipo de Transacción
 - 3 Sección (Ahorros, Préstamos, etc.)
 - 4 Operación (Suma o Resta)
 - 5 Seleccionar Cuenta Contable
- c) Si el usuario desea modificar información de un tipo de transacción tendrá que modificar los siguientes datos:
- 1 Tipo de Transacción (Descripción)
 - 2 Abreviatura del tipo de Transacción
 - 3 Sección (Ahorros, Préstamos, etc.)
 - 4 Operación (Suma o Resta)
 - 5 Seleccionar Cuenta Contable

- d) Si el usuario desea eliminar un tipo de transacción debe seleccionar el usuario y confirmar su eliminación.
- e) Al realizarse cualquiera de las acciones mencionadas anteriormente para los tipos de transacciones de la cooperativa se debe trabajar sobre la tabla tab_ahorros_tipo_transaccion, la misma que se encuentra estructurada de la siguiente forma:

```

tab_ahorros_tipo_transaccion
"cod_tipo_transaccion" : INT4
"tipo_transaccion" : VARCHAR(50)
"borrado" : BOOL
"seccion" : CHAR(1)
"operacion" : CHAR(1)
"abreviatura" : VARCHAR(30)
"cuenta_contable" : VARCHAR(20)

«PK» "pk_tab_ahorros_tipo_transaccio" : cod_tipo_transaccio
«FK» "fk_tab_ahorros_t_t_cc" : cuenta_contable

```

Figura 47: Diseño de la tabla tipos de transacciones
Fuente: Propia

| Campo | Tipo | Observación |
|----------------------|-------------|----------------------------|
| COD_TIPO_TRANSACCION | INTEGER | Clave primaria de la Tabla |
| TIPO_TRANSACCION | VARCHAR(50) | Describir el Concepto |
| SECCION | CHAR(1) | Seleccionar la sección |
| OPERACION | CHAR(1) | Seleccionar Operación |
| ABREVIATURA | VARCHAR(30) | Ingreso de Abreviatura |
| CUENTA _CONTABLE | VARCHAR(20) | Clave foránea |
| BORRADO | BOOLEAN | Campo para borrar |

Tabla 13: Capos de la tabla de tipos de transacciones
Fuente: Propia

3. Flujos Alternativos

- a) Se realiza la limpieza de pantallas para nuevos ingresos de cuentas de usuario.
- b) Se actualizan las listas después de ingresar, modificar o eliminar.
- c) Se puede cancelar la modificación o eliminación si no está seguro de hacerlo.
- d) Comunicar al usuario sobre errores de ingreso de datos, validaciones.

- e) Se debe ingresar información en todos los campos marcados como obligatorios, en estos campos el sistema no aceptara valores nulos.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe cumplir con el proceso de autenticación de usuario para acceder al menú administrar Sesiones.
- d) Debe tener permisos para visualizar, crear y alterar esta información.

5. Postcondiciones

- a) Los datos ingresados se registrarán en la base del sistema y en las tablas de auditoría, ver casos de uso del módulo de auditoría.

3.2.4. Descripción del módulo de Captaciones (Caja)

El módulo de captaciones es aquel donde se realizan todas las transacciones diarias recibidas por varios conceptos como depósitos, retiros, pagos de préstamos, transferencias, etc. El monto que deberá permanecer en caja será de 5000 dólares americanos como máximo, y los pagos en efectivo, retiros de ahorro, entrega de crédito, pago de bienes y servicios se podrá realizar hasta un máximo de 1000 dólares pasado este monto se entregara cheques para su retiro en bancos vinculados con la cooperativa.

3.2.5. Funcionamiento del módulo de Captaciones (Caja)

Solo un usuario tiene permisos acceso al módulo de caja podrá realizar las siguientes transacciones como se muestra a continuación.

La siguiente tabla muestra los ítems o funciones que realizara el sistema para el módulo de Captaciones.

| Paso | Acción |
|------------|--|
| 1 | Autenticarse como usuario Cajero |
| 2 | Ingresar al menú Caja |
| 3 | <p>Seleccionar una de las siguientes Opciones:</p> <ol style="list-style-type: none"> 1. Apertura de Caja 2. Apertura de Cuenta 3. Depósito de Ahorros 4. Retiros en Efectivo 5. Retiros con Cheque 6. Pago de Préstamos 7. Registro Otros Ingresos 8. Depósitos a Plazo Fijo 9. Arqueos de Caja 10. Consultas y Reportes 11. Backups y Respaldos |
| 3.1 | <p>Apertura de Caja</p> <ol style="list-style-type: none"> a) Seleccionar la Fecha de la apertura de Caja |
| 3.2 | <p>Apertura de Cuenta</p> <ol style="list-style-type: none"> a) Seleccionar el Cliente registrado en el día actual. b) Ingresar valores a los campos requeridos c) Realizar la Apertura de Cuenta d) Imprimir comprobante de transacción. |
| 3.3 | <p>Depósito de Ahorros</p> <ol style="list-style-type: none"> a) Seleccionar la Cuenta de Ahorro b) Ingresar valores a los campos requeridos c) Realizar el Depósito de Ahorros en Efectivo d) Imprimir en la libreta si lo desea e) Imprimir comprobante de transacción |
| 3.4 | <p>Retiros en Efectivo</p> <ol style="list-style-type: none"> a) Seleccionar la Cuenta de Ahorro b) Ingresar valores a los campos requeridos c) Realizar el Retiro de Ahorros en Efectivo d) Imprimir en la libreta si lo desea e) Imprimir comprobante de transacción |
| 3.5 | <p>Retiros con Cheque</p> <ol style="list-style-type: none"> a) Seleccionar la Cuenta de Ahorro b) Ingresar valores a los campos requeridos c) Realizar el Retiro de Ahorros con Cheque d) Imprimir en la libreta si lo desea e) Imprimir comprobante de transacción |
| 3.6 | Pago de Préstamos |

| | |
|-------------|--|
| | <ul style="list-style-type: none"> a) Seleccionar el Cliente b) Seleccionar el Préstamo a pagar del Cliente c) Ingresar valores a los campos requeridos d) Realizar el pago de Préstamo. e) Imprimir comprobante de transacción |
| 3.7 | <p>Registro Otros Ingresos</p> <ul style="list-style-type: none"> a) Seleccionar el socio b) Elegir la causa del Ingreso c) Ingresar los valore a los campos requeridos d) Realizar el registro del Ingreso e) Imprimir comprobante de transacción |
| 3.8 | <p>Depósitos a Plazo Fijo (DPF)</p> <p>3.7.1 Registro de Nuevo DPF</p> <ul style="list-style-type: none"> a) Seleccionar el socio b) Ingresar los valore a los campos requeridos c) Registrar el DPF d) Imprimir contrato del DPF <p>3.7.2 Cancelar DPF</p> <ul style="list-style-type: none"> a) Seleccionar el DPF a cancelar b) Si va a renovar ingresar los valores de los campos requeridos c) Registrar el pago y la cancelación del DPF d) Imprimir comprobante de pago. |
| 3.9 | <p>Arqueos de Caja</p> <ul style="list-style-type: none"> a) Seleccionar la Fecha del Arqueo b) Seleccionar el Usuario cajero que realizo las transacciones y la Sucursal de las mismas c) Imprimir arqueo de Caja |
| 3.10 | <p>Consultas y Reportes</p> <ul style="list-style-type: none"> a) Realiza consultas con diferentes parámetros sobre las cuentas de de los clientes y sus transacciones. |
| 3.11 | <p>Backups y Respaldos</p> <ul style="list-style-type: none"> a) Se saca respaldos de la información generada a la fecha |

Tabla 14: Funcionalidad de Modulo de Captaciones

Fuente: Propia

3.2.6. Descripción de los casos de uso del módulo de Captaciones

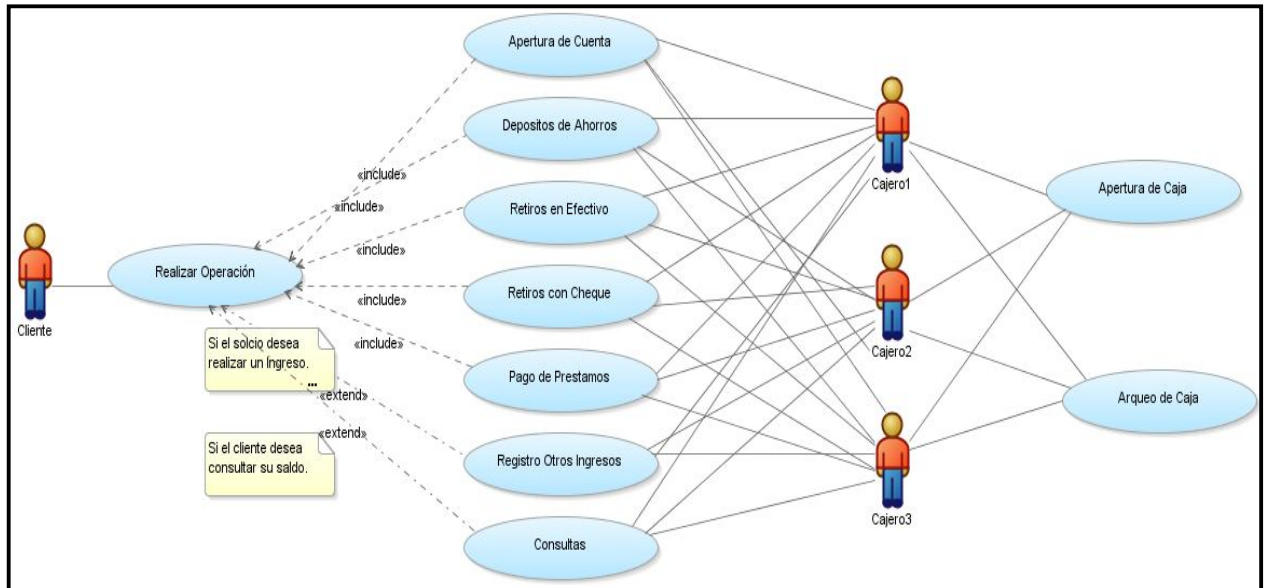


Figura 48: Casos de uso del módulo de Captaciones
Fuente: Propia

A continuación se muestra la descripción de los casos de uso más relevantes del módulo de Captaciones.

3.2.6.1. Especificación de Caso de Uso: Apertura de Caja

1. Descripción Breve

Este caso de uso describe el proceso que indica cómo realizar la Apertura de Caja del día para empezar a realizar transacciones sobre las cuentas.

2. Flujo Básico de Eventos

- a) Seleccionamos Apertura de Caja del menú Caja, el usuario podrá visualizar la interfaz para seleccionar la fecha de la apertura como se muestra a continuación.



Figura 49: Pantalla selección de la fecha de trabajo
Fuente: Propia

- b) El sistema cargara automáticamente la fecha seleccionada, el usuario y la sucursal, la interfaz para ingresar los datos de la apertura de caja se muestra en la siguiente imagen.

| Editar | Código Apertura | Fecha Apertura | Operador | Sucursal | Valor Apertura |
|--------|-----------------|----------------|----------|----------|----------------|
| | 14 | 2012-01-30 | bairon | lbarra | 3000.0 |
| | 13 | 2012-01-28 | bairon | lbarra | 5000.0 |

Figura 50: Pantalla de apertura de Caja
Fuente: Propia

- c) El usuario cajero ingresa el valor de la Apertura de Caja del día.
d) El usuario hace clic en el botón “Abrir Caja”.
e) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
f) Los datos ingresados se registraran en la tabla tab_caja_apertura la misma que se encuentra estructurada de la siguiente manera

| tab_caja_apertura | |
|---------------------------------|----------------|
| "cod_apertura" | : INT4 |
| "operador" | : INT4 |
| "oficina" | : INT4 |
| "fecha_apertura" | : DATE |
| "valor_apertura" | : FLOAT8 |
| "borrado" | : BOOL |
| «PK» "pk_tab_caja_apertura_cod" | : cod_apertura |
| «FK» "fk_tab_caja_apertura_op" | : operador |
| «FK» "fk_tab_caja_apertura_suc" | : oficina |

Figura 51: Diseño de la tabla apertura de caja
Fuente: Propia

| Campo | Tipo | Observación |
|----------------|-------------|----------------------------|
| COD_APERTURA | INTEGER | Clave primaria de la Tabla |
| OPERADOR | INTEGER | Clave Foránea |
| OFICINA | DATE | Clave Foránea |
| FECHA_APERTURA | INTEGER | Clave Foránea |
| VALOR_APERTURA | INTEGER | Clave Foránea |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 15: Campos de la tabla apertura de caja
Fuente: Propia

3. Flujos Alternativos

- a) El Usuario podrá realizar búsquedas de aperturas realizadas en fecha anteriores ingresando la fecha en el cuadro de texto del menú horizontal.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.

5. Postcondiciones

- a) La apertura realizada se visualizará en la lista después de su ingreso.

3.2.6.2. Especificación de Caso de Uso: Apertura de Cuenta

1. Descripción Breve

Este caso de uso describe el proceso para realizar la Apertura de Cuenta de clientes registrados en el día.

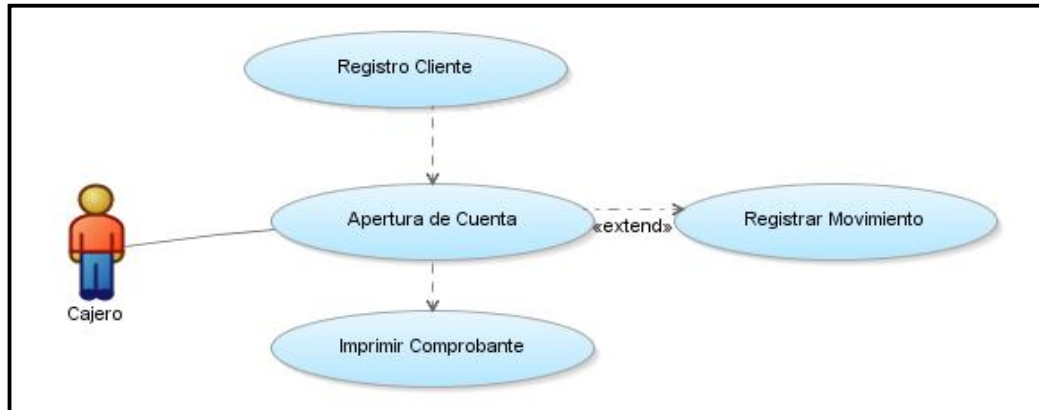


Figura 52: Caso de uso Apertura de Cuenta
Fuente: Propia

2. Flujo Básico de Eventos

- Seleccionamos Apertura de Cuenta del menú Caja, la interfaz para ingresar los datos de la Apertura de Cuenta se muestra en la siguiente imagen.

La interfaz de usuario muestra un formulario de 'APERTURA DE CUENTA' con los siguientes campos:

| | |
|----------------------------|--------|
| Codigo de Socio: (*) | 1029 |
| Sucursal: | Ibarra |
| Ahorros: (*) | 500.0 |
| Ahr. Cert. Aportación: (*) | 20.0 |
| Cuota de Ingreso: (*) | 5.0 |
| ===== | |
| Total Apertura: | 525.0 |
| Nro de Recibo: (*) | 365 |

En la parte superior derecha del formulario hay un botón 'Guardar Comprobante'. En la barra de navegación superior se ven los botones 'Abrir Cuenta' y 'Limpiar'.

Figura 53: Pantalla apertura de cuentas
Fuente: Propia

- b) El usuario selecciona al cliente ingresado en la fecha actual.
- c) El sistema carga los datos del cliente automáticamente.
- d) El usuario cajero ingresa los valores para ahorros, certificados de aportación, cuota de ingreso y el número de recibo.
- e) El sistema realiza la suma de los valores automáticamente y actualiza antes de realizar la transacción
- f) El usuario hace clic en el botón “Abrir Cuenta”.
- g) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- h) Los datos ingresados se registraran en la tabla tab_ahorros_movimientos la misma que se encuentra estructurada de la siguiente manera.

```

tab_ahorros_movimientos
"nro_movimiento" : INT4
"cod_tipo_transaccion" : INT4
"nro_cuenta" : INT4
"fecha_transaccion" : DATE
"valor_transaccion" : FLOAT8
"valor_efectivo" : FLOAT8
"valor_cheque" : FLOAT8
"saldo_actual" : FLOAT8
"operador" : INT4
"borrado" : BOOL
"cod_producto" : INT4
"nro_recibo" : VARCHAR(30)
"con_libreta" : CHAR(1)
"impreso" : CHAR(1)
"oficina" : INT4
"nro_linea" : INT4

«PK» "tab_ahorros_movimientos_pkey" : nro_movimiento, nro_cuenta, cod_producto
«FK» "fk_tab_ahor_ahor3" : cod_tipo_transaccion
«FK» "fk_tab_mov_sucr" : oficina
«FK» "fk_tab_ahor_tab_opera" : operador

```

Figura 54: Diseño de la tabla movimientos
Fuente: Propia

| Campo | Tipo | Observación |
|----------------------|---------|----------------------------|
| NRO_MOVIMIENTO | INTEGER | Clave primaria de la Tabla |
| COD_TIPO_TRANSACCION | INTEGER | Clave Foránea |
| NRO_CUENTA | INTEGER | Clave Foránea |
| COD_PRODUCTO | INTEGER | Clave Foránea |
| OPERADOR | INTEGER | Clave Foránea |

| | | |
|-------------------|-------------|--|
| OFICINA | INTEGER | Clave Foránea |
| FECHA_TRANSACCION | DATE | Registra fecha de Transacción |
| VALOR_TRANSACCION | FLOAT | Valor total de la transacción |
| VALOR_EFECTIVO | FLOAT | Valor en efectivo |
| VALOR_CHEQUE | FLOAT | Valor con cheque |
| SALDO_ACTUAL | FLOAT | Valor después de la transacción |
| NRO_RECIBO | VARCHAR(30) | Nº de recibo emitido |
| CON_LIBRETA | CHAR(1) | Marca si la transacción se realizó con libreta |
| IMPRESO | CHAR(1) | Marca si la transacción se imprimió |
| NRO_LINEA | INTEGER | Registra el número de línea de 1 a 29 |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 16: Campos de la tabla movimientos
Fuente: Propia

3. Flujos Alternativos

- a) El usuario selecciona el botón borrar para limpiar la pantalla y realizar otra apertura de cuenta si en caso lo necesita.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.
- d) El cliente debe estar registrado para lo cual se lo realiza como se explica en los casos de uso del módulo Clientes.
- e) El cliente debe registrarse en la fecha que se realizará la apertura de Cuenta.

5. Postcondiciones

- a) Los datos ingresados se podrán visualizar en el módulo ahorros como se indica en el caso de uso 001 UC Cuentas de Ahorros del módulo de Ahorros.
- b) Si desea se imprimirá el respectivo Comprobante de Pago.

3.2.6.3. Especificación de Caso de Uso: Depósitos de Ahorros

1. Descripción Breve

Este caso de uso describe el proceso para realizar Depósitos de Ahorros sobre las cuentas de ahorros de los clientes registrados.

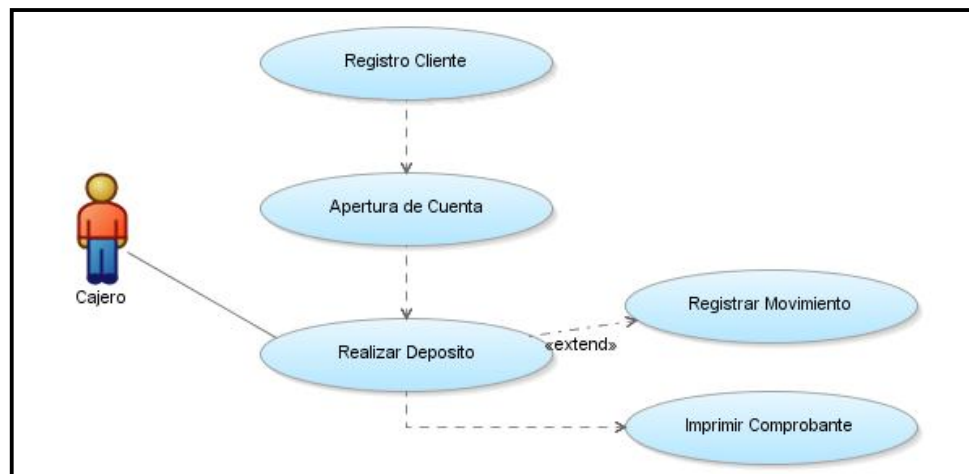


Figura 55: Caso de uso depósito de ahorros
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Depósito de Ahorros del menú Caja, la interfaz para ingresar los datos del Depósito de Ahorros se muestra en la siguiente imagen.

DEPÓSITOS DE AHORROS

| | |
|-----------------------|--|
| Numero de Cuenta: (*) | 101001029 |
| Codigo Socio: | 1029 |
| Ciente: | PICUASI DUQUE |
| | EDGAR DANIEL |
| Producto: | Cuenta de Ahorros |
| Oficina: | Ibarra |
| Saldo: | 500.0 |
| Efectivo: (*) | 200 |
| Cheques: (*) | 0.0 |
| ===== | |
| Total Depósito: | 200.0 |
| Nro de Recibo: (*) | 854 |
| Con Libreta: | <input type="radio"/> Si <input checked="" type="radio"/> No |

Figura 56: Pantalla de depósitos de ahorros
Fuente: Propia

- b)** El usuario ingresa y selecciona la cuenta de ahorros del cliente a realizar la transacción.
- c)** El sistema carga los datos del cliente automáticamente.
- d)** El usuario cajero ingresa los valores de depósitos en efectivo o en cheque y el número de recibo emitido.
- e)** El usuario deberá seleccionan si la transacción se registrara con libreta de ahorros o no.
- f)** El sistema realiza la suma de los valores automáticamente y actualiza el total del depósito antes de realizar la transacción
- g)** El usuario hace clic en el botón “Depositar” y se realizará la transacción respectiva.
- h)** El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- i)** Los datos ingresados se registraran en la tabla tab_ahorros_movimientos la misma que se encuentra estructurada como se indicó en el caso de uso anterior de este documento.

3. Flujos Alternativos

- a) El usuario selecciona el botón borrar para limpiar la pantalla y realizar otro Depósito de Ahorros de otros clientes.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.

5. Postcondiciones

- a) Los datos ingresados se podrán visualizar en el módulo ahorros como se indica en el caso de uso 001 UC Cuentas de Ahorros del módulo de Ahorros.
- b) Si desea se imprimirá el respectivo Comprobante de Pago.

3.2.6.4. Especificación de Caso de Uso: Retiros en Efectivo

1. Descripción Breve

Este caso de uso describe el proceso para realizar Retiros en Efectivo sobre las cuentas de ahorros de los clientes registrados.

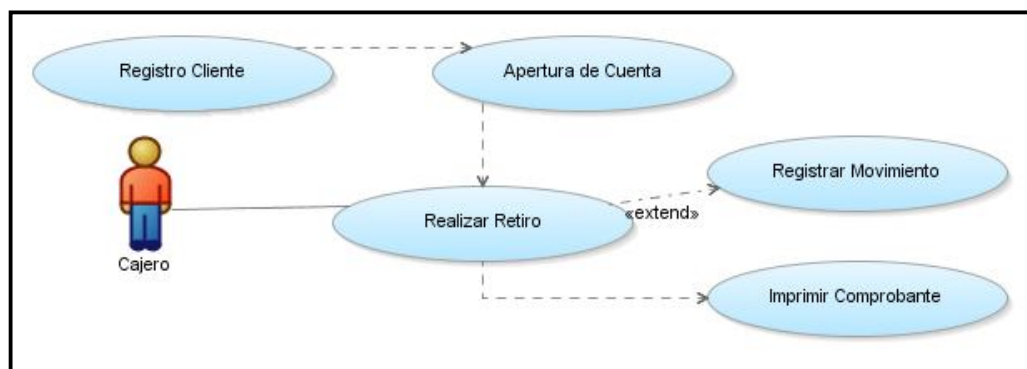


Figura 57: Caso de uso retiros de Ahorros
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Retiros en Efectivo del menú Caja, la interfaz para ingresar los datos del Retiro de Ahorros se muestra en la siguiente imagen.

 Si No'. The form has a 'Realizar Retiro' button and a 'Retirar' button in the top right corner."/>

RETIRO DE AHORROS EN EFECTIVO

Numero de Cuenta: (*) 101001029

Cedula Cliente: 1002954988

Nombres Cliente: PICUASI DUQUE
EDGAR DANIEL

Producto Cliente: Cuenta de Ahorros

Oficina Retiro: Ibarra

Saldo Disponible: 700.0

Efectivo a Retirar: (*) 200

Nro de Recibo: (*) 845

Con Libreta: Si No

Figura 58: Pantalla de retiros de ahorros
Fuente: Propia

- b) El usuario ingresa y selecciona la cuenta de ahorros del cliente a realizar la transacción.
- c) El sistema carga los datos del cliente automáticamente.
- d) El sistema muestra la fotografía del cliente la cual se guarda al momento de abrir una cuenta.
- e) El usuario cajero ingresa el valor de retiros en efectivo y el número de recibo emitido.
- f) El usuario seleccionan si la transacción se registrará con libreta de ahorros o no.
- g) El usuario hace clic en el botón “Retirar”.
- h) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- i) Los datos ingresados se registraran en la tabla tab_ahorros_movimientos la misma que se encuentra estructurada como se indico en el caso de uso anterior de este documento.

3. Flujos Alternativos

- a) El usuario selecciona el botón borrar para limpiar la pantalla y realizar otro Depósito de Ahorros de otros clientes.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.
- d) El cliente debe haber realizado la apertura de la Cuenta.

5. Postcondiciones

- a) Los datos ingresados se podrán visualizar en el módulo ahorros como se indica en el caso de uso 001 UC Cuentas de Ahorros del módulo de Ahorros.

3.2.6.5. Especificación de Caso de Uso: Pago de Préstamo

1. Descripción Breve

Este caso de uso describe el proceso para realizar Pagos de Préstamos.

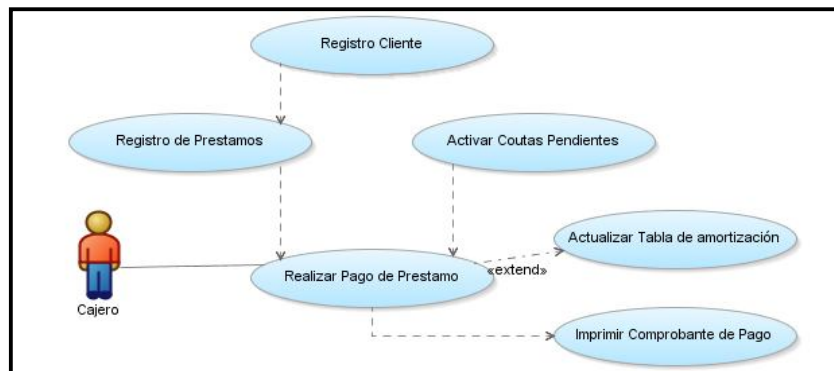


Figura 59: Caso de uso pago de préstamos
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Pago de Préstamo menú Caja y nos mostrará la interfaz para ingresar los datos del pago del préstamo como se muestra en la siguiente imagen.

The screenshot shows a web application window titled "PAGOS DE PRESTAMOS". At the top right, there is a "Limpiar" button. The main content area is titled "Ingreso de Datos:" and contains a dropdown menu for "Cuenta del Socio (*)". Below this is a sub-section titled "Préstamos" with a table header: "Pagar", "Número Préstamo", "Cantidad", "Linea de Crédito", and "Tabla de Amortización". Underneath the table is a form with several input fields: "Cuota #:", "Fecha Pago:" (with the value "30/01/2012"), "Capital a Pagar:" (0,00), "Interés por Mora:" (0,00), "Interés a Pagar:" (0,00), "Total a Pagar:" (0,00), and "Valor a Pagar (*)" (0,0). A blue "Pagar" button with a checkmark icon is located at the bottom center of the form.

Figura 60: Pantalla de pagos de Préstamos
Fuente: Propia

- b) El sistema presenta en pantalla la ventana, con los campos necesarios para realizar la transacción seleccionada, estos campos son los que se muestran en la imagen anterior, los campos marcados con el asterisco de color rojo son obligatorios de ingresar.
- c) El usuarios selecciona el préstamo a pagar y el sistema carga los datos de la cuota a cancelar.
- d) El usuario procede a ingresar la información en cada uno de los campos.
- e) El usuario hace clic en el botón "Pagar".
- f) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- g) Los datos ingresados se registraran en la tabla tab_cartera_tabla_amortizacion la misma que se encuentra estructurada de la siguiente manera

```

tab_cartera_tabla_amortizacion
"codigo_amortizacion" : INT4
"codigo_prestamo" : INT4
"cuota" : INT4
"fecha_pago" : DATE
"capital" : FLOAT8
"interes" : FLOAT8
"dividendo" : FLOAT8
"saldo" : FLOAT8
"estado" : VARCHAR(20)
"capital_pagado" : FLOAT8
"interes_pagado" : FLOAT8
"borrado" : BOOL
...
«PK» "pk_tabla_amortizacion" : codigo_amortizacion
«FK» "fk_codigo_usuario" : codigo_usuario
«FK» "fk_codigo_prestamo" : codigo_prestamo
«FK» "fk_codigo_oficina_amortizacion" : codigo_oficina

```

Figura 61: Diseño de la tabla amortizaciones para pago de préstamos
Fuente: Propia

| Campo | Tipo | Observación |
|------------------|-------------|-------------------------------|
| COD_AMORTIZACION | INTEGER | Clave primaria de la Tabla |
| COD_PRESTAMO | INTEGER | Clave Foránea |
| CUOTA | INTEGER | Nº de cuota cancelada |
| FECHA_PAGO | DATE | Fecha de Pago |
| CAPITAL | FLOAT | Capital a Pagar |
| INTERES | FLOAT | Interés a Pagar |
| DIVIDENDO | FLOAT | Suma C + I |
| SALDO | FLOAT | Cantidad Pendiente |
| ESTADO | VARCHAR(20) | Cambia según su fecha de pago |
| CAPITAL_PAGADO | FLOAT | Cantidad Pagada |
| INTERES_PAGADO | FLOAT | Cantidad Pagada |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 17: Campos de la tabla amortizaciones
Fuente: Propia

3. Flujos Alternativos

- a) El Usuario puede seleccionar cualquiera de los préstamos del cliente.
- b) Si el usuario desea realizar un pago de otro cliente debe seleccionar el botón limpiar para la limpieza de la pantalla.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá al sistema.

- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar esta información.
- d) El cliente debe tener registrado préstamos a su favor.

5. Postcondiciones

- a) Si desea se imprimirá el respectivo Comprobante de Pago.

3.2.7. Descripción del módulo de Ahorros

La Cooperativa de Ahorro y Crédito Unión Cochapamba, de conformidad a lo establecido en el Art. 66 de la ley de Cooperativas, recibirá depósitos de ahorros de sus socios. El concejo de administración debe establecer la normatividad y los incentivos a fin de estimular el ahorro, con el objeto de hacer participar activamente a los socios.

La cooperativa capta ahorros de las siguientes modalidades:

1. Ahorros a plazo fijo.
2. Ahorro a la vista
3. Certificados de Aportación.

La Cooperativa utilizará permanentemente los recursos de ahorro captado, canalizándolos hacia el crédito para los socios, para satisfacer sus necesidades y atender así la actividad principal.

3.2.8. Funcionamiento del módulo de Ahorros

Los usuarios que tienen permisos de acceso al módulo de Ahorros podrán realizar las siguientes transacciones como se muestra a continuación.

La siguiente tabla muestra los ítems o funciones que realizara el sistema para el módulo de Ahorros.

| Paso | Acción |
|------|---|
| 1 | Autenticarse como usuario con acceso al Módulo de Ahorros |
| 2 | Ingresar al menú Ahorros |
| 3 | <p>Seleccionar una de las siguientes Opciones:</p> <ol style="list-style-type: none"> 1. Cuentas de Ahorros 2. Otros Ingresos 3. Realizar Bloqueos y Desbloques 4. Depósitos a Plazo Fijo 5. Realizar Transferencias, Débitos y Créditos 6. Consultas y Reportes 7. Backups y Respaldos |
| 3.1 | <p>Cuentas de Ahorros</p> <ol style="list-style-type: none"> 1) Ver lista de clientes 2) De cada cliente se muestran sus productos registrados 3) De cada producto se muestran sus transacciones |
| 3.2 | <p>Otros Ingresos</p> <ol style="list-style-type: none"> 1) Lista de Otros Ingresos registrados 2) Buscar ingresos realizados |
| 3.3 | <p>Realizar Bloqueos y Desbloques</p> <ol style="list-style-type: none"> 1) Seleccionar la cuenta del cliente o socio 2) Seleccionar el tipo de bloqueo o desbloqueo a realizar 3) Ingresar los datos requeridos por el sistema. 4) Realizar transacción de bloqueo o desbloqueo de saldo |
| 3.4 | <p>Depósitos a Plazo Fijo</p> <ol style="list-style-type: none"> 1) Nuevo DPF <ol style="list-style-type: none"> a) Ingresar los valores a los campos requeridos b) Registrar el DPF c) Imprimir contrato del DPF 2) Cancelar DPF <ol style="list-style-type: none"> a) Si va a renovar ingresar los valores de los campos requeridos b) Registrar el pago y la cancelación del DPF c) Imprimir comprobante de pago. |
| 3.5 | <p>Realizar Transferencias, Débitos y Créditos</p> <ol style="list-style-type: none"> 1) Seleccionar el Tipo de Transferencia a Realizar <ol style="list-style-type: none"> a) Transferencia de Cuenta a Cuenta b) Transferencia de Cuenta de Préstamo. c) Pago de Préstamo por depósito en el Banco d) Nota de crédito por Depósito en el Banco e) Notas de Débito o Crédito 2) Ingresar los datos requeridos por el sistema. 3) Realizar las transacciones 4) Imprimir comprobantes de las Transacciones |

| | |
|------------|---|
| 3.6 | Consultas y Reportes <ol style="list-style-type: none"> 1) Realiza consultas y reporte sobre estados de cuentas de clientes. 2) Consultas sobre otros ingresos realizados en la institución. |
| 3.7 | Backups y Respaldos <ol style="list-style-type: none"> 1) Se saca respaldos de la información generada a la fecha para prevenir pérdidas de información. |

Tabla 18: Funcionalidad de Modulo de Administración y Seguridad
Fuente: Propia

3.2.9. Descripción de los casos de uso del módulo de Ahorros

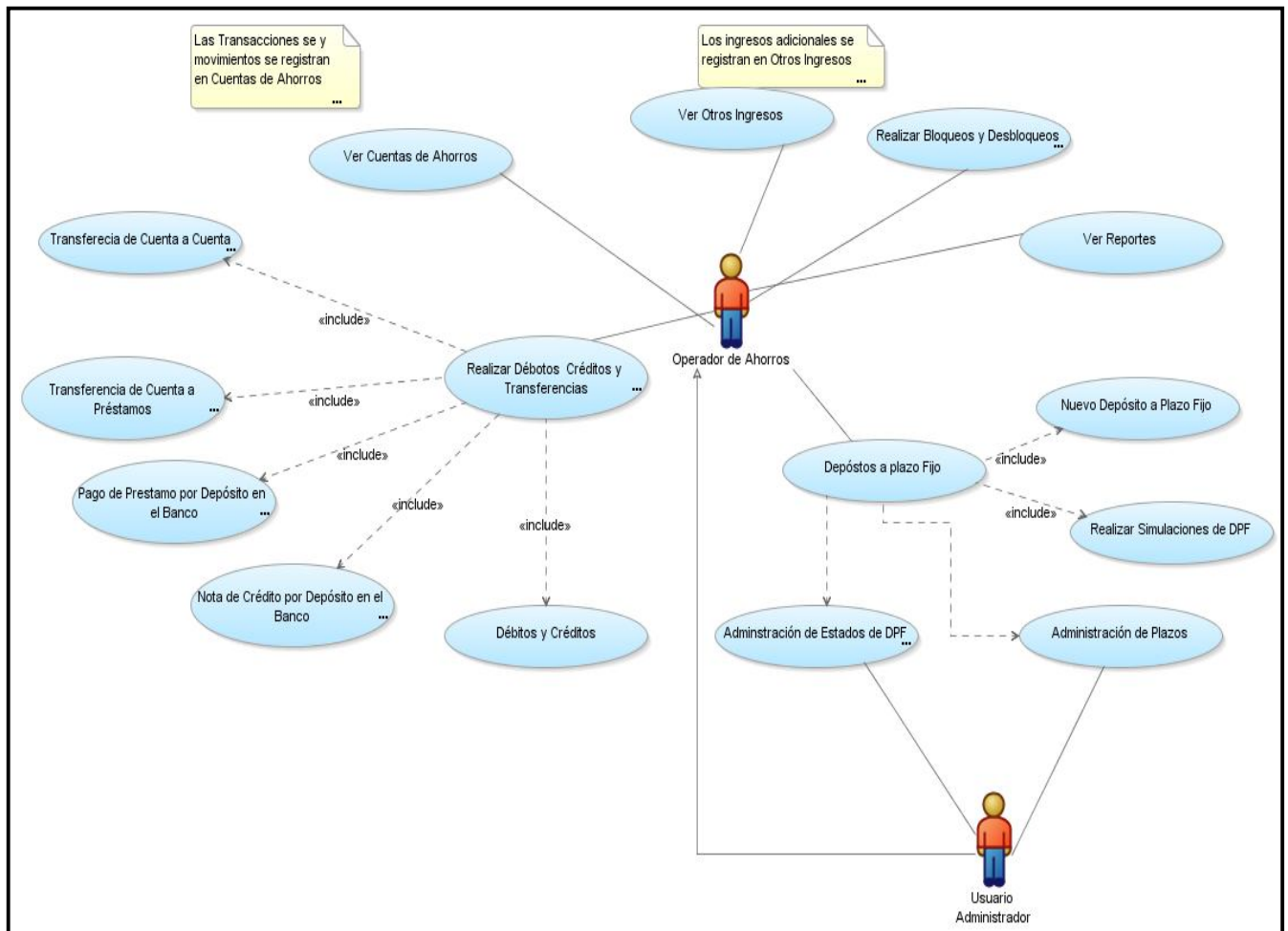


Figura 62: Casos de uso del módulo de ahorros
Fuente: Propia

A continuación se muestra la descripción de los casos de uso más relevantes del módulo de Ahorros.

3.2.9.1. Especificación de Caso de Uso: Cuentas de Ahorro

1. Descripción Breve

Este caso de uso se escribe el proceso que se realiza para poder visualizar la lista de clientes con sus respectivos productos y a su vez cada producto mostrara los movimientos generados al realizar un deposito, retiro, etc.

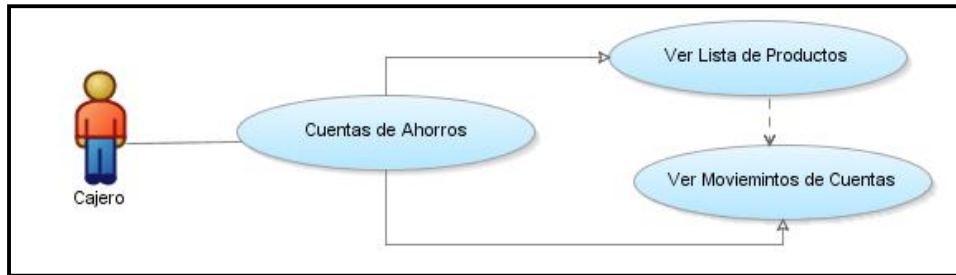


Figura 63: Caso de uso Cuentas de Ahorro
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Cuentas de Ahorro del menú Ahorros usuario podrá visualizar la lista de clientes registrados en el módulo Clientes (Ver casos de uso del módulo). La interfaz gráfica se muestra a continuación.

| Código Cliente | Cedula Cliente | Nombres | Apellidos | Fecha Registro | Productos Socio |
|----------------|----------------|-----------------|-----------------|----------------|-----------------|
| 1029 | 1002954988 | EDGAR DANIEL | PICUASI DUQUE | 2012-01-23 | |
| 1028 | 1002299566 | BAIRON PATRICIO | GUDINO VILLALBA | 2012-01-22 | |
| 1027 | 1000953669 | MARIA NIEVES | JUMA URCUANGO | 2005-02-18 | |
| 1026 | 1000882694 | JOSE MANUEL | JUMA URCUANGO | 2005-02-18 | |

Figura 64: Pantalla cuentas de ahorros
Fuente: Propia

- b) Para poder visualizar los productos de cada cliente seleccionamos la opción Productos Socio de la lista de clientes y a continuación se mostrara los productos de dicho cliente, como se muestra a continuación.

| Código Cliente | Cedula Cliente | Nombres | Apellidos | Fecha Registro | Productos Socio |
|----------------|----------------|--------------|--------------|----------------|-----------------|
| 740 | 1002469128 | LUIS ALFREDO | COLCHA CAIZA | 2011-11-21 | |

| N° Cuenta | Producto | Estado Cuenta | Saldo Total | Disponibles | Encajes | Bloqueados | Cheques | Interes | Movimientos |
|-----------|----------------------------|---------------|--------------|-------------|-----------|------------|---------|---------|-------------|
| 102001028 | Certificados de Aportacion | Activa | 20,00 \$ | 20,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | |
| 101001028 | Cuenta de Ahorros | Activa | 10.000,00 \$ | 9.500,00 \$ | 500,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | |

Figura 65: Pantalla de productos activos del Cliente
Fuente: Propia

- c) Al tener la lista de Productos de cada cliente podemos ver la lista de transacciones de cada producto y esto se lo realiza seleccionando la opción de Movimientos de la lista de productos, a continuación se muestra la interfaz gráfica de movimientos pertenecientes a las transacciones realizadas en el módulo de Caja (Ver casos de uso módulo Caja).

| N° Movimiento | N° de Cuenta | Producto | Tipo Transacción | Fecha Transacción | Valor Transacción | Valor Efectivo | Valor Cheque | Valor Actual | Operador |
|---------------|--------------|-------------------|----------------------------|-------------------|-------------------|----------------|--------------|--------------|----------|
| 5 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 10.000,00 \$ | baliron |
| 4 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 9.000,00 \$ | baliron |
| 3 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 8.000,00 \$ | baliron |
| 2 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 2.000,00 \$ | 2.000,00 \$ | 0,00 \$ | 7.000,00 \$ | baliron |
| 1 | 101001028 | Cuenta de Ahorros | APERTURA DE CUENTA | 2012-01-22 | 5.000,00 \$ | 5.000,00 \$ | 0,00 \$ | 5.000,00 \$ | baliron |

Figura 66: Pantalla de lista de movimientos de una cuenta de ahorros
Fuente: Propia

- d) El registro de esta información se la realiza utilizando las siguientes tablas: tab_clientes_cliente, tab_ahorros_productos_socio y tab_ahorros_movimientos las mismas que se encuentran estructuradas como se muestra a continuación:



Figura 67: Diseño de las tablas para registro de cuentas de ahorros
Fuente: Propia

3. Flujos Alternativos

- El usuario podrá realizar búsquedas de los clientes por cedula, apellidos y fecha de registro para agilizar las búsquedas de información.
- En los movimientos se puede realizar búsqueda personalizada para poder ver las transacciones de fechas anteriores.

4. Precondiciones

- Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- Debe cumplir con el proceso de autenticación de usuario para acceder al menú Cuentas de Ahorro.
- Debe tener permisos para visualizar esta información.

5. Postcondiciones

N/A

3.2.9.2. Especificación de Caso de Uso: Débitos y Créditos

1. Descripción Breve

Este caso de uso describe el proceso para realizar un Débito o un Crédito a una cuenta de Ahorros por los conceptos especificados en el caso de uso 007 UC Administrar Parámetros Generales del módulo de administración y seguridad.

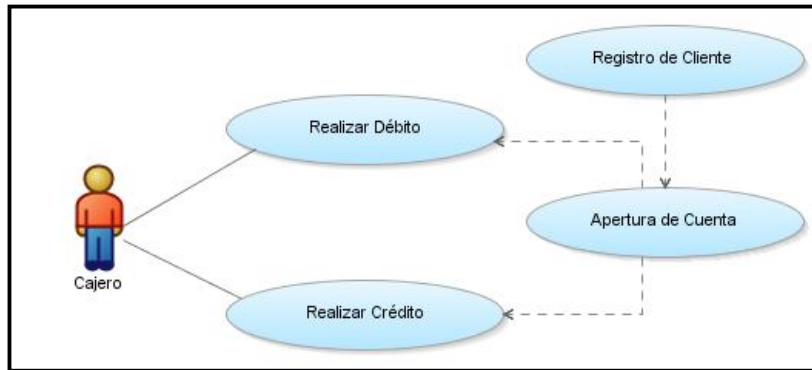


Figura 68: Cado de uso realizar débitos y créditos
Fuente: Propia

2. Flujo Básico de Eventos

- Seleccionamos Débitos/Créditos/Transf del menú Ahorros, el usuario podrá visualizar la interfaz gráfica especificada en el caso de uso 004 UC Débitos, Créditos y Transferencias del módulo de Ahorros.
- Seleccionamos Débitos/Créditos del menú circular o del menú lateral y nos mostrará la interfaz con la lista de transacciones realizadas como se muestra en la siguiente imagen.

La interfaz muestra una barra superior con botones: Guardar, Crédito, Débito, Notas DIC, Limpiar y Buscar. Debajo, un encabezado indica 'Lista de Notas de Débito y Crédito'. El contenido principal es una tabla con las siguientes columnas: Código, Causa Débito/Crédito, Tipo Causal, Cuenta Ahorros, Cliente, Valor y Fecha.

| Código | Causa Débito/Crédito | Tipo Causal | Cuenta Ahorros | Cliente | Valor | Fecha |
|--------|-----------------------|-------------|----------------|---------------------------------|--------|------------|
| 1 | Pago de Interes | credito | 101001028 | GUDINO VILLALBA BAIRON PATRICIO | 500.0 | 2012-01-26 |
| 2 | Compra de Formularios | debito | 101001028 | GUDINO VILLALBA BAIRON PATRICIO | 1000.0 | 2012-01-26 |

Debajo de la tabla hay un control de paginación con botones de navegación.

Figura 69: Pantalla para realizar Débitos y Créditos
Fuente: Propia

- c) El usuario debe seleccionar del menú horizontal la opción que desea realizar como guardar, realizar débito, realizar crédito o ver la lista de transacciones.
- d) El sistema presenta en pantalla la ventana, con los campos necesarios para realizar la transacción seleccionada, estos campos son los que se muestran en la siguiente imagen:

Figura 70: Pantalla de ingreso de información para acreditación
Fuente: Propia

- e) El usuario procede a ingresar la información en cada uno de los campos.
- f) El usuario hace clic en el botón “guardar”.
- g) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- h) Los datos ingresados se registrarán en la tabla `tab_ahorros_debitos_creditos` la misma que se encuentra estructurada de la siguiente manera

```

tab_ahorros_debitos_creditos
"cod_deb_cred" : INT4
"cod_causa" : INT4
"nro_cuenta" : INT4
"valor" : FLOAT8
"fecha" : DATE
"borrado" : BOOL
"cod_producto" : INT4

«PK» "pk_tab_ahorros_debitos_credito" : cod_deb_cred
«FK» "fk_tab_ahor_ref4" : cod_causa
«FK» "fk_tab_ahor_prod" : nro_cuenta, cod_producto

```

Figura 71: Diseño de la tabla débitos y créditos
Fuente: Propia

| Campo | Tipo | Observación |
|--------------|-------------|----------------------------------|
| COD_DEB_CRED | INTEGER | Clave primaria de la Tabla |
| COD_CAUSA | INTEGER | Clave Foránea |
| NRO_CUENTA | INTEGER | Clave Foránea |
| COD_PRODUCTO | INTEGER | Clave Foránea |
| VALOR | FLOAT | Registros de Cantidad |
| FECHA | DATE | Registra la fecha de transacción |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 19: Campos de la tabla débitos y créditos

Fuente: Propia

3. Flujos Alternativos

- a) El usuario podrá realizar búsquedas a la tabla y ver transacciones realizadas en fechas anteriores dependiendo los parámetros de búsqueda.
- b) El Usuario podrá seleccionar cualquiera de las opciones que se encuentran visibles en el menú horizontal.
- c) Si no desea realizar ninguna transacción puede seleccionar del menú la opción regresar.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar esta información.

5. Postcondiciones

- a) Puede visualizar su transacción utilizando la opción de lista en el menú horizontal.
- b) Si desea realizar otra transacción diferente puede hacerlo utilizando el botón regresar y visualizara nuevamente el menú principal.

3.2.9.3. Especificación de Caso de Uso: Notas de Crédito por Depósito en Bancos

1. Descripción Breve

Este caso de uso describe el proceso para realizar Notas de Crédito por Depósito en el Banco a una determinada cuenta de Ahorros.

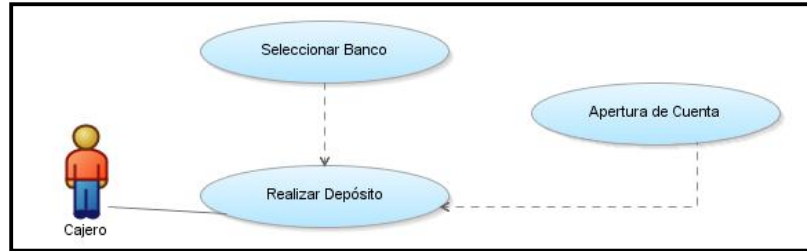


Figura 72: Caso de uso nota de crédito por depósito en bancos
Fuente: Propia

2. Flujo Básico de Eventos

- Seleccionamos Débitos/Créditos/Transf del menú Ahorros, el usuario podrá visualizar la interfaz gráfica especificada en el caso de uso 004 UC Débitos, Créditos y Transferencias del módulo de Ahorros.
- Seleccionamos Notas de Crédito por Depósito en el Banco del menú circular o del menú lateral y nos mostrará la interfaz para ingresar los datos del depósito como se muestra en la siguiente imagen.

Regresar Guardar Depósito Listado Limpiar

NOTA DE CRÉDITO POR DEPÓSITO EN EL BANCO

Cta Ahorros: (*)

Número de Cliente:

Producto:

Ahorro disponible:

Número Depósito: (*)

Valor Depósito: (*)

Fecha del Depósito:

Nombre del Banco: (*)

Figura 73: Pantalla de crédito por depósitos en el banco
Fuente: Propia

- c) El sistema presenta en pantalla la ventana, con los campos necesarios para realizar la transacción seleccionada, estos campos son los que se muestran en la imagen anterior.
- d) El usuario procede a ingresar la información en cada uno de los campos.
- e) El usuario hace clic en el botón “guardar”.
- f) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- g) Los datos ingresados se registraran en la tabla tab_ahorros_depositos_banco la misma que se encuentra estructurada de la siguiente manera.

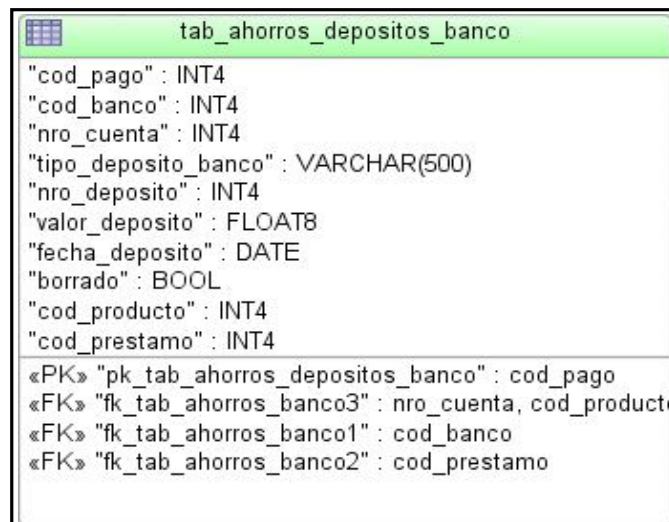


Figura 74: Diseño de la tabla depósitos en bancos
Fuente: Propia

| Campo | Tipo | Observación |
|---------------------|--------------|------------------------------|
| COD_PAGO | INTEGER | Clave primaria de la Tabla |
| COD_BANCO | INTEGER | Clave Foránea |
| NRO_CUENTA | INTEGER | Clave Foránea |
| COD_PRODUCTO | INTEGER | Clave Foránea |
| COD_PRESTAMO | INTEGER | Clave Foránea |
| TIPO_DEPOSITO_BANCO | VARCHAR(500) | Describir el Depósito |
| NRO_DEPOSITO | INTEGER | Nº Depósito (Comprobante) |
| VALOR_DEPOSITO | FLOAT | Cantidad Depositada |
| FECHA_DEPOSITO | DATE | Fecha se realizó el Depósito |
| BORRADO | BOOLEAN | Campo para borrar registro |

Figura 75: Campos de la tabla depósitos en el banco
Fuente: Propia

3. Flujos Alternativos

- a) El Usuario podrá seleccionar cualquiera de las opciones que se encuentran visibles en el menú horizontal.
- b) Si no desea realizar ninguna transacción puede seleccionar del menú la opción regresar.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar esta información.

5. Postcondiciones

- a) Puede visualizar su transacción utilizando la opción de lista en el menú horizontal.
- b) Si desea realizar otra transacción diferente puede hacerlo utilizando el botón regresar y visualizara nuevamente el menú principal.

3.2.9.4. Especificación de Caso de Uso: Transferencia de Cuenta a Préstamos

1. Descripción Breve

Este caso de uso describe el proceso para realizar Transferencias de Cuenta de ahorros a préstamos seleccionando la cuenta del cliente y préstamo a cancelar.

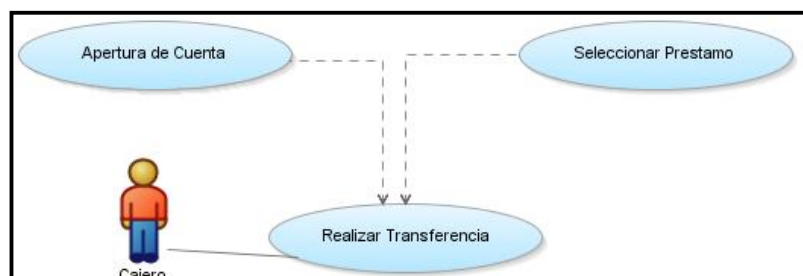


Figura 76: Caso de uso Transferencias de ahorros a préstamos
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Débitos/Créditos/Transf del menú Ahorros, el usuario podrá visualizar la interfaz gráfica especificada en el caso de uso 004 UC Débitos, Créditos y Transferencias del módulo de Ahorros.
- b) Seleccionamos Transferencia de Cuenta a Préstamos del menú circular o del menú lateral y nos mostrará la interfaz para ingresar los datos del depósito como se muestra en la siguiente imagen.

Regresar Nueva Listado Limpiar

TRANSFERENCIAS DE CUENTAS DE AHORROS A PRESTAMOS

Cuenta Origen: (*) 101001028

GUDINO VILLALBA BAIRON PATRICIO

Número de Cliente: 1028

Producto: Cuenta de Ahorros

Ahorro disponible: 10000.0

Prestamo Destino: (*) 1

Nº de Socio: 0

Nombres:

Valor a Pagar: 0,00

| Prestamo | Nº de Cuota | Capital | Interes | Mora | Cancelar |
|----------|-------------|---------|---------|------|----------|
| 0 | | 0,00 | 0,00 | 0,00 | 0,00 |

Ingrese Valor a Pagar

Valor Transferencia: (*) 0.0

Realizar Transferencia

Figura 77: Pantalla de transferencia de cuentas de ahorros a préstamos
Fuente: Propia

- a) El sistema presenta en pantalla la ventana, con los campos necesarios para realizar la transacción seleccionada, estos campos son los que se muestran en imagen anterior, los campos marcados con el asterisco de color rojo son obligatorios.
- b) El usuario procede a ingresar la información en cada uno de los campos.
- c) Al seleccionar el préstamo a cancelar el sistema mostrará automáticamente la información sobre la cantidad que el usuario debe cancelar.
- d) El usuario hace clic en el botón “Realizar Transferencia”.

- e) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
- f) Los datos ingresados se registraran en la tabla tab_ahorros_transferencias la misma que se encuentra estructurada de la siguiente manera.



Figura 78: Diseño de la tabla transferencias
Fuente: Propia

| Campo | Tipo | Observación |
|----------------------|---------|----------------------------|
| COD_TRANSFERENCIA | INTEGER | Clave primaria de la Tabla |
| TIPO_TRANSFERENCIA | INTEGER | Clave Foránea |
| FECHA_TRANSFERENCIA | DATE | Fecha de la Transferencia |
| NRO_CUENTA_ORIGEN | INTEGER | Clave Foránea |
| COD_PRODUCTO | INTEGER | Clave Foránea |
| VALOR_TRANSFERENCIA | FLOAT | Cantidad a Transferir |
| NRO_CUENTA_DESTINO | INTEGER | Clave Foránea |
| COD_PRODUCTO_DESTINO | INTEGER | Clave Foránea |
| NRO_PRESTAMO_DESTINO | INTEGER | Clave Foránea |
| BORRADO | BOOLEAN | Campo para borrar registro |

Tabla 20: Campos de la tabla transferencias
Fuente: Propia

3. Flujos Alternativos

- a) El Usuario podrá seleccionar cualquiera de las opciones que se encuentran visibles en el menú horizontal.
- b) Si no desea realizar ninguna transacción puede seleccionar del menú la opción regresar.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.

5. Postcondiciones

- a) Puede visualizar su transacción utilizando la opción de listado en el menú horizontal.
- b) Si desea realizar otra transacción diferente puede hacerlo utilizando el botón regresar y visualizara nuevamente el menú principal.
- c) Si desea se imprimirá el respectivo Comprobante de Pago.

3.2.9.5. Especificación de Caso de Uso: Transferencia de Cuenta a Cuenta

1. Descripción Breve

Este caso de uso describe el proceso para realizar Transferencia de Cuenta a Cuenta seleccionando la cuenta origen y la cuenta destino.

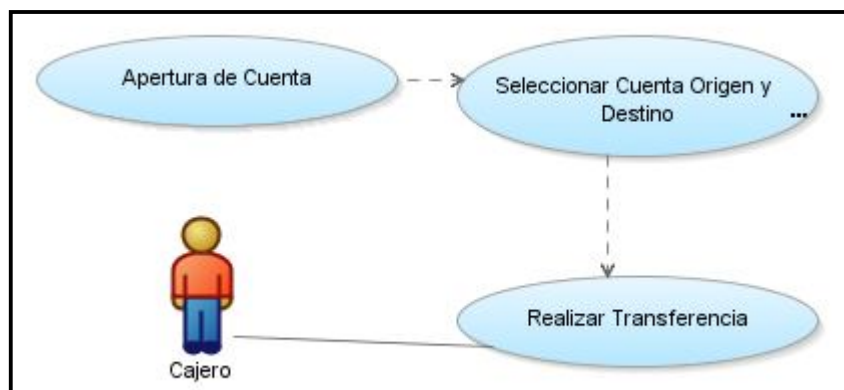


Figura 79: Caso de uso transferencia de ahorros a ahorros
Fuente: Propia

2. Flujo Básico de Eventos

- a) Seleccionamos Débitos/Créditos/Transf del menú Ahorros, el usuario podrá visualizar la interfaz gráfica especificada en el caso de uso 004 UC Débitos, Créditos y Transferencias del módulo de Ahorros.
- b) Seleccionamos Transferencia de Cuenta a Cuenta del menú circular o del menú lateral y nos mostrará la interfaz para ingresar los datos del depósito como se muestra en la siguiente imagen.

Regresar Guardar Nueva Listado Limpiar

TRANSFERENCIAS DE CUENTA A CUENTA DE AHORROS

Cuenta Origen: (*) 101001028

BAIRON PATRICIO GUDINO VILLALBA

Número de Cliente: 1028

Producto: Cuenta de Ahorros

Ahorro disponible: 10000,00

Cuenta Destino: (*)

Número de Cliente: 0

Producto:

Ahorro: 0,00

Valor a Transferir: (*) 0,0

Figura 80: Pantalla de transferencias de ahorros a ahorros
Fuente: Propia

- c) El sistema presenta en pantalla la ventana, con los campos necesarios para realizar la transacción seleccionada, estos campos son los que se muestran en imagen anterior, los campos marcados con el asterisco de color rojo son obligatorios de ingresar.
- d) El usuario procede a ingresar la información en cada uno de los campos.
- e) El usuario selecciona la cuenta de ahorro origen y el sistema carga automáticamente la información de la cuenta.
- f) El usuario selecciona la cuenta de ahorro destino y el sistema carga automáticamente la información de la cuenta.
- g) El usuario hace clic en el botón “Guardar”.
- h) El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.

3. Flujos Alternativos

- a) El Usuario podrá seleccionar cualquiera de las opciones que se encuentran visibles en el menú horizontal.
- b) Si no desea realizar ninguna transacción puede seleccionar la opción regresar.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.

5. Postcondiciones

- a) Puede visualizar su transacción utilizando la opción de listado en el menú horizontal.
- b) Si desea realizar otra transacción diferente puede hacerlo utilizando el botón regresar y visualizara nuevamente el menú principal.
- c) Si desea se imprimirá el respectivo Comprobante de Pago.

3.2.9.6. Especificación de Caso de Uso: Depósitos a Plazo Fijo

1. Descripción Breve

Este caso de uso se escribe el proceso que se realiza para poder visualizar la lista de clientes con sus respectivos Depósitos a Plazo Fijo y detalles del depósito.

2. Flujo Básico de Eventos

- a) Seleccionamos Plazos Fijos del menú Ahorros y Depósitos Plazo Fijo como submenú de Plazos Fijos.

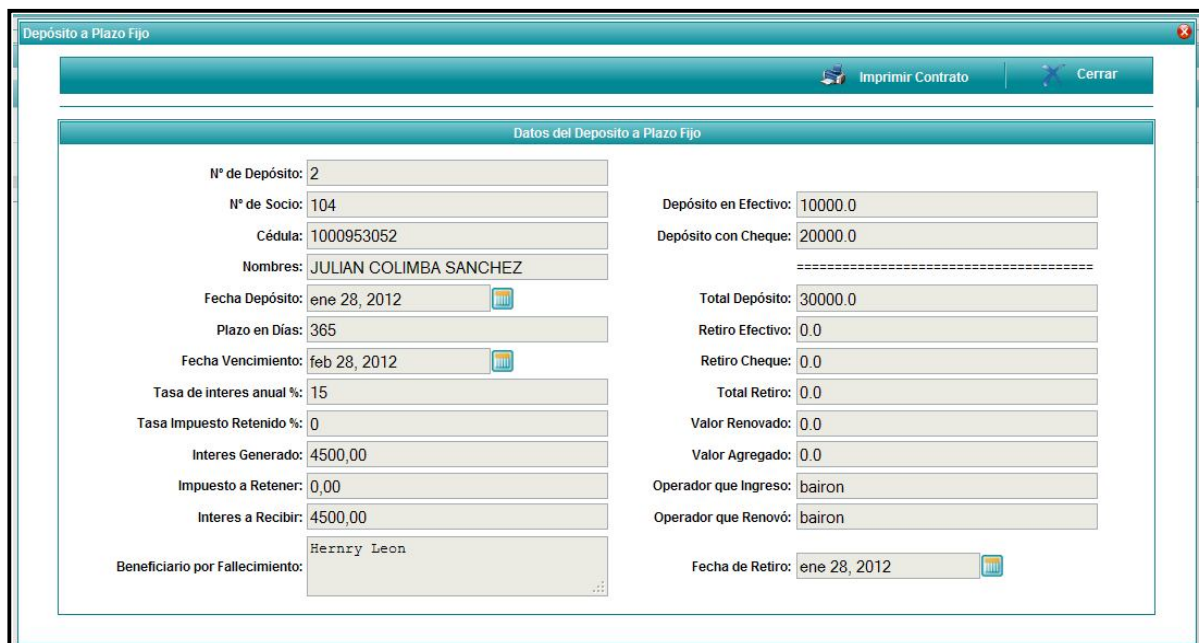


The screenshot shows a software interface for 'Depósitos a Plazo Fijo'. At the top right, there is a button labeled 'Nuevo DPF'. Below it is a table with the following data:

| Pagar DPF | Nombre Socio | Nro Depósito | Fecha de Depósito | Fecha de Vencimiento | Valor | Estado | Ver Datos |
|-----------|------------------------|--------------|-------------------|----------------------|----------|---------|-----------|
| | JULIAN COLIMBA SANCHEZ | 2 | 2012-01-28 | 2012-02-28 | 34500.00 | Vigente | |
| | MARISOL SIMBAÑA IRALDA | 1 | 2012-01-28 | 2012-02-28 | 7700.00 | Vigente | |

Figura 81: Pantalla de depósitos a plazo fijo
Fuente: Propia

- b) El usuario puede ingresar un nuevo Depósito a Plazo Fijo seleccionando el botón nuevo y se sigue las instrucciones que se indica en el caso de uso 0011 UC Nuevo Depósito a Plazo Fijo.
- c) Si el DPF cumplió con el plazo fijado al realizar el depósito el usuario puede seleccionar la opción Pagar DPF de la lista que se muestra.
- d) Para cancelar un DPF el usuario debe ingresar los datos requeridos como la forma de pago y los campos obligatorios.



The screenshot shows a detailed form for 'Depósito a Plazo Fijo'. At the top right, there are buttons for 'Imprimir Contrato' and 'Cerrar'. The form is titled 'Datos del Depósito a Plazo Fijo' and contains the following fields:

| | | | |
|---------------------------------|------------------------|-----------------------|--------------|
| Nº de Depósito: | 2 | Depósito en Efectivo: | 10000.0 |
| Nº de Socio: | 104 | Depósito con Cheque: | 20000.0 |
| Cédula: | 1000953052 | ===== | |
| Nombres: | JULIAN COLIMBA SANCHEZ | Total Depósito: | 30000.0 |
| Fecha Depósito: | ene 28, 2012 | Retiro Efectivo: | 0.0 |
| Plazo en Días: | 365 | Retiro Cheque: | 0.0 |
| Fecha Vencimiento: | feb 28, 2012 | Total Retiro: | 0.0 |
| Tasa de interes anual %: | 15 | Valor Renovado: | 0.0 |
| Tasa Impuesto Retenido %: | 0 | Valor Agregado: | 0.0 |
| Interes Generado: | 4500,00 | Operador que Ingreso: | bairon |
| Impuesto a Retener: | 0,00 | Operador que Renovó: | bairon |
| Interes a Recibir: | 4500,00 | Fecha de Retiro: | ene 28, 2012 |
| Beneficiario por Fallecimiento: | Henry Leon | | |

Figura 82: Pantalla de registros de depósitos a plazo fijo
Fuente: Propia

- e) Para poder visualizar la lista de Depósitos a Plazo Fijo se realiza una consulta a la tabla `tab_ahorros_depositospf` la cual se encuentra estructurada de la siguiente forma:

| CAMPO | TIPO |
|-------------------|---------|
| cod_deposito | INTEGER |
| cod_estado | INTEGER |
| cod_plazo | INTEGER |
| cod_cliente | INTEGER |
| operador_ingreso | INTEGER |
| operador_renovo | INTEGER |
| fecha_deposito | DATE |
| oficina | INTEGER |
| plazo_dias | INTEGER |
| fecha_vencimiento | DATE |
| total_deposito | FLOAT |
| deposito_efectivo | FLOAT |
| deposito_cheque | FLOAT |
| retiro_efectivo | FLOAT |
| retiro_cheque | FLOAT |
| valor_renovar | FLOAT |
| valor_agregar | FLOAT |
| fecha_retiro | DATE |
| Interés_anual | INTEGER |
| impuesto | INTEGER |
| beneficiario | TEXT |
| borrado | BOOLEAN |

| CAMPO | TIPO |
|------------------------------|------|
| "cod_deposito" : INT4 | |
| "cod_estado_dpf" : INT4 | |
| "cod_plazo" : INT4 | |
| "operador_renovo" : INT4 | |
| "cod_cliente" : INT4 | |
| "operador_ingreso" : INT4 | |
| "fecha_deposito" : DATE | |
| "plazo_dias" : INT4 | |
| "fecha_vencimiento" : DATE | |
| "total_deposito" : FLOAT8 | |
| "deposito_efectivo" : FLOAT8 | |
| "deposito_cheque" : FLOAT8 | |
| "retiro_efectivo" : FLOAT8 | |
| "retiro_cheque" : FLOAT8 | |
| "valor_renovar" : FLOAT8 | |
| "valor_agregar" : FLOAT8 | |
| "fecha_retiro" : DATE | |
| "interes_anual" : INT4 | |
| "impuesto" : INT4 | |
| "beneficiario" : TEXT | |
| "borrado" : BOOL | |
| "oficina" : INT4 | |

| | |
|--|--|
| «PK» "tab_ahorros_depositospf_pkey" : cod_deposito | |
| «FK» "fk_tab_ah_dp3" : cod_estado_dpf | |
| «FK» "fk_tab_ah_dp2" : cod_plazo | |
| «FK» "fk_tab_ah_dp5" : operador_renovo | |
| «FK» "fk_tab_ah_dp4" : operador_ingreso | |
| «FK» "fk_tab_ah_dp1" : cod_cliente | |
| «FK» "fk_tab_ah_dp6" : oficina | |

Figura 83: Diseño y campos de la tabla depósitos a plazo fijo
Fuente: Propio

3. Flujos Alternativos

- El usuario podrá seleccionar cualquier opción de las que se muestra en la interfaz como Pagar DPF, Ver Datos o Nuevo DPF.
- Si selecciona Ver Datos el usuario podrá volver a imprimir el contrato realizado al realizar el depósito.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar esta información.
- d) Si selecciona la opción Pagar DPF primero debe cumplirse el plazo fijado caso contrario mostrara un mensaje indicando que el plazo aún no se ha cumplido.

5. Postcondiciones

- a) Cuando Depósito a plazo fijo ya ha sido pagado ya este se marcara como cancelado y ya no se podrá volver a cancelar solo se podrá visualizar la información generada cuando se realizó el DPF.

3.2.9.7. Especificación de Caso de Uso: Consultas y Reportes

1. Descripción Breve

Este caso de uso describe el proceso que se ejecuta para realizar reportes acerca registros almacenados en el módulo de ahorros. Este proceso se ejecutará si el usuario cajero o administrador desea generar estas consultas.

2. Flujo Básico de Eventos

- a) El usuario cajero puede generar estos reportes dependiendo del requerimiento necesario, es decir realizar 4 tipos de consultas como estados de cuenta, saldos de ahorros, transacciones de ahorros y saldos de otros ingresos.
- b) Seleccionar Reportes del menú Ahorros y seleccionar la opción Estados de Cuenta de Reportes y se mostrara la siguiente interfaz para seleccionar e ingresar los datos de la cuenta a ver el estado.

- c) Seleccionar Reportes del menú Ahorros y seleccionar la opción Saldos de Ahorros de Reportes y se mostrara la siguiente interfaz.



Figura 84: Pantalla para reportes de saldos de ahorros
Fuente: Propia

- d) Seleccionar Reportes del menú Ahorros y seleccionar la opción Transacciones de Ahorros de Reportes y se mostrara la siguiente interfaz.



Figura 85: Pantalla para reportes de transacciones de ahorros
Fuente: Propia

- e) Seleccionar Reportes del menú Ahorros y seleccionar la opción Saldos Otros Ingreso de Reportes y se mostrara la siguiente interfaz.



Figura 86: Pantalla para reportes de saldos totales de Ahorros
Fuente: Propia

3. Flujos Alternativos

- a) Seleccionar las fechas como crea necesario para su informe.

4. Precondiciones

- a) Para poder acceder al sistema la PC cliente debe estar comunicada en red con el Servidor y tener instalado un navegador web con el cual accederá a la interfaz gráfica del Sistema.
- b) El usuario debe estar registrado en el Sistema, el registro se lo realiza por el administrador del sistema.
- c) Debe tener permisos para visualizar e ingresar esta información.

5. Postcondiciones

N/A

Sistema Integrado de Información Financiera FINANSYS



Diseño y Desarrollo del Aplicativo

- 1) Introducción
- 2) Fase de Incepción
 - 2.1 Documento de Visión
 - 2.2 Plan de Desarrollo de Software
 - 2.3 Actas de Trabajo
- 3) Fase de Elaboración
 - 3.1 Arquitectura
 - 3.2 Especificación de Casos de Uso
- 4) Fase de Construcción
 - 4.1 Modelo de Datos
 - 4.2 Plan de Pruebas
 - 4.3 Lista de Riesgos
- 5) Fase de Transición
 - 5.1 Manual de Instalación

Capítulo 4: DISEÑO Y DESARROLLO DEL APLICATIVO

4. DISEÑO Y DESARROLLO DEL APLICATIVO

4.1. Introducción

Para el diseño y desarrollo del aplicativo Módulos de Administración y Seguridad, Ahorros y Captaciones para la Cooperativa de Ahorro y Crédito Unión Cochapamba se ha utilizado la metodología de desarrollo RUP (Rational Unified Process).

El proceso iterativo de RUP se organiza y separa sus actividades en fases, las cuales muestran el desarrollo correcto y ordenado de una aplicación.⁷

A continuación se describen las cuatro fases:

- I. Incepción.-** Durante la fase inicial se concibe la idea central del producto, se arma el documento de visión, se revisan y confirma nuestro entendimiento sobre los objetivos centrales del negocio.
Debemos entender los argumentos en favor de porqué el proyecto debe intentarse. La fase de incepción establece la viabilidad del producto y delimita el alcance del proyecto.
- II. Elaboración.-** Durante la fase de elaboración la mayoría de los Casos de Uso son especificados en detalle y la arquitectura del sistema es diseñada.
Se identifican los riesgos significativos y se preparan el calendario, el equipo de trabajo y el costo del proyecto.
- III. Construcción.-** Desarrollar el producto y evolucionar la visión, la arquitectura y los planes hasta que el producto en una primera versión esté listo para ser enviado a la comunidad de usuarios es que el foco del producto se mueve de la arquitectura de base a un sistema lo suficientemente completo como para llevarlo al usuario. Se realiza el diseño para llevarlo a código fuente.

⁷The Rational Unified Process, An Introduction.” de Philipp Kruchten.

IV. Transición.-En la fase de transición el objetivo es garantizar que los requisitos se han cumplido, con la satisfacción de las partes interesadas. Esta fase a menudo se inicia con una versión beta de la aplicación. Otras actividades incluyen la preparación del ambiente, se completan, se identifican y corrigen defectos. Esta fase termina con la capacitación o sociabilización del producto.

4.2. Fase de Incepción

En esta fase se describe principalmente el alcance del proyecto a través del documento de Visión, se determino los requerimientos a través de actas de trabajo y se desarrollo un plan de desarrollo de software.

4.2.1. Documento de Visión

4.2.1.1. Introducción

4.2.1.1.1. Propósito

El propósito de este documento es definir a alto nivel los requerimientos del Sistema Integrado de Información Financiera - Cooperativa Unión Cochapamba (FINANSYS).

El sistema FINANSYS se encargará de realizar todo el proceso financiero de la cooperativa Unión Cochapamba de manera detallada y automática.El detalle de cómo el sistema FINANSYS cubrirá las necesidades de los usuarios se especifica en los Casos de Uso, que son información adicional no especificada en este documento.

4.2.1.1.2. Alcance

Este documento de visión se aplica al Sistema Integrado de Información Financiera - Cooperativa Unión Cochapamba que será implementado por estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas - Universidad Técnica del Norte.

4.2.1.1.3. Definiciones, Siglas y Abreviaturas

Ver Glosario.

4.2.1.1.4. Referencias

- a) Glosario
- b) Acta de trabajo No. 01
- c) Resumen del Modelo de Casos de Uso

4.2.1.2. Posicionamiento

4.2.1.2.1 Oportunidad de negocio

A partir de los procedimientos ya establecidos en la cooperativa, y como parte de la renovación y automatización establecido por los directivos de la Cooperativa, se determina la creación del sistema FINANSYS que permita mejorar toda la gestión de las actividades relacionadas al manejo de la información financiera generada en la entidad.

El manejo de la información financiera generada por la Cooperativa en la actualidad se realiza en base a un sistema implementado en lenguaje C, el cual presenta limitaciones, porque no controla y administra todos los procesos de forma unificada empleando mucho tiempo en procesar la información, esto provoca realizar procesos manuales que dificultan la agilidad con la que se debería trabajar una institución como esta, dichos procesos manuales se sustentan en la elaboración de documentos físicos y digitales que impiden un flujo de información adecuado para la Cooperativa.

Luego de un análisis realizado por los desarrolladores, se pudo establecer que el sistema que actualmente cubre las necesidades de la Cooperativa presenta algunas falencias o dificultades:

1. No existe un gestor de base de datos

El sistema actual usa archivos visibles al usuario para el manejo de la información lo que hace que este sistema sea por demás vulnerable y susceptible a cualquier tipo de error.

2. El sistema no es parametrizable

Se realizan modificaciones continuas al programa fuente lo cual provocando duplicidad de esfuerzos y un gasto.

3. Tecnología desactualizada

La tecnología de la aplicación que se mantiene en funcionamiento es desactualizada, pues ha sido implementada en una herramienta que discontinuó su uso.

4. Falta de control y auditoríaEl sistema informático actual, carece de controles y pistas de auditoría que permitan validar la información y establecer la eficiencia y eficacia con que se está operando.

5. Seguridad deficiente

El sistema actual tiene falencias de seguridad importantes que comprometen la confidencialidad de la información.

Con referencia a lo expuesto anteriormente, la directiva general de la cooperativa ha planteado rediseñar la aplicación informática existente para permitir trabajar de una manera más eficiente e integrada en un único sistema financiero.El nuevo sistema FINANSYS entonces brindará las facilidades necesarias para el manejo de la información financiera generada por la cooperativa.Por otro lado, rediseñar e implementar un nuevo sistema, obliga a que se cree en la nueva aplicación la mayoría de los del sistema antiguo, como también los nuevos módulos que se desarrollen con el fin de obtener una aplicación eficiente, estos módulos son:

- a) Módulo de Administración
- b) Módulo Clientes.
- c) Módulo de Contabilidad.
- d) Módulo de Préstamos y Cartera.
- e) Módulo de Ahorros.
- f) Módulo de Caja.
- g) Módulo de Auditoría.

4.2.1.2.2 Definición del problema

| | |
|-----------------------|--|
| El problema de | Redundancia de procesos en el desarrollo de actividades diarias. No poseer un sistema integrado que administre todos los procesos de gestión de la información financiera La aplicación actual carece de seguridad. Aplicación construida con tecnología que ya no tiene soporte. |
|-----------------------|--|

| | |
|-------------------------------------|--|
| Que afecta a | Todos los usuarios de la Cooperativa involucrados con la gestión de la información financiera. Personal que tiene dificultad para mantener el sistema en producción. |
| El impacto de ello es | Falta de información consistente de la gestión financiera. Existen muchas actividades y procesos manuales que no permiten una gestión eficiente. Inconsistencia en la información generada. |
| Una solución exitosa debería | Solucionar los requerimientos internos de los involucrados en el proceso de la gestión financiera. Cubrir las necesidades de integración Automatizar algunos de los procesos que hasta ahora son realizados de forma manual. Brindar información confiable sobre el manejo financiero de la cooperativa de manera fácil y oportuna. Implementar una solución informática de calidad soportada por una metodología eficiente de desarrollo de software. |

Tabla 21: Definición del Problema
Fuente: Metodología RUP

4.2.1.3. Descripción de los interesados y usuarios

Las siguientes tablas indican la responsabilidad de interesados y usuarios.

4.2.1.3.1 Resumen de los interesados

Interesados son todas aquellas personas directamente involucradas en la definición y alcance del proyecto. A continuación se presenta la lista de los interesados:

| Nombre | Descripción | Responsabilidad |
|--------------------------|--|--|
| Coordinador del proyecto | Responsable del proyecto a nivel directivo de la Cooperativa de Ahorros y Crédito Unión Cochapamba. | Establecer los lineamientos generales para el desarrollo del proyecto. Coordinar a nivel directivo los diferentes requerimientos que surjan en el desarrollo del sistema. Proveer de los recursos necesarios para el correcto desarrollo del proyecto. |
| Responsable del proyecto | Responsable del proyecto por parte de la Cooperativa (Desarrollador de la tesis). | Responsable del análisis y diseño del proyecto. Gestiona el correcto desarrollo del proyecto en lo referente a la construcción e implantación. |
| Responsable funcional | Responsables del proyecto por parte de cada uno de los departamentos de la Cooperativa. | Responsables de coordinar con los diferentes usuarios la correcta determinación de los requerimientos y la correcta concepción del sistema. |
| Jefe Departamental | Responsable de proporcionar la información para el registro de productos, clientes, préstamos y demás datos financieros. | Definir la estructura de la información que se utilizará para el registro de productos, clientes, préstamos y demás datos financieros que se registraran en la Cooperativa. |

Tabla 22: Resumen de Interesados

Fuente: Metodología RUP

4.2.1.3.2 Resumen de los usuarios

Los usuarios son todas aquellas personas involucradas directamente en el uso del sistema FINANSYS. A continuación se presenta una lista de los usuarios:

| Nombre | Descripción | Responsabilidad |
|-------------------------------------|--|---|
| Administrador del sistema | Persona del Centro de Cómputo que administra el sistema FINANSYS. | Administrar eficazmente el sistema (creación de nuevos usuarios y gestionar acceso a usuarios, facilitar mantenimiento al sistema frente a nuevos requerimientos). |
| Administrador funcional del sistema | Persona de la Cooperativa de Ahorros y Crédito Unión Cochapamba que administra el sistema FINANSYS. | Administrar funcionalmente el sistema: creación de nuevos productos, definir los tipos de transacciones, nuevos grupos de clientes, tipos de préstamos, etc. |
| Usuario del sistema | Personal que labora en los diferentes departamentos de la Cooperativa de Ahorros y Crédito Unión Cochapamba que administra el sistema FINANSYS | Ingresan la información referente a cada uno de los departamentos de la Cooperativa como son registro de información personal para el ingreso de nuevos clientes, créditos, retiros , etc. |
| Usuario de gestión del sistema | Personal de la Cooperativa de Ahorro Crédito Unión Cochapamba | Validar la información proveniente de los diferentes departamentos de la Cooperativa generada a través del FINANSYS. Consolidar la información. Aprobación de Socios y Créditos |

Tabla 23: Resumen de los Usuarios

Fuente: Metodología RUP

4.2.1.3.3 Entorno actual de los usuarios

- a) El personal de los diferentes departamentos de la Cooperativa serán usuarios del sistema FINANSYS, y beneficiará así a toda la cooperativa ya que permitirá registrar y llevar un control adecuado de todos los procesos y movimientos financieros que se realizan diariamente en cada unos de los departamentos de la Cooperativa.

- b) Los usuarios manejan la información de cada módulo de forma independiente debido a que no existe un sistema que integre todos los módulos en un solo sistema.
- c) Debido a que es muy costoso adquirir licencias de software privativo el sistema será implementado en su totalidad con software libre y el lineamiento general es tener las aplicaciones en plataforma web.
- d) Actualmente existen sistemas para instituciones financieras como Bancos y Cooperativas de Ahorro y Crédito pero hay que tomar en cuenta que cada institución tiene lineamientos diferentes por lo que hace difícil la adquisición de estos sistemas ya elaborados, lo que conlleva a las instituciones a diseñar su propio sistema que se ajuste a sus necesidades.
- e) Los Módulos de Ahorros, Captaciones, Clientes, Administración y Seguridad, Cartera, Auditoria y Contabilidad formarán parte del Sistema Integrado de Información Financiero, los mismos que deberán interactuar entre si.

4.2.1.3.3.1 Coordinador de Proyecto

| | |
|---------------------------|--|
| Representante | Flandes Ibarra |
| Descripción | Responsable a nivel directivo del proyecto por parte de la Cooperativa de Ahorro y Crédito Unión Cochapamba. |
| Tipo | Gerente de la Cooperativa. |
| Responsabilidades | Establecer los lineamientos generales para el desarrollo del proyecto. Coordinar a nivel directivo los diferentes requerimientos que surjan en el desarrollo del sistema. |
| Criterios de éxito | Mantener activo el sistema luego de su implementación. Mantener la funcionalidad del sistema. |
| Implicación | Revisor de la Administración |
| Entregables | N/A |
| Comentarios | Mantener una relación constante con el desarrollo del proyecto. Brindar apoyo a nivel gerencial cuando sea necesario. |

Tabla 24: Perfil de Coordinador del Proyecto
Fuente: Metodología RUP

4.2.1.3.3.2 Responsable del proyecto

| | |
|---------------------------|---|
| Representante | Esteban Gudiño, Bairon Gudiño y Edgar Picuasi. |
| Descripción | Responsables del proyecto por parte de la Cooperativa de Ahorro y Crédito Unión Cochapamba. |
| Tipo | Analistas de sistemas |
| Responsabilidades | Responsable del análisis, diseño e implementación del proyecto. Gestiona el correcto desarrollo del proyecto en lo referente a la construcción e implantación. |
| Criterios de éxito | Cumplir con el cronograma determinado. Cumplir con los requerimientos establecidos para el sistema. Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos. |
| Implicación | Jefe de proyecto (Project Manager) |
| Entregables | Documento de visión Resumen del modelo de casos de uso Especificaciones del modelo de casos de uso Diseño ER de la base de datos y el diccionario. Especificaciones complementarias |

Tabla 25: Perfil Responsable del Proyecto
Fuente: Metodología RUP

4.2.1.3.3.3 Responsable funcional

| | |
|--------------------------|--|
| Representante | Irene |
| Descripción | Responsables del proyecto por parte de la Cooperativa de Ahorro y Crédito Unión Cochapamba. |
| Tipo | Experto en el tema. |
| Responsabilidades | Responsable de coordinar con los diferentes usuarios la correcta determinación de los requerimientos y la correcta concepción del sistema. Coordinar las pruebas de validación del nuevo sistema. |

| | |
|---------------------------|--|
| | Coordinar y asegurar la capacitación de los usuarios. |
| Criterios de éxito | Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos. |
| Implicación | Aprueba las especificaciones funcionales y las pruebas realizadas. |
| Entregables | Documento de revisión de las especificaciones funcionales. Documento de revisión de las pruebas funcionales |
| Comentarios | |

Tabla 26: Perfil responsable Funcional
Fuente: Metodología RUP

4.2.1.3.4 Perfiles de usuario

Los perfiles de cada uno de los usuarios que tendrán acceso al sistema se especifican a continuación en las siguientes tablas.

4.2.1.3.4.1 Administrador del sistema

| | |
|---------------------------|--|
| Representante | N/A |
| Descripción | Persona del Centro de Cómputo que administrará el sistema. |
| Tipo | Operador, Analista de Sistemas |
| Responsabilidades | Administrar funcionalmente el sistema: gestionar acceso a usuarios, dar mantenimiento al sistema frente a nuevos requerimientos. |
| Criterios de éxito | N/A |
| Implicación | N/A |
| Entregables | Bitácora de control de nuevos requerimientos. Bitácora de control de incidencias del nuevo sistema. |
| Comentarios | N/A |

Tabla 27: Perfil Administrador del Sistema
Fuente: Metodología RUP

4.2.1.3.4.2 Administrador funcional del sistema

| | |
|---------------------------|--|
| Representante | Irene |
| Descripción | Persona del Centro de Cómputo de la Cooperativa que administra el FINANSYS |
| Tipo | Experto en el Tema |
| Responsabilidades | Creación de nuevas cuentas de usuarios, definición de periodos contables, etc. |
| Criterios de éxito | N/A |
| Implicación | N/A |
| Entregables | N/A |
| Comentarios | N/A |

Tabla 28: Perfil Administrador Funcional del Sistema

Fuente: Metodología RUP

4.2.1.3.4.3 Usuario del sistema

| | |
|---------------------------|--|
| Representante | N/A |
| Descripción | Personal de los diferentes departamentos CUC que harán uso de cada uno de los módulos del FINANSYS. |
| Tipo | Personal de los diferentes departamentos de la CUC. |
| Responsabilidades | Ingresar la información concerniente en cada uno de los módulos del FINANSYS. Realizar consultas de información de clientes. Realizar transacciones, etc. |
| Criterios de éxito | N/A |
| Implicación | N/A |
| Entregables | N/A |
| Comentarios | N/A |

Tabla 29: Perfil del Usuario del Sistema

Fuente: Metodología RUP

4.2.1.3.5 Necesidades de los interesados y usuarios

| Necesidades | Prioridad | Inquietudes | Solución Actual | Solución propuesta |
|--|-----------|--|---|--|
| Diseñar un sistema que facilite la consolidación e integración de información mediante la unificación de módulos y automatización de procesos de la CUC. | Alta | El sistema debe consolidar la información para facilitar registro y administración de información. | Sistema desarrollado en C++ pero no es un sistema que integra todos los módulos en un solo sistema. | Implementar el módulo de Ahorros con herramientas de software actuales que nos permita integrar todos los módulos en un solo sistema. |
| Elaborar el sistema utilizando herramientas y software libre que facilite y agilice su desarrollo. | Alta | Se debe utilizar software libre para el desarrollo del sistema. | N/A | Desarrollar el sistema utilizando las herramientas de desarrollo como la plataforma JEE, frameworks JSF, Hibernate, rich faces, servidor de aplicaciones apache, estándares abiertos como CSS. |

| Necesidades | Prioridad | Inquietudes | Solución Actual | Solución propuesta |
|---|-----------|---|--|--|
| La interfaz del sistema debe ser fácil de manejar, cumpliendo con todos los requerimientos establecidos. | Alta | Cumplir con todos los requerimientos de los usuarios. | N/A | Desarrollo con la ayuda de los expertos en cada uno de los departamentos de la CUC. |
| Lograr la implementación del sistema en el menor tiempo posible para su utilización desde el próximo año. | Alta | Cumplir con las especificaciones de los usuarios. | Actualmente se trabaja con el sistema por módulos independientes . | Trabajar con el FINANSYS desde el año 2011 ya con el sistema implementado y en funcionamiento. |

Tabla 30: Necesidades de los interesados y usuarios
Fuente: Metodología RUP

4.2.1.3.6 Alternativas y competencia

Adquirir un sistema desarrollado externamente por casas Desarrolladoras de Software. Se ha mostrado interés en buscar alternativas externas para solucionar los diversos requerimientos, pero en la actualidad no existen herramientas en el mercado que se adapten a las necesidades específicas de la Cooperativa.

4.2.1.4 Vista General del Producto

Esta sección provee información a alto nivel de las funciones del sistema a implantar y de las interfaces con otras aplicaciones existentes.

4.2.1.4.1 Perspectiva del producto

La perspectiva del producto indica los módulos de los cuales estará compuesto el sistema y el orden de funcionamiento.

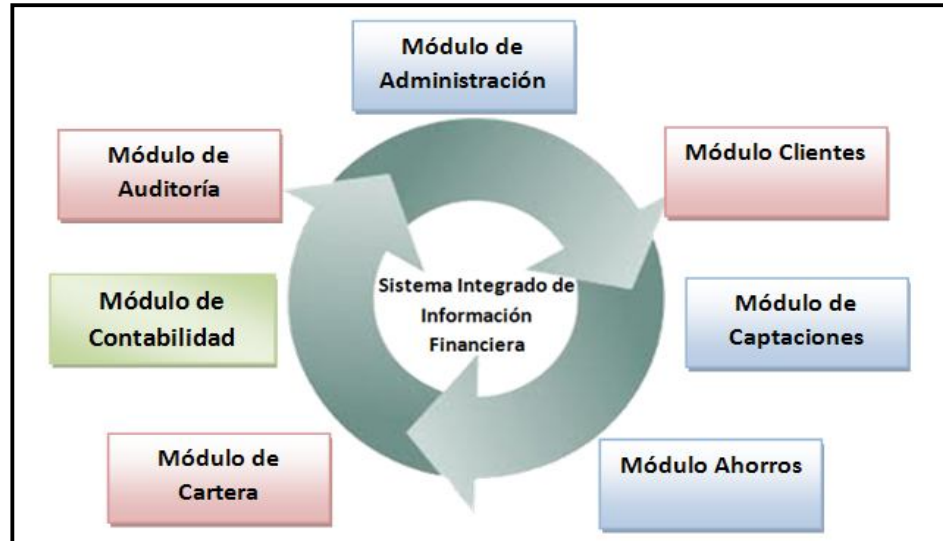


Figura 87: Perspectiva del Producto
Fuente: Propia

4.2.1.4.2 Resumen de capacidades

FINANSYS, mejorará todo el proceso de gestión financiera y permitirá además:

- Eliminar la elaboración manual de varios procesos redundantes.
- Controlar la seguridad en el manejo de información.

4.2.1.4.2.1 Módulo de Ahorros

| Beneficios para el usuario | Características que lo Soportan |
|---|--|
| El registro de la información se realiza con mayor rapidez. | Debido a su facilidad de manejo y entendimiento se podrá ingresar la información con mayor rapidez. El responsable del Manejo del módulo Ahorros tendrá una herramienta de registro y manejo de información congruente y sincronizada con el sistema integrado. |

| | |
|---|--|
| El acceso a la información y acceso a consultas se realizan con mayor facilidad. | Al contar con un sistema computacional, el tiempo de respuesta a una consulta se mejora Brindará diferentes reportes y funciones de consulta. |
| Se tendrá alta disponibilidad. | El acceso a la información a través del sistema Web permitirá a los usuarios un acceso inmediato desde cualquier punto de la intranet de la cooperativa. |
| El módulo de Ahorros y su información interactúa con el resto de módulos del sistema. | Oportunidad de mantener un control sobre la información generada por los módulos que interactúan con el módulo de Ahorros. |

Tabla 31: Resumen de Capacidades Módulo de Ahorros
Fuente: Metodología RUP

4.2.1.4.2.2 Módulo de Captaciones

| Beneficios para el usuario | Características que lo soportan |
|--|---|
| El registro de transacciones se realiza con mayor seguridad. | El usuario cajero podrá visualizar mensajes si los datos ingresados no son validos y cuando las transacciones se han realizado correctamente. El sistema permitirá visualizar la imagen del cliente para constatar quien realiza la transacción. |
| Fácil acceso a la información y acceso a consultas se realizan con rapidez | Permite mostrar al cliente saldos disponibles y préstamos pendientes para realizar sus transacciones. |
| Se tendrá alta disponibilidad de información. | El usuario podrá visualizar la información necesaria para realizar cada una de las transacciones que permite el módulo de Ahorros. |
| El módulo de Caja unifica la información. | Al ser el módulo donde se realizan la mayoría de movimientos de una institución financiera este genera y recoge información de cada módulo. |

Tabla 32: Resumen de Capacidades Módulo de Captaciones
Fuente: Metodología RUP

4.2.1.4.2.3 Módulo de Administración y Seguridad

| Beneficios para el usuario | Características que lo soportan |
|--|--|
| Control de Accesos al Sistema. | El usuario Administrador podrá ingresar visualizar la hora y fecha de acceso de usuarios al Sistema. |
| Creación de Cuentas de Usuarios con roles. | Permite mostrar al cliente saldos disponibles y préstamos pendientes para realizar sus transacciones. Los roles se asignaran dependiendo de la tarea asignada al usuario. |
| Modificación de Cuentas y Contraseñas | El usuario Administrador podrá modificar las cuentas de los usuarios que por alguna razón necesitan una nueva cuenta o contraseña para el ingreso al sistema. |
| Activar y Desactivar cuentas de Usuarios. | El usuario Administrador podrá dar de alta o baja a un usuario cuando este ya no se encuentre laborando en la institución. |
| Reportes de Usuarios | Se podrá realizar reportes de usuarios existentes y de usuarios que han ingresado al sistema. |

Tabla 33: Resumen de Capacidades Módulo de Administración y Seguridad
Fuente: Metodología RUP

4.2.1.4.3 Suposiciones y dependencias

Para el correcto uso e implantación del sistema FINANSYS, la cooperativa deberá contratar los servicios necesarios para que todas las sucursales tengan una conexión a la intranet de la cooperativa.

4.2.1.4.4 Costos y precios

| Equipos | | | | |
|---------|----------------------|-------------------|---------------|------------|
| N° | Descripción | Costo Cooperativa | Costo Tesista | Costo Real |
| 1 | Servidor HP SmartBuy | 1562,97 | 0.00 | 0.00 |
| 2 | Computador portátil | 0.00 | 800 | 0.00 |
| | Subtotal | 1562,97 | 800 | 0.00 |

| Software | | | | |
|------------------------------|--|--------------------------|----------------------|-------------------|
| N° | Descripción | Costo Cooperativa | Costo Tesista | Costo Real |
| 1 | IDE Netbeans | 0.00 | 0.00 | 0.00 |
| 2 | Servidor de aplicaciones Apache tomcat | 0.00 | 0.00 | 0.00 |
| 3 | Base de datos Postgres | 0.00 | 0.00 | 0.00 |
| 4 | Framework Hibernate | 0.00 | 0.00 | 0.00 |
| 5 | JDK 6.18 y JRE 6.18 | 0.00 | 0.00 | 0.00 |
| | Subtotal | 0.00 | 0.00 | 0.00 |
| Capacitaciones | | | | |
| N° | Descripción | Costo Cooperativa | Costo Tesista | Costo Real |
| 1 | Cursos de Capacitación Continua FICA | 0.00 | 80.00 | 80.00 |
| 2 | Cursos externos | 0.00 | 150.00 | 150.00 |
| | Subtotal | 0.00 | 230.00 | 230.00 |
| Materiales de Oficina | | | | |
| N° | Descripción | Costo Cooperativa | Costo Tesista | Costo Real |
| 1 | Resmas de hojas de papel bond | 0.00 | 20.00 | 20.00 |
| 2 | Copias de documentos y libros | 0.00 | 50.00 | 50.00 |
| 3 | DVDs | 0.00 | 10.00 | 10.00 |
| 4 | Memoria Flash | 0.00 | 30.00 | 30.00 |
| 5 | Libros | 0.00 | 80.00 | 80.00 |
| 6 | Internet | 0.00 | 40.00 | 40.00 |
| 7 | Cartuchos y recargas de tinta | 0.00 | 50.00 | 50.00 |
| 8 | Anillados y empastados | 0.00 | 120.00 | 120.00 |
| 9 | Lápices, esferos y cuadernos | 0.00 | 5.00 | 5.00 |
| | Subtotal | 0.00 | 405.00 | 405.00 |

| Varios | | | | |
|--------|---------------------|-------------------|---------------|------------|
| N° | Descripción | Costo Cooperativa | Costo Tesista | Costo Real |
| 1 | Movilizaciones | 0.00 | 200.00 | 200.00 |
| 2 | Imprevistos | 0.00 | 50.00 | 50.00 |
| | Subtotal | 0.00 | 250.00 | 250.00 |
| | TOTAL COSTOS | 1562.97 | 1685.00 | 885.00 |

Tabla 34: Costos del Proyecto
Fuente: Propia

4.2.1.4.5 Licenciamiento e instalación

Dado que se usará software libre no se requiere ningún tipo de licencia.

La instalación del Sistema será realizada por los estudiantes que han desarrollado el sistema.

4.2.1.5 Características del producto

4.2.1.5.1 Facilidad de acceso y uso

FINANSYS será desarrollado utilizando tecnología Web, desarrollando la parte visual con RichFaces optimizando de esta forma el rendimiento y facilidad de uso por parte de los usuarios, creando pantallas con una presentación clara y atractiva gracias a títulos barras de estado, íconos intuitivos, tipografía legible.

4.2.1.5.2 Unificación de la información

Unos de los principales objetivos del FINANSYS es registrar y unificar la información de la Cooperativa y proporcionar herramientas con criterios de búsqueda parametrizables, ahorrando mucho tiempo a la hora de obtener la información.

4.2.1.5.3 Mejor control y validación de la información

Al tratarse de un sistema financiero, la validación de los datos es un proceso muy importante a tener en cuenta en FINANSYS, validando correctamente los flujos de datos de entrada, para verificar que el tipo de datos sea el esperado y avisando al usuario cuando exista algún error. Logrando que la información guardada sea congruente y adecuada.

4.2.1.6 Restricciones

Por cuanto la Cooperativa en la actualidad no cuenta con sucursales, se sugiere un entorno de red local o intranet, por tal motivo el sistema no será subido al internet porque para que el sistema esté en línea se necesitan las debidas seguridades para funcionar en ese entorno.

4.2.1.7 Rangos de calidad

El desarrollo del Sistema FINANSYS se elaborará siguiendo la Metodología de Desarrollo de Software RUP, contemplando los parámetros que la metodología define para asegurar la producción de software de alta calidad y que se ajuste a las necesidades de los usuarios finales a tiempo.

4.2.1.8 Precedencia y Prioridad

- 1) Las prioridades del sistema son:
- 2) Mantener la información unificada y congruente.
- 3) Realizar de forma correcta los procesos financieros de la Cooperativa.
- 4) Interactuar entre los diferentes módulos del sistema.
- 5) Proporcionar una herramienta que tenga una interfaz amigable con el usuario.
- 6) Desarrollar un sistema multiusuario con el respectivo control de acceso y privilegios a cada usuario de acuerdo al rol que éste tenga.
- 7) Desarrollar métodos fáciles de búsqueda de información

4.2.1.9 Otros requerimientos del producto

4.2.1.9.1 Requisitos del Sistema.

Para realizar el sistema se necesita cumplir con los requisitos detallados a continuación:

- 1) Un computador servidor con las siguientes aplicaciones y características:
 - a) Sistema operativo CentOS.
 - b) Base de datos PostgreSQL versión 8.4 sobre un servidor CentOS.
 - c) Servidor web Apache Tomcat versión 6.0.26 en un servidor CentOS.
 - d) Puerto de red.
- 2) Red de área local.
- 3) Navegador Web Firefox desde la versión 4.
- 4) No requiere una conexión a internet.
- 5) Facilidad de uso.
- 6) Que sea administrable.

4.2.2 Plan de Desarrollo de Software

4.2.2.1 Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto FINANSYS para la cooperativa de ahorro y crédito “Unión Cochapamba” (en adelante CUC). Este documento presenta una visión general del enfoque de desarrollo propuesto.

El desarrollo de FINANSYS utilizará la metodología de unificación de procesos, la misma que permitirá utilizar las normas para definir el proyecto y de esta manera organizar el desarrollo la documentación generada.

El enfoque del desarrollo constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables). Este documento es a su vez uno de los artefactos de RUP.

4.2.2.1.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para definir, planificar y controlar el desarrollo de un sistema admisible. En él se describe el enfoque de desarrollo del software.

Participantes:

Los usuarios del Plan de Desarrollo del Software son:

- a) Analistas del proyecto, los mismos que definen los tiempos, esquemas, recursos y herramientas a utilizarse en el desarrollo del cronograma de actividades.
- b) Los miembros del equipo de desarrollo que ejecutan y desarrollan el contenido de la planificación del cronograma.

4.2.2.1.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo de FINANSYS. El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base principal de cada etapa. Para la versión 1.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos por medio del stakeholder representante de la “CUC” para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

4.2.2.2 Vista General del Proyecto

4.2.2.2.1 Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han realizado con el stakeholder de la “CUC” desde el inicio del proyecto, Dra. Irene de la Cruz y Sr. Flandes Ibarra.

La “CUC” como entidad financiera busca satisfacer las necesidades económicas de sus socios, en las diferentes actividades tales como: Ahorro, Crédito, Vivienda, Transporte, Salud, Educación, Agricultura y Ganadería, en una institución de estas características es previsible la adaptación a los nuevos sistemas de información y a la evolución tecnológica. Por ello, la “CUC” considera necesario el desarrollo de un nuevo sistema integrado de información financiera dado que el actual sistema usado presenta varias deficiencias entre ellas, está desarrollado con tecnología antigua, cada módulo funciona por separado esto provoca que muchos de los procesos existentes se repitan, entre otros, de ahí la necesidad de implementar un sistema informático integral basado en una arquitectura moderna, utilizando software libre y de esta manera contribuir con el desarrollo y crecimiento de la cooperativa.

El proyecto detallará la propuesta para el desarrollo de los módulos, cómo interactúan entre sí, como se conforman para un buen funcionamiento y con ello cumplir los objetivos planteados en el documento “Visión”.

a) Módulo de Administración y Seguridad:

1. Crear y administrar tipos de usuario.
2. Crear y administrar roles que se asignan a los usuarios.
3. Crear y administrar usuarios del sistema.
4. Crear y administrar sucursales de la cooperativa.
5. Crear y administrar parámetros del módulo de Clientes.
6. Crear y administrar parámetros del módulo de Captaciones.
7. Crear y administrar parámetros del módulo de Ahorros.
8. Crear y administrar parámetros del módulo de Cartera.
9. Control de acceso y autenticación de usuarios.
10. Administración de sesiones.

b) Módulo de Ahorros:

1. Registros de Cuentas de Ahorros.
2. Registro de Otros Ingresos.
3. Realizar Bloqueos y Desbloques.
4. Realizar Depósitos a Plazo Fijo.

5. Realizar Transferencias, Débitos y Créditos.
6. Realizar Consultas y Reportes.
7. Generar Backups y Respaldos

c) Módulo de Captaciones:

1. Realizar Apertura de Caja.
2. Realizar Apertura de Cuenta.
3. Realizar Depósito de Ahorros.
4. Realizar Retiros en Efectivo.
5. Realizar Retiros con Cheque.
6. Realizar Cobro de Préstamos.
7. Realizar Registro Otros Ingresos.
8. Realizar Arqueos de Caja.
9. Realizar Consultas y Reportes.
10. Generar Backups y Respaldos

4.2.2.2.2 Suposiciones y Restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la “CUC” son:

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- a) Sistemas seguros: protección de la información.
- b) Seguridad de transacciones e intercambio de información
- c) Adaptación a la normativa de Protección de Datos
- d) Adaptabilidad y la facilidad de uso.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

4.2.2.2.3 Entregables del proyecto

En la metodología RUP, a lo largo de todo el desarrollo del software se van generando documentos entregables, que son documentos explicativos de cada actividad. A continuación se describen cada uno de los artefactos que serán generados y utilizados por el proyecto.

En la metodología de RUP, todos los documentos entregables, se manejan versiones de cada uno porque pueden ir variando a lo largo de todo el proceso de elaboración del proyecto.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Glosario

Es un entregable que contiene una recopilación de términos poco conocidos o de difícil interpretación con su respectiva explicación o significado.

3) Modelo de Casos de Uso

El modelo de Casos de Uso presenta una descripción de los pasos para llevar a cabo algún proceso o funcionalidad del sistema. Se representa mediante Diagramas de Casos de Uso.

4) Visión

Este documento presenta el enfoque del producto desde la perspectiva del cliente, detallando las necesidades y características del producto.

5) Especificaciones de Casos de Uso

Para aquellos casos de uso que su funcionalidad no sea evidente, se realiza una explicación detallada utilizando un documento plantilla; dicha plantilla incluye según la especificación del caso de uso: flujo básico de eventos, flujo alternativo de eventos, precondiciones y pos condiciones.

6) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea de las interfaces que proveerá el sistema.

7) Modelo de Datos

Este documento describe la representación lógica de la estructura de los datos almacenada en una base de datos, describiendo los tipos de datos y la forma de relacionarse con la demás tablas, así como también las restricciones de integridad.

8) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración.

Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

9) Lista de Riesgos

Este documento contiene la lista de los riesgos conocidos en el proyecto, organizados descendientemente de acuerdo a la importancia que tenga el riesgo; También contiene acciones específicas de contingencia o mitigación del riesgo.

10) Manual de Instalación

Este documento proporciona una guía detallada de los pasos necesarios para la correcta instalación del sistema.

11) Producto

El sistema FINANSYS se entregará almacenado en un CD con los mecanismos apropiados para su instalación.

12) Actas de Trabajo

Las actas de trabajo contienen las actividades, responsabilidades, observaciones y revisiones que se realicen en el desarrollo del proyecto.

4.2.2.2.4 Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

4.2.2.3 Organización del Proyecto

Se muestra como está estructurado el proyecto indicando participantes, interfaces y roles.

4.2.2.3.1 Participantes en el Proyecto

El personal involucrado en el desarrollo del sistema son los siguientes:

Analistas.- Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final.

Programadores.- Los conocimientos que deben tener son: amplio conocimientos en el lenguaje de programación JAVA, administración de la base de datos PostgreSQL, diseño UML, metodología RUP.

4.2.2.3.2 Interfaces Externas

La CUC proporcionará los requisitos del sistema y los encargados de evaluar los artefactos de acuerdo a cada subsistema.

El equipo de desarrollo interactuará activamente con ellos para la especificación y validación de los artefactos generados.

4.2.2.3.3 Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

| Puesto | Responsabilidad |
|-------------|---|
| Analista | Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. |
| Programador | Son los responsables del desarrollo del proyecto, desarrollando los módulos del proyecto, creando versiones de los módulos hasta llegar a la versión final. |

Tabla 35: Roles y Responsabilidades

Fuente: Metodología RUP

4.2.2.4 Gestión del Proceso

Se muestra que actividades se realizaran en cada fase de la metodología RUP.

4.2.2.4.1 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

4.2.2.4.2 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar)

| Fase | Nro. Iteraciones | Duración |
|----------------------|------------------|------------|
| Fase de Inicio | 1 | 5 semanas |
| Fase de Elaboración | 1 | 8 semanas |
| Fase de Construcción | 2 | 16 semanas |
| Fase de Transición | 1 | 4 semanas |

Tabla 36: Planificación de fases
Fuente: Metodología RUP

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

| Descripción | Hito |
|----------------|---|
| Fase de Inicio | En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión, previamente establecidos los requerimientos en las actas de trabajo. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase. |

| | |
|----------------------|--|
| Fase de Elaboración | En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de unasemana. |
| 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. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta. |
| Fase de Transición | En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto. |

Tabla 37: Hitos Finales de cada Fase
Fuente: Metodología RUP

4.2.2.4.2.1 Objetivos de las Iteraciones

| Fase | Iteración | Descripción | Hitos Asociados | Objetivos |
|--------------|--|---|---------------------------------------|--|
| Incepción | Iteración Inicial | Definición del modelo de negocio y plan de proyecto | Actas de Trabajo Documento de Visión. | Clasificar de forma general los requerimientos del Usuario. Determinar el alcance e impacto del proyecto. Determinar la factibilidad del proyecto desde el punto de vista del negocio. |
| Elaboración | Desarrollar Prototipo | Desarrollar el prototipo de la Arquitectura. Definir y diseñar todos los casos de uso. | Documento de Arquitectura. | Definir arquitectura y herramientas. Realizar la lista de riegos técnicos. Prototipo temprano para revisión de usuarios. |
| Construcción | Primera Iteración: Desarrollar versión beta del producto. | Implementar y probar los casos de uso para la versión beta. | Versión beta del producto | Implementación de todas las características claves obtenidas de la arquitectura realizada previamente y desde la perspectiva del usuario. |
| | Segunda Iteración: Desarrollar el reléase inicial. | Implementación y desarrollo de los casos de uso restantes, corregir errores y observaciones de la versión beta. | Producto (Software) | Software revisado por los usuarios involucrados en cada uno de los módulos del sistema. Alta calidad del software con |
| | Tercera Iteración: Desarrollo del sistema completo. | Corregir y mejorar los defectos del reléase inicial. Desarrollo del sistema completo. | Producto (Software) | Toda la funcionalidad del sistema completa para su liberación. |
| Transición | Liberar el Software | Empaqueta e instalar el reléase. | Software liberado | Sistema en producción. |

Tabla 38: Objetivos de las Iteraciones
Fuente: Metodología RUP

4.2.2.4.2.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración.

Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

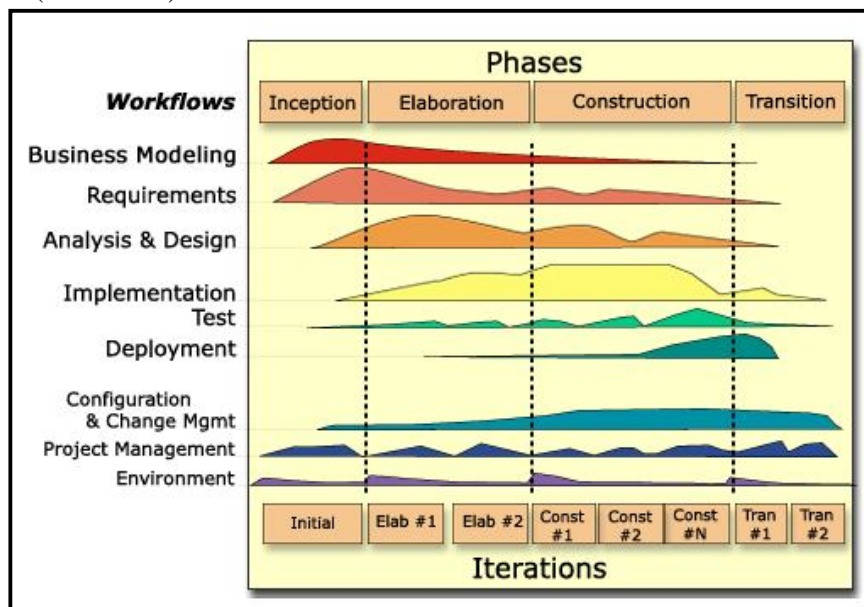


Figura 88: Calendario de las principales tareas del proyecto
Fuente: Metodología RUP

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

| Disciplinas / Artefactos generados o modificados durante la Fase de Inicio | Comienzo | Aprobación |
|---|---------------------------|--------------------------|
| Modelado del Negocio | | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1 14/10 – 20/10 | Semana 3 28/10 – 3/11 |
| Requisitos | | |
| Glosario | Semana 1 14/10 – 20/10 | Semana 3 28/10 – 3/11 |
| Visión | Semana 2 21/10 – 27/10 | Semana 3 28/10 – 3/11 |
| Modelo de Casos de Uso | Semana 3 28/10 – 3/11 | siguiente fase |
| Especificación de Casos de Uso | Semana 3 28/10 – 3/11 | siguiente fase |
| Análisis / Diseño | | |
| Modelo de Análisis / Diseño | Semana 2 21/10 – 27/10 | siguiente fase |
| Modelo de Datos | Semana 2 21/10 – 27/10 | siguiente fase |
| Implementación | | |
| Modelo de Implementación | Semana 3 28/10 – 3/11 | siguiente fase |
| Pruebas | | |
| Casos de Pruebas Funcionales | Semana 3 28/10 – 3/11 | siguiente fase |
| Gestión de Cambios y Configuración | Durante todo el proyecto | |
| Gestión del proyecto | | |
| Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones | Semana 1 14/10 – 20/10 | Semana 3 28/10 – 3/11 |
| Ambiente | Durante todo el proyecto | |

Tabla 39: Calendario Fase de Inicio

Fuente: Metodología RUP

| Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración | Comienzo | Aprobación |
|--|---------------------------|-------------------|
| Modelado del Negocio | | |
| Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio | Semana 1 14/10 – 20/10 | aprobado |
| Requisitos | | |
| Glosario | Semana 1 14/10 – 20/10 | aprobado |

| | | |
|---|---------------------------|------------------------------|
| Visión | Semana 2 21/10 – 27/10 | aprobado |
| Modelo de Casos de Uso | Semana 3 28/10 – 3/11 | Semana 5 11/12 – 17/12 |
| Especificación de Casos de Uso | Semana 3 28/10 – 3/11 | Semana 5 11/12 – 17/12 |
| Análisis / Diseño | | |
| Modelo de Datos | Semana 2 21/10 – 27/10 | Revisar en cada iteración |
| Implementación | | |
| Modelo de Implementación | Semana 3 28/10 – 3/11 | Revisar en cada iteración |
| Pruebas | | |
| Casos de Pruebas Funcionales | Semana 3 28/10 – 3/11 | Revisar en cada iteración |
| Gestión de Cambios y Configuración | Durante todo el proyecto | |
| Gestión del proyecto | | |
| Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones | Semana 4 4/11 – 10/11 | Revisar en cada iteración |
| Ambiente | Durante todo el proyecto | |

Tabla 40: Calendario Fase de Elaboración
Fuente: Metodología RUP

4.2.2.4.3 Seguimiento y Control del Proyecto

4.2.2.4.3.1 Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito.

Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

4.2.2.4.3.2 Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por la persona encargada del seguimiento y control asignada en la institución.

4.2.2.4.3.3 Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y listas de verificación incluidas en RUP.

4.2.2.4.3.4 Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

4.2.2.4.3.5 Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada por Solicitud aprobada.

4.2.2.5 Referencias

- a) Proyecto SIGPRE Automatización del proceso para la gestión de la reforma presupuestaria de EMELNORTE de Edwin Roberto López Hinojosa.
- b) Sistema para Gestión de Artículos Deportivos LSI de Patricio Letelier Torres y César López Rodríguez.
- c) Visita Virtual de la infraestructura física y tecnológica de la Universidad Técnica del Norte de Christian Guerrón y Roger Catucumbá.
- d) Documentación de Rational Unified Process, manuals de ayuda, tutoriales, etc.

4.2.3 Actas de Trabajo

En las actas de trabajo se registraron los requerimientos especificados en cada una de las visitas realizadas a la cooperativa para el levantamiento de procesos e información.

4.2.3.1 Acta de Trabajo N° 1 requerimientos módulos del sistema

| ACTA DE TRABAJO No. 1 | | |
|---|-----------|-------|
| Proyecto: Desarrollo e implementación de los módulos de administración, captaciones y ahorros para la cooperativa de ahorro y crédito “Unión Cochapamba” | | |
| Tema a tratar: Levantamiento de requerimientos funcionales de cada módulo a implementar | | |
| Fecha: 14/07/2011 | | |
| Participantes: | | |
| Nombre | Cargo | Firma |
| Sr. Flandes Ibarra | Gerente | |
| Sta. Ana Ramírez | Cajera | |
| Dra. Irene Mena | Contadora | |
| Esteban Gudiño | Tesista | |
| Observaciones: | | |
| Para el desarrollo del sistema integrado de información financiera para la cooperativa de ahorro y crédito “Unión Cochapamba” se recopilaran los datos necesarios para estructurar una buena estructura y diseño de cada módulo. | | |
| Se obtiene los siguientes requerimientos y se tratan los siguientes temas: | | |
| <ul style="list-style-type: none">• Revisión del sistema anterior y preguntas de los módulos que contiene el sistema.• Se identifica los módulos que conformara el sistema integrado.• Se obtiene las relaciones que mantendrán los módulos.• Identificación de las dificultades del sistema anterior.• Propuestas para definir el nombre del sistema.• Propuesta para desarrollo web con herramientas libres. | | |
| Compromisos adquiridos: | | |
| Indicar los beneficios del uso de herramientas libres para el desarrollo del sistema, a los directivos de la Cooperativa. | | |

4.2.3.2 Acta de Trabajo N° 2 requerimientos cuentas de usuario

| ACTA DE TRABAJO No. 2 | | |
|--|-----------|-------|
| Proyecto: Desarrollo e implementación de los módulos de administración, captaciones y ahorros para la cooperativa de ahorro y crédito “Unión Cochabamba” | | |
| Tema a tratar: Levantamiento de requerimientos funcionales del módulo de administración y seguridad. | | |
| Fecha: 16/07/2011 | | |
| Participantes: | | |
| Nombre | Cargo | Firma |
| Sr. Flandes Ibarra | Gerente | |
| Dra. Irene Mena | Contadora | |
| Esteban Gudiño | Tesista | |
| Observaciones: | | |
| <p>Para el desarrollo del módulo administración y seguridad se obtienen los requerimientos basándose en la funcionalidad de otros sistemas de autenticación.</p> <p>Se obtienen los siguientes requerimientos para autenticación y administración de usuarios:</p> <ul style="list-style-type: none">• Solo un usuario de tipo administrador podrá administrar usuarios.• El módulo de administración permitirá crear, borrar, modificar y visualizar usuarios.• A cada usuario se le asignara un tipo de usuario.• A los tipos de usuarios se le asignara un rol el cual limitara el acceso a los módulos seleccionados en el rol.• La pantalla principal constara con sistema de autenticación de usuarios.• Para la autenticación el usuarios deberá ingresar el Login, password y seleccionar la sucursal donde se realizaran las transacciones.• Dependiendo el tipo de usuarios se cargaran los menús de los módulos a los que tendrá acceso el usuario logeado.• Cada usuario debe tener un perfil de usuario donde podrá cambiar sus datos personales.• Para la administración de roles se deben registrar los siguientes datos<ul style="list-style-type: none">- Nombres de Usuario- Login- Contraseña- Correo- Seleccionar el tipo de usuario | | |
| Compromisos adquiridos: | | |
| Mantener la privacidad de la información para cada usuario. | | |

4.2.3.3 Acta de Trabajo N° 9 requerimientos cuentas de ahorro.

| ACTA DE TRABAJO No. 9 | | |
|--|-----------|-------|
| Proyecto: Desarrollo e implementación de los módulos de administración, captaciones y ahorros para la cooperativa de ahorro y crédito “Unión Cochabamba” | | |
| Tema a tratar: Levantamiento de requerimientos funcionales del módulo de ahorros | | |
| Fecha: 16/07/2011 | | |
| Participantes: | | |
| Nombre | Cargo | Firma |
| Srta. Anita Ramírez | Cajera | |
| Dra. Irene Mena | Contadora | |
| Esteban Gudiño | Tesista | |
| Observaciones: | | |
| <p>Para el desarrollo del módulo ahorros se obtienen los requerimientos basándose en la funcionalidad del sistema que actualmente funciona en la Cooperativa.</p> <p>Se obtienen los siguientes requerimientos para las cuentas de ahorros</p> <ul style="list-style-type: none"> • Solo un usuario que tenga permiso de acceso al módulo de ahorros podrá visualizar la lista de clientes con sus respectivos productos y movimientos. • Debe permitir realizar búsquedas de clientes por los siguientes parámetros: <ul style="list-style-type: none"> - Por código - Por Apellido - Por cedula - Por fecha de ingreso. • Cada cliente permitirá visualizar la lista de productos. • Cada producto de cada cliente debe permitir visualizar la lista de movimientos. • Para la lista de movimientos debe existir búsquedas de movimientos anteriores por los siguientes parámetros: <ul style="list-style-type: none"> - Por código - Por tipo de transacción - Por fecha de transacción | | |
| Compromisos adquiridos: | | |
| Visualizar el registro de productos y sus movimientos de cada cliente | | |

El resto de actas de trabajo se muestran en el apartado **C.1** del ANEXO C de los entregables de RUP.

4.3 Fase de Elaboración

En esta fase se realiza el diseño básico de la arquitectura a implementar y se realiza los casos de uso con su respectiva descripción de cada uno de ellos.

4.3.1 Arquitectura

A continuación se presenta el documento de arquitectura generado por la metodología RUP, el mismo que se describe la forma general del sistema.

4.3.1.1 Introducción

4.3.1.1.1 Propósito

El presente documento presenta una vista de alto nivel de la arquitectura del Sistema Integrado de Información Financiera (FINANSYS), objetivos, restricciones, los casos de uso más relevantes, los patrones de arquitectura aplicados y las principales decisiones sobre el diseño del sistema.

4.3.1.1.2 Alcance

El alcance de la arquitectura es definir los componentes arquitectónicos del sistema para tener una visión clara del sistema que describa los aspectos funcionales del mismo.

4.3.1.2 Representación de la Arquitectura

El presente documento presenta la arquitectura como una serie de vistas; vista de casos de uso, vista de procesos, vista de despliegue y vista de implementación. Los modelos han sido desarrollados usando el Visio, Excel, UML y JDeveloper.

4.3.1.3 Objetivos y Restricciones de la Arquitectura

Existen requerimientos y restricciones de relevancia para la definición de la arquitectura:

1. Diseño basado en capas de propósito claro y con alto grado de cohesión.

2. Desacoplamiento entre capas que permita el fácil reemplazo de los mismos.
3. Capas altamente reutilizables.
4. Todos los requerimientos descritos en el documento de Visión deben ser tomados en consideración para el desarrollo de la arquitectura definida.

4.3.1.4 Vista de Casos de Uso

Los casos de uso y los procesos de cada módulo ayudan a ver claramente cuál será el funcionamiento del sistema.

4.3.1.4.1 Procesos

4.3.1.4.1.1 Proceso Módulo Administración y Seguridad

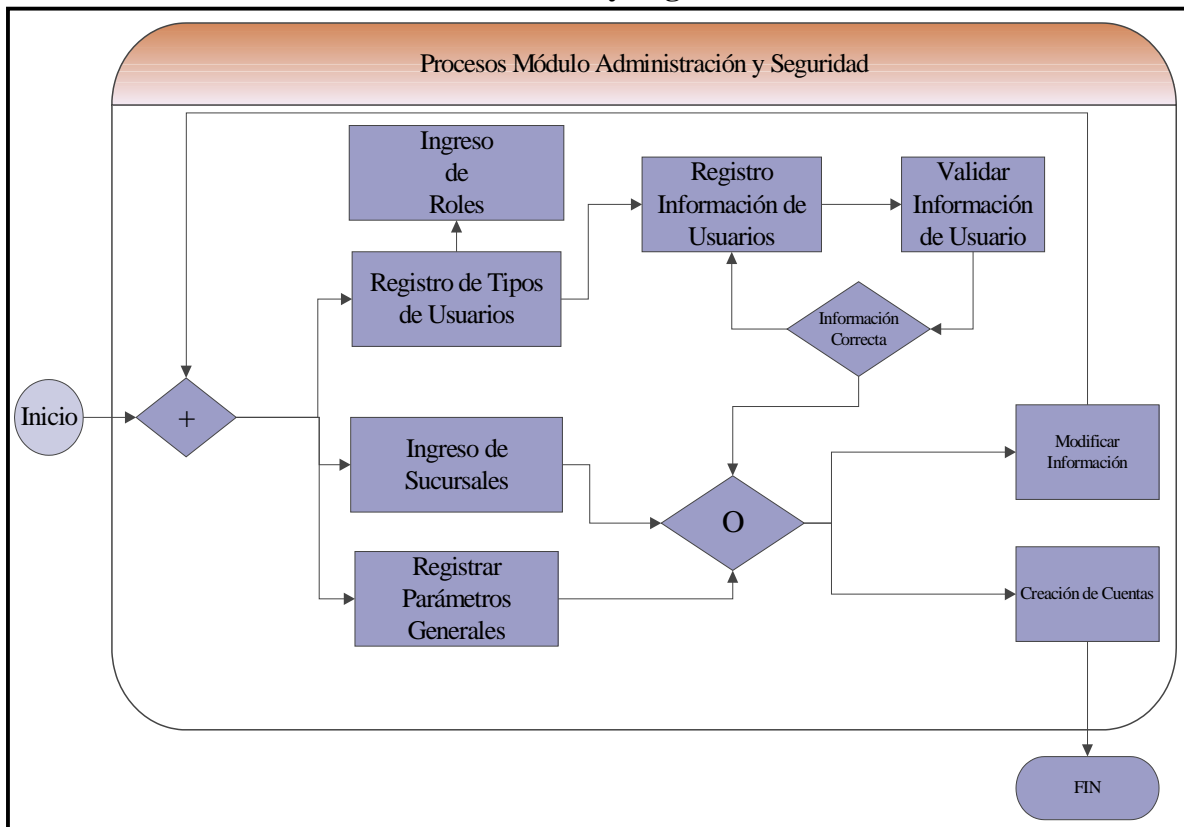


Figura 89: Procesos del Módulo Administración y Seguridad
Fuente: Propia

4.3.1.4.1.2 Proceso Módulo de Ahorros

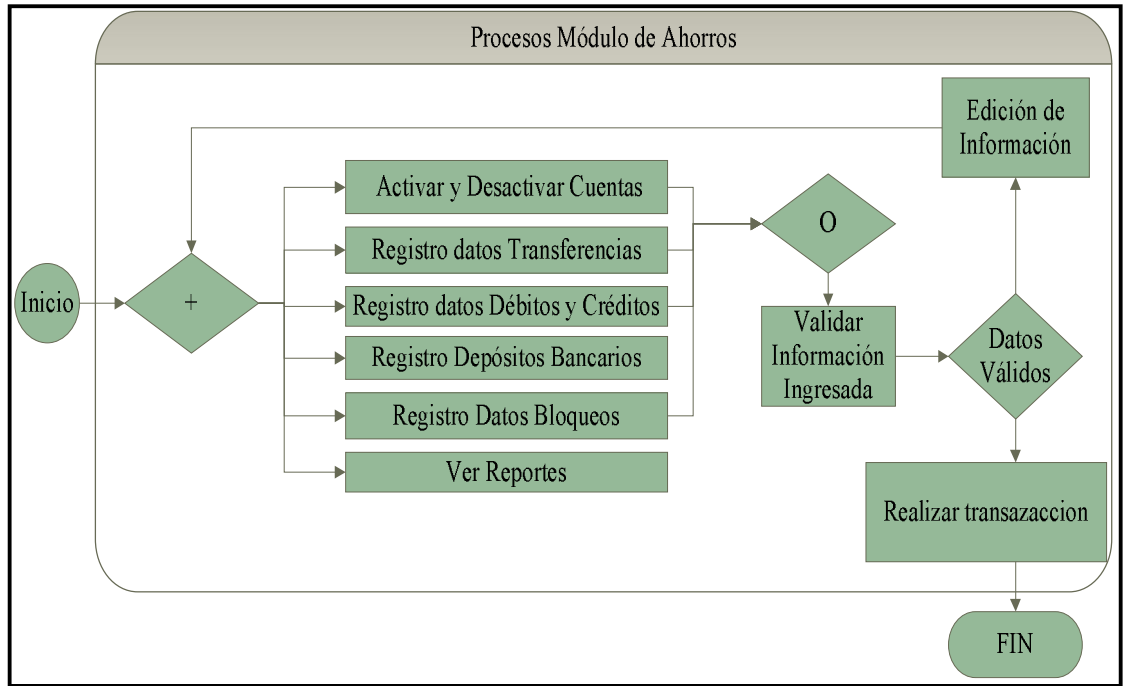


Figura 90: Procesos del Módulo Ahorros
Fuente: Propia

4.3.1.4.1.3 Proceso Módulo Cajero

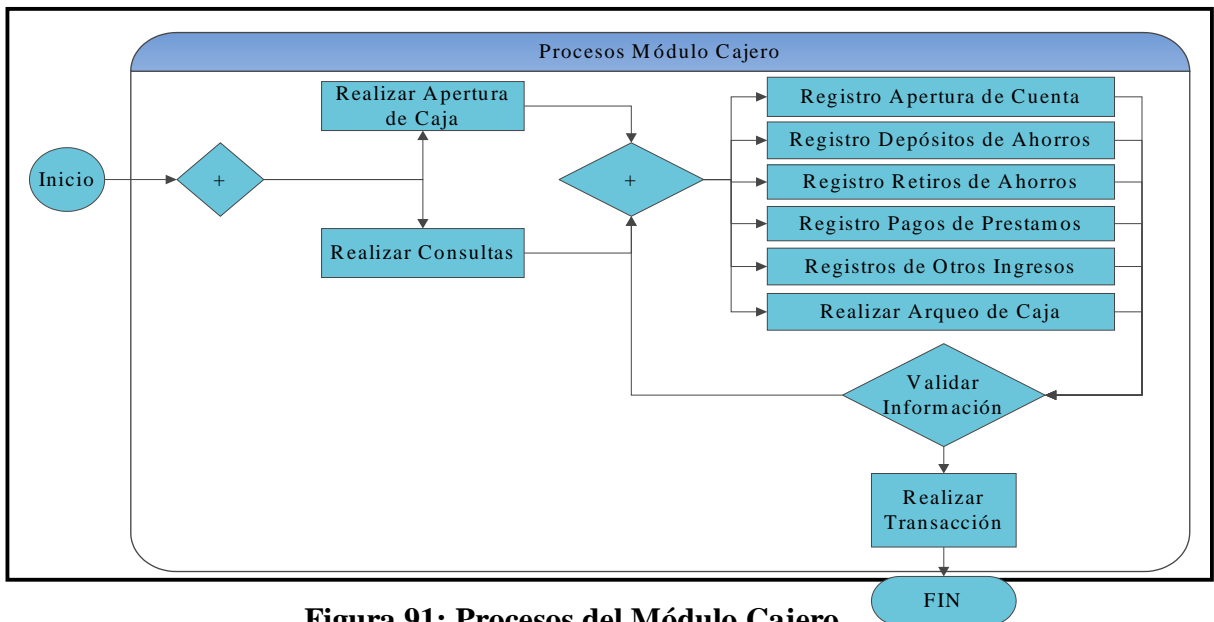


Figura 91: Procesos del Módulo Cajero
Fuente: Propia

4.3.1.4.2 Actores

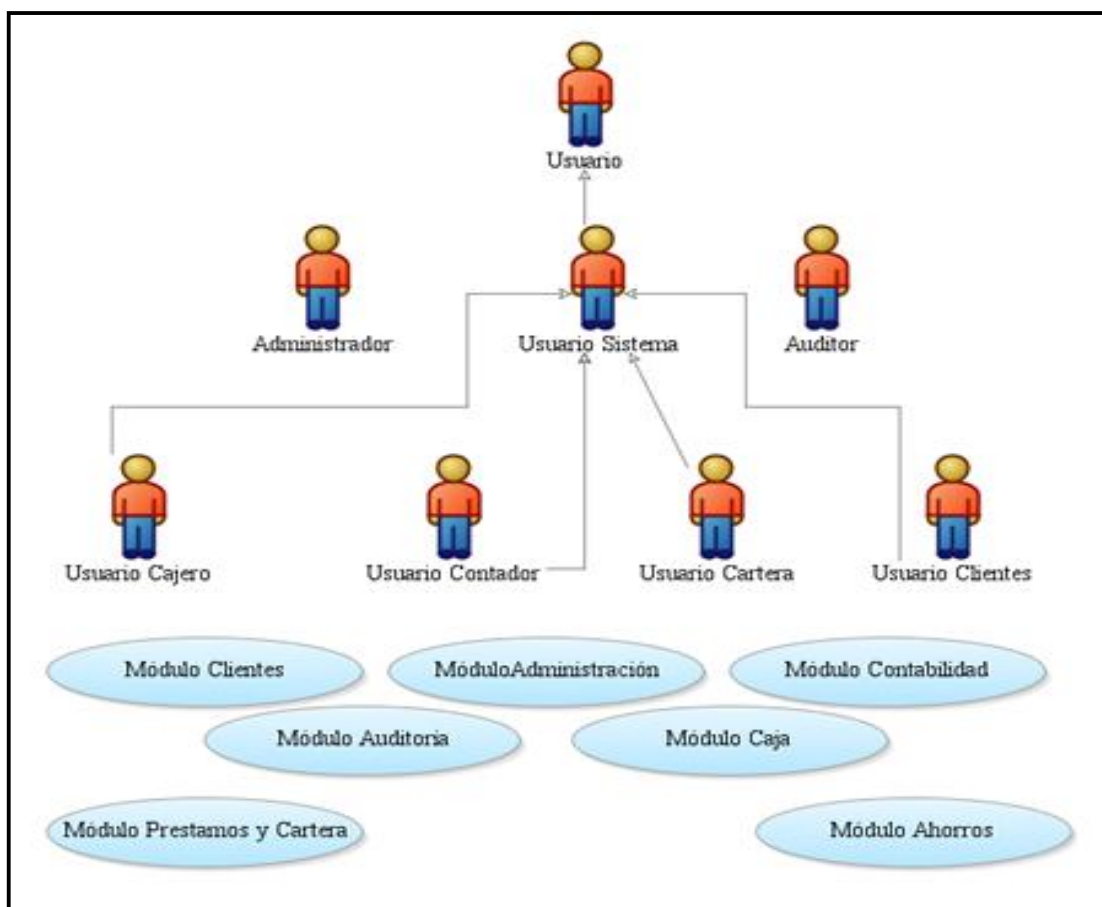


Figura 92: Actores

Fuente: Propia

| Nro. | Stakeholder | Descripción |
|------|------------------|--|
| 1 | Usuario | Usuario general del sistema que puede ingresar requerimientos y consultar la base de datos |
| 2 | Administrador | Administrador del sistema quién gestiona los objetos y recursos del sistema |
| 4 | Usuario Auditor | Realiza consultas de ingresos y salidas de usuarios, movimientos (ingreso, modificación, eliminación) de la información del sistema. |
| 5 | Usuario Contador | Procesa todas las operaciones contables que implica el sistema financiero. |
| 6 | Usuario Clientes | Se encarga de actividades relacionadas con clientes nuevos y existentes. |

| | | |
|---|-----------------|--|
| 7 | Usuario Cartera | Analiza la simulación, viabilidad, registro y asignación de préstamos a los clientes. |
| 8 | Usuario Cajero | Se encarga de realizar todas las transacciones de los clientes en las actividades de depósitos retiros, pagos de préstamos, etc. |

Tabla 41: Descripción de Clientes.
Fuente: Propia

4.3.1.4.3 Modelos de Casos de Negocio

EL modelo de casos de negocio presenta una visión global del sistema desde el punto de vista del negocio.

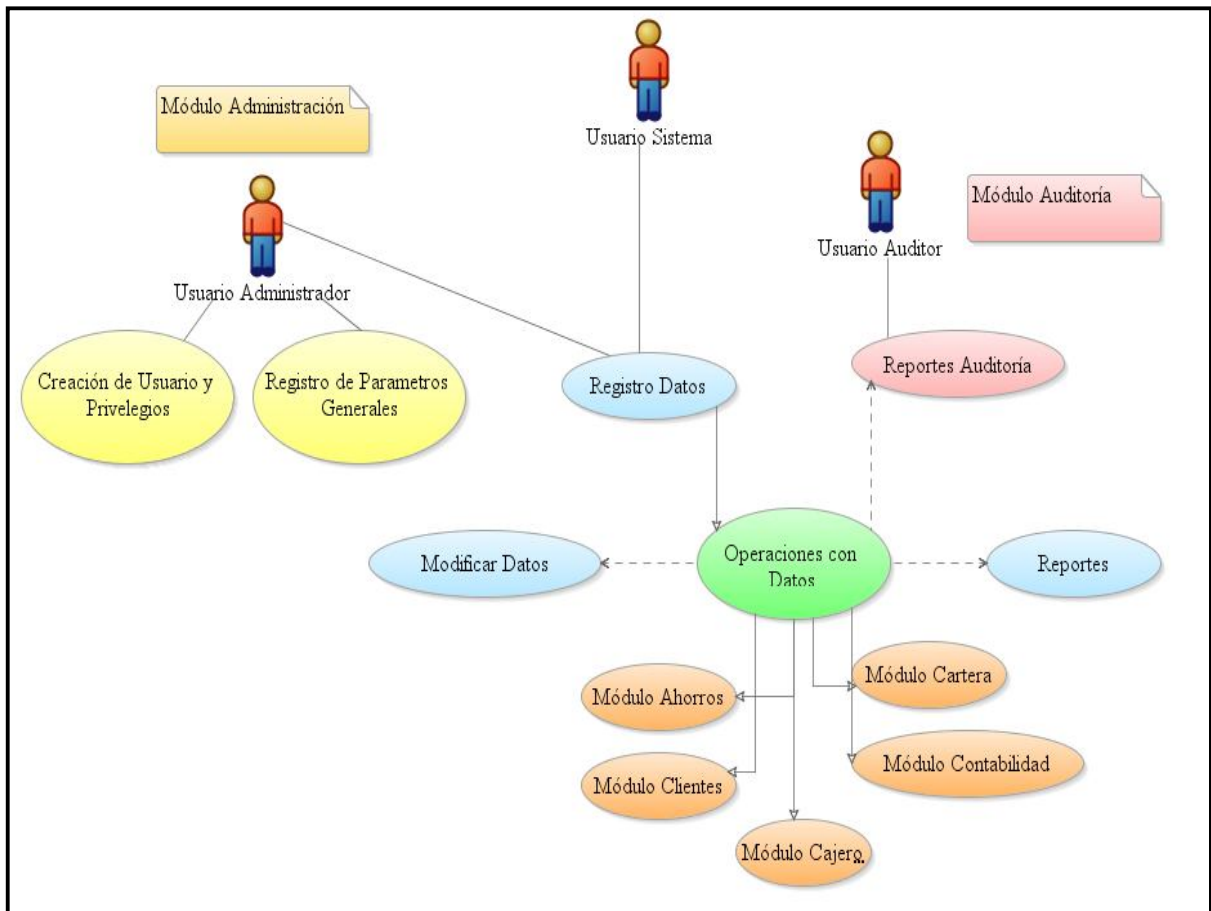


Figura 93: Modelos de Casos de Negocio
Fuente: Propia

4.3.1.4.4 Modelos de Casos de Uso

El gráfico con los casos de uso del sistema se encuentra en la sección de anexos al final de este documento.

4.3.1.4.5 Prioridad de Casos de Uso

| Caso de Uso | Prioridad para el Negocio | Prioridad Técnica |
|---|---------------------------|-------------------|
| 1. MÓDULO DE ADMINISTRACIÓN Y SEGURIDAD | | |
| 1. Crear y administrar tipos de usuarios | Baja | Alta |
| 1. Crear y administrar roles | Media | Alta |
| 2. Crear y administrarusuarios | Alta | Alta |
| 3. Crear y administrarsucursales | Alta | Media |
| 4. Crear y administrar parámetros de Captaciones (Caja) | Alta | Media |
| 5. Crear y administrar Parámetros de Ahorros | Alta | Media |
| 6. Control y Autenticación de Usuarios | Alta | Alta |
| 7. Administrar Sesiones | Baja | Media |
| 2. MÓDULO DE AHORROS | | |
| 1. Registrar Cuentas de Ahorros | Media | Media |
| 2. Registrar Otros Ingresos | Media | Media |
| 3. Realizar Bloqueos y Desbloques | Alta | Media |
| 4. Realizar Depósitos a Plazo Fijo | Alta | Media |
| 5. Realizar Transferencias, Débitos y Créditos | Alta | Alta |
| 6. Realizar Consultas y Reportes | Media | Baja |
| 7. Generar Backups y Respaldos | Alta | Media |
| 3. MÓDULO CAPTACIONES(CAJA) | | |
| 1. Realizar Apertura de Caja | Alta | Media |
| 2. Realizar Apertura de Cuenta | Alta | Alta |
| 3. Realizar Depósito de Ahorros | Alta | Alta |
| 4. Realizar Retiros en Efectivo | Alta | Alta |
| 5. Realizar Retiros con Cheque | Alta | Alta |
| 6. Realizar Cobro de Préstamos | Alta | Alta |
| 7. Realizar Registro Otros Ingresos | Alta | Alta |
| 8. Realizar Arqueos de Caja | Alta | Media |
| 9. Realizar Consultas y Reportes | Media | Media |
| 10. Generar Backups y Respaldos | Alta | Media |

Tabla 42: Prioridad de Casos de Uso

Fuente: Propia

4.3.1.4.6 Modelos de Caso de Uso

A continuación se presenta el gráfico de los casos de uso de cada uno de los módulos.

4.3.1.4.6.1 Modelo de Caso de Uso Módulo Administración y Seguridad.

La siguiente figura representa el modelo de casos de uso del Módulo de Administración y Seguridad.

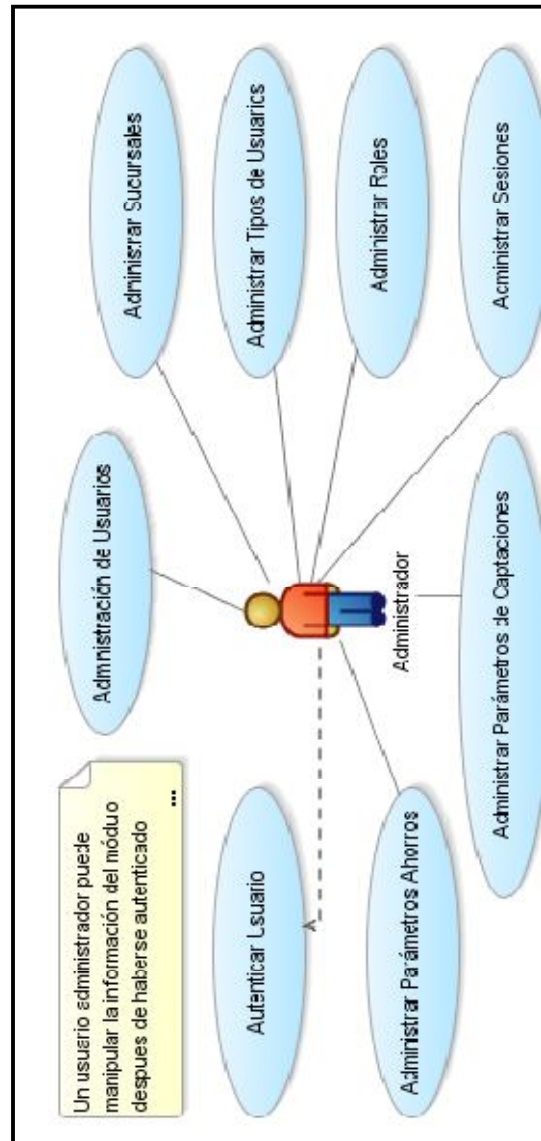


Figura 94: Casos de Uso del Módulo de Administración y Seguridad
Fuente: Propia

4.3.1.4.6.2 Modelo de Caso de Uso Módulo de Ahorros

La siguiente figura representa el modelo de casos de uso del Módulo de Ahorros.

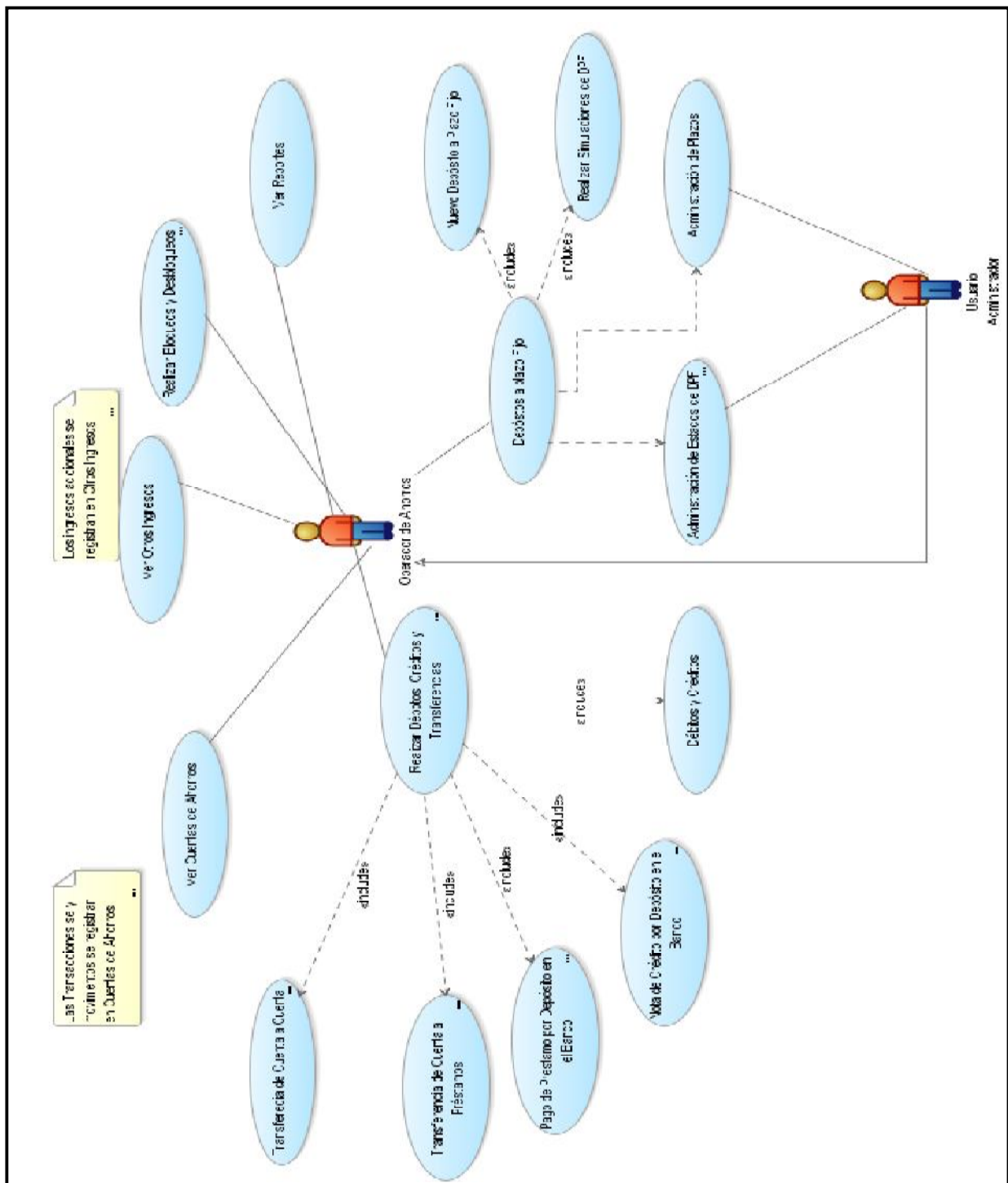


Figura 95: Casos de Uso del Módulo de Ahorros

Fuente: Propia

4.3.1.4.6.3 Modelo de Caso de Uso Módulo Cajero

La siguiente figura representa el modelo de casos de uso del Módulo de Ahorros.

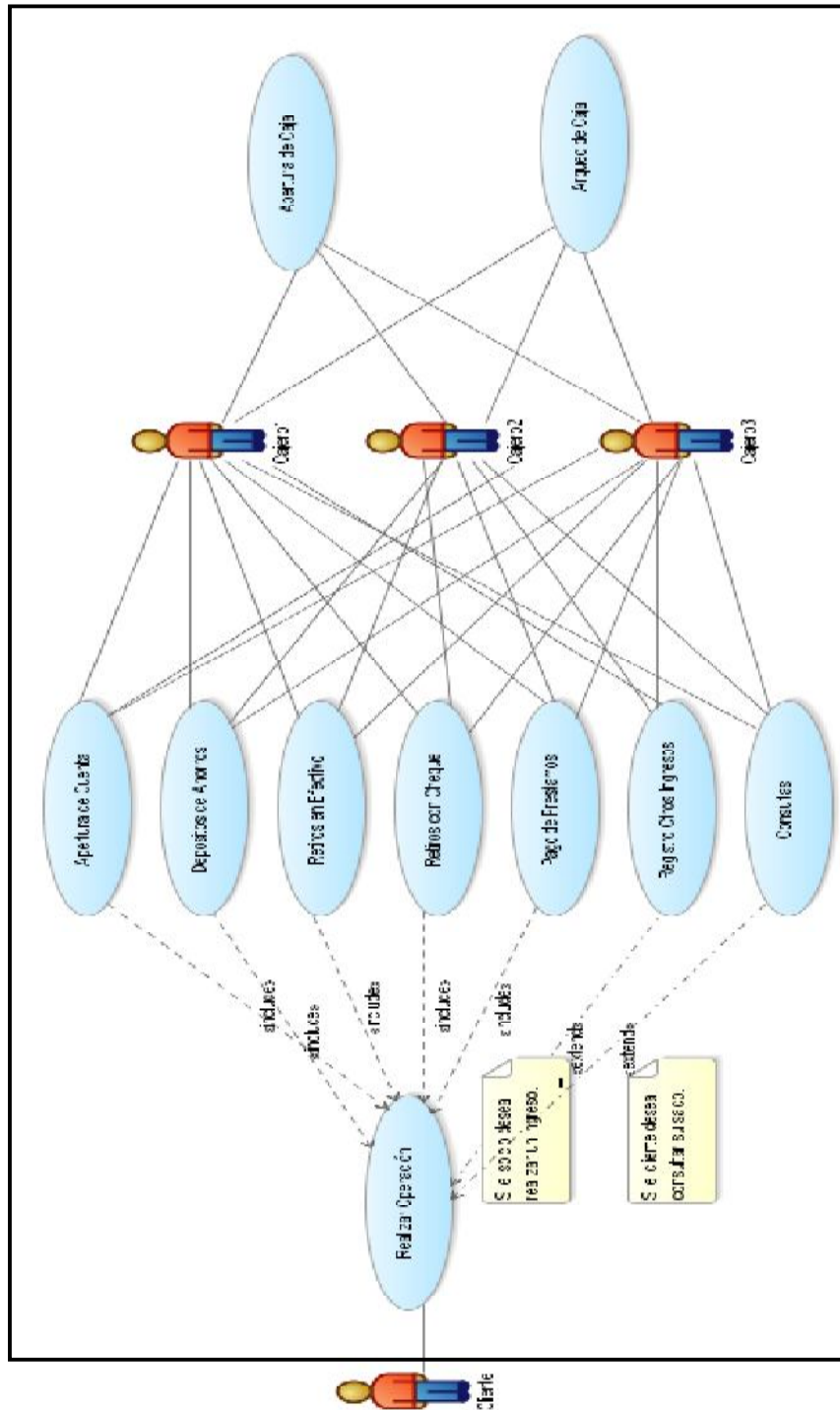


Figura 96: Casos de Uso del Módulo Cajero
Fuente: Propia

4.3.1.4.7 Descripción de los Casos de Uso más relevantes

A continuación se describen los casos de uso considerados más importantes de cada módulo del sistema.

4.3.1.4.7.1 Descripción Módulo de Administración y Seguridad

a) Crear y Administrar de Usuarios

Permitir al usuario Administrador crear cuentas para nuevos usuarios, cuentas con las cuales pueden acceder al sistema a través de un Login y password los cuales se especifican al momento de crear una cuenta. Ofrece la funcionalidad de eliminar, modificar y crear cuentas de usuario además de ver el listado de cuentas creadas.

b) Crear y Administrar de Roles

Permite asignar un conjunto de permisos a un usuario, los cuales especifican si su poseedor dispone de acceso a una determinada función del sistema o a una parte de la interfaz de usuario del sistema. Se pueden asignar varios roles a un mismo usuario. De este modo, se garantiza que el usuario disponga del conjunto de permisos definidos por tales roles. Los roles son asignados por el administrador del Sistema y las funciones que permite realizar son: eliminar, modificar y crear nuevos roles para nuevos tipos de usuarios.

c) Crear y Administrar de Sucursales

Se pueden crear nuevas sucursales si la institución lo requiere, esto les permitirá ampliar el mercado a otros lugares de la provincia. Permite la funcionalidad de crear nuevas sucursales, eliminar, modificar y ver la lista de sucursales existentes.

d) Crear y Administrar Tipos de Usuarios

Los roles que los usuarios pueden desempeñar se asignaran a un tipo de usuario. Los tipos de Usuarios nos permiten categorizar en grupos a los usuarios y así asignarles los roles correspondientes.

e) Administración de Parámetros Generales

El administrador del sistema podrá ver, crear, eliminar y modificar un sinnúmero de parámetros generales que se especifican en cada módulo pero además de eso podrá crear, eliminar y modificar los tipos de transacciones que se utilizan según las necesidades del caso al momento de realizar una transacción o movimiento en la institución.

4.3.1.4.7.2 Descripción Módulo de Ahorros

a) Registrar Cuentas de Ahorros

Permite ver la lista de clientes existentes, productos de cada cliente y los movimientos que se generan de cada producto al realizar una transacción o un movimiento financiero.

b) Registrar Otros Ingresos

Permite registrar ingresos extras que ingresan a la institución como el valor de libretas de ahorro, mantenimiento de cuenta, etc. Los cuales son ingresados por los clientes y socios de la institución.

c) Realizar Depósitos a Plazo Fijo

Permite realizar depósitos a plazo fijo si los socios lo desean dependiendo de las tasas de interés fijadas y de las políticas de la institución. Ofrece la funcionalidad de crear nuevos DPF, cancelar DPF, Renovar DPF y ver la lista de DPF.

d) Realizar Transferencias, Débitos y Créditos

Permite realizar las siguientes operaciones con las cuentas de clientes y préstamos realizados.

1. Notas de Débitos y créditos
2. Transferencias de Saldo
3. Notas de Créditos por depósitos en Bancos
4. Pagos de préstamos por depósitos en Bancos.

4.3.1.4.7.3 Descripción Módulo Cajero

a) Realizar Apertura de Cuenta

Permite realizar apertura de cuentas para clientes nuevos que requieren de este producto.

b) Realizar Depósito de Ahorros

Permite realizar depósitos de ahorros, los mismos que pueden realizarse en efectivo o con cheque dependiendo de la necesidad.

c) Realizar Retiros en Efectivo

Permite realizar retiros de ahorros, los mismos que se realizaran en efectivo y si los montos son mayores de \$500 se procede a realizar un retiro con cheque.

d) Realizar Retiros con Cheque

Permite realizar retiros con cheque, la institución emite un cheque para su retiro desde bancos externos.

e) Realizar Cobro de Préstamos

Permite realizar pagos de préstamos de los clientes dependiendo de la forma de pago.

4.3.1.5 Vista de Restricciones

En esta vista se presentan las restricciones normativas, de estándares y de tecnología, a las cuales está sujeto tanto el proceso de desarrollo como el producto desarrollado, incluidas en las categorías soporte, implementación e interfaces.

4.3.1.5.1 Normativas

Existen restricciones y normativas, dictadas por el gobierno para entidades financieras las cuales el sistema actual no le permite a la Cooperativa acoplarse en el manejo de la información financiera generada por la institución.

4.3.1.5.2 Licenciamiento

Dado que el Sistema financiero a desarrollar utiliza en su totalidad varias herramientas, estándares y tecnologías libres implementadas en java con licencia GNU GPL, el Sistema financiero no necesita inversión alguna en el licenciamiento de las herramientas a ser utilizadas.

4.3.1.5.3 Estándares

UML.- Todo artefacto utilizado para comunicación y documentación, tanto entre miembros del equipo de desarrollo como con los usuarios, está basado en UML.

Interfaz Web.- La interfaz de usuario debe estar orientada a la web. Debe ser posible visualizar el contenido utilizando cualquiera de los browsers más difundidos

4.3.1.5.4 Tecnología

El desarrollo completo del Sistema debe realizarse bajo los siguientes parámetros:

| Elemento | Tecnología (Frameworks) | Observaciones |
|---------------|-------------------------|--|
| Base de Datos | PostgreSQL | Gestor de bases de datos, de gran escalabilidad, multiplataforma, estable y confiable, diseñado para ambientes de alto volumen, libre bajo licencia BSD. |
| Framework | JSF | Tecnología y Framework para aplicaciones java de entornos web, con un modelo de programación de capas, potente por su fácil desarrollo, desempeño, adopción. |
| | RichFaces | Librería de código abierto basada en Java para componentes visuales en JSF, que permite crear aplicaciones web con Ajax |
| | MVC | A arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en capas. |
| IDE | NetBeans | Entorno de Desarrollo integrado libre, principalmente para el lenguaje Java. |
| Servidor Web | Tomcat | Servidor Web libre, compatible con Java, muy fiable, multiplataforma. |
| Tipo MIME | CSS | Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XHTML, separando la estructura y presentación de un documento. |

| | | |
|-------------------------|--------------------------|---|
| | HTML | Lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto |
| | JavaScript | Lenguaje de programación interpretado, usado principalmente para mejorar la interfaz de usuario y web dinámicas. |
| Control de Versiones | CVS | Mantiene el registro de todo el trabajo y los cambios en los ficheros que forman parte de un proyecto, permitiendo que distintos desarrolladores colaboren. |
| Sistema Operativo | De Desarrollo: Fedora | Distribución Linux muy estable, de propósitos generales basada en RPM respaldada por Red Hat. |
| | De Producción: Centos | Distribución Linux popular en el segmento de servidores web, derivada y con total compatibilidad de Red Hat. |
| Herramienta de Reportes | JasperReports | Herramienta escrita en Java para la creación de informes con contenido enriquecido. |
| Metodología | RUP | Metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos |

Tabla 43: Tecnologías Empleadas.

Fuente: Propia

4.3.1.5.5 Sistemas Existentes

A continuación se muestra la situación actual de la Cooperativa en el uso de sistemas financieros.

4.3.1.5.5.1 Sistema Financiero CUC

La CUC (Cooperativa Unión Cochabamba) en la actualidad trabaja con un sistema programado en C++, El cual se encarga en la totalidad de procesar toda la información financiera generada, sin embargo dicho sistema presenta limitaciones, porque no controla y

administra todos los procesos de forma unificada, además los datos no se guardan en ningún gestor de base de datos sino en archivos planos.

4.3.1.6 Vistas Lógicas

FINANSYS comprende 3 vistas principales como son las siguientes: Arquitectura del sistema, arquitectura lógica y arquitectura de los módulos.

4.3.1.6.1 Arquitectura del Sistema

FINANSYSY será desarrollado utilizando tecnología Web, desarrollando la parte visual con RichFaces optimizando de esta forma el rendimiento y facilidad de uso por parte de los usuarios, creando pantallas con una presentación clara y atractiva gracias a títulos barras de estado, íconos intuitivos, tipografía legible, etc.

La figura que se muestra a continuación indica la arquitectura básica de desarrollo y separa sus funcionalidades en tres capas diferenciando claramente el funcionamiento y el proceso del sistema. Esta lógica se basa en el patrón arquitectónico MVC (Modelo Vista Controlador).

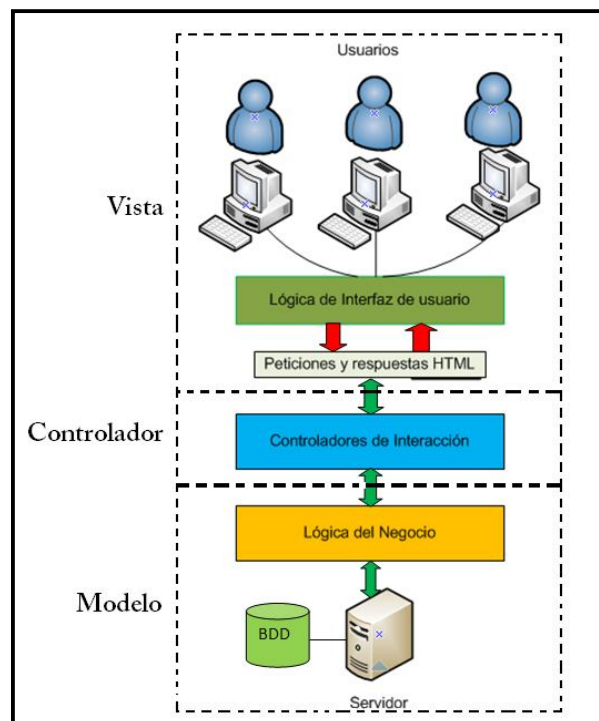


Figura 97: Patrón Arquitectónico MVC.
Fuente: Propia

La Arquitectura MVC fue planteada para disminuir el esfuerzo y complejidad de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos logrando con esto una buena funcionalidad y sobre todo entendibles lo que permite fácilmente la creación de nuevas funciones.

4.3.1.6.2 Arquitectura Lógica

Como refinamiento de la arquitectura del sistema especificada anteriormente, a continuación se muestra la arquitectura lógica del sistema.

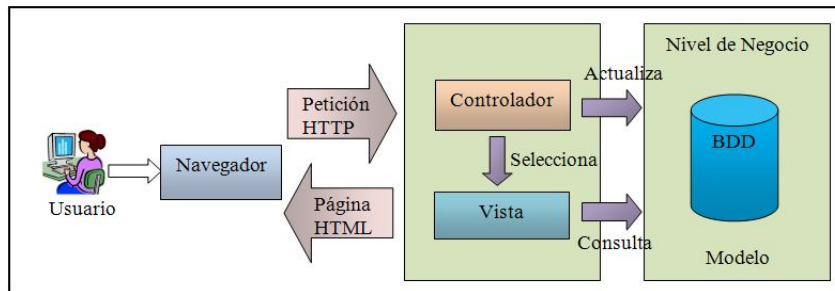


Figura 98: Arquitectura Lógica
Fuente: Propia

4.3.1.6.2.1 Interfaz de Usuario

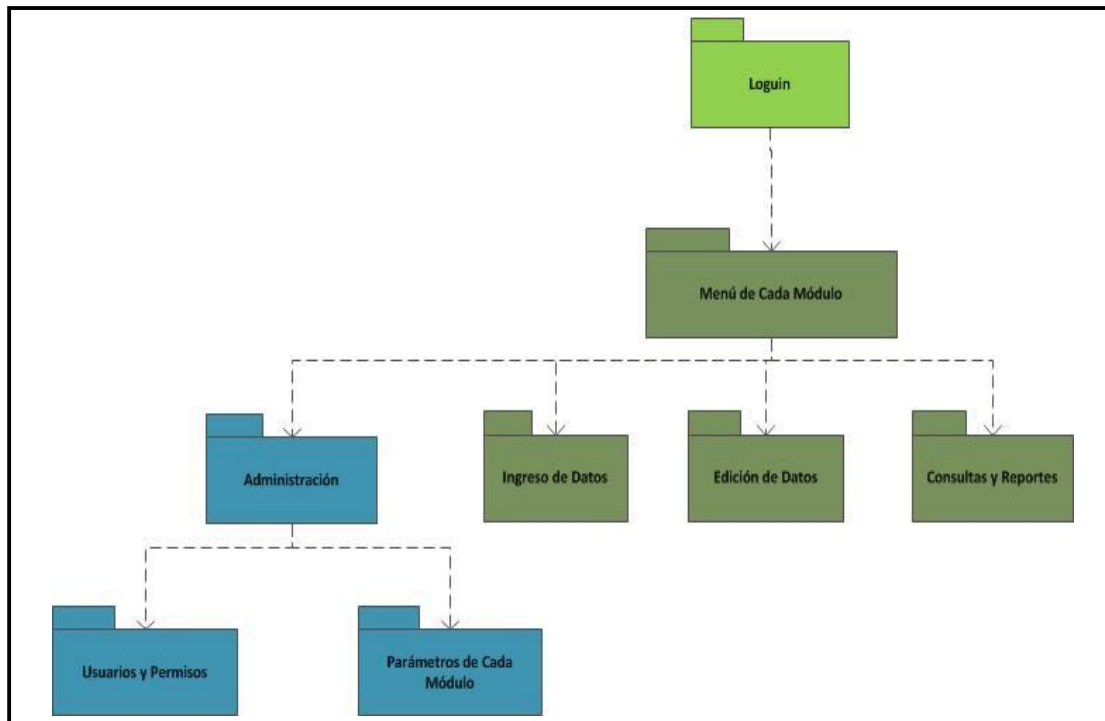


Figura 99: Interfaz de Usuario
Fuente: Propia

4.3.1.6.3 Servicios del Sistema

Al tratarse de un sistema financiero, la validación de los datos es un proceso muy importante a tener en cuenta en FINANSYS, validando correctamente los flujos de datos de entrada, para verificar que el tipo de datos sea el esperado y avisando al usuario cuando exista algún error. Logrando que la información guardada sea congruente y adecuada.

4.3.1.6.4 Infraestructura

Corresponde a los requerimientos de infraestructura que necesita el sistema FINANSYS para su funcionamiento.

4.3.1.6.4.1 Aplicación

La aplicación será publicada en el servidor Web Tomcat, y estará accesible a los usuarios del sistema a través de la red local.

4.3.1.6.4.2 Datos

Para mantener la información estable y confiable se guarda una la base de datos PostgreSQL.

4.3.1.6.4.3 Sistema Operativo

La aplicación será instalada y configurada en una máquina con Centos como su sistema operativo.

4.3.1.6.5 Vista de Datos

La Siguiete vista muestra gráficamente la estructura de la base de datos, presentando las tablas con sus respectivos campos y las relaciones existentes entre tablas.

4.3.1.7 Vista de Despliegue

El siguiente diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen el sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Como se puede observar en la figura, un usuario podrá acceder a la aplicación mediante un browser siendo especificado Firefox como browser necesario el cual ejecutará la página Jsf-RichFace con extensión jsp, el mismo que muestra la portada de login que permite el acceso al sistema. La conexión a la base de datos se realiza a través de Hibernate (Herramienta ORM).

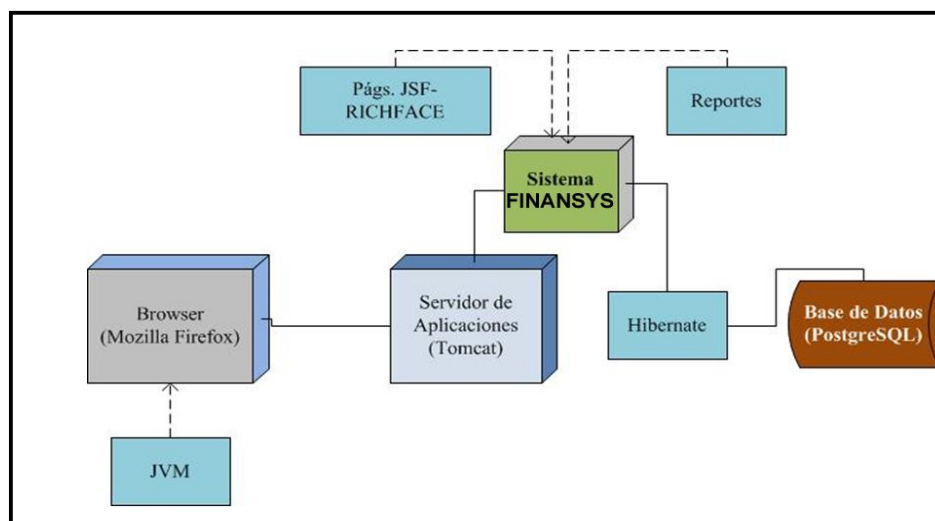


Figura 100: Vista de Despliegue.
Fuente: Propia

4.3.2 Especificación de Casos de Uso

En la especificación de cada caso de uso se indica el funcionamiento detallada de cada modulo.

4.3.2.1 Documentos

Los Documentos generados en la especificación de casos de uso se muestran en el ANEXO C de los entregables de RUP.

4.4 Fase de Construcción

En esta fase de construcción se realiza principalmente diseño e implementación, aquí se evoluciona del prototipo inicial a un producto operacional con requerimientos especificados en la fase 2. Las características de la aplicación son desarrolladas e integradas al producto, además de ser extensamente probadas para su primera versión.

Los documentos generados en este documento son el modelo de datos, plan de pruebas y la lista de riesgos como se muestran a continuación.

4.4.1 Modelo de Datos

Indica como está estructurado los diagramas de bases de datos de cada uno de los módulos especificados en el desarrollo del presente proyecto.

4.4.1.1 Introducción

4.4.1.1.1 Propósito

El propósito de este documento es definir los modelos de datos de los módulos de Ahorros, Capacaciones y Administración que conforman el Sistema Integrado de Información Financiera.

4.4.1.1.2 Alcance

Se aplica a los modelos de datos de los módulos de Ahorros, Capacaciones y Administración que conforman el Sistema Integrado de Información Financiera que será desarrollado por tesista encargado de estos módulos.

4.4.1.1.3 Definiciones, Siglas y Abreviaturas

Ver Glosario.

4.4.1.1.4 Referencias

1. Glosario
2. Resumen de los Requerimientos de los Interesados
3. Resumen del Modelo de Casos de Uso

4.4.1.2 Modelos de Datos

A continuación se presenta el modelo de datos de los módulos de Ahorros, Capacaciones y Administración que conforman el Sistema Integrado de Información Financiera.

4.4.1.2.1 Modelo de Datos Módulo de Administración

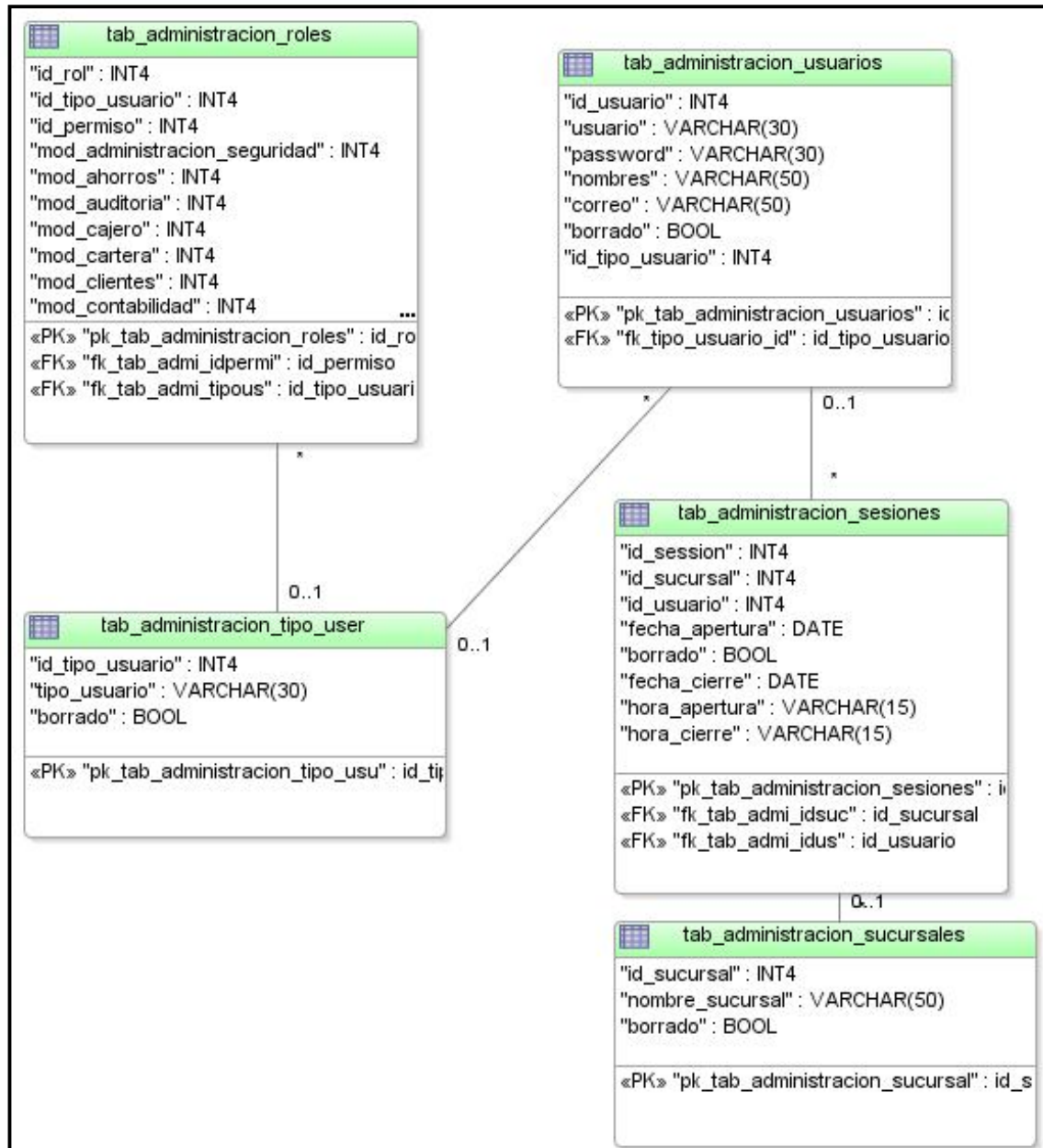


Figura 101: Modelo de Datos Módulo de Administración
Fuente: Propia

4.4.1.2.2 Modelo de Datos Módulo de Ahorros

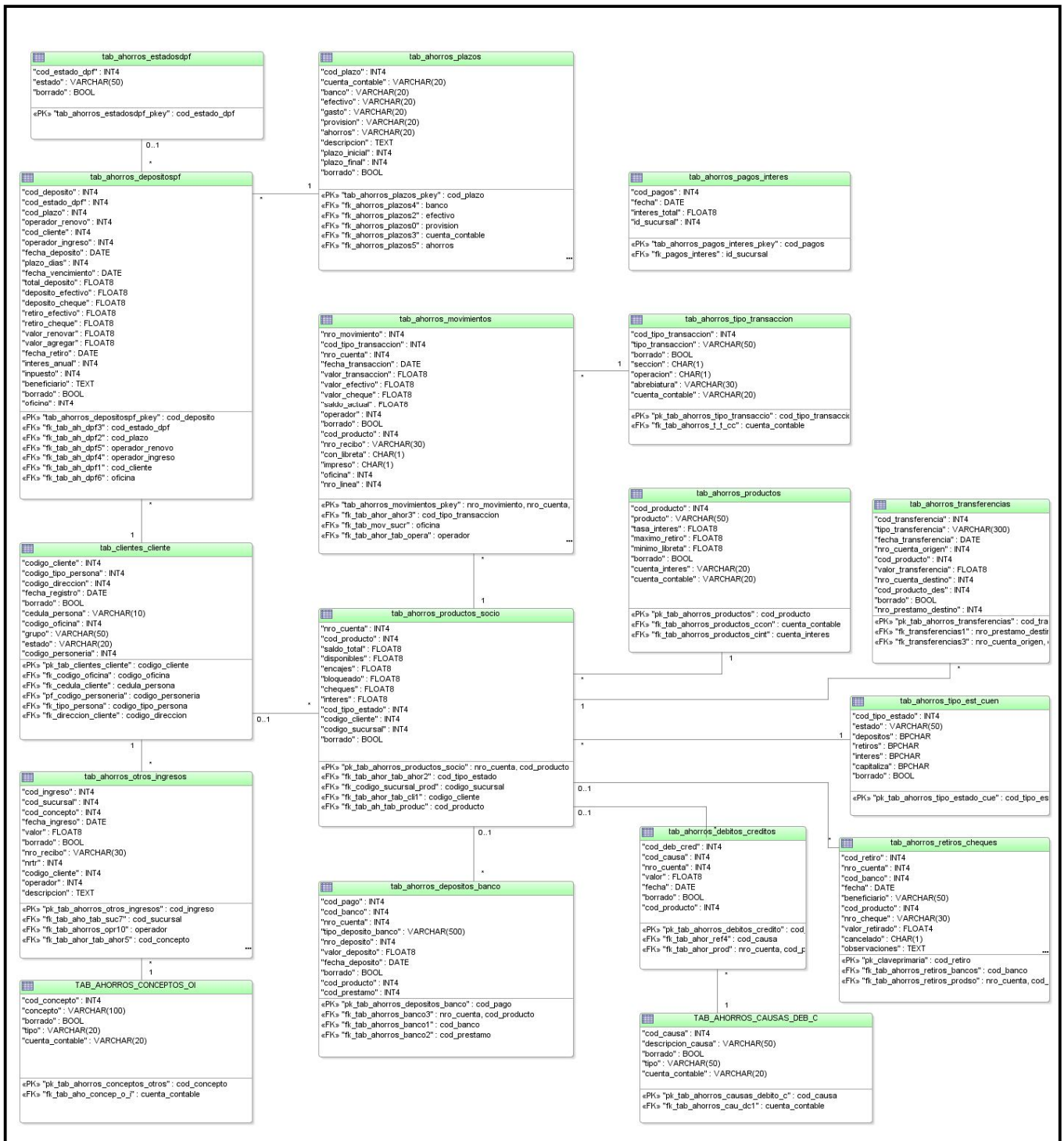


Figura 102: Modelo de Datos del Módulo de Ahorros

Fuente: Propia

4.4.1.2.3 Modelo de Datos Módulo de Captaciones (Caja)

Debido a que en el módulo de Captaciones (Caja) solo se realiza transacciones, se ha creado una sola tabla para el control de aperturas de Cajas.

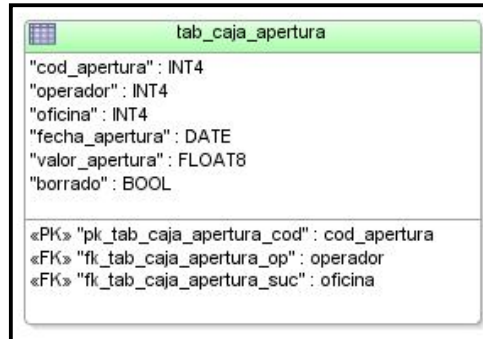


Figura 103: Modelo de Datos Módulo de Captaciones (Caja)
Fuente: Propia

4.4.2 Plan de Pruebas

4.4.2.1 Introducción

El presente documento presenta el Plan de Pruebas como respuesta al proyecto FINANSYS para la cooperativa de ahorro y crédito “Unión Cochapamba” (en adelante CUC).

4.4.2.1.1 Propósito

El propósito del Plan de Pruebas consiste en la creación de pruebas rutinarias a FINANSYS con el objetivo de identificar posibles errores e inconsistencias.

4.4.2.1.2 Alcance

Este Plan de Pruebas aplica a todos los componentes necesarios para registrar, modificar o borrar información de FINANSYS durante el proceso financiero.

4.4.2.2 Estrategia de Pruebas

La estrategia define como se realizarán las pruebas, las consideraciones principales para la estrategia de prueba son las técnicas a ser usadas y el criterio para saber cuándo las pruebas están completas.

4.4.2.2.1 Tipos de Pruebas

4.4.2.2.1.1 Pruebas de Integridad de Datos

| | |
|--------------------------------|---|
| Objetivo: | Asegurar la integridad de datos |
| Técnica: | <ol style="list-style-type: none">1. Registrar datos con tipos válidos.2. Registrar datos en entidad que tengan relación con otras.3. Revisar el esquema de base de datos para asegurarse que los datos se han guardado satisfactoriamente y de acuerdo a los estándares definidos. |
| Criterio de completitud | Todos los métodos de acceso y procesos de la Base de datos funcionan como fueron diseñados. |

| | |
|-----------------------------------|---|
| Consideraciones especiales | <ol style="list-style-type: none"> 1. Se debe utilizar un conjunto pequeño de datos para incrementar la visibilidad de cualquier evento anormal o inesperado. 2. Los datos de pruebas deberían ser reales y de uso común. |
|-----------------------------------|---|

Tabla 44: Pruebas de Integridad de Datos
Fuente: Metodología RUP

4.4.2.2.1.2 Pruebas del Sistema

| | |
|-----------------------------------|---|
| Objetivo: | Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. |
| Técnica: | <p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ol style="list-style-type: none"> 1. Los resultados esperados ocurren cuando se utiliza un dato válido. 2. Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido. 3. Cada regla de negocios es aplicada adecuadamente. |
| Criterio de completitud | <ol style="list-style-type: none"> 1. Todas las pruebas planeadas han sido ejecutadas. 2. Todos los defectos que se identificaron han sido tenidos en cuenta. |
| Consideraciones especiales | Considerar aspectos que impactan la implementación y ejecución de las pruebas del Sistema |

Tabla 45: Pruebas del Sistema
Fuente: Metodología RUP

4.4.2.2.1.3 Pruebas del Ciclo del negocio

| | |
|----------------------------------|---|
| Objetivo: | Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo. |
| Descripción de la prueba: | Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en él a través del tiempo. Debería identificarse un periodo, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un periodo de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales. |
| Técnicas: | <p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ol style="list-style-type: none"> 1. Incremente el número de veces en que una función es |

| | |
|-----------------------------------|--|
| | <p>ejecutada para simular diferentes usuarios sobre un periodo especificado</p> <ol style="list-style-type: none"> 2. Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo. 3. Todas las funciones ocurren en un periodo de tiempo serán ejecutadas en el tiempo apropiado. 4. Los resultados esperados ocurren cuando los datos válidos son usados. 5. Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido. 6. Cada regla de negocios es aplicada adecuadamente. |
| Criterio de completitud | <ol style="list-style-type: none"> 7. Todas las pruebas planeadas han sido ejecutadas. 8. Todos los defectos que se identificaron han sido tenidos en cuenta. |
| Consideraciones especiales | <ol style="list-style-type: none"> 1. Las fechas y eventos del sistema pueden requerir actividades especiales de soporte. 2. Se requiere un modelo de negocios para identificar requisitos y procedimientos de prueba apropiados. |

Tabla 46: Pruebas del Ciclo de Negocio
Fuente: Metodología RUP

4.4.2.2 Pruebas de Interfaz de Usuario

| | |
|----------------------------------|---|
| Objetivo: | <p>Verificar lo siguiente:</p> <p>La navegación a través de los objetos de la prueba refleja la funcionalidad del proyecto; Se realiza una navegación de todos los menús y los formularios de cada submenú.</p> <p>Los elementos de los formularios tales como botones, íconos, alertas, etc. Deben mantener un solo formato.</p> |
| Descripción de la prueba: | <p>Las pruebas de interfaz de usuario verifican la adecuada interacción del usuario con el software. El objetivo es verificar que cada interfaz corresponda con la acción que realiza y que la interfaz tenga una adecuada navegación.</p> |
| Técnicas: | <p>Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema.</p> <p>Los usuarios del sistema son reales y trabajan en su área de trabajo normal.</p> <p>Los desarrolladores no están presentes.</p> <p>Los usuarios son advertidos que el sistema puede fallar.</p> |

| | |
|-----------------------------------|--|
| Criterio de completitud | Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos para el sistema. |
| Consideraciones especiales | Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados. |

Tabla 47: Pruebas de Interfaz de Usuario
Fuente: Metodología RUP

4.4.2.2.3 Pruebas de Desempeño

| | |
|----------------------------------|--|
| Objetivo: | Validar y verificar los requisitos de desempeño especificados para el sistema. Validar el tiempo de respuesta para las transacciones o procesos bajo las siguientes condiciones: Volumen normal anticipado. Volumen máximo anticipado. |
| Descripción de la prueba: | Las pruebas de desempeño miden los tiempos de respuesta que tiene el sistema y otros aspectos sensibles al tiempo. Las pruebas se realizan varias veces, cambiando entre una y otra cargas diferentes. La prueba inicial debe ejecutarse con una carga similar a la esperada, mientras que la prueba final se ejecuta utilizando una carga máxima esperada. Las pruebas se pueden utilizar para calibrar el desempeño del sistema en función de condiciones como el hardware, usuarios, etc. Algunas características que pueden afectar el desempeño son: <ol style="list-style-type: none"> 1. Cuellos de botella en el disco. 2. Cuellos de botella en el CPU. 3. Capacidad de almacenamiento. 4. Capacidades físicas del hardware donde está alojado el sistema. 5. Virus informáticos. 6. Congestión en la red de datos. |
| Técnicas: | <ol style="list-style-type: none"> 1. Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema. 2. Los usuarios del sistema son reales y trabajan en su área de trabajo normal. 3. Se pide a los usuarios que todos accedan a una determinada acción o proceso al mismo tiempo. 4. Los desarrolladores no están presentes. 5. Los usuarios son advertidos que el sistema puede fallar. |
| Criterio de completitud | Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos. |
| Consideraciones | Se debe establecer el mecanismo de comunicación entre los usuarios y los |

| | |
|------------|---|
| especiales | desarrolladores para que los errores que se detecten puedan ser solucionados. |
|------------|---|

Tabla 48: Pruebas de Desempeño

Fuente: Metodología RUP

4.4.2.2.4 Pruebas de Seguridad y Control de Acceso

| | |
|-----------------------------------|---|
| Objetivo: | <p>Seguridad en el Funcionamiento y Datos, verificar que los usuarios puedan acceder solo aquellas funciones y datos para los cuales se le ha otorgado permisos al momento de crear su perfil de usuario.</p> <p>Seguridad en Administración del Sistema, comprobar que solo aquellos usuarios con permisos privilegiados puedan acceder a las funciones parametrizables del sistema y opciones del sistema.</p> |
| Descripción de la prueba: | <p>Las pruebas se realizará enfocándonos en dos aspectos principales:</p> <ol style="list-style-type: none"> 1. Seguridad en la aplicación controlando el acceso a determinada información y funciones del negocio, y 2. Seguridad del sistema realizando registro de accesos de usuarios al sistema. |
| Técnicas: | <ol style="list-style-type: none"> 1. Seguridad de Datos y Funciones, identificar los tipos de usuarios y asignar funciones a las que tiene acceso el tipo de usuarios para asignarle al usuario. 2. Realizar pruebas para cada tipo de usuario y verificar los permisos creado transacciones para cada tipo de usuario. 3. Re direccionamiento a la página de autenticación si el usuario aún no se ha logeado o registrado para el acceso al sistema. 4. Modificar los tipos de usuario y verificar si los permisos han cambiado para el usuario. |
| Criterio de completitud | Para cada tipo de dato se puede asignar las funciones y datos apropiados para su desempeño. |
| Consideraciones especiales | <p>El acceso al sistema debe ser revisado con el administrador de la red y de la base de datos.</p> <p>Con esto podremos visualizar cualquier anomalía.</p> |

Tabla 49: Pruebas de Seguridad y Control de Acceso

Fuente: Metodología RUP

4.4.2.3 Herramientas

| Tarea | Herramientas |
|------------------------------|---|
| Registro de Defectos | Microsoft Word |
| Otras Herramientas de Prueba | SQLPLUS |
| Gestión de Proyecto | JDEVELOPER, Microsoft Project, Microsoft Word |
| Herramientas DBMS | PGADMIN III |

Tabla 50: Herramientas
Fuente: Metodología RUP

4.4.2.4 Recursos

| Rol | Recurso Requerido | Responsabilidad Específica |
|--------------------------|---|--|
| Administrador de Pruebas | Esteban Gudiño, Bairon Gudiño Edgar Picuasi | Proveer las directrices de las pruebas. Adquirir los recursos Necesarios. |
| Diseñador de Pruebas | Esteban Gudiño, Bairon Gudiño Edgar Picuasi | Identificar y priorizar las pruebas. Generar Plan de Pruebas |
| Gestión de Proyecto | Esteban Gudiño, Bairon Gudiño Edgar Picuasi | Responsables de ejecutar las pruebas, registro de resultados. |
| Administrador de BDD | Esteban Gudiño, Bairon Gudiño Edgar Picuasi | Administrar y asegurar los datos de pruebas. |

Tabla 51: Recursos
Fuente: Metodología RUP

4.4.2.5 Entregables

| Entregable | Propietario | Revisión/Distribución |
|-----------------|---|----------------------------|
| Plan de Pruebas | Esteban Gudiño, Bairon Gudiño Edgar Picuasi | Coordinadores del proyecto |

Tabla 52: Entregables
Fuente: Metodología RUP

4.4.3 Lista de Riesgos

| Ranking | Descripción del Riesgo e Impacto | Estrategia de reducción del riesgo |
|---------|--|---|
| 7 | La liberación del sistema FINANSYS podría no estar lista para el mes de enero del 2012, mes en que se inicia un nuevo periodo contable. | Aumentar el esfuerzo en el desarrollo del sistema. |
| 7 | Que la cooperativa no haga la adquisición del servidor a tiempo. | Recordar a la administración de la cooperativa la adquisición del servidor. |
| 5 | Que las características del servidor donde será alojado el sistema, no cumpla con los requerimientos necesarios para un correcto funcionamiento del sistema. | Antes de su uso especificar los debidos requerimientos de hardware y software. |
| 3 | Abandono del proyecto por parte de los desarrolladores. | Aumentar el compromiso de terminar todo el sistema. |
| 3 | Incompatibilidad del navegador de internet de los usuarios. | Instalar el navegador compatible con el sistema. |
| 3 | Modificación de los requerimientos puestos en marcha. | Realizar actas de reunión de trabajo, definir bien los requerimientos y que el interesado apruebe dicha acta con la respectiva firma. |
| 3 | Saturación de la red informática por parte de los usuarios en actividades ajenas al sistema. | Sugerir políticas de uso de internet. |
| 2 | Presencia de virus en las computadores de los usuarios finales. | Instalar un antivirus y actualizar constantemente la base de datos del anti virus. |

Tabla 53: Lista de Riesgo
Fuente: Metodología RUP

4.5 Fase de Transición

El Objetivo de esta etapa de RUP es lograr que el usuario quede satisfecho con el trabajo realizado y lograr un producto final tan rápidamente y costo efectivo como sea posible.

En esta fase se realiza la entrega del Sistema, se realiza capacitaciones y se entrega el manual de instalación.

4.5.1 Manual de Instalación

A continuación se detalla como instalar cada una de las herramientas que son necesarias para el presente proyecto.

4.5.1.1 Introducción

4.5.1.1.1 Copiar el software al escritorio

En el CD del proyecto se encuentra una carpeta con todo el software necesario, copiar todos esos archivos y pegarlos en el escritorio.

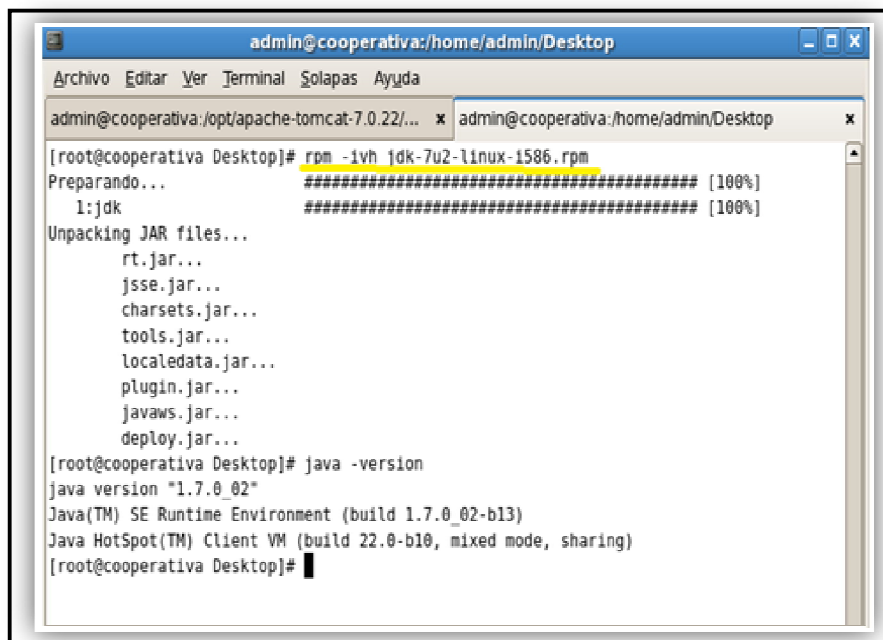
Se abre una consola, debe de autenticarse como súper usuario (root), acceder desde consola al directorio del escritorio (`cd /home/admin/Desktop`) y dar todos los permisos a todos los archivos (`chmod 777 *`).

Luego de realizar este paso, estamos listos para proceder a instalar el software.

4.5.1.1.2 Instalación de la máquina virtual de Java

Desde la consola se debe digitar el siguiente comando:

```
#rpm -ivh jdk-7u2-linux-i586.rpm
```

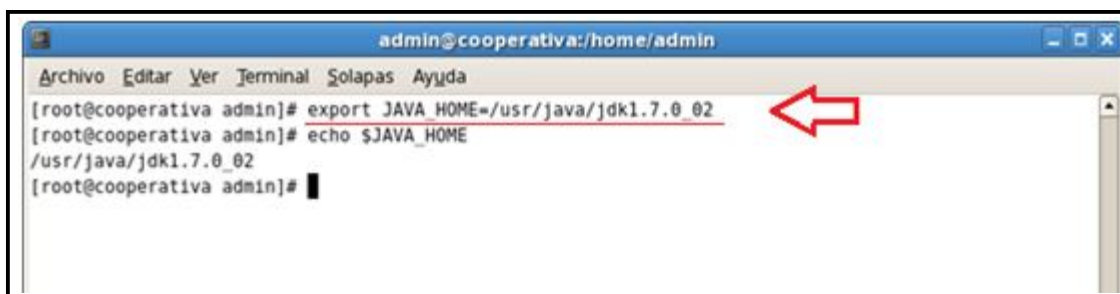
```
admin@cooperativa:/home/admin/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
admin@cooperativa:/opt/apache-tomcat-7.0.22/... x admin@cooperativa:/home/admin/Desktop x
[root@cooperativa Desktop]# rpm -ivh jdk-7u2-linux-1586.rpm
Preparando... ##### [100%]
1:jdk ##### [100%]
Unpacking JAR files...
rt.jar...
jsse.jar...
charsets.jar...
tools.jar...
localedata.jar...
plugin.jar...
javaws.jar...
deploy.jar...
[root@cooperativa Desktop]# java -version
java version "1.7.0_02"
Java(TM) SE Runtime Environment (build 1.7.0_02-b13)
Java HotSpot(TM) Client VM (build 22.0-b10, mixed mode, sharing)
[root@cooperativa Desktop]#
```

Figura 104: Instalación de la máquina virtual de Java.
Fuente: Propia

El resultado debe ser como se muestra en la Figura 53. Digitando el comando `java -version` podemos comprobar la versión del java que acabamos de instalar.

4.5.1.1.3 Configuración de JAVA_HOME

En el paso anterior, el `jdk` se instala en: `/usr/java/jdk.1.7.0_02`, el siguiente paso es configurar el `JAVA_HOME` asignando la ruta donde se instaló el `jdk` como se indica en la figura:



```
admin@cooperativa:/home/admin
Archivo Editar Ver Terminal Solapas Ayuda
[root@cooperativa admin]# export JAVA_HOME=/usr/java/jdk1.7.0_02
[root@cooperativa admin]# echo $JAVA_HOME
/usr/java/jdk1.7.0_02
[root@cooperativa admin]#
```

Figura 105: Configuración de la variable JAVA_HOME
Fuente: Propia

Del mismo modo se procede a configurar el PATH de la siguiente forma:



```
admin@cooperativa:/home/admin
Archivo Editar Ver Terminal Solapas Ayuda
[root@cooperativa admin]# export PATH=$PATH:/usr/java/jdk1.7.0_02
[root@cooperativa admin]# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/home/admin/bin:/usr/java/jdk1.7.0_02/bin:/usr/java/jdk1.7.0_02
[root@cooperativa admin]#
```

Figura 106: Configuración del PATH de java
Fuente: Propia

4.5.1.1.4 Instalación de Postgres

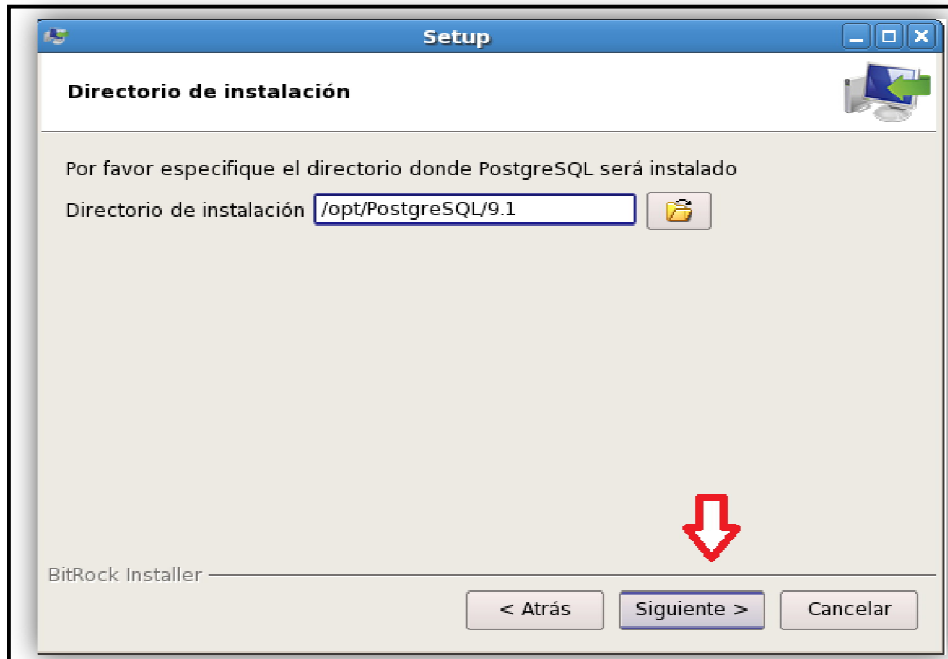
Desde la consola se debe digitar el siguiente comando y nos aparecerá el asistente de instalación de Postgres:

```
# ./postgresql-9.1.2-1-linux.run
```

A continuación se carga un asistente gráfico de instalación, los pasos son los siguientes:



Figura 107: Asistente de instalación de Postgres.
Fuente: Propia



**Figura 108: Seleccionar el directorio de Instalación de Postgres.
Fuente: Propia**



**Figura 109: Seleccionar el directorio de datos de Postgres.
Fuente: Propia**

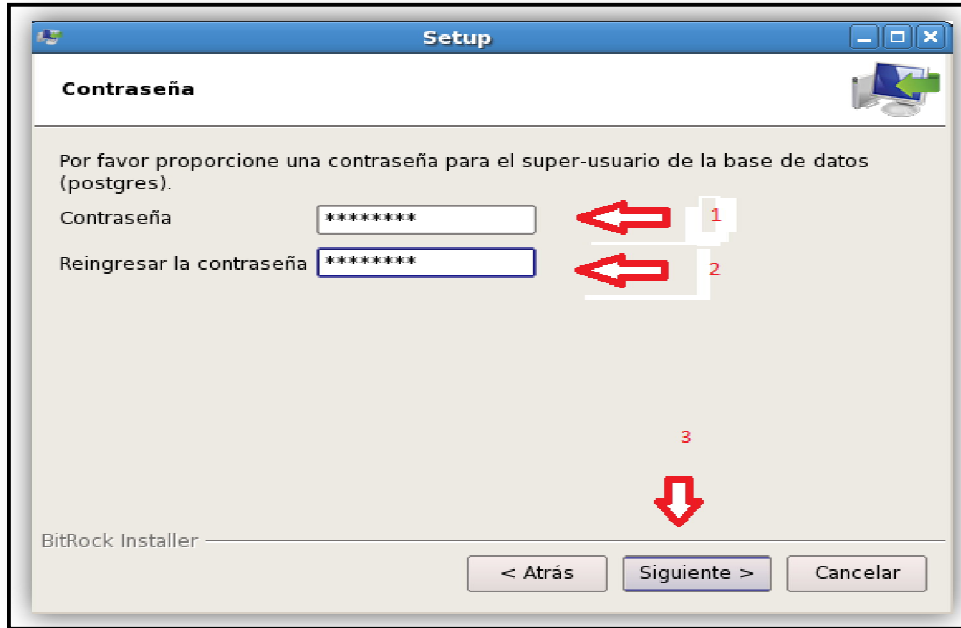


Figura 110: Ingresando la contraseña de root de Postgres.
Fuente: Propia

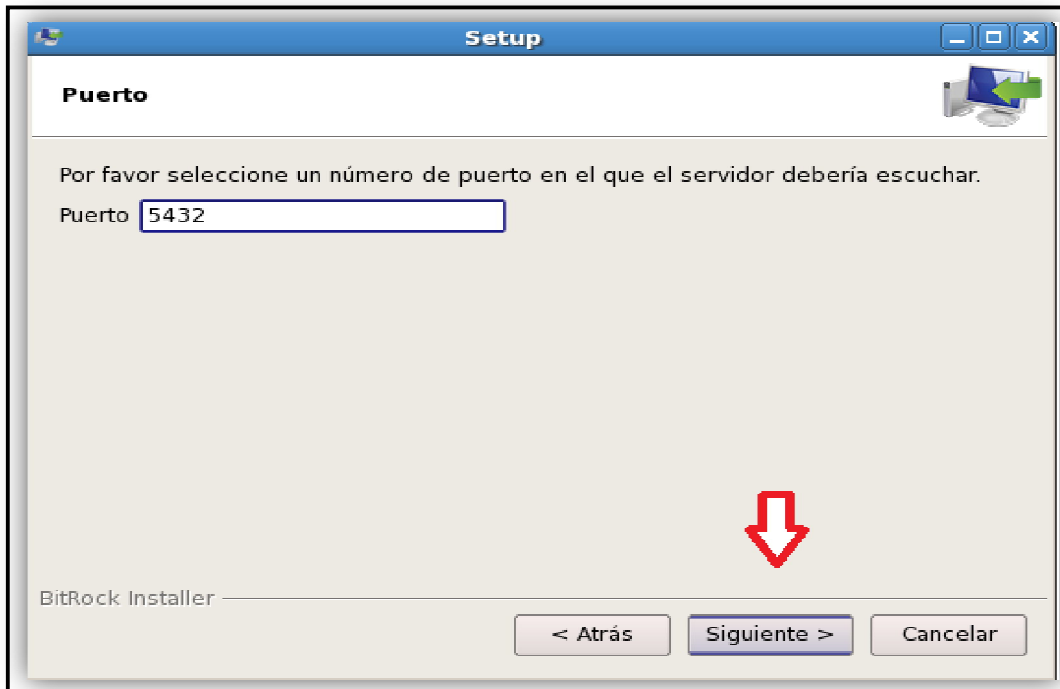
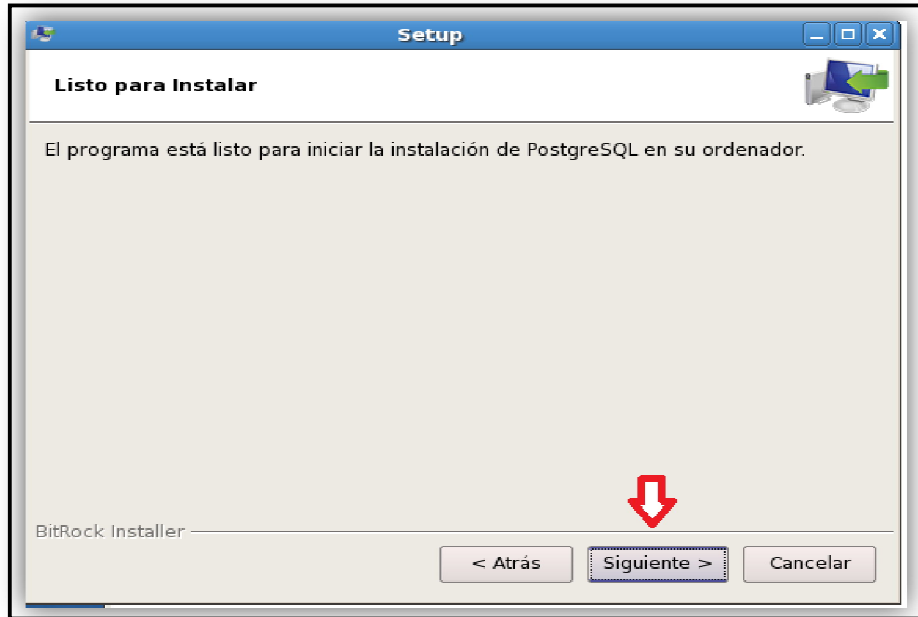
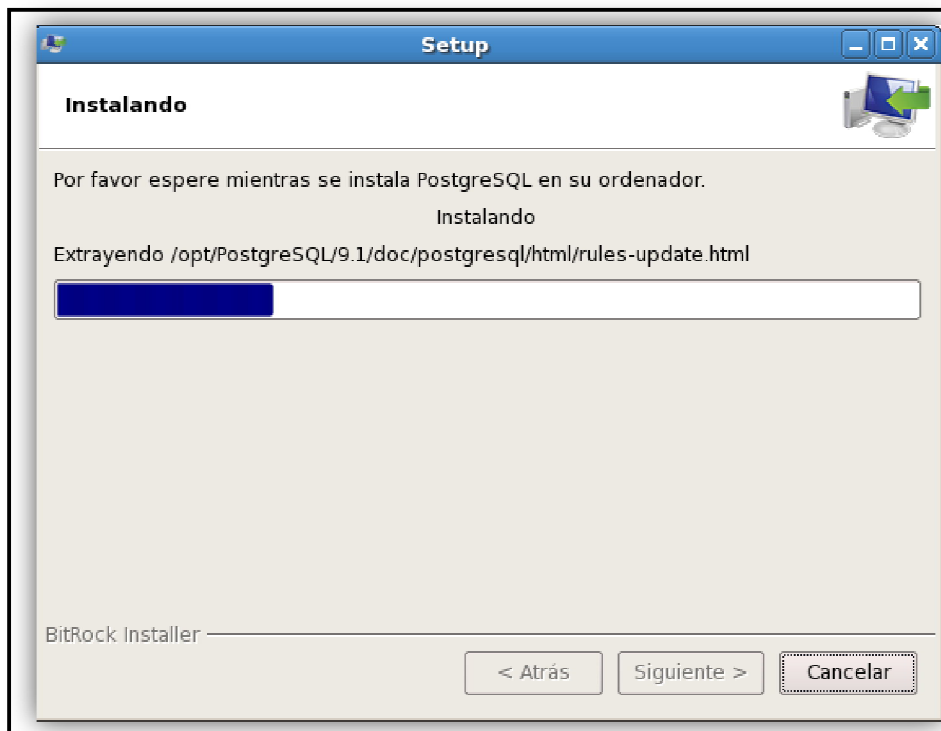


Figura 111: Seleccionar el puerto que utilizará Postgres.
Fuente: Propia



**Figura 112: El programa listo para instalar.
Fuente: Propia**



**Figura 113: Instalación en progreso de Postgres.
Fuente: Propia**

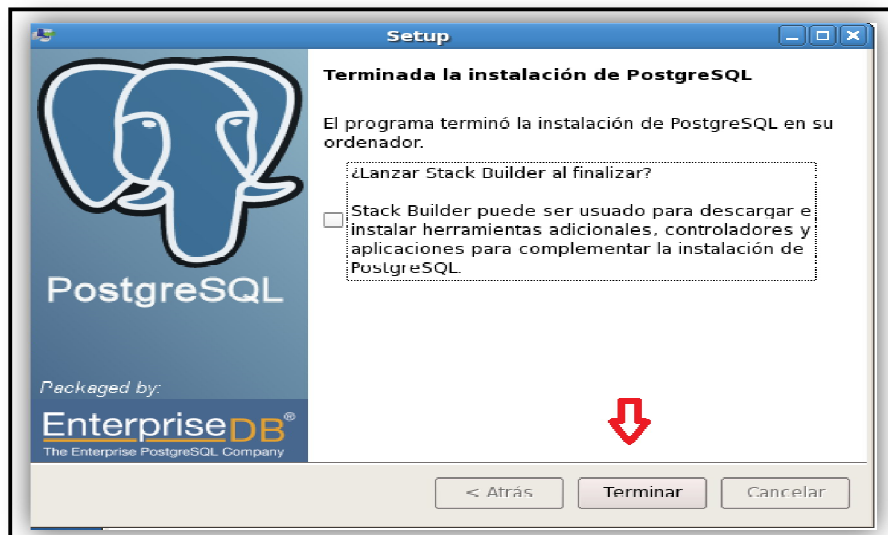


Figura 114: Terminar la instalación de Postgres.
Fuente: Propia

Una vez instalado la base de datos, procedemos a abrir PgAdmin III, para realizar acciones en el motor de base de datos.

Se debe autenticar con el password proporcionado en el proceso de instalación.

Dentro de PgAdmin III, dar click derecho en Login Roles y seleccionar New Login Role como muestra la imagen.

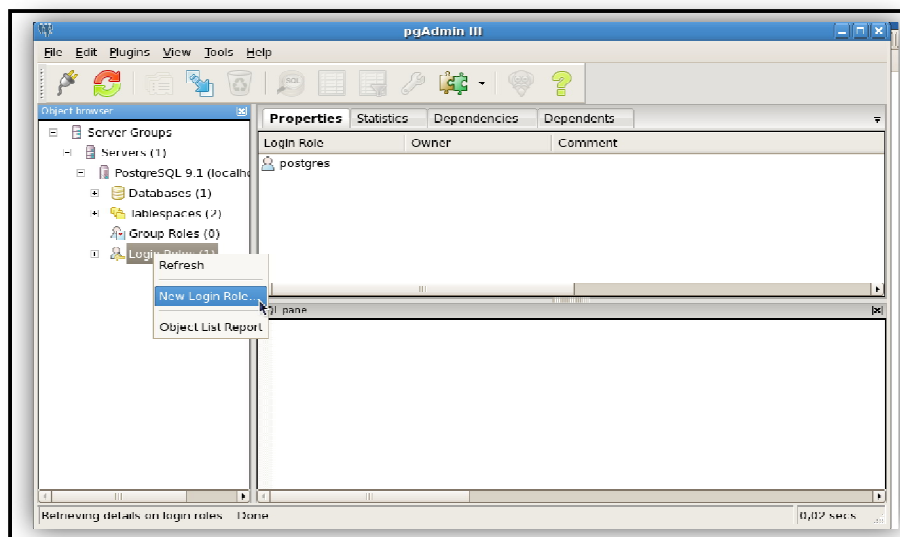


Figura 115: Creación de nuevo usuario.
Fuente: Propia

En Role name escribir finansys.

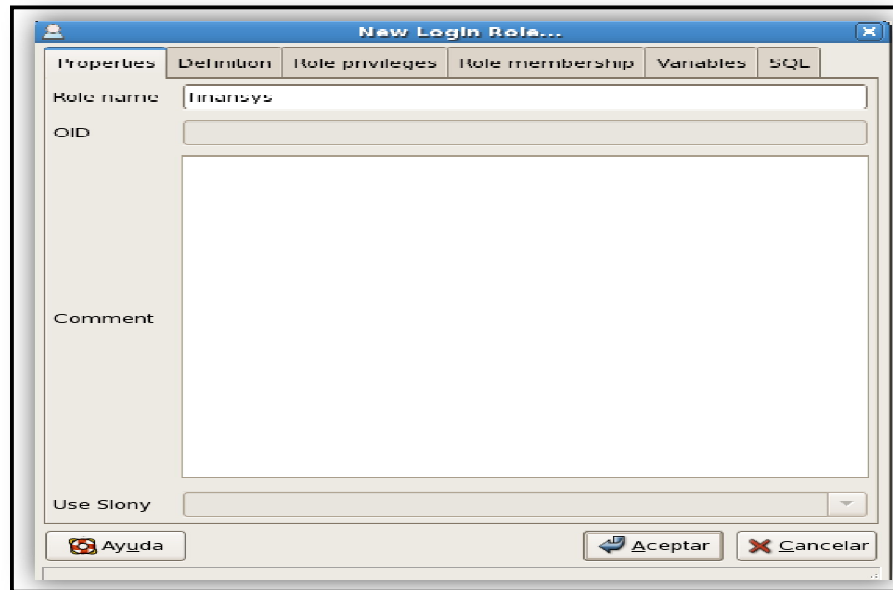


Figura 116: Creación de nuevo usuario.
Fuente: Propia

Luego en la pestaña Definition asignar el password que tendrá dicho usuario.

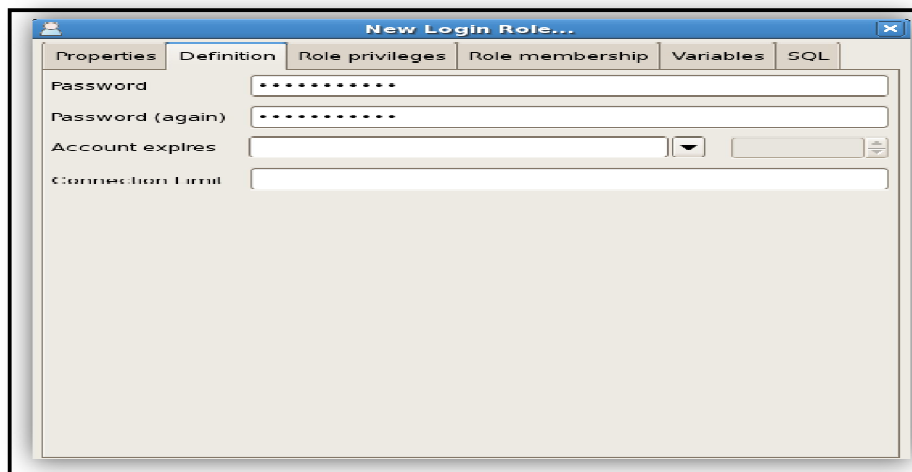


Figura 117: Creación de nuevo usuario.
Fuente: Propia

De igual forma se procede a crear una nueva base de datos dando click derecho en la opción Databases y seleccionando la opción new database. Crear la base de datos con el nombre finansys y elegir al usuario creado como se muestra en el gráfico.

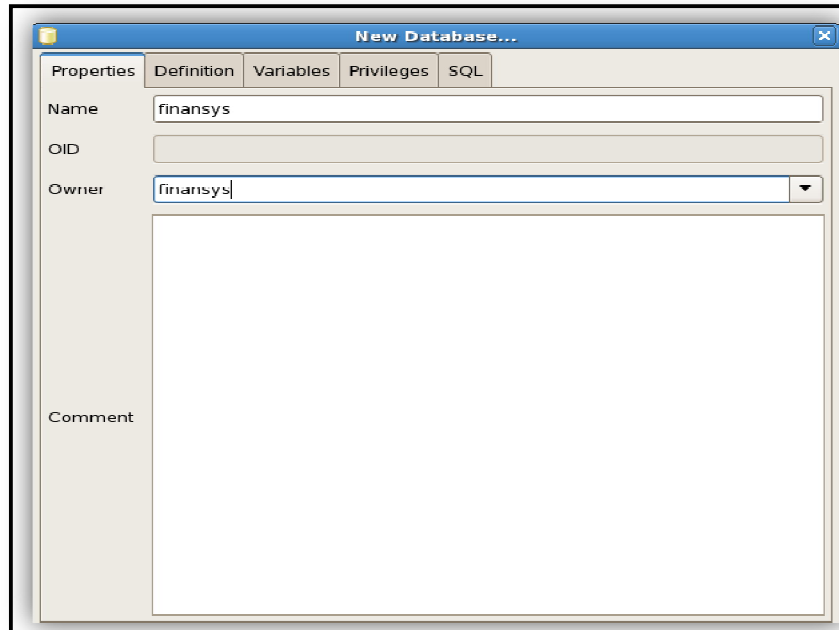


Figura 118: Creación de una nueva base de datos.
Fuente: Propia

Por último se procede a cargar el backup, el mismo que contiene el script de las tablas y algunos datos básicos que necesita el sistema. En la base de datos creada, dar click derecho y seleccionar la opción Restore.

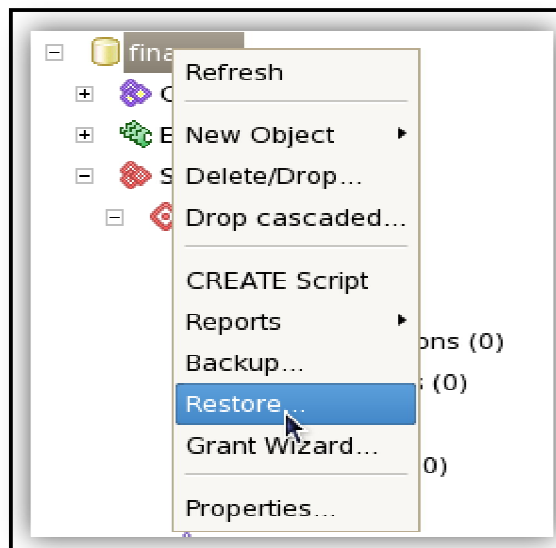


Figura 119: Subir backup a la base de datos.
Fuente: Propia

A continuación en la ventana que se carga, primero se debe seleccionar un archivo con extensión .backup, asignarle el usuario creado y presionar el botón Restore tal como se muestra en la imagen.

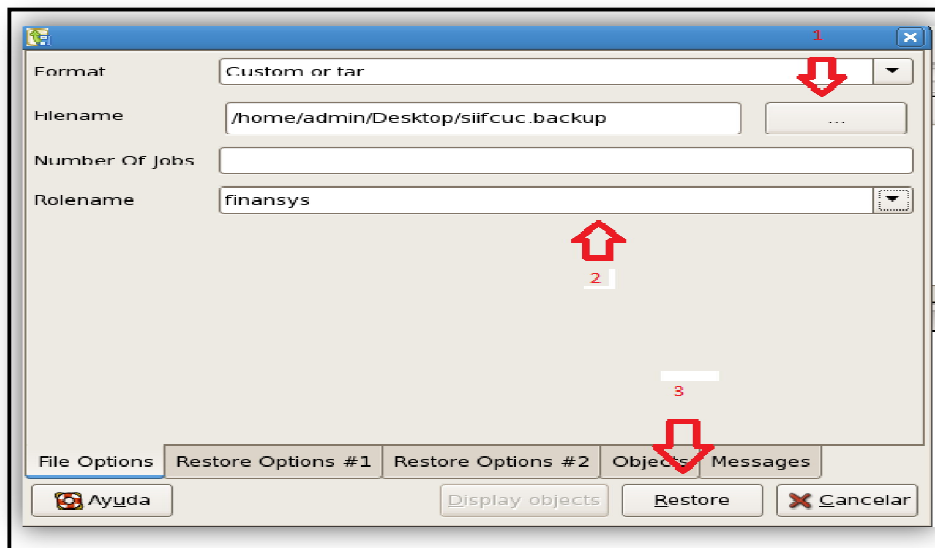


Figura 120: Subir backup a la base de datos.
Fuente: Propia

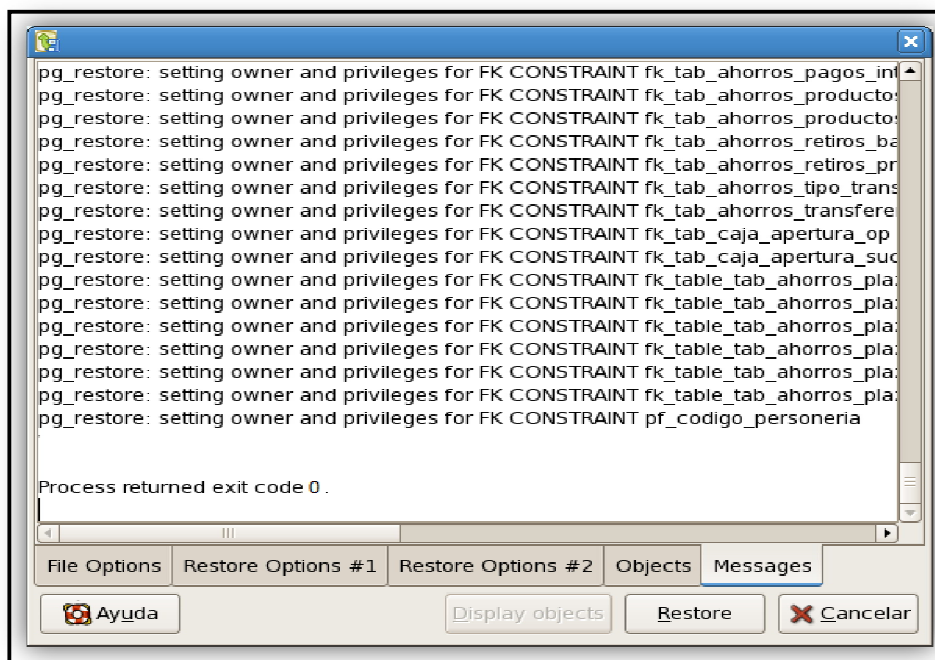


Figura 121: Subir backup a la base de datos.
Fuente: Propia

4.5.1.1.5 Instalación y Configuración de Apache Tomcat.

En el escritorio, donde se copiaron todos los archivos, existe un archivo comprimido llamado `apache-tomcat-7.0.22-x86.zip`, se accede nuevamente a la consola que estaba abierta y ubicada en el directorio del escritorio, para descomprimir el archivo de digita el siguiente comando:

```
# unzip apache-tomcat-7.0.22-x86.zip
```

Al concluir la ejecución del comando anterior, se crea en el escritorio una carpeta llamada `apache-tomcat-7.0.22`:

```
# mv apache-tomcat-7.0.22 /opt/
```

```
# cd /opt/ apache-tomcat-7.0.22/bin
```

```
# chmod 777 *
```

Hasta ahí ya está el apache tomcat listo, para iniciar el servicio hay que ejecutar el fichero `startup.sh`. Para hacer que el servicio se inicie automáticamente al inicial el sistema operativo se debe seguir los siguientes pasos:

```
# cd /etc/init.d/
```

```
# vi apache
```

El script `apache` debe quedar de la siguiente forma

```
#!/bin/sh
# Script ejemplo para arranque de servicios en /etc/init.d/
#

case "$1" in
start)
echo "Iniciando servicio..."
/opt/ apache-tomcat-7.0.22/bin/startup.sh
;;
stop)
echo "Deteniendo servicio..."
/opt/ apache-tomcat-7.0.22/bin/shutdown.sh
;;
*)
echo "Modo de empleo: /etc/init.d/mi_script {start|stop}"
exit 1
;;
esac
exit 0
```

Posteriormente, hemos de crear un enlace simbólico en el runlevel correspondiente para que se ejecute cada vez que arranquemos la máquina.

```
#ln -s /etc/init.d/apache /etc/rc3.d/S98apache
```

Reiniciar el servidor para verificar que el servicio si se inicia automáticamente al inicial el sistema. Una vez iniciado el sistema, para verificar que el servidor Apache Tomcat este funcionando correctamente, en el path en un web browser digitar localhost y debe cargar la siguiente página:

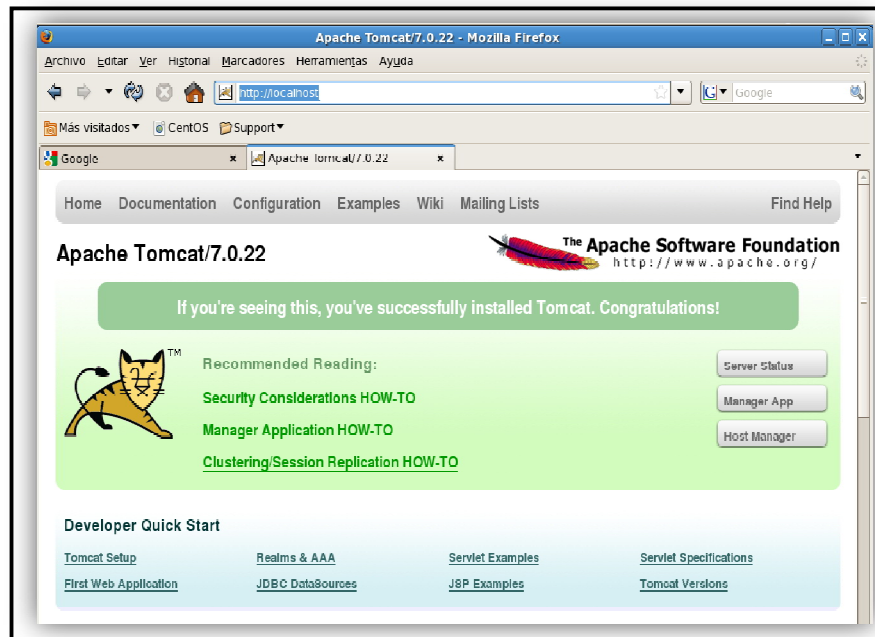


Figura 122: Página de administración de Apache Tomcat
Fuente: Propia

Por último el archivo con extensión .war ubicado en el escritorio, pegarlo en:

```
/opt/apache-tomcat-7.0.22/webapp/
```

Para verificar que la aplicación funcione correctamente en el navegador firefox digitar `http://localhost/FINANSYS`

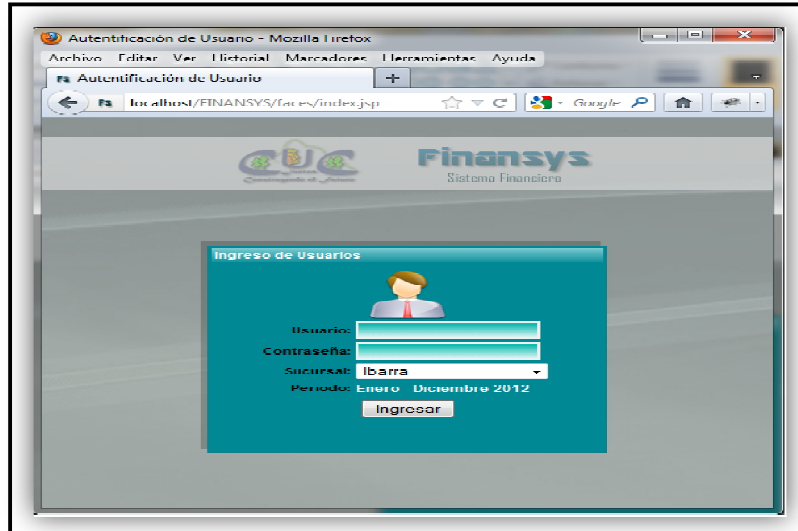


Figura 123: Página de inicio del sistema.

Fuente: Propia

4.5.1.1.6 Configuración del SSL para HTTPS

Después de configurar correctamente el JAVA_HOME, se procede a generar una clave con el algoritmo de encriptación RSA, a dicha clave debemos asignarle un alias; Se debe configurar de la siguiente manera:

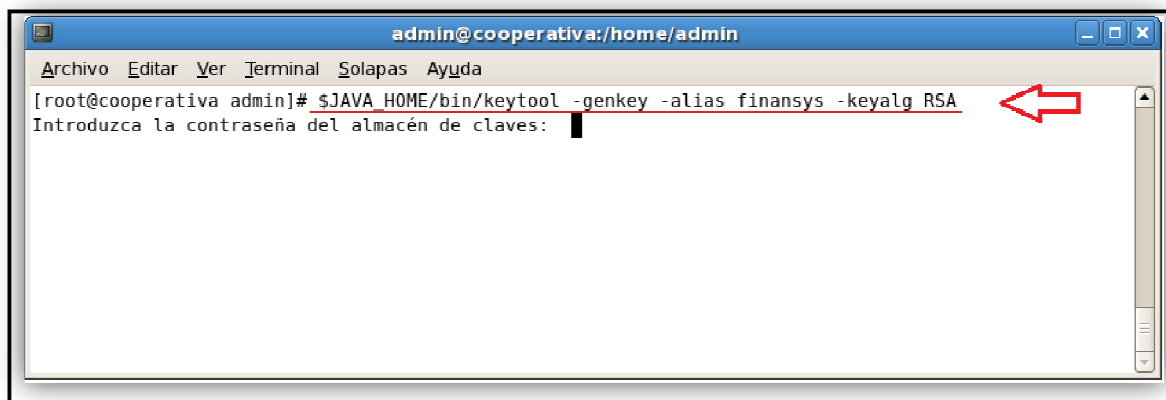


Figura 124: Generación de la clave.

Fuente: Propia

Luego se debe ingresar una contraseña, se debe completar unos datos que le va solicitando:

```

admin@cooperativa:/home/admin
Archivo Editar Ver Terminal Solapas Ayuda

[root@cooperativa admin]# $JAVA_HOME/bin/keytool -genkey -alias finansys -keyalg RSA
Introduzca la contraseña del almacén de claves:
¿Cuáles son su nombre y su apellido?
[Unknown]: Bairon Gudino
¿Cuál es el nombre de su unidad de organización?
[Unknown]: FINANSYS
¿Cuál es el nombre de su organización?
[Unknown]: FINANSYS
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: IBARRA
¿Cuál es el nombre de su estado o provincia?
[Unknown]: IMBABURA
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: EC
¿Es correcto CN=Bairon Gudino, OU=FINANSYS, O=FINANSYS, L=IBARRA, ST=IMBABURA, C=EC?
[no]: si

Introduzca la contraseña de clave para <finansys>
(INTRO si es la misma contraseña que la del almacén de claves): _____
Volver a escribir la contraseña nueva: _____
[root@cooperativa admin]#

```

Figura 125: Parámetros de la clave.
Fuente: Propia

Para terminar la configuración del SSL, se procede a configurar el servidor apache tomcat. El Archivo server.xml ubicado en la carpeta /opt/apache-tomcat-7.0.22/conf, debe quedar de la siguiente forma:

```

admin@cooperativa:/home/admin
Archivo Editar Ver Terminal Solapas Ayuda

-->
<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation -->

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="${user.home}/.keystore"
keystorePass="changeit"/>

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

-- INSERT --

```

Descomentar y configurar ésta sección de la siguiente forma

Figura 126: Configuración del server.xml
Fuente: Propia

Sistema Integrado de Información Financiera

FINANSYS



Conclusiones y Recomendaciones

- 1) Conclusiones
- 2) Recomendaciones

Capítulo 5: CONCLUSIONES Y RECOMENDACIONES

5. Conclusiones y Recomendaciones

5.1. Conclusiones

- 1) El desarrollo de estos módulos permitió alcanzar los objetivos planteados.
- 2) Se creó e implementó los módulos descritos en este proyecto, con los cuales se pudo contar con un sistema integrado de calidad para la administración financiera de la Cooperativa Unión Cochapamba.
- 3) Todas las instituciones financieras deben realizar sus procesos en un sistema informático integrado el cual permita realizar transacciones en forma automática, rápida y eficiente.
- 4) La implementación del módulo de Administración permitió definir la funcionalidad y la agrupación de los usuarios en perfiles asignando las funcionalidades específicas para cada usuario del sistema.
- 5) RUP es un proceso de Ingeniería de Software que proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en el transcurso de desarrollo de un proyecto de software.
- 6) El Software Libre aporta de muchas formas al desarrollo y crecimiento de las pequeñas y medianas empresas, proporcionando programas de excelente calidad que satisfacen las necesidades de la mayoría de los usuarios, haciendo de ésta una estrategia para poder competir en esta carrera tecnológica frente al software propietario.

5.2.Recomendaciones

- 1) Es importante tener en cuenta el funcionamiento de cada módulo y la relación que existe entre ellos para el desarrollo de un sistema integrado ya que de esto depende que se obtenga un producto de calidad.
- 2) En todo sistema o aplicación es recomendable la implementación de un módulo de administración el cual nos permita mantener una correcta gestión de cuentas y roles de usuario para mantener un nivel de privacidad en la información.
- 3) Se recomienda utilizar la metodología RUP en proyectos de Desarrollo de Software para mantener la información ordenada y documentada durante todo el proceso de desarrollo e implementación, utilizando las plantillas que proporciona RUP, ya que facilitan y garantizan el producto software.
- 4) Al ver el funcionamiento y resultado obtenido después de haber implementado estos módulos se recomienda utilizar software libre para el desarrollo y la implementación de aplicaciones o sistemas web, ya que el software libre mantiene las características del software comercial.

**Sistema Integrado de Información Financiera
FINANSYS**



GLOSARIO

6.1.Introducción.

El presente glosario forma parte del sistema “FINANSYS” y es una recopilación de términos empleados en todo el sistema.

Este glosario no es un diccionario informático; representa una guía general de las definiciones y conceptos que se usan con frecuencia y tienen una relevante importancia para la adecuada comprensión dentro del sistema.

6.2.Propósito

La función principal del presente glosario es proporcionar un estándar de términos y conceptos asociadas a los términos técnicos del sistema.

6.3.Referencias.

Éste documento hace referencia a términos empleados en toda la documentación del sistema.

6.4.Glosario

Actions.- Métodos de capa de vista, para el uso de componentes en páginas jsp.

AJAX (Asynchronous JavaScript and XML). -Es una técnica de desarrollo web para crear aplicaciones interactivas

API.- Interfaz de Programación de Aplicaciones (**Application Programming Interface**) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Artefacto.- Un artefacto es un producto tangible resultante del proceso de desarrollo de software. En ocasiones un artefacto puede referirse a un producto terminado, pero más habitualmente se refiere a la documentación generada a lo largo del desarrollo del producto.

Beans.- Son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

Buteo Boot.- Es la secuencia de arranque, es el proceso que inicia el sistema operativo cuando el usuario enciende una computadora. Se encarga de la inicialización del sistema y de los dispositivos.

ByteCodes.- Es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable.

Casos de Uso.- Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

CSS.- El nombre hojas de estilo en cascada viene del inglés **Cascading Style Sheets**.

CRUD.- Es el acrónimo de crear, obtener, actualizar y eliminar, es decir las funciones básicas en bases de datos o en la capa de persistencia de un sistema de software.

DTD.- Significa en inglés “Document Type Definition” y es una descripción de estructura y sintaxis de un documento XML

FINANSYS.- Nombre del sistema financiero desarrollado.

Framework.- Es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

GNU/Linux.- GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU.

HTML.- (Siglas de **H**yper **T**ext **M**arkup **L**anguage) Lenguaje de marcado de hipertexto.

Jdeveloper.- Es un entorno de desarrollo integrado desarrollado por Oracle.

JSF.- Es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

Live CD.- Es un sistema operativo almacenado en un medio extraíble, que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora.

Mapping o Mapeo.- Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

On-the-fly.- Sobre marcha, en proceso

Open Source.- Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente.

ORM.- Es el Mapeo Objeto-Relacional (**Object-Relational Mapping**) es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

POO.- Programación Orientada a Objetos.

RichFaces.- Framework de código abierto que añade la capacidad de ajax en aplicaciones JSF sin recurrir a JavaScript.

Rup.- El Proceso Unificado de Rational (**Rational Unified Process**) es un proceso de desarrollo de software y junto con el UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SELinux.- Es una característica de seguridad de Linux que provee una variedad de políticas de seguridad.

Servlets.- Los servlets, son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

skin.- Mascara, interfaz visual de distintas características.

Skinnability.- Cambiar de skin.

SMB.- Server Message Block es un Protocolo de red que permite compartir archivos e impresoras entre nodos de una red.

Tag.- Etiqueta. Las etiquetas le dicen al browser las instrucciones y características necesarias para presentar la página en la pantalla.

UML.- Lenguaje Unificado de Modelado (**Unified Modeling Language**). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

Visio.- Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más.

Vista de Deployment.- es un tipo de diagrama UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

YUM.- Es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software.

Sistema Integrado de Información Financiera FINANSYS



REFERENCIAS

7. Referencias

7.1.Leyes para Sistemas Financieros

- a) Super Intendencia de Bancos del Ecuador. (29 de Diciembre de 2009). *Ley de Cooperativas*. Recuperado el 2011, de http://www.sbs.gob.ec/medios/PORTALDOCS/downloads/normativa/decreto_194_cooperativas_29_dic_09.pdf
- b) LA COMISIÓN DE LEGISLACIÓN. (2012). *LEY GENERAL DE INSTITUCIONES DEL SISTEMA FINANCIERO*. Obtenido de 05
- c) Ley general de instituciones del sistema financiero. (1 de Enero de 2010). Quito, Pichincha, Ecuador.
- d) Normas de Basilea, A. S. (2004). *Normas de Basilea*. Recuperado el 2010, de LA NORMATIVA DE BASILEA Y SU PAPEL DENTRO DE LAS NORMAS FINANCIERAS DEL ECUADOR: www.dspace.espol.edu.ec/bitstream

7.2.Sistemas Operativos

- a) Anónimo. (2012). *Sistemas Operativos*. Recuperado el Enero de 2012, de <http://www.masadelante.com/faqs/sistema-operativo>
- b) Centos. (2012). *CentOS Community Enterprise Operating System*. Recuperado el Febrero de 2012, de <http://wiki.centos.org/es>
- c) CentOS. (06 de Marzo de 2008). *DVD Linux Magazine*. Recuperado el 15 de Mayo de 2010, de <http://www.linux-magazine.es/issue/16/DVD.pdf>
- d) Fedora. (01 de Enero de 2012). *Fedora, Libertad, Amistad y Características*. Recuperado el 2012, de <http://fedoraproject.org/es/features/>
- e) Oliva, J. (Marzo de 2009). *CentOS-es*. Recuperado el 2010, de Marzo: <http://lists.centos.org/pipermail/centos-es/2007-March/000317.html>

7.3.Seguridad Web

- a) Wikilearning, D. S. (18 de Octubre de 2005). *wikilearning*. Recuperado el 2011, de http://www.wikilearning.com/curso_gratis/protocolos_seguros_para_el_web/6091-4

7.4.Bases de Datos

- a) Anónimo, B. d. (2009). *El rincon del vago*. Recuperado el 2011, de <http://html.rincondelvago.com/concepto-de-base-de-datos.html>
- b) Peña, I. (2011 de Julio de 2011). *XperimentoS*. Obtenido de <http://www.xperimentos.com/>
- c) PostgreSQL. (02 de Enero de 2012). *Introducción a Postgres*. Recuperado el 2012, de http://www.postgresql.org.pe/articulos/introduccion_a_postgresql.pdf

7.5.Herramientas de Desarrollo

- a) affiliates, O. C. (2012). *Bienvenido a NetBeans y www.netbeans.org*. Recuperado el 2012, de http://netbeans.org/index_es.html
- b) Anónimo. (2 de Septiembre de 2011). *Dos Ideas*. Recuperado el 2011, de Arquitectura de tres Capas: <http://www.dosideas.com/cursos/mod/resource/view.php?id=10>
- c) Anónimo. (s.f.). *Sicuma*. Obtenido de Tutoriales de JSF: <http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>
- d) Apache POI Project. (26 de Febrero de 2012). *Apache POI*. Obtenido de <http://jakarta.apache.org/poi/>
- e) Bruno Lowagie, P. S. (2010-2011). *iText Software BVBA* . Obtenido de <http://itextpdf.com/>
- f) Chirita, T. D. (2007). *The Definitive Guide to Jasper Reports*. Apress.
- g) Commons Beanutils. (24 de Marzo de 2010). *Apache Commons*. Recuperado el 2011, de <http://commons.apache.org/beanutils/>
- h) Commons Collections. (15 de Abril de 2008). *Apache Commons*. Recuperado el 2011, de <http://jakarta.apache.org/commons/collections/>
- i) Commons Digester. (13 de Diciembre de 2011). *Apache Commons*. Recuperado el 2011, de <http://commons.apache.org/digester/>
- j) Commons Logging. (5 de Marzo de 2008). *Apache Commons*. Recuperado el 2011, de <http://commons.apache.org/logging/>
- k) Domínguez-Dorado, M. (2005). *NetBeans IDE* . Madrid: Iberprensa.

- d)** Java, Wikipedia. (07 de Febrero de 2012). *Java (lenguaje de programación)*. Recuperado el 2012, de http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29
- m)** JBoss.org. (2011). *RichFaces Developer Guide*. Obtenido de http://docs.jboss.org/richfaces/latest_3_3_X/en/devguide/html/Introduction.html
- n)** Roman, J. (20 de Noviembre de 2007). *Jasper Report Open Source*. Recuperado el 2010, de <http://jsanroman.net/2007/11/%C2%BFque-es-jasper-reports-2/>
- o)** Sekler, G. Z. (2012). *JSp, JSF and Tomcat Web Development*. Apress.
- p)** Shaikovsky, M. K. (2012). *Practical RichFaces*. Apress.
- q)** Wikimedia Proyect. (28 de Mayo de 2009). *Wikipedia*. Recuperado el 2011, de <http://es.wikipedia.org/wiki/JavaEE>
- r)** Wikipedia. (01 de Enero de 2012). *Wikipedia - Tomcat*. Recuperado el 2012, de <http://es.wikipedia.org/wiki/Tomcat>
- s)** Birnam, S. *Java Distribuido*. McGrawHill.
- t)** Mendoza, I. M. *php con PostgreSQL 8*. Grupo Editorial Megabyte.
- u)** Paul KIMMEL, T. J. *Manual de UML*. McGrawHill.
- v)** Richard Monson-Haefel, O. R. *Enterprise JavaBeans 2da Edición*.
- w)** Rueda, F. *Sistemas Operativos*. McGRAW-HILL.
- x)** Smith, T. N. *JavaScript - Quinta Edicion*.

Sistema Integrado de Información Financiera FINANSYS



ANEXOS

8. ANEXOS

A continuación se muestra la lista de anexos que estarán presentes en este documento y otros adicionales en el CD adjunto.

- **Anexos al presente documento.**

 - Anexo A:** Anteproyecto

 - Anexo B:** Manual de Usuario

 - Anexo C:** Diagrama general de la base de datos Finansys

 - Anexo D:** Diagrama general de casos de uso del sistema Finansys

- **Anexos en el CD**

 - Anexo A: Manual Técnico**

 - Anexo B: Artículo Científico**

 - Anexo C: Entregables de RUP**

 - C1:** Actas de trabajo

 - C2:** Casos de uso del módulo de Administración y Seguridad Captaciones

 - C3:** Casos de uso del módulo de Captaciones

 - C4:** Casos de uso del módulo de Ahorros

MANUAL DE USUARIO

1. Introducción

El presente manual ha sido elaborado para permitir el uso apropiado del Sistema integrado de información financiera en lo que se refiere a los módulos de Ahorros, Captaciones y Administración y Seguridad. Se presentan las pantallas correspondientes a cada menú, acompañado de la respectiva aclaración de sus funciones.

2. Ingreso al Sistema

Para iniciar es necesario el explorador de internet llamado Mozilla Firefox debido a que es uno de los mejores navegadores que tiene soporte para componentes JSF y java. Al iniciar el sistema será re direccionado al inicio de sesión como muestra la figura siguiente:



Figura 1. Pantalla de inicio de Sesión

En la pantalla de autenticación solicitará ingresar el nombre de usuario, la contraseña, seleccionar la sucursal donde iniciara la sesión y pulsar el botón ingresar para ingresar a la pantalla principal del Sistema Integrado de Información Financiera (FINANSYS), debido a que es un sistema integrado en la pantalla principal del sistema se muestran todos los módulos que componen al sistema como se muestra en la Figura 2.



Figura 2. Pantalla principal del Sistema

3. Manejo del Módulo de Captaciones (Caja)

Las opciones que se presentan en el menú principal del modulo de Captaciones se muestran en la figura 3 las cuales son las siguientes:

- 1.** Apertura de Cuenta
- 2.** Depósitos de Ahorros
- 3.** Retiros de Ahorros
- 4.** Retiros con Cheque
- 5.** Pagos de Prestamos
- 6.** Otros Ingresos
- 7.** Depósitos a Plazo Fijo
- 8.** Apertura de Caja
- 9.** Arqueo de Caja
- 10.** Consultas
 - a)** Depósitos de Ahorros
 - b)** Pagos de Prestamos
 - c)** Otros ingresos

d) Códigos de Socios



Figura 3. Opciones del Menú de Captaciones (Caja)

3.1 Apertura de Cuentas

Al seleccionar la opción apertura de cuentas se mostrará una pantalla como la que se muestra en la figura 4 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) El usuario selecciona al cliente ingresado en la fecha actual.
- 2) El usuario ingresa los valores para ahorros, certificados de aportación, cuota de ingreso y el número de recibo.
- 3) Hacer click en el botón “Abrir Cuenta”.
- 4) Hacer click en el botón “Limpiar” para borrar los datos de la pantalla.
- 5) Seleccionar si o no para imprimir el comprobante de la transacción.

APERTURA DE CUENTA

Guardar Comprobante

Codigo de Socio: (*) 1029

Sucursal: lbarra

Ahorros: (*) 500.0

Ahr. Cert. Aportación: (*) 20.0

Cuota de Ingreso: (*) 5.0

=====
 Total Apertura: 525.0

Nro de Recibo: (*) 365

Figura 4. Pantalla de Apertura de Cuenta

3.2 Depósitos de Ahorro

Al seleccionar la opción Depósito de Ahorros se mostrará una pantalla como la que se muestra en la figura 5 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la cuenta de ahorros del cliente a realizar la transacción.
- 2) Ingresar los valores de depósitos en efectivo o en cheque y el número de recibo.
- 3) Seleccionar si la transacción se registrará en la libreta de ahorros o no.
- 4) El usuario hace clic en el botón “Depositar”.
- 5) Hacer clic en el botón “Limpiar” para borrar los datos de la pantalla.
- 6) Seleccionar si o no para imprimir el comprobante de la transacción.

DEPÓSITOS DE AHORROS

Realizar Deposito

Numero de Cuenta: (*) 101001029

Codigo Socio: 1029

Cliente: PICUASI DUQUE
EDGAR DANIEL

Producto: Cuenta de Ahorros

Oficina: lbarra

Saldo: 500.0

Efectivo: (*) 200

Cheques: (*) 0.0

=====
 Total Depósito: 200.0

Nro de Recibo: (*) 854

Con Libreta: Si No

Figura 5. Pantalla de Depósito de Ahorros

3.3 Retiro de Ahorros

Al seleccionar la opción Retiro de Ahorros se mostrará una pantalla como la que se muestra en la figura 6 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la cuenta de ahorros del cliente a realizar la transacción.
- 2) Ingresar el valor de retiros en efectivo y el número de recibo emitido.
- 3) Seleccionar si la transacción se registrará en la libreta de ahorros o no.
- 4) El usuario hace clic en el botón “Retirar”.
- 5) Hacer click en el botón “Limpiar” para borrar los datos de la pantalla.
- 6) Seleccionar si o no para imprimir el comprobante de la transacción.

RETIRO DE AHORROS EN EFECTIVO

Numero de Cuenta: (*) 101001029

Cedula Cliente: 1002954988

Nombres Cliente: PICUASI DUQUE

Producto Cliente: Cuenta de Ahorros

Oficina Retiro: Ibarra

Saldo Disponible: 700.0

Efectivo a Retirar: (*) 200

Nro de Recibo: (*) 845

Con Libreta: Si No

Figura 6. Pantalla de Retiro de Ahorros

3.4 Retiro con Cheque

Al seleccionar la opción Retiro con Cheque se mostrará una pantalla como la que se muestra en la figura 7 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la cuenta de ahorros del cliente a realizar la transacción.
- 2) Ingresar el valor de retiros en efectivo y el número de recibo emitido.
- 3) Seleccionar si la transacción se registrará en la libreta de ahorros o no.
- 4) Seleccionar el banco del cual se realizará el retiro, ingresa el número de cheque emitido y el beneficiario.

- 5) El usuario hace clic en el botón “Retirar”.
- 6) Seleccionar si o no para imprimir el comprobante de la transacción.

RETIRO DE AHORROS CON CHEQUE

Numero de Cuenta: (*) 101001029

Cedula Cliente: 1002954988

Nombres Cliente: PICUASI DUQUE
EDGAR DANIEL

Producto Cliente: Cuenta de Ahorros

Oficina Retiro: Ibarra

Saldo Disponible: 700.0

Valor a Retirar: (*) 500

Nro de Recibo: (*) 78

Con Libreta: Si No

Seleccione un Banco: (*) 1

Beneficiario: (*) Esteban Duque

Nro de Cheque: (*) 2566
Banco del Pichincha
1314465-4

Figura 7. Pantalla de Retiro con Cheque

3.5 Pago de Préstamos

Al seleccionar la opción Pago de Préstamos se mostrará una pantalla como la que se muestra en la figura 8 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar el préstamo a pagar y el sistema carga los datos de la cuota a cancelar.
- 2) Ingresar el valor a pagar del préstamo.
- 3) El usuario hace clic en el botón “Pagar”.
- 4) Hacer click en el botón “Limpiar” para borrar los datos de la pantalla.

Figura 8. Pantalla de Pago de Préstamos

3.6 Otros Ingresos

Al seleccionar la opción Otros Ingresos se mostrará una pantalla como la que se muestra en la figura 9 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar el número del cliente o socio del cual se va a realizar la transacción.
- 2) Ingresar la cantidad del ingreso, el número de recibo emitido y la descripción.
- 3) Seleccionar el concepto del ingreso.
- 4) Hacer click en el botón “Ingresar”.
- 5) Hacer click en el botón “Limpiar” para borrar los datos de la pantalla.

Depositar Limpiar

REGISTRO DE OTROS INGRESOS Realizar Deposito

Código de Socio (*) 1029

Cedula Cliente: 1002954988

Nombres Cliente: PICUASI DUQUE
EDGAR DANIEL

Oficina Retiro: Ibarra

Valor Transacción (*) 20

Nro de Recibo (*) 452

Descripción (*) Compra de Formularios de Préstamos

Seleccione una Opción (*) 1

- 1 CUOTA DE INGRESO 5.6.90.03
- 2 MOVILIZACION PERSONAL 5.1.02.05

Foto del Cliente

Figura 9. Pantalla de Registro de Otros Ingresos

3.7 Depósitos a Plazo Fijo

Al seleccionar la opción Depósitos a Plazo Fijo se mostrará una pantalla como la que se muestra en la figura 10 donde el sistema mostrara la lista de DPF:

Nuevo DPF

Depósitos a Plazo Fijo

| Pagar DPF | Nombre Socio | Nro Depósito | Fecha de Depósito | Fecha de Vencimiento | Valor | Estado | Ver Datos |
|-----------|------------------------|--------------|-------------------|----------------------|----------|---------|-----------|
| | JULIAN COLIMBA SANCHEZ | 2 | 2012-01-28 | 2012-02-28 | 34500,00 | Vigente | |
| | MARISOL SIMBAÑA IRALDA | 1 | 2012-01-28 | 2012-02-28 | 7700,00 | Vigente | |

Figura 10. Lista de Depósitos a Plazo Fijo

- 1) Para ingresar un nuevo DPF seleccionar el botón nuevo DPF y se mostrara la pantalla como se muestra en la figura 11 donde se ingresará la información de la siguiente manera.
- 2) Para cancelar un DPF el usuario debe ingresar los datos requeridos como la forma de pago y los campos obligatorios.

- 3) Proceder a ingresar la información en cada uno de los campos.
- 4) Hacer clic en el botón “Realizar Depósito” cuando ya se hayan ingresado datos.

| Datos del Deposito a Plazo Fijo | |
|---------------------------------|------------------------|
| N° de Depósito: | 2 |
| N° de Socio: | 104 |
| Cédula: | 1000953052 |
| Nombres: | JULIAN COLIMBA SANCHEZ |
| Fecha Depósito: | ene 28, 2012 |
| Plazo en Días: | 365 |
| Fecha Vencimiento: | feb 28, 2012 |
| Tasa de interes anual %: | 15 |
| Tasa Impuesto Retenido %: | 0 |
| Interes Generado: | 4500,00 |
| Impuesto a Retener: | 0,00 |
| Interes a Recibir: | 4500,00 |
| Beneficiario por Fallecimiento: | Henry Leon |
| Depósito en Efectivo: | 10000.0 |
| Depósito con Cheque: | 20000.0 |
| ===== | |
| Total Depósito: | 30000.0 |
| Retiro Efectivo: | 0.0 |
| Retiro Cheque: | 0.0 |
| Total Retiro: | 0.0 |
| Valor Renovado: | 0.0 |
| Valor Agregado: | 0.0 |
| Operador que Ingresó: | bairon |
| Operador que Renovó: | bairon |
| Fecha de Retiro: | ene 28, 2012 |

Figura 11. Pantalla de Ingreso de Depósito a Plazo Fijo

3.8 Apertura de Caja

Al seleccionar la opción Apertura de Caja se mostrará una pantalla como la que se muestra en la figura 12 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la fecha actual para la apertura de caja.
- 2) Ingresar el valor de la Apertura de Caja del día.
- 3) El usuario hace clic en el botón “Abrir Caja”.
- 4) Buscar aperturas de Caja de fechas anteriores.

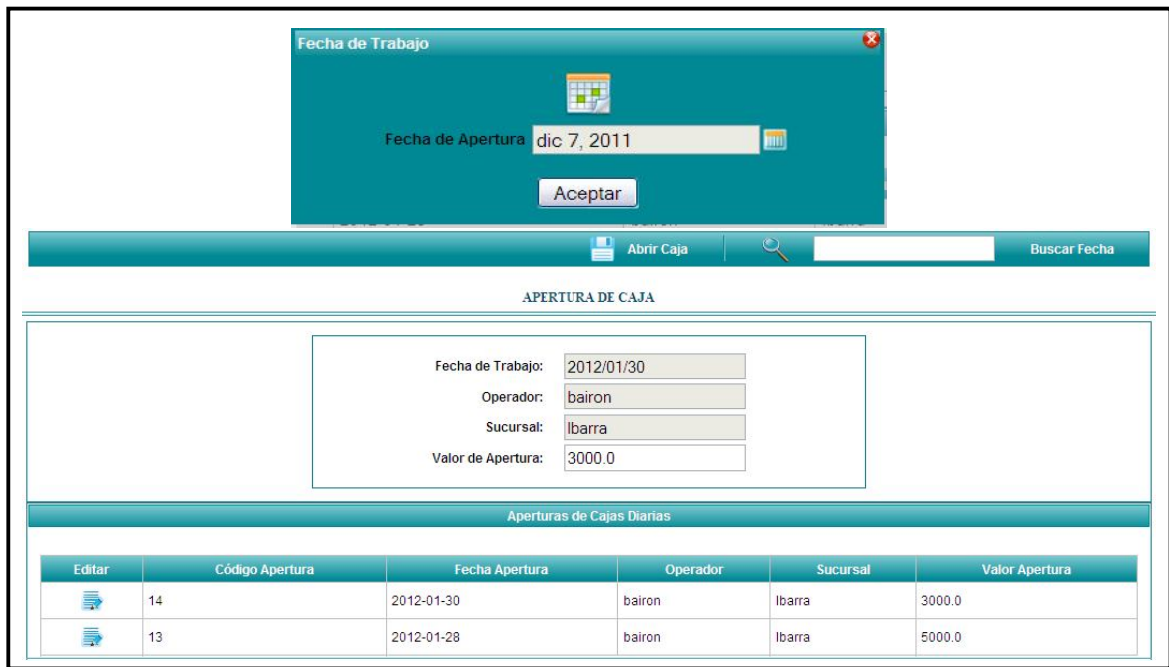
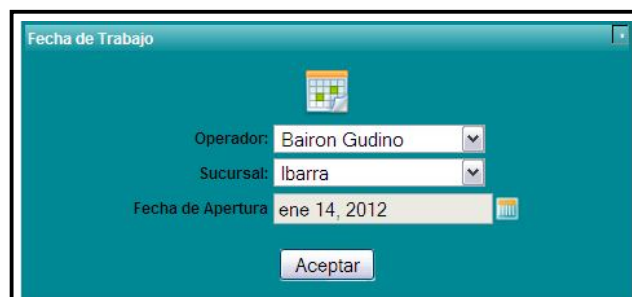


Figura 12. Pantalla de Apertura de Caja

3.9 Arqueo de Caja

Al seleccionar la opción Arqueo de Caja se mostrará una pantalla como la que se muestra en la figura 13 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la fecha, nombre y la sucursal donde se realizaron las transacciones.
- 2) Hacer click en el botón “Aceptar”
- 3) Hacer click en el botón “Imprimir” para ver detalladamente el arqueo de caja.



The screenshot shows a software interface titled 'ARQUEO DE CAJA'. At the top right, there is a 'Imprimir' button. The main content is a table with two columns: 'Tipo de Transacción' and 'Valor Total'. The table lists various financial transactions and their corresponding values.

| Tipo de Transacción | Valor Total |
|----------------------------------|-------------|
| Valor de Apertura de Caja: | 3000,00 |
| Depósito. de Ahorro en Efectivo: | 720,00 |
| Depósito. de Ahorro con Cheques: | 0,00 |
| Retiro. de Ahorro en Efectivo: | 0,00 |
| Retiro. de Ahorro con Cheques: | 0,00 |
| Pagos de Prestamos: | 0,00 |
| Registro Otros Ingresos: | 5,00 |
| DPF. Depositos en Efectivo: | 0,00 |
| DPF. Depositos con Cheque: | 0,00 |
| DPF. Retiros en Efectivo: | 0,00 |
| DPF. Retiros con Cheque: | 0,00 |
| Valor Final de la Caja: | 3725,00 |

Figura 13. Pantalla de Arqueo de Caja

3.9 Consultas

Al seleccionar la opción Consultas se mostrará un sub menú donde se podrá elegir cualquier opción de estas:

1. Depósitos de Ahorros
2. Pagos de Prestamos
3. Otros ingresos
4. Códigos de Socio

Estas consultas son generadas de transacciones realizadas durante el día actual, las cuales indican detalladamente los movimientos realizados.

4. Manejo del Módulo de Ahorros

Las opciones que se presentan en el menú principal del modulo de Ahorros se muestran en la figura 14 las cuales son las siguientes:

- 1) Cuentas de Ahorro
- 2) Otros Ingresos
- 3) Débitos/Créditos
- 4) Bloqueos/Desbloques
- 5) Plazos Fijos
- 6) Reportes



Figura 14. Figura de Opciones del Menú de Ahorros

4.1 Cuentas de Ahorro

Al seleccionar la opción Cuentas de Ahorro se mostrará una pantalla como la que se muestra en la figura A15 donde el sistema mostrara la lista de clientes donde se podrá realizar las siguientes acciones:

| LISTA DE CLIENTES | | | | | |
|-------------------|----------------|-----------------|-----------------|----------------|-----------------|
| Código Cliente | Cedula Cliente | Nombres | Apellidos | Fecha Registro | Productos Socio |
| 1029 | 1002954988 | EDGAR DANIEL | PICUASI DUQUE | 2012-01-23 | |
| 1028 | 1002299566 | BAIRON PATRICIO | GUDINO VILLALBA | 2012-01-22 | |
| 1027 | 1000953669 | MARIA NIEVES | JUMA URCUANGO | 2005-02-18 | |
| 1026 | 1000882694 | JOSE MANUEL | JUMA URCUANGO | 2005-02-18 | |

Figura 15. Figura de Lista de Clientes

- 1) Seleccionar la opción buscar para encontrar un cliente específico.
- 2) Seleccionar el botón Productos Socio de un cliente específico de la lista, y se visualizará la lista de producto que tiene el cliente seleccionado como se muestra en la figura 16.

| Código Cliente | Cedula Cliente | Nombres | Apellidos | Fecha Registro | Productos Socio |
|----------------|----------------|--------------|--------------|----------------|-----------------|
| 740 | 1002469128 | LUIS ALFREDO | COLCHA CAIZA | 2011-11-21 | |

| Lista de Productos | | | | | | | | | |
|--------------------|----------------------------|---------------|--------------|-------------|-----------|------------|---------|---------|-------------|
| N° Cuenta | Producto | Estado Cuenta | Saldo Total | Disponibles | Encajes | Bloqueados | Cheques | Interes | Movimientos |
| 102001028 | Certificados de Aportacion | Activa | 20,00 \$ | 20,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | |
| 101001028 | Cuenta de Ahorros | Activa | 10.000,00 \$ | 9.500,00 \$ | 500,00 \$ | 0,00 \$ | 0,00 \$ | 0,00 \$ | |

Figura 16. Figura de Lista de Productos de un Cliente

- 3) Seleccionar el botón Movimientos de un producto específico de la lista, y se visualizará la lista de movimientos pertenecientes a la cuenta o producto seleccionado como se muestra en la figura 17.

| Movimientos de la Cuenta Seleccionada | | | | | | | | | |
|---------------------------------------|--------------|-------------------|----------------------------|---------------------------------|----------------------------------|----------------|--------------|--------------|----------|
| Area de Búsquedas | | | | | | | | | |
| | | | Buscar por Código: | Buscar por Tipo de Transacción: | Buscar por Fecha de Transacción: | | | | |
| | | | <input type="text"/> | <input type="text"/> | <input type="text"/> | | | | |
| N° Movimiento | N° de Cuenta | Producto | Tipo Transacción | Fecha Transacción | Valor Transacción | Valor Efectivo | Valor Cheque | Valor Actual | Operador |
| 5 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 10.000,00 \$ | bairon |
| 4 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 9.000,00 \$ | bairon |
| 3 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 1.000,00 \$ | 1.000,00 \$ | 0,00 \$ | 8.000,00 \$ | bairon |
| 2 | 101001028 | Cuenta de Ahorros | DEPOSITOS EN CTAS. AHORROS | 2012-01-22 | 2.000,00 \$ | 2.000,00 \$ | 0,00 \$ | 7.000,00 \$ | bairon |
| 1 | 101001028 | Cuenta de Ahorros | APERTURA DE CUENTA | 2012-01-22 | 5.000,00 \$ | 5.000,00 \$ | 0,00 \$ | 5.000,00 \$ | bairon |

Figura 17. Figura de Lista de Movimientos de un Producto

A4.2 Otros Ingresos

Al seleccionar la opción Otros Ingresos se mostrará una pantalla como la que se muestra en la figura 18 donde el sistema mostrara la lista de clientes que han realizado un ingreso en la cooperativa y se podrá realizar las siguientes acciones:

- 1) Realiza búsquedas de ingresos por código, sucursal. Conceptos o fechas.

| Código | Sucursal | Concepto | Fecha | N° Recibo | Valor | Socio | NRTR |
|--------|----------|------------------|------------|-----------|-------|-------|------|
| 1 | Ibarra | CUOTA DE INGRESO | 2012-01-22 | 123 | 5.0 | 1028 | 1 |

Figura 18. Figura de Otros Ingresos

4.3 Débitos/Créditos/Transferencias

Al seleccionar la opción Débitos/Créditos se mostrará una pantalla como la que se muestra en la figura 19 donde el sistema mostrara otras acciones como se muestra a continuación.

- 1) El usuario debe seleccionar la una de las siguientes transacciones.
 - Notas de Débito o Crédito por Cuenta
 - Nota de Crédito pos depósito en el Banco
 - Pago de Prestamos por Depósito en el Banco
 - Transferencia de Cuenta de Ahorros a Cuenta de Ahorros
 - Transferencia de Cuenta de Ahorros a Prestamos
 - Activar y Desactivar una Cuenta.
- 2) Ingresar los datos en todos los campos requeridos
- 3) Seleccionar cuentas y préstamos de socios o clientes
- 4) Seleccionar las causas que determinan una transacción
- 5) Hacer click en el los botones “Guardar” o “Pagar”
- 6) Seleccionar si o no si desea imprimir comprobante de transacción

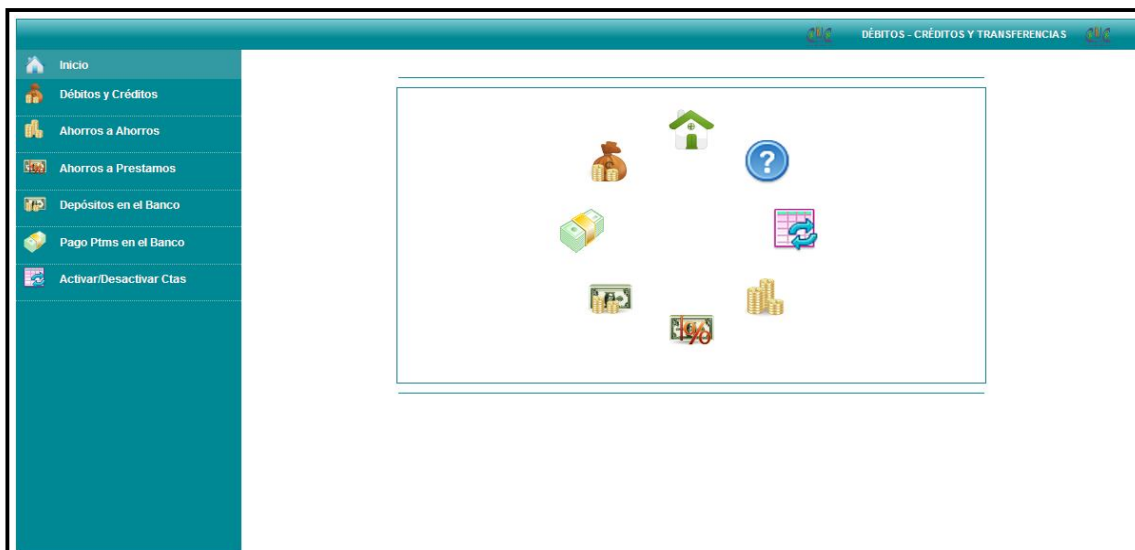


Figura 19. Figura de Débitos/Créditos/Transferencias

4.4 Bloqueos/Desbloques

Al seleccionar la opción Bloqueos/Desbloques se mostrará una pantalla como la que se muestra en la figura A20 donde el sistema solicitará que se ingrese los siguientes datos de la siguiente manera:

- 1) Seleccionar la opción bloqueos, desbloques o cheques.
- 2) Seleccionar la cuenta del cliente a bloquear el saldo.
- 3) Ingresar el valor a bloquear o desbloquear.
- 4) Click en el botón “Bloquear Saldo” o “Desbloquear Saldo” según sea el caso.
- 5) Click en el botón Limpiar para borrar la pantalla.

The screenshot shows a web application interface for blocking/unblocking funds. The title is "BLOQUEOS POR ENCAJES". The form contains the following fields and values:

| | |
|-----------------------|-----------|
| Número de Cuenta: (*) | 101001028 |
| Número de Cliente: | 1028 |
| Disponibles: =====> | 9500.0 |
| Encajes: =====> | 500.0 |
| Cheques: =====> | 0.0 |
| Bloqueados: =====> | 0.0 |
| Interes: =====> | 0.0 |

Client Information:
GUDINO VILLALBA BAIRON PATRICIO
Producto: Cuenta de Ahorros

Valor a Bloquear: (*) 20

Buttons: Bloquear Saldo, Limpiar

Figura 19. Figura de Débitos/Créditos/Transferencias

Diagrama general de la Base de Datos

Diagrama general de casos de uso