



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

TEMA:

“ANÁLISIS COMPARATIVO DE FRAMEWORK SOFTWARE LIBRE
PARA EL DESARROLLO DE APLICACIONES DE ESCRITORIO
EN JAVA”

TESIS DE GRADO

Previa a la obtención del título de:
INGENIEROS EN SISTEMAS INFORMÁTICOS

PRESENTADO POR:

JOSÉ LUIS MANZANO ZUMBA
ÁNGEL PATRICIO COBO RODRÍGUEZ

RIOBAMBA-ECUADOR

2014

AGRADECIMIENTO

Dejamos constancia de nuestros más sinceros y profundos agradecimientos:

A la Escuela Superior Politécnica de Chimborazo y sus docentes quienes nos brindaron todos sus conocimientos, necesarios para obtener una educación de calidad.

A cada uno de nuestros familiares quienes nos apoyaron incondicionalmente y nos ayudaron a culminar con éxito esta investigación.

A nuestro Director de Tesis el Ing. Raúl Rosero y asesora Ing. Landy Ruíz quienes con sus conocimientos y experiencias nos guiaron y prestaron todo su tiempo en el desarrollo de esta investigación.

DEDICATORIA

La presente tesis la dedico:

A mis padres Luis y Olga quienes me apoyaron incondicionalmente para obtener una profesión y en los buenos y malos momentos me han sabido dar su cariño, comprensión y apoyo.

A mis hermanos Patricia, Silvia, Ruth y Alex quienes han sido un pilar fundamental en la consecución de esta meta que sin duda formaran parte de mi vida profesional.

A mis cuñados Byron y Gonzalo y a mis sobrinas Doménica y Valentina quienes fueron fuente de inspiración para haber culminado con éxito mi carrera profesional.

José

El presente trabajo de tesis lo dedico:

A Carmen y Manuel mis padres, quienes son los pilares fundamentales de apoyo en mi vida.

A Oswaldo mi hermano, un ejemplo de trabajo y superación para mi.

A mi familia y amigos que durante este proceso para conseguir ésta meta siempre estuvieron presentes con sus valiosos consejos y apoyo.

Angel

FIRMAS RESPONSABLES Y NOTAS

NOMBRES	FIRMAS	FECHA
Ing. Iván Menes DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Ing. Jorge Huilca P. DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS
Ing. Raúl Rosero M. DIRECTOR DE TESIS
Ing. Landy Ruíz M. MIEMBRO DEL TRIBUNAL
DIR. DPTO. CENTRO DOCUMENTACIÓN
NOTA DE LA TESIS.....		

RESPONSABILIDAD DE LOS AUTORES

Nosotros, José Luis Manzano Zumba y Ángel Patricio Cobo Rodríguez, somos responsables de las ideas, doctrinas y los resultados expuestos en esta tesis, y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica De Chimborazo.

José Luis Manzano Zumba

Ángel Patricio Cobo Rodríguez

ÍNDICE DE ABREVIATURAS

- API:** Application Programming Interface
- AWT:** Abstract Widget Toolkit
- DAO:** Data Acces Object
- GPL:** General Public Licence
- GUI:** Graphic User Interface
- HTTP:** Protocolo de transporte de Hyper Texto
- IDE:** Entorno de desarrollo Integrado
- JAR:** Java Archive
- JDK:** Java Developed Kit
- JVM:** Java Virtual Machine
- JSON:** JavaScript Object Notation
- JPF:** Java Plugin Framework
- JSPF:** Java Simple Plugin Framework
- MVC:** Modelo Vista Controlador
- ORM:** Object Record Mapper
- OSGI:** Open Service Gateway Initiative
- POO:** Programación Orientada a Objetos
- RCP:** Rich Cliente Platform
- RMI:** Remote Method Invocation
- SISCOOP:** Sistema Informático de Servicios Cooperativos
- SWT:** Standard Widget Toolkit
- SOA:** Arquitectura Orientada a Servicios
- UML:** Unified Modeling Language

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS RESPONSABLES Y NOTAS

RESPONSABILIDAD DE LOS AUTORES

ABREVIATURAS

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

INTRODUCCIÓN

CAPÍTULO I	- 20 -
MARCO REFERENCIAL	- 20 -
1.1. Antecedentes	- 20 -
1.2. Justificación.....	- 22 -
1.3. Objetivos	- 26 -
1.3.1. Objetivo General	- 26 -
1.3.2. Objetivos Específicos	- 27 -
1.4. Hipótesis	- 27 -
CAPÍTULO II	- 28 -
MARCO TEÓRICO	- 28 -
2.1. Arquitectura Modular	- 28 -
2.1.1. Definición	- 28 -
2.1.2. Características	- 28 -
2.1.2.1. Módulo	- 28 -
2.1.2.2. Modularidad.....	- 29 -
2.1.2.3. Desarrollo Modular	- 29 -
2.1.3. Estándares existentes	- 30 -
2.1.3.1. OSGi	- 30 -
2.1.3.2. JPF (Java Plugin Framework).....	- 34 -

2.1.3.3. JSPF (Java Simple Plugin Framework)	- 35 -
2.1.4. Ventajas	- 35 -
2.1.5. Desventajas.....	- 36 -
2.2. Desarrollo de software usando frameworks.....	- 36 -
2.2.1. Definición	- 36 -
2.2.2. Arquitectura.....	- 37 -
2.2.2.1. Controlador.....	- 38 -
2.2.2.2. Modelo	- 38 -
2.2.2.3. Vista	- 38 -
2.2.3. Características	- 38 -
2.2.4. Tipos de frameworks	- 39 -
2.2.4.1. System infrastructure frameworks	- 39 -
2.2.4.2. Middleware integration frameworks	- 40 -
2.2.4.3. Enterprise application frameworks.....	- 40 -
2.2.4.4. White Box, Frameworks de Caja Blanca.....	- 40 -
2.2.4.5. Black Box, Frameworks de Caja Negra	- 42 -
2.2.4.6. Grey Box, Frameworks de Caja Gris	- 43 -
2.2.4.7. Visual Builders y Soporte de Lenguaje	- 43 -
2.2.5. Ventajas	- 44 -
2.2.6. Desventajas.....	- 44 -
2.3. Pluggins	- 45 -
2.3.1. Definición	- 45 -
2.3.2. Funcionalidades	- 45 -
2.4. Frameworks Software Libre para Aplicaciones de Escritorio en Java	- 45 -
2.4.1. Estudio previo de NetBeans (RCP)	- 46 -
2.4.1.1. Definición	- 46 -
2.4.1.2. Características	- 46 -
2.4.1.3. Ventajas	- 47 -
2.4.1.4. Desventajas.....	- 47 -
2.4.2. Estudio previo de Eclipse (RCP).....	- 47 -
2.4.2.1. Definición	- 47 -
2.4.2.2. Características	- 48 -
2.4.2.3. Ventajas	- 48 -
2.4.2.4. Desventajas.....	- 48 -

2.4.3. Estudio previo de GlueMvc.....	- 49 -
2.4.3.1. Definición	- 49 -
2.4.3.2. Características	- 49 -
2.4.3.3. Ventajas	- 49 -
2.4.3.4. Desventajas.....	- 50 -
2.4.4. Estudio previo de JMeter	- 50 -
2.4.4.1. Definición	- 50 -
2.4.4.2. Características	- 51 -
2.4.4.3. Ventajas	- 51 -
2.4.4.4. Desventajas.....	- 52 -
2.4.5. Estudio previo de OpenSwing	- 52 -
2.4.5.1. Definición	- 52 -
2.4.5.2. Características	- 53 -
2.4.5.3. Ventajas	- 53 -
2.4.5.4. Desventajas.....	- 54 -
2.5. Calidad Del Software.....	- 54 -
2.5.1. Introducción.....	- 54 -
2.5.2. Definición	- 54 -
2.5.3. La Norma ISO 9126.....	- 55 -
CAPÍTULO III	- 62 -
ANÁLISIS COMPARATIVO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES DE ESCRITORIO EN JAVA	- 62 -
3.1. Introducción.....	- 62 -
3.2. Determinación de los parámetros de preselección de los frameworks.....	- 62 -
3.2.1. Selección de los parámetros para la preselección de los frameworks	- 63 -
3.2.2. Valoración de los parámetros	- 63 -
3.3. Comparación de los frameworks	- 64 -
3.3.1. Producto.....	- 64 -
3.3.1.1. Análisis de los indicadores de Madurez del Producto	- 64 -
3.3.1.1.1. Tiempo en el mercado	- 64 -
3.3.1.1.2. Versiones del producto	- 65 -
3.3.1.1.3. Resultados	- 67 -
3.3.1.1.4. Interpretación	- 68 -
3.3.2. Facilidades	- 69 -

3.3.2.1. Análisis de los indicadores de facilidades para el desarrollo.....	- 69 -
3.3.2.1.1. Soporte con otras tecnologías y plugins	- 69 -
3.3.2.1.2. Documentación.....	- 70 -
3.3.2.1.3. Comunidad	- 71 -
3.3.2.1.4. Resultados	- 71 -
3.3.2.1.5. Interpretación	- 72 -
3.3.2.1.6. Resultados final del análisis realizado	- 73 -
3.3.3. Diagrama general con porcentajes de los resultados	- 74 -
3.3.3.1. Conclusiones de los resultados obtenidos.....	- 74 -
3.4. Estudio de frameworks seleccionados.....	- 75 -
3.4.1. Netbeans (RCP)	- 75 -
3.4.1.1. Introducción.....	- 75 -
3.4.1.2. Definición	- 76 -
3.4.1.3. Arquitectura	- 76 -
3.4.1.4. Características	- 78 -
3.4.2. Eclipse RCP	- 79 -
3.4.2.1. Introducción.....	- 79 -
3.4.2.2. Definición	- 79 -
3.4.2.3. Arquitectura	- 80 -
3.4.2.4. Características	- 81 -
3.5. Análisis comparativo de los frameworks seleccionados.....	- 82 -
3.5.1. Determinación de los parámetros de evaluación de los frameworks.....	- 82 -
3.5.2. Selección de características, subcaracterísticas y métricas para el análisis de los módulos de prueba.....	- 82 -
3.5.3. Determinación de la importancia de las características y subcaracterísticas a evaluar	- 84 -
3.5.4. Determinación de los módulos a probar	- 85 -
3.5.4.1. Módulo: Contabilidad	- 85 -
3.5.4.2. Descripción del módulo de prueba	- 85 -
3.5.4.3. Módulo Contabilidad.....	- 85 -
3.5.5. Análisis comparativo de los frameworks Netbeans (RCP) y Eclipse (RCP) .	- 86 -
3.5.6. Funcionalidad	- 87 -
3.5.6.1. Análisis de las subcaracterísticas de Funcionalidad	- 87 -
3.5.6.1.1. Adecuación.....	- 87 -
Interpretación de resultados	- 89 -

3.5.6.1.2. Exactitud.....	- 89 -
Interpretación de resultados	- 91 -
3.5.6.2. Confiabilidad.....	- 91 -
3.5.6.2.1. Madurez	- 91 -
Interpretación de resultados	- 93 -
3.5.6.2.2. Tolerancia ante fallos.....	- 93 -
Interpretación de resultados	- 95 -
3.5.6.2.3. Recuperabilidad.....	- 95 -
Interpretación de resultados	- 97 -
3.5.6.3. Usabilidad.....	- 97 -
3.5.6.3.1. Comprensibilidad.....	- 97 -
Interpretación de resultados	- 99 -
3.5.6.3.2. Operabilidad.....	- 99 -
Interpretación de resultados	- 101 -
3.5.6.4. Eficiencia.....	- 101 -
3.5.6.4.1. Rendimiento	- 101 -
Interpretación de resultados	- 103 -
3.5.6.4.2. Utilización de recursos.....	- 103 -
Interpretación de resultados	- 105 -
3.5.6.5. Tabulación de resultados.....	- 105 -
3.5.7. Diagrama general de resultados.....	- 107 -
3.5.8. Análisis de los resultados obtenidos.....	- 111 -
3.6. COMPROBACION DE LA HIPOTESIS.....	- 112 -
3.6.1. Hipótesis	- 112 -
3.6.2. Tipo de hipótesis	- 112 -
3.6.3. Determinación de variables	- 112 -
3.6.4. Operacionalización Conceptual	- 113 -
3.6.5. Operacionalización Metodológica	- 114 -
3.7. Conclusión	- 119 -
CAPÍTULO IV	- 120 -
IMPLEMENTACION DE LA APLICACIÓN	- 120 -
4.1. Introducción.....	- 120 -
4.2. Fase Inicial	- 121 -
4.2.1. Destinatarios	- 121 -

4.2.2. Definición del problema	- 121 -
4.2.3. Visión	- 121 -
4.2.4. Usuarios	- 123 -
4.2.5. Alcance	- 123 -
4.3. Descripción y características del producto.....	- 124 -
4.3.1. Descripción general del producto	- 124 -
4.3.2. Características del producto	- 124 -
4.4. Requerimientos de documentación	- 125 -
4.4.1. Manual de usuario.....	- 125 -
4.5. Herramientas.....	- 125 -
4.6. Requerimientos Funcionales	- 125 -
4.7. Requerimientos No Funcionales.....	- 126 -
4.8. Riesgos	- 127 -
4.8.1. Identificación del Riesgo.....	- 127 -
4.8.2. Análisis de Riesgos	- 128 -
4.8.3. Criterios de valoración de la exposición al riesgo	- 130 -
4.8.4. Planeación y Programación del Riesgo	- 132 -
4.9. Recursos Físicos.....	- 137 -
4.9.1. Hardware.....	- 137 -
4.9.2. Software	- 138 -
4.10. Diagrama Gantt y Presupuesto	- 139 -
4.11. Planificación	- 139 -
4.11.1. Diseño Conceptual	- 139 -
4.11.2. Especificación Funcional	- 139 -
4.11.3. Casos de Uso.....	- 139 -
4.11.4. Diseño Lógico.....	- 141 -
4.11.5. Diagramas de Secuencia.....	- 141 -
4.11.6. Diagramas de Clases	- 142 -
4.11.7. Diseño de Interfaces de Usuario.....	- 144 -
4.11.8. Diseño Físico.....	- 152 -
4.11.9. Modelo Físico de la Base de Datos	- 154 -
4.11.10. Diccionario de datos	- 155 -
4.11.11. Arquitectura del sistema	- 155 -
4.12. Entregables	- 156 -

4.12.1. Sprint 1.....	- 159 -
4.12.2. BurnDown Chart.....	- 161 -
4.13. Pruebas.....	- 161 -

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla II.I Ventajas de la especificación OSGi.	- 31 -
Tabla II.II Capas de OSGi.....	- 33 -
Tabla III.III Parámetros para la preselección de los frameworks	- 63 -
Tabla III.IV Valoración de los parámetros para los frameworks.....	- 63 -
Tabla III.V Tiempo en el mercado de los frameworks	- 65 -
Tabla III.VI Versiones de los frameworks	- 65 -
Tabla III.VII Resultado final del análisis Producto	- 67 -
Tabla III.VIII Soporte de tecnologías y plugins de los frameworks.....	- 69 -
Tabla III.IX Cantidad de documentación obtenida por los buscadores	- 70 -
Tabla III.X Cantidad de comunidades obtenida por los buscadores	- 71 -
Tabla III.XI Resultado del análisis Facilidades	- 71 -
Tabla III.XII Resultado final para la selección de los frameworks	- 73 -
Tabla III.XIII Selección de características, subcaracterísticas y métricas.....	- 83 -
Tabla III.XIV Importancia de las características y subcaracterísticas a evaluar	- 84 -
Tabla III.XV Equivalencia del peso	- 85 -
Tabla III.XVI Valores obtenidos para la métrica adecuación.....	- 88 -
Tabla III.XVII Valores obtenidos para la métrica exactitud.....	- 90 -
Tabla III.XVIII Valores obtenidos para la métrica madurez	- 92 -
Tabla III.XIX Valores obtenidos para la métrica tolerancia.....	- 94 -
Tabla III.XX Valores obtenidos para la métrica recuperabilidad	- 96 -
Tabla III.XXI Valores obtenidos para la métrica comprensibilidad	- 98 -
Tabla III.XXII Valores obtenidos para la métrica operabilidad	- 100 -
Tabla III.XXIII Valores obtenidos para la métrica rendimiento	- 102 -
Tabla III.XXIV Valores obtenidos para la métrica utilización de recursos	- 104 -
Tabla III.XXV Promedio de calidad para los módulos de prueba.....	- 106 -
Tabla III.XXVI Operacionalización Conceptual.....	- 113 -
Tabla III.XXVII Operacionalización Metodológica	- 114 -
Tabla III.XXVIII Promedio de calidad para los módulos de prueba.....	- 116 -

Tabla IV.XXXVII	Usuarios del sistema.....	- 123 -
Tabla IV.XXXVIII	Product Backlog.....	- 123 -
Tabla IV.XXXIX	Identificación del Riesgo.....	- 128 -
Tabla IV.XXX	Valoración del Riesgo.....	- 128 -
Tabla IV.XXXI	Probabilidad.....	- 129 -
Tabla IV.XXXII	Impacto del Riesgo.....	- 129 -
Tabla IV.XXXIII	Riesgo – Impacto.....	- 130 -
Tabla IV.XXXIV	Exposición al riesgo.....	- 130 -
Tabla IV.XXXV	Impacto – Probabilidad.....	- 131 -
Tabla IV.XXXVI	Resumen del Riesgo.....	- 131 -
Tabla IV.XXXVII	Prioridades del Riesgo.....	- 132 -
Tabla IV.XXXVIII	Gestión del riesgo 1.....	- 133 -
Tabla IV.XXXIX	Gestión del riesgo 2.....	- 134 -
Tabla IV.XL	Gestión del riesgo 3.....	- 135 -
Tabla IV.XLI	Gestión del riesgo 4.....	- 136 -
Tabla IV.XLII	Gestión del riesgo 5.....	- 137 -
Tabla IV.XLIII	Hardware utilizado.....	- 138 -
Tabla IV.XLIV	Software utilizado.....	- 138 -
Tabla IV.XLV	Caso de uso esencial de transacciones de caja.....	- 140 -
Tabla IV.XLVI	Sprint_Backlog.....	- 156 -
Tabla IV.XLVII	Sprint 1.....	- 159 -

ÍNDICE DE GRÁFICOS

Figura II.1:	Capas de OSGI	- 31 -
Figura II.2:	Componentes de JPF.....	- 34 -
Figura II.3:	Arquitectura (MVC) de un framework.....	- 37 -
Figura II.4:	Calidad del producto software y estándar ISO 9126	- 56 -
Figura II.5:	Características y subcaracterísticas de calidad del software	- 61 -
Figura II.6:	Características de la calidad en uso	- 61 -
Figura III.7:	Valoración final de cada framework	- 74 -
Figura III.8:	Logotipo Netbeans RCP.....	- 75 -
Figura III.9:	Arquitectura de la plataforma de desarrollo NetBeans RCP.....	- 76 -
Figura III.10:	Contenedor de ejecución NetBeans	- 77 -
Figura III.11:	Interacción de los módulos de una aplicación en NetBeans Platform.....	- 77 -
Figura III.12:	Logotipo Eclipse RCP	- 79 -
Figura III.13:	Elementos básicos de Eclipse RCP	- 81 -
Figura III.14:	Modelo MOSCA	- 87 -
Figura III.15:	Métrica Adecuación.....	- 89 -
Figura III.16:	Métrica exactitud	- 91 -
Figura III.17:	Métrica Madurez.....	- 93 -
Figura III.18:	Métrica No fallos.....	- 95 -
Figura III.19:	Métrica tiempo promedio de falla	- 97 -
Figura III.20:	Métrica comprensibilidad.....	- 99 -
Figura III.21:	Métrica comprensibilidad.....	- 101 -
Figura III.22:	Subcaracterística rendimiento.....	- 103 -
Figura III.23:	Subcaracterística utilización de recurso memoria	- 105 -
Figura III.24:	Resultados totales	- 107 -
Figura III.25:	Características del estándar ISO	- 108 -
Figura III.26:	Resultado final medición	- 110 -
Figura III.27:	Resultados totales	- 117 -
Figura III.28:	Características del estándar ISO	- 118 -
Figura III.29:	Resultado final medición	- 118 -
Figura IV.30:	Fases de la metodología SCRUM.....	- 120 -
Figura IV.31:	Diagrama de caso de uso de transacciones de caja	- 140 -
Figura IV.32:	Diagrama de secuencia de Transacciones de caja	- 142 -
Figura IV.33:	Diagrama de clases de transacciones caja.....	- 143 -
Figura IV.34:	Interfaz de loguin de usuarios	- 144 -
Figura IV.35:	Interfaz de inicio con listado de socios.....	- 144 -
Figura IV.36:	Interfaz de ingreso de nuevo socio	- 145 -

Figura IV.37:	Formulario de ingreso de datos del representante.	- 145 -
Figura IV.38:	Formulario de ingreso de datos del conyugue.	- 146 -
Figura IV.39:	Formulario de búsqueda de socios.	- 146 -
Figura IV.40:	Interfaz de gestión de cuentas	- 147 -
Figura IV.41:	Interfaz de búsqueda de cuentas	- 147 -
Figura IV.42:	Interfaz de gestión de registros	- 148 -
Figura IV.43:	Interfaz de búsqueda de registros.....	- 148 -
Figura IV.44:	Interfaz de gestión de Registros_Cuentas	- 149 -
Figura IV.45:	interfaz para la creación de asientos contables	- 149 -
Figura IV.46:	Interfaz de gestión de roles de usuario	- 150 -
Figura IV.47:	Interfaz de gestión de permisos de usuarios	- 150 -
Figura IV.48:	Interfaz de asignación de permisos a los roles	- 151 -
Figura IV.49:	Interfaz de gestión de usuarios	- 151 -
Figura IV.50:	Interfaz para generar los reportes de cada módulo	- 152 -
Figura IV.51:	Diagrama de despliegue transacciones caja	- 153 -
Figura IV.52:	Modelo físico de la base de datos.....	- 154 -
Figura IV.53:	Arquitectura del sistema.....	- 155 -
Figura IV.54:	BurnDown Chart Sprint 1	- 161 -

INTRODUCCIÓN

El presente trabajo investigativo trata del **Análisis Comparativo De Framework Software Libre Para El Desarrollo De Aplicaciones De Escritorio En Java y su posterior aplicación en el desarrollo del aplicativo para la Cooperativa de Ahorro y Crédito Chunchi Ltda.**, utilizando un framework dentro del ámbito de la programación y desarrollo, el mismo que cuenta con un sinnúmero de librerías, clases y arquitecturas disponibles para aplicarlas facilitando así la creación de los productos Software.

Entre los frameworks de software libre que nos facilitan el desarrollo de aplicaciones de escritorio en Java tenemos a: Netbeans (RCP), Eclipse (RCP), JMeter, GlueMvc y OpenSwing, de los cuales se ha seleccionado a Netbeans (RCP) y Eclipse (RCP) para realizar un estudio más detallado y poder realizar el análisis comparativo entre estos y así poder seleccionar el más adecuado para desarrollar el aplicativo "SISCOOP" para la Cooperativa de Ahorro y Crédito Chunchi Ltda.

A continuación se describirá el desarrollo de la tesis, contenido en cuatro capítulos:

El Capítulo I, Marco Referencial; describe las razones de la investigación, justificación, objetivos e hipótesis.

El Capítulo II, Marco Teórico; describe los fundamentos teóricos de la investigación.

El Capítulo III, Análisis Comparativo; describe la construcción del prototipo con cada framework de estudio, la definición de parámetros de calidad, análisis comparativo de cada framework de estudio y la demostración de la hipótesis.

El Capítulo IV, Implementación de la Aplicación; describe la construcción de la parte aplicativa de la tesis SISCOOP (Sistema Informático de Servicios Cooperativos) utilizando la metodología SCRUM.

Por último se encuentran las conclusiones, recomendaciones y anexos del presente trabajo de investigación.

El trabajo realizado servirá como base para escoger un framework de software libre adecuado sobre el cuál desarrollar una aplicación de escritorio en Java, obteniendo la máxima calidad de ella.

CAPÍTULO I

MARCO REFERENCIAL

1.1. Antecedentes

Junto al desarrollo del software surge un nuevo concepto denominado Framework como una estructura de soporte definida para proyectos de software en donde su principal actividad es organizar el desarrollo. En otras palabras un Framework representa una arquitectura de software que modela las relaciones generales, provee una estructura y una metodología de trabajo, con lo cual podemos decir que los Framework fueron diseñados con la intención de facilitar en gran medida el desarrollo de software, permitiendo que tanto desarrolladores como diseñadores se enfoquen más en identificar requerimientos del software en lugar de “luchar” horas con la programación de la aplicación.

Muchas de las aplicaciones de escritorios tienen similares requerimientos técnicos tales como los siguientes:

- Interfaces de usuario consistentes

- Extensibilidad
- Visualización de datos
- Configuraciones
- Sistema de ayuda
- Mecanismo de Distribución
- Actualizaciones en línea
- Soporte multiplataforma

En cada nuevo desarrollo ir cumpliendo estos requerimientos técnicos resulta costoso, superfluo y aburrido. Por estas razones se han ido desarrollando varios frameworks que proveen a los desarrolladores abordar todos estos requerimientos técnicos y que pueden ahorrar años de tiempo de desarrollo.

Siguiendo esta tendencia actual, el desarrollo de aplicaciones de escritorio en el lenguaje de programación Java también se ha visto favorecido por varias plataformas libres como NetBeans (RCP), Eclipse (RCP), JMeter, OpenSwing, GlueMvc entre otras; que además de ser herramientas IDE se pueden usar como frameworks y reutilizar sus componentes gráficos, arquitectura, librerías o API e incluso combinar las características de las dos plataformas para construir potentes aplicaciones de escritorio tipo cliente enriquecido, y es lo que han venido realizando importantes empresas, organizaciones gubernamentales, etc. a nivel mundial como: Oracle, IBM, La NASA, etc. en centenares de proyectos con licencias libres y comerciales destinados al ámbito de la planificación, simulación, diseño CAD, etc. que se los puede encontrar en las páginas web de sus respectivas comunidades de desarrollo.

En la actualidad en la ESPOCH existen una gran variedad de investigaciones referente a los frameworks; pero relacionados con las aplicaciones web; como Joomla, Spring, Zend, Simfony etc, siendo casi nulas las investigaciones realizadas en lo referente a frameworks de software libre para aplicaciones de escritorio en Java, existiendo únicamente opiniones de expertos y abundante información técnica al respecto en internet en las páginas oficiales que desarrollan y dan soporte a cada uno de los frameworks.

1.2. Justificación

El desarrollo de software avanza muy rápido en el tiempo, es decir mientras se está implementando una aplicación ya se está pensando en implementar otra, es por esta razón que hoy en día muchas de las empresas optan por utilizar diferentes tecnologías y frameworks que les permiten ser más productivos y alcanzar éxito en las entrega del aplicativo.

Actualmente existe un número considerable de frameworks para el desarrollo de aplicaciones de escritorio basadas en el lenguaje Java que siguen una determinada arquitectura (modular o capas) o metodología de desarrollo otorgando las siguientes ventajas: una mayor productividad del desarrollador, una mejora de la arquitectura de la aplicación final y también se resuelven problemas recurrentes que están presentes en la construcción de nuevas aplicaciones de escritorio. Por lo tanto se hace importante realizar un análisis de las características de los frameworks y contrastarlas estableciendo parámetros de comparación para el software desarrollado con ellos.

En vista que existen numerosos criterios difundidos en Internet acerca de las bondades de los frameworks para el desarrollo de aplicaciones de escritorio en Java de los miembros de cada una de las comunidades que desarrollan los mismos. Se pretende que el presente análisis comparativo permita a varios desarrolladores y programadores tener un enfoque práctico sobre estas herramientas existentes, para seleccionar el mejor framework para el desarrollo de software; con el propósito de obtener productos de calidad.

El uso de frameworks de software libre permite eliminar costos de licenciamiento para el desarrollador y como consecuencia se reduce costos en el momento de incorporar una solución informática en instituciones públicas o privadas en las distintas fases de su desarrollo ayudando especialmente a las más pequeñas que se encuentran en etapas de crecimiento y fortalecimiento, ya que no cuentan con los recursos económicos suficientes para adquirir soluciones tecnológicas propietarias o pagadas.

Debido a que la Cooperativa de Ahorro y Crédito Chunchi Ltda lleva dos años y medio de funcionamiento en el mercado, no posee un sistema informático propio para sus actividades operativas como gestión de crédito, manejo de las transacciones de caja y manejo de contabilidad, frente a estas dificultades esta institución se ve en la necesidad de implantar un sistema propio de acuerdo a sus necesidades para realizar las actividades anteriormente descritas ya que al momento se encuentran arrendando el sistema CONEXUS Milenium cuyos costos son elevados e influyen considerablemente en los gastos de la institución.

El sistema a desarrollarse en la Cooperativa de Ahorro y Crédito Chunchi Ltda contendrá los siguientes módulos:

Módulo 1: Módulo de Gestión de Procesos de Crédito

- Análisis de la capacidad de pago de los clientes
- Registro de solicitudes de crédito
- Generación de tablas presuntivas
- Liquidación de créditos
- Seguimiento y cobranza de la cartera de crédito
- Reclasificación diaria de la cartera
- Cobros de cuotas
- Cobros de intereses
- Cobros de multas
- Cobros de comisiones
- Generación de notificaciones por atraso
- Renovación de créditos
- Registro de garantes

Módulo 2: Módulo de Reportes de Crédito

- Clasificación de carteras de crédito de acuerdo a la Superintendencia de Bancos
- Clientes puntuales
- Cartera vencida
- Cartera que no devenga interés
- Cartera que devenga interés

- Cartera contaminada
- Clasificación del crédito de acuerdo al monto
- Emisión de pagares
- Generación de tablas de pagos

Módulo 3: Módulo de Gestión de procesos de Caja

- Registro de nuevos socios
- Recepción de depósitos
- Recepción de retiros
- Aperturas de libretas
- Manejo de inversiones y pólizas
- Cobro de servicios bajo convenio
- Arqueo de caja
- Pago de intereses a los ahorros
- Pago de intereses a las pólizas

Módulo 4: Módulo de reportes de Caja

- Cierre diario
- Estado de cuenta
- Impresión de Libretas

Módulo 5: Módulo de Gestión de procesos de Contabilidad

- Registro de un asiento contable
- Conciliación bancaria
- Manejo de Depreciaciones y Amortizaciones

- Cierre del año

Módulo 6: Módulo de reportes de Caja

- Balance General
- Estado de resultados
- Estado de cambio de patrimonio
- Diario por cuenta

Módulo 7: Módulo de gestión de usuarios

- Registro de nuevos usuarios
- Asignación de roles y permisos a los usuarios
- Auditoria de usuarios

Módulo 8: Módulo de reportes de usuarios

- Usuarios registrados
- Accesos de usuarios en determinada fecha
- Accesos fallidos

Manual de usuario

Manual Técnico

Manual de Instalación del sistema

1.3. Objetivos

1.3.1. Objetivo General

Analizar los frameworks de software libre para el desarrollo de aplicaciones de escritorio en Java aplicado a la Cooperativa de Ahorro y Crédito Chunchi Ltda.

1.3.2. Objetivos Específicos

- Analizar los frameworks de software libre que dan soporte para el desarrollo de aplicaciones de escritorio en Java.
- Definir criterios técnicos de preselección de los frameworks para construir los módulos de prueba.
- Establecer parámetros de comparación dirigido hacia los módulos de prueba a desarrollarse; para determinar el mejor framework de software libre de creación de aplicaciones de escritorio en Java.
- Realizar módulos de prueba en los frameworks preseleccionados para establecer la comparación y determinar cuál es el mejor framework en lo relacionado a la calidad del producto final.
- Implantar una aplicación informática de escritorio usando el mejor framework de software libre en Java, para la Cooperativa de Ahorro y Crédito Chunchi Ltda.

1.4. Hipótesis

El framework software libre para aplicaciones de escritorio en Java Netbeans RCP permitirá obtener un producto de mayor calidad que el frameworks Eclipse RCP.

CAPÍTULO II

MARCO TEÓRICO

2.1. Arquitectura¹ Modular

2.1.1. Definición

Una arquitectura modular permite que se diseñe módulos para ser cambiados y mejorados con el tiempo sin subvaloración de la funcionalidad del sistema en su conjunto. En este sentido, la arquitectura de un sistema modular es "tolerante a la incertidumbre" y "da la bienvenida a los experimentos "en el diseño de módulos [2].

2.1.2. Características

2.1.2.1. Módulo

Un módulo es un auto-contenido de componentes de software con el que otros módulos se pueden comunicar a través de interfaces bien definidas, tras la que sus implementaciones están ocultas. Cada característica de grano grueso de

¹ De acuerdo a (8) La Arquitectura de Software especifica los elementos del código fuente, la forma en que interactúan, los patrones de composición, y restricciones a los patrones

una aplicación se crea dentro de un módulo, lo que cada módulo de una responsabilidad claramente definida dentro de la aplicación [12].

2.1.2.2. Modularidad

Un sistema complejo se dice que muestra la modularidad en el diseño si sus partes pueden ser diseñadas de forma independiente, pero que trabajan juntas para apoyar al conjunto.

Los sistemas pueden tener diferentes estructuras modulares y diferentes grados de interdependencia entre sus respectivos componentes. Sin embargo, las diferentes partes de un sistema modular deben ser compatibles. La compatibilidad está asegurada por las reglas de diseño arquitectónico que los desarrolladores de seguir

2.1.2.3. Desarrollo Modular

Según el editor de la revista internacional *Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO*, frecuentemente los programadores de software no visualizan la magnitud de las aplicaciones que están desarrollando; por ejemplo, se comienza a crear una aplicación de tipo Web o escritorio y es entonces cuando se percibe la magnitud del proyecto por la cantidad de paquetes y clases que se generan para la aplicación, adicionalmente de las clases proporcionadas por las librerías de terceros. Una solución a esta problemática es el desarrollo de aplicaciones en base a módulos o servicios utilizando tecnologías conocidas como plataformas de cliente enriquecidas (RCP, sus siglas en inglés), que separan las funcionalidades de una aplicación permitiendo el desarrollo paralelo o colaborativo integrándose de forma transparente en una sola aplicación. Para

ello existen una serie de proyectos y especificaciones que permiten implementar el desarrollo modular en las aplicaciones tales como OSGi (Open Services Gateway Initiative), la plataforma de desarrollo de NetBeans, JPF (Java Plugin Framework) y JSPF (Java Simple Plugin Framework), entre otros [3].

2.1.3. Estándares existentes

2.1.3.1. OSGi

La implementación de un desarrollo modular basado en servicios es una de las novedades aplicadas al desarrollo de sistemas Web o de escritorio complejos con un gran número de clases interactuando. OSGi ha sido creado para generar un entorno de software de colaboración. Con frecuencia se crea una aplicación sin tener dimensionada la cantidad de funcionalidades que se implementarán a futuro. Es por ello que este tipo de marco de trabajo (framework) brinda la posibilidad de controlar las aplicaciones de una manera transparente y sin problemas de integración. Un ejemplo de una implementación modular es la de los repositorios de las distribuciones de Linux como Ubuntu y Linux Mint, en donde son descargados los paquetes necesarios para instalar una nueva aplicación, actualizando, moviendo o eliminando paquetes de forma transparente, en donde programadores de diversas áreas brindan de forma colaborativa a la comunidad de la distribución Linux la posibilidad de actualizar automáticamente la versión del sistema operativo.

Sin duda las ventajas de la integración de la especificación OSGi, descritas en la tabla II.I, se ven reflejadas en el rápido desarrollo de

aplicaciones. En la figura II.1 se presentan las capas de OSGi que se desarrollan por encima de la plataforma Java y que proporcionan la capacidad de desarrollo modular. Estas capas se describen en la tabla II.I.

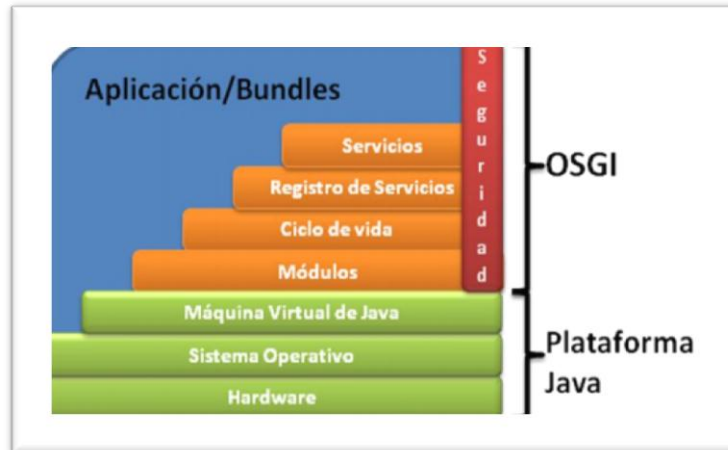


Figura II.1: Capas de OSGi

Fuente: Meza Ortiz, Carlos Omar, Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO

En la tabla II.I se observa las ventajas de la especificación OSGi.

Tabla II.I Ventajas de la especificación OSGi.

Ventaja	Descripción
Reducción de complejidad	OSGi se basa en módulos o paquetes llamados bundles que ocultan sus propiedades a otros paquetes, con la capacidad de actualización, instalación o desinstalación en tiempo de ejecución. Además, por su diseño modular, es mucho más claro detectar los errores dentro de la aplicación.
Reuso	El modelo de componentes OSGi hace que sea fácil

	<p>de utilizar componentes de terceros en una aplicación.</p> <p>Un número creciente de proyectos de código abierto y comercial proporcionan sus JAR bajo las especificaciones de OSGi.</p>
Desarrollo	<p>Capacidad para definir tu aplicación en partes independientes de forma clara y explícita, analizar, entender y resolver requerimientos dada la independencia entre los paquetes para vistas como funcionalidades dentro del sistema permitiendo el desarrollo colaborativo, y por consecuencia, disminuyendo el tiempo de producción.</p>
Herramientas	<p>La especificación permite la creación de múltiples implementaciones del framework para proporcionar libertad de elección. Algunos proyectos de código abierto son: Apache Félix, Equinoccio de Eclipse, Knopflerfish, entre otros.</p>

Fuente: Meza Ortiz, Carlos Omar, Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO

En la tabla II.II se describen las capas que conforman el estándar OSGI.

Tabla II.II Capas de OSGi

Capa	Descripción
Módulos	En su nivel más bajo, la especificación OSGi define un modelo de implementación de los módulos basados en Java. La unidad de despliegue en OSGi se conoce como un conjunto. En lugar de crear un mecanismo de implementación completamente nueva, aprovecha el formato de archivos jar para crear los bundles.
Ciclo de Vida	Una vez que el paquete se instala en un marco de OSGi, el ciclo de vida OSGi gobierna el estado del paquete. Un paquete se puede instalar, iniciar, detener y desinstalar desde el marco, siguiendo el ciclo de vida prescritos por la especificación OSGi.
Registro de Servicio	OSGi proporciona también un registro de servicios con la que se puede publicar y/o consumir servicios a través de la máquina virtual de Java. Por lo tanto, OSGi es a veces descrito como "SOA en una JVM."
Servicios	OSGi define también varios servicios básicos que se pueden proporcionar en el marco. Estos incluyen servicios de HTTP, de configuración, etc.
Seguridad	Esta capa se asegura que los bundles sean mostrados mediante una autenticación con firma digital. Además, la capa de seguridad puede apoyar a los permisos de Java para controlar la carga y ejecución de las clases del paquete.

Fuente: Meza Ortiz, Carlos Omar, Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO

2.1.3.2. JPF (Java Plugin Framework)

JPF proporciona un motor de tiempo de ejecución que descubre de forma dinámica y carga "plug-ins". Un plug-in es un componente de estructura que se describe a JPF con un "manifiesto". JPF mantiene un registro de plug-ins disponibles y las funciones que proporcionan (a través de puntos de extensión y extensiones) [13].

Uno de los objetivos principales de JPF es que la aplicación (y su usuario final) no debe pagar ninguna penalización en el rendimiento de memoria o de plug-ins que se instalan, pero no se utiliza. Plug-ins se añaden a la del registro en la solicitud de puesta en marcha o mientras la aplicación se está ejecutando pero no se cargan hasta que se les llama. En la figura II.2 se observa los componentes de Java Plug-in Framework.

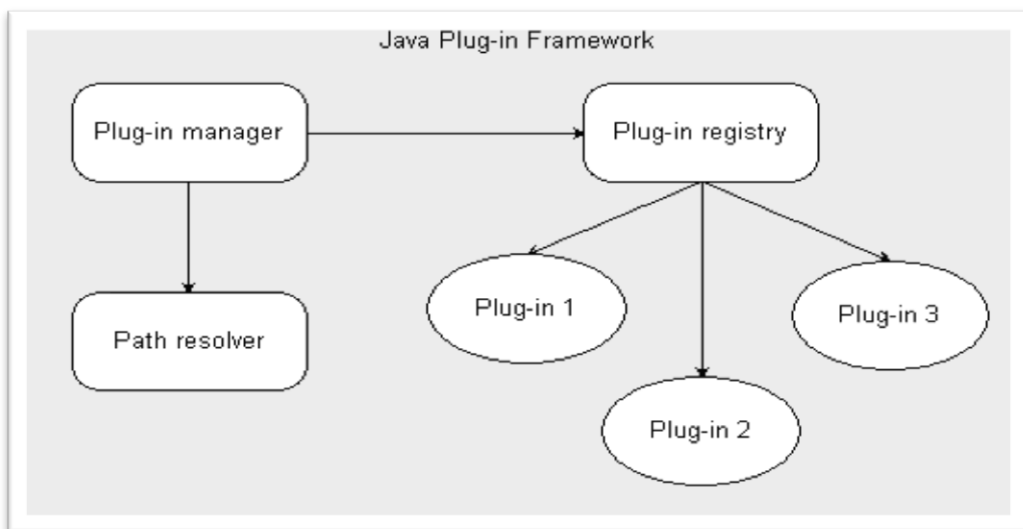


Figura II.2: Componentes de JPF

Fuente: <http://jpf.sourceforge.net/resources/images/jpf-diagram.png>

2.1.3.3. JSPF (Java Simple Plugin Framework)

El Marco de Java Plugin simple fue construido para reducir el tiempo de desarrollo al tiempo que aumenta la mantenibilidad del código de las pequeñas y proyectos de tamaño mediano. Oculta completamente los detalles de implementación de los componentes. Sólo el uso de sus interfaces.

Los componentes pueden ser cargados con sólo dos líneas de código. En gran medida basada en la anotación: Tema PluginImplementation @, InjectPlugin@, @ pluginLoaded, @ @ temporizador y, a través del uso de genéricos por lo general es de tipo seguro. Plugins adicionales para exportar otros plugins de JavaScript, JSON, LipeRMI, XMLRPC, placer o XMLRPC ERMI. Plugins se puede descubrir en la red local utilizando ZeroConf [19].

2.1.4. Ventajas

- Según la arquitectura modular en el desarrollo de software, deja abierta la posibilidad de enriquecer el sistema a medida que sea necesario, de este modo el cliente no verá limitado su desempeño ni tendrá comprometida su información en caso de necesitar actualizar con nuevos módulos el sistema [19].
- Según criterio de la Modularización se vuelve más importante a medida que aumenta la complejidad de una aplicación. La vieja programación y metodologías enfrentan una lucha para mantener el ritmo continua reducción de los ciclos de producción, aumentando gradualmente un conjunto de características, y al mismo tiempo la importancia creciente de la calidad del software [1].

- En particular, la presión de tiempo en el mercado exige un diseño de la aplicación sólida, proporcionando clara extensión puntos, los componentes de acoplamiento flexible, y un mecanismo de control de versiones fiables. Sin una base sólida sobre una arquitectura sostenible, el incremento de nuevas funciones en una versión introduce un nuevo nivel de caos, lo que lleva a un escenario en el que las mejoras y correcciones de errores resulta imposible de implementar, ya que cada cambio hace que cada vez se produzcan más efectos secundarios perjudiciales.

2.1.5. Desventajas

- Al tratarse de desarrollo de software basado en reutilización las actualizaciones de los componentes adquiridos o módulos que no se dispone de información no están en manos de los desarrolladores del sistema [7].

2.2. Desarrollo de software usando frameworks

2.2.1. Definición

La palabra de origen inglés "framework" se define como un grupo estandarizado de conceptos, prácticas y criterios para dirigir una problemática particular, además sirve como guía para enfrentar y resolver nuevos problemas similares características [4].

En el desarrollo de software, un framework es una composición conceptual y tecnológica con un soporte bien definido, habitualmente con módulos de

software concretos, en base a la cual otro proyecto de software puede ser fácilmente organizado y desarrollado [1].

Habitualmente los frameworks incluyen bibliotecas, soporte de programas y lenguaje interpretado, y otras herramientas, para facilitar a desarrollar y concatenar los diferentes componentes de un proyecto software.

2.2.2. Arquitectura

La arquitectura de los frameworks están basados en el modelo MVC (Modelo=>Vista=>Controlador), ya que los desarrolladores deben fragmentar la programación.

En la figura II.3 podemos apreciar los componentes básicos que se utilizan en la arquitectura básica de la mayor parte de frameworks.

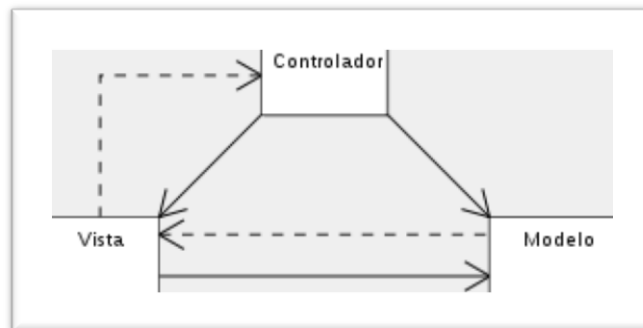


Figura II.3: Arquitectura (MVC) de un framework

Fuente: http://commons.wikimedia.org/wiki/File:ModelViewControllerDiagram_es.svg

Otros aspectos básicos que se debe contemplar en la implementación de nuestro sistema son:

2.2.2.1. Controlador

Este componente controla el acceso (inclusive todo) a la aplicación, esto suele incluir: archivos, scripts, o programas; el todo tipo de información que admite la interfaz. Así se diversifica el contenido de forma dinámica y estática; y sólo se debe controlar algunos aspectos.

2.2.2.2. Modelo

El modelo maneja todas operaciones lógicas, y manipulación de información (enviada por el controlador). Cada miembro es llamado cuidadosamente, con su nombre correcto y al comienzo con su verdadera naturaleza.

2.2.2.3. Vista

La vista dibuja o expresa la forma final de los datos, es la interfaz gráfica con la que interactúa el usuario final del programa (GUI).

2.2.3. Características

Un framework es un diseño orientado a objetos, aunque no necesariamente su implementación esté realizada en un lenguaje orientado a objetos. Generalmente se diferencian dos partes de un framework, una parte llamada “virtual engine”, que es estática y no cambia independientemente de la aplicación, mientras que las clases abstractas que son utilizadas como interfaces hacia la aplicación son también llamadas “plug points” o “hot spots”, estas clases son implementadas usualmente utilizando delegados o polimorfismo, hacen que el framework pueda ser extendido y adaptado para satisfacer las necesidades específicas de una aplicación [18].

Aunque también usualmente se incluyen clases concretas que pueden ser instanciadas para poner a trabajar el framework directamente sin ninguna modificación. De esta manera un framework no solamente se centra en el concepto de reutilización de código de la orientación a objetos, sino también amplía esto de manera que se busca la reutilización del diseño de un sistema, sobre la reutilización de código. Las características de un framework dependerán del tipo de aplicación para el que el framework fue desarrollado, pero en general un framework debe tener dos características básicas:

Debe ser lo suficientemente simple para que el desarrollador pueda aprender a utilizarlo correctamente.

Debe proveer suficiente funcionalidad y puntos de entrada para que pueda ser utilizado rápidamente, además de poder ser personalizado sin mucho esfuerzo.

2.2.4. Tipos de frameworks

Aunque los principios en los que el diseño de un framework y los beneficios de su utilización son independientes del dominio de aplicaciones a los que son aplicados, se puede clasificar a los frameworks tomando en cuenta el ámbito en el que se desenvuelven.

2.2.4.1. System infrastructure frameworks

Frameworks para infraestructura de un Sistema. Estos frameworks simplifican el desarrollo de aplicaciones, ya sea frameworks de comunicaciones, interfaces de usuario, etc. Este tipo de frameworks son internos a la organización y no son vendidos a los consumidores directamente.

2.2.4.2. Middleware integration frameworks

Frameworks de Integración. Estos frameworks son comúnmente utilizados para integrar aplicaciones distribuidas y componentes. Son diseñados para mejorar la habilidad de los desarrolladores de software de modularizar, reutilizar, y extender la infraestructura de las aplicaciones para trabajar en ambientes distribuidos. Estos frameworks pueden ser vendidos a los consumidores, que usualmente son otras organizaciones dedicadas al desarrollo de software.

2.2.4.3. Enterprise application frameworks

Frameworks para Aplicaciones Empresariales. Estos frameworks tratan con muchos dominios de aplicaciones, como telecomunicaciones, aeronáutica, finanzas. En comparación con los frameworks de infraestructura o integración, son costosos de desarrollar o de adquirir. Los frameworks empresariales soportan el desarrollo de aplicaciones de usuario directamente, en contraste a los frameworks de integración e infraestructura, que se enfocan en el desarrollo interno a una organización. Independientemente del ámbito, los frameworks pueden ser clasificados dependiendo de las técnicas que se utilicen para extenderlos, o personalizarlos.

2.2.4.4. White Box, Frameworks de Caja Blanca

Este tipo de frameworks descansan fuertemente en las características de la programación orientada a objetos, como son herencia y polimorfismo, más la habilidad de sobrecargar funcionalidad genérica con específica para adquirir las características necesarias de extensibilidad. Esto genera un acoplamiento muy fuerte entre la superclase y su derivada, en términos de un íntimo conocimiento de su implementación y además que al instanciar una subclase

automáticamente se crea una instancia de su superclase. Este fuerte acoplamiento es un factor que afecta mucho a la flexibilidad y extensibilidad, y estas son las dos razones principales para el desarrollo de un framework.

Un framework de caja blanca implica que el consumidor del framework debe tener acceso al código fuente, o en su defecto utilizar herencia para agregar la funcionalidad requerida. De cualquier manera se necesita un conocimiento amplio acerca de la implementación del framework. Este nivel de conocimiento requerido hace que los frameworks de caja blanca tengan curvas de aprendizaje bastante largas, además se requiere documentación muy detallada, y ejemplos de su utilización, para utilizarlos de una manera eficiente y efectiva.

Usualmente se dice que un framework de caja blanca está en la “adolescencia”, ya que se está separando la parte estática del framework de los puntos de ingreso para su extensión. En las empresas de desarrollo, se suele decir que a partir de este punto un framework tiene vida por sí mismo, y debería tener su propio equipo de desarrollo y manejo. De esta manera con un nuevo proyecto, durante las fases de análisis, diseño e implementación, los frameworks serán extendidos para cumplir con las características específicas del proyecto, y el equipo de desarrollo del framework puede ser notificado de cualquier error que se encuentre.

Lo óptimo sería que uno de los miembros del equipo encargado del desarrollo del framework se incluya en todas las implementaciones, para promover un

bucle de retroalimentación continuo. Luego de que un proyecto se ha finalizado, deberá empezar una fase en la que se determine las partes del framework que deberán ser cambiadas, para una mejora continua del framework.

2.2.4.5. Black Box, Frameworks de Caja Negra

Los frameworks de caja negra soportan la extensión de sus funciones definiendo interfaces para componentes que pueden ser enlazados al framework vía composición de objetos. La funcionalidad existente es reutilizada definiendo componentes que utilicen una interface particular, e integrando estos componentes en el framework. Este modelo de composición actualmente es muy aplicado. Los grandes componentes de negocios, son desarrollados al rededor de los conceptos de interfaces, composición y delegación. La mayor parte de los modelos de componentes hoy en día son desarrollados de esta manera. Además de agregar una gran ventaja ya que los componentes desarrollados de esta manera son independientes del lenguaje de programación. Los frameworks de caja negra son tomados como frameworks “maduros” al contrario de frameworks de caja blanca, ya que una vez que el framework es lo suficientemente estable, se lo puede volver estático, esto es que no se requiere herencia para agregar funcionalidad. Cualquier configuración es hecha a partir de interfaces, y “proveedores” pueden agregar cualquier nuevo comportamiento.

Los frameworks de caja blanca además requieren que los desarrolladores tengan un conocimiento muy amplio acerca de la estructura interna del

framework, aunque los frameworks de caja blanca son ampliamente utilizados, se tiende a producir sistemas que están fuertemente acoplados a detalles específicos del framework. En contraste con los de caja blanca los frameworks de caja negra son estructurados usando composición de objetos y delegación en reemplazo a la herencia. Como resultado los frameworks de caja negra son generalmente más fáciles de utilizar y extender que los frameworks de caja blanca. Pero los frameworks de caja negra son más difíciles de desarrollar ya que requieren que sus desarrolladores definan interfaces y puntos de ingreso que anticipen un rango muy amplio de usos potenciales.

2.2.4.6. Grey Box, Frameworks de Caja Gris

Realizando un análisis de la evolución de los framework, las cajas negras y blancas son los extremos del diseño y utilización de un framework. Muchos frameworks probablemente vivan en la región entre estos dos extremos, como frameworks de caja gris. Los frameworks de caja gris intentan sacar provecho de los beneficios de ambos tipos de framework, mientras intentan evitar las limitaciones de ambos. Realizando un análisis del ciclo de vida de los distintos tipos de frameworks los frameworks de caja negra representan la etapa de madurez de un framework, pero muchas veces hay limitaciones tanto de tiempo como de dinero que impiden alcanzar esta etapa, por lo que el desarrollo del Framework queda “a medio camino” como un Framework de caja gris.

2.2.4.7. Visual Builders y Soporte de Lenguaje

Cuando un framework alcanza la más alta madurez, es decir el tope de su evolución. Las aplicaciones son “armadas” como un rompecabezas, con una herramienta que permita el diseñar el sistema de una manera visual, y un

lenguaje a nivel macro de la aplicación. Con este nivel de soporte el framework está listo para ser movido fuera de las manos de un desarrollador de aplicaciones, hacia un usuario final del sistema. Un ejemplo de un visual builder y un framework que esté completamente maduro sería el que un experto en el tema de seguros, en lugar de interactuar con un desarrollador, lo haga como una herramienta como Visual Basic.

El podría crear una aplicación de seguros, basada en un framework específico para el dominio. Seleccionar un consumidor, una póliza de seguros, que proveen las interfaces necesarias, escribir unas pocas líneas en un lenguaje natural para procesar los eventos deseados, y luego un botón de Generar y Distribuir. Si esto suena difícil de crear sería comparable al hablar de Delphi a un programador de ensamblador de los años 60.

2.2.5. Ventajas

Entre las principales ventajas de usar los frameworks tenemos [19]:

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona, podríamos decir que tendríamos que llenar un esqueleto.
- Minimiza tiempos de desarrollo / acorta el tiempo “time to market”
- Reduce los riesgos del desarrollo
- Proporciona una arquitectura consistente entre aplicaciones

2.2.6. Desventajas

Como desventajas de utilizar un framework tenemos las siguientes [21]:

- Limitación de la flexibilidad

- Dificultad de aprendizaje
- Reducción de la creatividad

2.3. Pluggins

2.3.1. Definición

Un plugin o complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión.

2.3.2. Funcionalidades

Los Plugins permiten:

- Que los desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones
- Reducir el tamaño de la aplicación
- Separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de software

2.4. Frameworks Software Libre para Aplicaciones de Escritorio en Java

En la actualidad existe un sinnúmero de frameworks libres para el desarrollo de aplicaciones de escritorio en Java.

Por lo que para realizar el estudio preliminar de los frameworks libres Java se han seleccionado los más populares según la búsqueda realizada en la web, como son: NetBeans (RCP), Eclipse (RCP), GlueMvc, JMeter y OpenSwing.

2.4.1. Estudio previo de NetBeans (RCP)

2.4.1.1. Definición

Según el editor de la revista La plataforma de NetBeans (RCP) ofrece las características de una aplicación de cliente enriquecida en base a su arquitectura modular formada por un conjunto de API's, en donde cada módulo se identifica como una nueva funcionalidad, teniendo la posibilidad de un desarrollo paralelo ya que solo se requiere integrar el módulo a la aplicación en desarrollo [3].

También de acuerdo a La plataforma NetBeans es un marco genérico para aplicaciones Swing. Que proporciona la "la base" que, antes, todos los desarrolladores tenían que escribir ellos mismos estado de ahorro, la conexión de las acciones de los elementos del menú, los elementos de la barra de herramientas y atajos de teclado, la gestión de ventanas, y así sucesivamente [6].

2.4.1.2. Características

Entre las características que se mencionan según los editores de la revista de ciencia y tecnología [3] se encuentran:

- Marco de trabajo para la interfaz gráfica basada en el paquete Swing.
- Editor de datos.
- Personalización de pantalla.
- Asistentes.
- Sistema de datos.

- Internacionalización.
- Ayuda del sistema.
- Facilidad de integración.
- Alto grado de cohesión y coherencia.

2.4.1.3. Ventajas

- La construcción de una aplicación de escritorio sobre la plataforma NetBeans(RCP) puede ahorrar años de tiempo de desarrollo [6].
- Simplifica drásticamente el desarrollo de aplicaciones de escritorio debido a que provee un número de técnicas, patrones y todos los compones Swing² [1].

2.4.1.4. Desventajas

- Bajo rendimiento de Swing aunque está mejorando.

2.4.2. Estudio previo de Eclipse (RCP)

2.4.2.1. Definición

Eclipse se inició originalmente como un sistema modular de aplicación IDE. En el año 2004, se publicó la versión de Eclipse 3.0 que podía ser re-utilizado para construir aplicaciones independientes basadas en la misma tecnología que el IDE de Eclipse [5].

Entonces podemos decir que Eclipse RCP es una plataforma para construir y desplegar aplicaciones ricas de cliente. Incluye Equinox, una estructura de componentes basados en el estándar OSGi, la capacidad de desplegar aplicaciones nativas de interfaz gráfica de usuario a una variedad de sistemas

² Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas según [http://es.wikipedia.org/wiki/Swing_\(Java\)](http://es.wikipedia.org/wiki/Swing_(Java))

operativos de escritorio, como Windows, Linux y Mac OSX y un mecanismo de actualización integrado para la implementación de aplicaciones de escritorio desde un servidor central.

2.4.2.2. Características

Entre las principales características de Eclipse RCP podemos mencionar las siguientes [22]:

- Apariencia nativa
- Extensibilidad
- Servicios comunes para aplicaciones
- Puede ser usado como punto de integración

2.4.2.3. Ventajas

Entre las principales ventajas podemos citar las siguientes:

- Proporciona un proyecto vacío que se puede modificar para crear aplicaciones.
- Se puede reutilizar la ayuda en los programas.
- Se puede crear su propia aplicación independiente, sin necesidad de utilizar cualquiera de los menús que existen en el IDE.
- Tiene un enfoque modular muy fuerte que permite el diseño de sistemas basado en componentes.

2.4.2.4. Desventajas

- Está basado en las librerías SWT, como alternativa a Swing.
- No tiene mucha flexibilidad al momento de incorporar nuevos plugins.
- Necesita mucho tiempo para el aprendizaje sino se tiene experiencia.

2.4.3. Estudio previo de GlueMvc

2.4.3.1. Definición

Gluemvc es un framework para el desarrollo de aplicaciones de escritorio en Java para pequeñas y medianas aplicaciones basadas en el patrón de diseño MVC(modelo-vista-controlador), y que está disponible bajo los términos de la licencia LGPL (Lesser General de licencia Public) GNU (GNU No es Unix) [14].

2.4.3.2. Características

- Gluemvc es un Framework que engloba algunos patrones muy utilizados para desarrollar J2EE.
- Contiene sus propios patrones como: BusinessService - estándar que se utiliza para procesar la información de un objeto dado y lo distribuirá a sus respectivos DAO, el BusinessRule - las reglas de negocio son los "pasos" que el caso de uso debe pasar para ser eficaz y el DAO (Data Access Object) - un patrón que es muy útil para cualquier aplicación para acceder a los datos.

2.4.3.3. Ventajas

- La implementación de un programa utilizando Gluemvc es una tarea muy sencilla.
- La configuración de Gluemvc es bastante simple.
- No es necesario instalar Gluemvc ya que funciona como una biblioteca de desarrollo.

2.4.3.4. Desventajas

- Se necesita crear un archivo de configuración para obtener los servicios, las reglas de negocio y las clases de persistencia.

2.4.4. Estudio previo de JMatter

2.4.4.1. Definición

Las aplicaciones basadas en JMatter están escritas en el lenguaje de programación Java. Esto implica principalmente que estas aplicaciones pueden ejecutarse en varios sistemas operativos de destino, desde computadoras Windows, para Mac, a la variedad de sistemas operativos GNU / Linux por ahí, así como otros sabores de Unix como Solaris y BSD [11].

Aplicaciones de diseño JMatter producen aplicaciones de software donde se conserva todos los datos a una base de datos relacional de fondo. Las bases de datos proporcionan potentes capacidades de consulta que son la base de las necesidades de muchos negocios. El marco JMatter utiliza la hibernación 5 marcos para comunicarse con bases de datos back-end. Esto permite la independencia de base de datos marco.

Las aplicaciones JMatter son compatibles con un número de sistemas de base de datos final. Se ha probado con PostgreSQL, MySQL, Oracle y bases de datos como las nuevas y más ligeros sistemas de bases de datos como H2 y hsqldb. Estas bases de datos finales están disponibles para un gran número de sistemas operativos.

Por el lado del cliente, JMatter aprovecha el Kit de herramientas de Java Swing para producir aplicaciones de cliente enriquecido. Estas interfaces de usuario son independientes de la plataforma.

2.4.4.2. Características

Entre las principales características de JMatter podemos citar las siguientes:

- Una rica Interfaz Grafica de Usuario (GUI) construida sobre Swing con soporte para operaciones CRUD.
- Persistencia: utilizando Hibernate.
- Autenticación: habilidad para login y administración de usuarios.
- Soporte para la construcción, invocación y persistencia de consultas en la GUI.
- Soporte en la construcción de wizards, calendarios y más.

2.4.4.3. Ventajas

- Reducción dramática en el tiempo de desarrollo en comparación con los métodos tradicionales en aproximadamente un factor de 10.
- Disminución del número de líneas de código necesarias para implementar una aplicación de software.
- Está escrito en el lenguaje de programación Java por lo que las aplicaciones de software se puede ejecutar una variedad de sistemas operativos, incluyendo la gran cantidad de GNU / Linux OS, Solaris, Mac OS X, y los diferentes sabores de Windows.

2.4.4.4. Desventajas

- JMeter es un framework que no es muy utilizado en la actualidad por los desarrolladores.

2.4.5. Estudio previo de OpenSwing

2.4.5.1. Definición

OpenSwing es una suite de código abierto de componentes gráficos avanzados construidos sobre la base de herramientas Swing, estos componentes son más sofisticados que los previstos con Swing y se puede manipular directamente en el diseñador de interfaz de usuario de un IDE. También es un framework que proporciona un mecanismo de unión de los datos (data binding) entre los componentes gráficos y el modelo de datos, basados en el paradigma MVC. El modelo de datos se basa en Java Beans (POJOs) y es compatible con todos los componentes OpenSwing, tales como el control para tablas de datos, control de árbol o un contenedor genérico de controles [10].

OpenSwing se puede utilizar para desarrollar aplicaciones Java de escritorio o aplicaciones dinámicas de Internet, gracias a una capa de software que permite conectar los componentes gráficos a través de HTTP a una capa del lado del servidor, tales como la Sprint o directamente Hibernate o Apache Cayenne herramientas ORM o iBatis de datos o JPA. EJB 3.0 y RMI también está disponible para la conexión del lado del cliente componentes OpenSwing de lógica de negocio de servidores de aplicaciones.

2.4.5.2. Características

OpenSwing requiere JVM 1.4.x o superior Incluye algunas bibliotecas de terceros muy comunes de código abierto:

- HSSF - se utiliza para generar automáticamente informes en formato Excel (XLS), el contenido de las tablas o cuadros
- iText - se utiliza para generar automáticamente informes en formato PDF y RTF, desde el contenido de las tablas o cuadros
- Log4J – para el registro de mensajes en el lado del servidor
- JCalendar - se usa para incluir un calendario dentro del control gráfico de fechas
- Java Database Connection Pool – para la administración de conexiones a la base de datos

2.4.5.3. Ventajas

- OpenSwing permite crear aplicaciones Java con arquitecturas que tienen varias capas.
- Por medio de OpenSwing se pueden desarrollar aplicaciones web.
- OpenSwing proporciona también las clases del lado del servidor, no conectadas directamente a las clases de la capa de presentación y que se puede aplicar con aplicaciones web de tres capas.
- OpenSwing ofrece una clase de utilidad que permite mapear objetos a instrucciones SQL que se utilizan para recuperar datos de una base de datos relacional o para insertar / actualizar los registros

2.4.5.4. Desventajas

- La capa del lado del servidor proporcionado por OpenSwing esta limitado a un servlet de Java.
- Cada capa en el framework depende de las capas subyacentes de la estructura.

2.5. Calidad Del Software

2.5.1. Introducción

En la actualidad las empresas desarrollan una gran cantidad de productos software que son liberados a sus clientes, por esta razón se hace necesario que los productos sean confiables y duraderos para que su producto software tenga un alto grado de aceptación en el mercado y así generar ingresos en beneficio de la empresa. Es así que varias empresas han desarrollado metodologías, procesos y técnicas que permiten a las empresas y sus desarrolladores brindar un producto de calidad a sus clientes aplicando una serie de etapas definidos en estos.

2.5.2. Definición

La calidad del software son características propias del software aquellas que tú quieres controlar y asegurar, el software es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla. El software puede tener errores, incidencias pero no son similares a lo que cualquier equipo de carácter físico.

La calidad del software se encuentra casi a la par de la calidad tradicional, ligeramente detrás debido a que la calidad tradicional tiene varias décadas de

historia, mientras que la calidad de software tiene entre 50 y 30 años de haber surgido [9].

2.5.3. La Norma ISO 9126

El estandar ISO 9126 es una norma internacional para la evaluación de software. La norma se divide en cuatro partes que aborda, respectivamente, los siguientes temas: modelo de calidad, métricas externas, métricas internas, y la calidad en el uso de métricas. La primera parte del ISO 9126, denominada ISO 9126-1 es una extensión del trabajo previo realizado por McCall (1977), Boehm (1978), FURPS y otros en la definición de un conjunto de características de calidad de software [17].

La ISO 9126-1 representa la última investigación (y permanente) en la caracterización de software para los fines de control de calidad, aseguramiento de la calidad del software de software y mejora de procesos software (SPI).

Este artículo define las características señaladas en la norma ISO 9126-1. Las otras partes de la norma ISO 9126, en relación con indicadores o mediciones de estas características, son esenciales para SQC, SQA y SPI, pero la preocupación principal de este artículo es la definición básica de la ISO 9126 del Modelo de Calidad.

La documentación ISO 9126 en sí, a partir de la documentación oficial de la norma ISO 9126, sólo puede ser comprado y está sujeto a derechos de autor.

El modelo de calidad del software ISO 9126-1 identifica 6 principales características de calidad, a saber:

- Funcionalidad
- Confiabilidad

- Facilidad de uso
- Eficiencia
- Capacidad de mantenimiento
- Portabilidad

En la figura II.4 se observa la subdivisión del estándar ISO 9126.

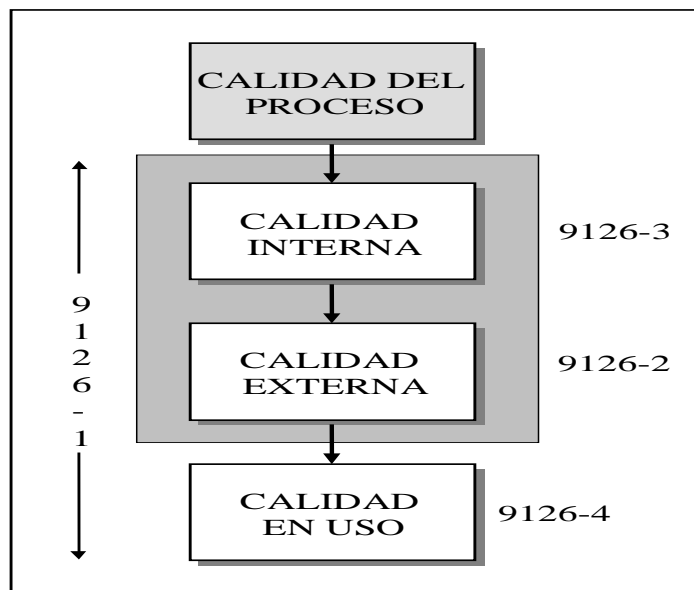


Figura II.4: Calidad del producto software y estándar ISO 9126

Fuente: http://docs.universidadecotec.edu.ec/tareas/2013E/COM420/alum/2013290373_4863_2013E_COM420_deber_de_tecnicas_iso.docx

Estas características se dividen en subcaracterísticas, una tabla de alto nivel se muestra a continuación. Es en el nivel subcaracterísticas que se produce la medición de SPI. Las principales características del modelo de ISO9126-1 calidad, se pueden definir de la siguiente manera:

Funcionalidad

La funcionalidad es el objetivo esencial de cualquier producto o servicio. Para ciertos artículos esto es relativamente fáciles de definir, por ejemplo el ancla de

un barco tiene la función de mantener un buque en un lugar determinado. Mientras más funciones un producto tenga, por ejemplo, las de un cajero automático, entonces es más complicado definir su funcionalidad [17].

Para el software se puede especificar una lista de funciones, es decir, una de ventas sistemas de procesamiento de pedidos debe ser capaz de grabar la información del cliente de modo que se puede utilizar para hacer referencia a un pedido de cliente. Un sistema de órdenes de venta también debe proporcionar las siguientes funciones:

- Registro de órdenes de venta del producto, precio y cantidad.
- Calcular el precio total.
- Calcular impuesto sobre las ventas correspondiente.
- Calcular la fecha existencia para su entrega, con base en el inventario.
- Generar órdenes de compra cuando la acción cae por debajo de un determinado umbral.

La lista sigue y sigue, pero el principal punto a tener en cuenta es que la funcionalidad se expresa como un conjunto de funciones esenciales que el producto de software proporciona. También es importante tener en cuenta que la presencia o ausencia de estas funciones en un producto de software pueden ser verificadas como ya sea existente o no, en que es un valor booleano (ya sea una respuesta sí o no). Las otras características de software mencionados (es decir, facilidad de uso) sólo están presentes en algún grado, es decir, no una simple tiene o no tiene. Muchas personas se confunden entre el proceso de funcionalidad en general (en la que el software juega un papel) y la funcionalidad del software.

Esto es en parte debido al hecho de que los diagramas de flujo de datos (DFD) y otras herramientas de modelado pueden representar la funcionalidad de proceso (como un conjunto de datos de entrada y conversiones en datos de salida) y la funcionalidad del software. Considere la posibilidad de un proceso de órdenes de venta, que cuenta con manuales y componentes de software. Una de las funciones del proceso de pedido de ventas podría ser para grabar la orden de venta, sino que podría implementar un archivador copia de las órdenes reales y sólo utilizar el software para el cálculo de la fecha, el impuesto y el precio. De esta manera la funcionalidad del software está limitada a esas funciones de cálculo. SPI o Software Process Improvement es diferente de la mejora de procesos en general o Reingeniería de Procesos , La ISO 9126-1 y otros modelos de calidad del software no ayudan a medir los costes\beneficios totales del proceso , pero sólo del componente de software.

La relación entre la funcionalidad del software dentro de un proceso de negocio general está fuera del ámbito de aplicación de la norma ISO 9126 y es sólo la funcionalidad del software , o propósito esencial del componente de software , que es de interés para la norma ISO 9126.

Después de la funcionalidad, hay otros 5 atributos de software que caracterizan a la utilidad del software en un entorno dado.

Cada una de las siguientes características sólo se puede medir (y se supone que existe), cuando la funcionalidad de un sistema dado está presente. De esta manera, por ejemplo, un sistema puede no poseer características de facilidad de uso, si el sistema no funciona correctamente (los dos simplemente no van juntos).

Confiabilidad

Una vez que un sistema de software está funcionando, tal como se especifica, y se entrega la característica de fiabilidad define la capacidad del sistema para mantener su prestación del servicio bajo condiciones definidas para períodos de tiempo definidos. Un aspecto de esta característica es que la tolerancia a fallos es la capacidad de un sistema para soportar fallo de un componente. Por ejemplo, si la red se cae durante 20 segundos y luego vuelve el sistema debe ser capaz de recuperarse y continuar funcionando [17].

Usabilidad

Usabilidad sólo existe con respecto a la funcionalidad y se refiere a la facilidad de uso para una función dada. Por ejemplo, una función de un cajero automático para dispensar dinero en efectivo es conforme a lo solicitado.

La colocación de cantidades habituales en la pantalla de selección, es decir, \$ 20.00, \$ 40.00, \$ 100.00, etc, no afecta a la función de la ATM, pero se refiere a la usabilidad de la función. La capacidad de aprender a utilizar un sistema (capacidad de aprendizaje) es también un importante subcaracterística de usabilidad [17].

Eficiencia

Esta característica se refiere a los recursos del sistema utilizados cuando se proporciona la funcionalidad requerida.

La cantidad de espacio en disco, memoria, red, etc es un buen indicador de esta característica. Al igual que con un número de estas características, existen superposiciones. Por ejemplo, la facilidad de uso de un sistema se ve influida por el rendimiento del sistema, en el que si un sistema le toma tres horas para

responder el sistema no sería fácil de usar aunque la cuestión esencial es una característica de rendimiento o eficiencia [17].

Mantenibilidad

La capacidad para identificar y corregir un error en un componente de software es lo que las características de la mantenibilidad abordan. En otros modelos de calidad de software de esta característica se hace referencia como la compatibilidad.

La Mantenibilidad se ve afectada por la legibilidad o la complejidad, así como la modularización del código. Cualquier cosa que ayude con la identificación de la causa de un fallo y a continuación se fijan el fallo es la preocupación de la mantenibilidad.

También la capacidad de verificar (o prueba) un sistema, es decir, la capacidad de prueba, es una de las subcaracterísticas de la mantenibilidad [17].

Portabilidad

Esta característica se refiere a lo bien que el software puede adoptar los cambios en su entorno o con sus requisitos. Las subcaracterísticas de esta característica incluyen la capacidad de adaptación.

Diseño orientado a objetos y prácticas de implementación pueden contribuir a la medida en que esta característica está presente en un sistema dado [17].

En la figura II.5 se observa las características y subcaracterísticas de la calidad del software.

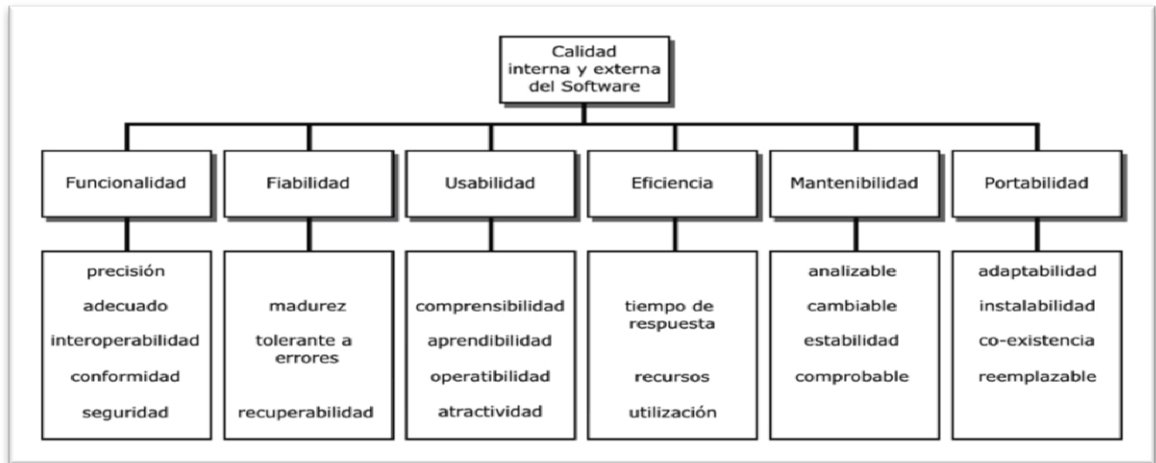


Figura II.5: Características y subcaracterísticas de calidad del software

Fuente: <http://www.grihocitools.udl.cat/mpiuu/usabilidad/calidad.html>

En la calidad de uso, se definen cuatro características: efectividad, productividad, seguridad de uso y satisfacción como se puede visualizar en la figura II.6.



Figura II.6: Características de la calidad en uso

Fuente: <http://4.bp.blogspot.com/-CIIBoz1tpal/UT-41v0fUaI/AAAAAAAAACE/CmQWNIXCFkY/s320/FUNCIONALIDAD.png>

CAPÍTULO III

ANÁLISIS COMPARATIVO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES DE ESCRITORIO EN JAVA

3.1. Introducción

En este capítulo se efectúa la investigación de los frameworks existentes y la selección de dos de ellos que sean maduros, documentados y que cumplan con la mayoría de características. Estas características brindan estabilidad a la tecnología pudiéndose apreciar por la cantidad de versiones y el grado de magnitud del cambio, contando además con el respaldo de una amplia comunidad de usuarios, de libros y tutoriales, así como de otros tipos de documentos útiles que sirvan como guía para la resolución de un problema, reduciendo así la curva de aprendizaje.

3.2. Determinación de los parámetros de preselección de los frameworks

A continuación se definen los parámetros que permiten preseleccionar los frameworks para el desarrollo de aplicaciones de escritorio en Java que es el objeto de análisis de este trabajo de investigación.

3.2.1. Selección de los parámetros para la preselección de los frameworks

En la tabla III.III se observa los parámetros que servirán para la preselección de los frameworks que se analizarán en este trabajo investigativo.

Tabla III.III Parámetros para la preselección de los frameworks

PARAMETRO	VARIABLE	VALOR
1.- PRODUCTO	Madurez del producto	<ul style="list-style-type: none">➤ Tiempo en el mercado➤ Versiones del producto
2.- FACILIDADES	Facilidad para el desarrollo	<ul style="list-style-type: none">➤ Soporte con otras tecnologías y plugins➤ Documentación➤ Comunidad

Elaborado por: Los autores

3.2.2. Valoración de los parámetros

Para analizar y evaluar los frameworks como: NetBeans (RCP), Eclipse (RCP), JMeter, GlueMvc y OpenSwing se tomarán los siguientes valores que valorarán a cada uno de los parámetros. En la tabla III.IV se observa la valoración de los parámetros para los frameworks.

Tabla III.IV Valoración de los parámetros para los frameworks

VALOR	SIGNIFICADO	PORCENTAJE
1	MALO	25%
2	REGULAR	50%
3	BUENO	75%
4	EXELENTE	100%

Elaborado por: Los autores

Para obtener resultados cuantitativos y cualitativos y poder llegar a una conclusión final durante el proceso de análisis y evaluación de los frameworks se utilizara la siguiente fórmula:

$$CTP = \sum_{i=1}^n \frac{VPE_i}{n}$$

Donde:

CTP: Cantidad total del parámetro

i: Número del parámetro evaluado

n: Total de parámetros evaluados

VPE: Valor del parámetro evaluado

3.3. Comparación de los frameworks

3.3.1. Producto

3.3.1.1. Análisis de los indicadores de Madurez del Producto

3.3.1.1.1. Tiempo en el mercado

Analiza si el framework es suficientemente sólido y efectivo, estableciendo el tiempo que se encuentra en el mercado tomando en cuenta así la experiencia. En la tabla III.V se puede observar el tiempo en años de los frameworks en el mercado.

Tabla III.V Tiempo en el mercado de los frameworks

FRAMEWORK	FECHA PRIMERA VERSION	FECHA ULTIMA VERSION	TIEMPO (AÑOS)
NetBeans RCP	Junio 2003	Octubre 2013	10
Eclipse RCP	Junio 2004	Junio 2013	9
GlueMvc	Enero 2006	Agosto 2013	7
OpenSwing	Abril 2007	Julio 2013	6
JMatter	Febrero 2008	Abril 2009	5

Elaborado por: Los autores

3.3.1.1.2. Versiones del producto

Analiza la cantidad de versiones que se lanzaron al mercado con sus nuevas mejoras y corrección de errores. En la tabla III.VI se puede ver la cantidad y el nombre de las diferentes versiones que han lanzado los frameworks.

Tabla III.VI Versiones de los frameworks

FRAMEWORK	VERSIONES DEL PRODUCTO
Netbeans RCP	NetBeans 7.4 15 de octubre de 2013, NetBeans 7.3.1 12 de junio de 2013, NetBeans 7.3.2 1 de febrero de 2013, NetBeans 7.2 noviembre de 2012, NetBeans 7.1.2 Mayo de 2012, NetBeans 7.0.1 01 de agosto de 2011, NetBeans 7.0 20 de abril de 2011, NetBeans 6.9.1 4 de

	<p>agosto de 2010, NetBeans 6.9 15 de junio de 2010, NetBeans 6.8 10 de diciembre de 2009, NetBeans 6.7.1 27 de julio de 2009, NetBeans 6.7 29 de junio de 2009, NetBeans 6.5 25 de noviembre de 2008, NetBeans 6.1 28 de abril de 2008, NetBeans 6.0 3 de diciembre de 2007, NetBeans 5.5.1 24 de mayo de 2007, NetBeans 5.5 30 de octubre de 2006, NetBeans 5.0 enero de 2006, NetBeans 4.1 mayo de 2005, NetBeans 4.0 diciembre de 2004, NetBeans 3.6 abril de 2004, NetBeans 3.5 junio de 2003</p> <p>22 versiones</p>
Eclipse RCp	<p>Kepler 26 de junio de 2013 4.3, Juno 27 de junio de 2012 4.2, Indigo 22 de junio de 2011 3.7, Helios 23 junio de 2010 3.6, Galileo 24 de junio de 2009 3.5, Ganymede 25 junio de 2008 3.4, Europa 29 de junio de 2007 3.3, Callisto 30 de junio de 2006 3.2, Eclipse 3.1 28 de junio 2005 3.1, Eclipse 3.0 28 de junio de 2004 3.0</p> <p>10 versiones</p>
JMatter	<p>jmatter-20090430, jmatter-20090405, jmatter-20081002, jmatter-20080902, jmatter-20080811, jmatter-20080627, jmatter-20080409, jmatter-20080320, jmatter-20080221, jmatter-20080211, jmatter-20080204</p> <p>11 versiones</p>
GlueMvc	<p>gluemvc 0.1.9 Enero 2006, gluemvc Agosto 2013</p> <p>2 versiones</p>

OpenSwing	OpenSwing 1.0.5, OpenSwing 1.0.6, OpenSwing 1.0.7, OpenSwing 1.0.8, OpenSwing 1.0.9, OpenSwing 1.1, OpenSwing 1.1.1, OpenSwing 1.1.2, OpenSwing 1.1.3, OpenSwing 1.1.4, OpenSwing 1.1.5, ... OpenSwing 2.2.2 117 versiones
-----------	---

Elaborado por: Los autores

3.3.1.1.3. Resultados

En la tabla III.VII se observa el resultado final del análisis del parámetro Producto.

Tabla III.VII Resultado final del análisis Producto

FRAMEWORK INDICADOR	NetBeans	Eclipse	JMatter	GlueMvc	OpenSwing
Tiempo en el mercado	4	4	1	3	2
Versiones del producto	4	3	2	1	4
TOTAL/4	4	3,5	1,5	2	3

Elaborado por: Los autores

$$CTP_{Netbeans} = \sum_{i=1}^2 4 + 4 = \frac{8}{2} = 4$$

$$CTP_{Eclipse} = \sum_{i=1}^2 4 + 3 = \frac{7}{2} = 3,5$$

$$CTP_{J\text{Matter}} = \sum_{i=1}^2 1 + 2 = \frac{3}{2} = 1,5$$

$$CTP_{\text{GlueMvc}} = \sum_{i=1}^2 3 + 1 = \frac{4}{2} = 2$$

$$CTP_{\text{OpenSwing}} = \sum_{i=1}^2 2 + 4 = \frac{6}{2} = 3$$

3.3.1.1.4. Interpretación

Netbeans RCP y Eclipse RCP se encuentran 9 años y 8 años en el mercado respectivamente son los más sólidos y utilizados en la actualidad, además poseen varias versiones en las cuales se han corregido y realizado mejoras al producto.

OpenSwing y JMeter son los que menos tiempo se encuentran en el mercado, pero OpenSwing ha lanzado un sinnúmero de versiones lo que da a entender que el producto tiene muchos errores y todavía no alcanza una madurez adecuada. JMeter tiene pocas versiones pero todavía no es muy aceptado por los desarrolladores.

GlueMvc ha estado disponible durante 7 años, pero hasta la fecha solo ha lanzado 2 versiones por lo que se podría que su desarrollo está paralizado ya que no ha alcanzado una buena aceptación o existe desconocimiento por parte de los desarrolladores.

3.3.2. Facilidades

3.3.2.1. Análisis de los indicadores de facilidades para el desarrollo

3.3.2.1.1. Soporte con otras tecnologías y plugins

Este indicador analiza la facilidad que tiene el framework para añadir funciones adicionales distintas a las nativas que tiene el framework. En la tabla III.VIII se observa el soporte de tecnologías y plugins que tienen los frameworks.

Tabla III.VIII Soporte de tecnologías y plugins de los frameworks

FRAMEWORK	SOPORTE CON OTRAS TECNOLOGIAS Y PLUGINS
NetBeans (RCP)	Puede ser mejorado y aumentado por el uso de módulos (plug-ins) que permiten extender sus características principales, además por ser un producto libre y gratuito permite la integración con otras tecnologías.
Eclipse(RCP)	Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Permite la integración con varias tecnologías.
JMatter	JMatter provee todas aquellas cosas que son comunes en las aplicaciones, como cuadros de autenticación, validación, persistencia etc., por esta razón no posee plug-ins lo que dificulta extender sus características además no puede integrarse fácilmente con otras tecnologías.
GlueMvc	Debido a que GlueMvc se instala como una biblioteca en su IDE favorito no se puede añadir plug-ins adicionales a este, pero si puede integrarse con algunas tecnologías fácilmente.
OpenSwing	Puede ser mejorado con algunos (plug-ins) de terceros muy comunes y de código abierto, y puede integrarse fácilmente con otras tecnologías como ORM y RMI.

Elaborado por: Los autores

3.3.2.1.2. Documentación

Analiza la cantidad y la calidad de la documentación existente para el aprendizaje de cada framework y así poder resolver dudas y problemas.

Es así que para tomar en cuenta la cantidad de documentación existente de cada framework se ha tomado en cuenta a los tres motores de búsqueda de internet más usados en el mundo, como son Google, Yahoo y Bing según el NetMarketShared (2013). Las consultas realizadas consistieron en verificar el número de resultados que arrojaban cada uno de los buscadores utilizando la siguiente sintaxis: nombre del framework + framework + java.

A continuación se ilustra en la tabla III.IX el orden de los frameworks de acuerdo al número de resultados arrojados por los buscadores:

Tabla III.IX Cantidad de documentación obtenida por los buscadores

FRAMEWORK	BUSCADOR			
	Google	Yahoo	Bing	TOTAL
Eclipse RCP	20.100.000	3,280,000	3.000.000	23.100.000
Netbeans RCP	3.650.000	1,800,000	1.650.000	5.300.000
JMatter	52.500	2,53	2.300	54.803
OpenSwing	9.390	4,33	4.300	13.694
GlueMvc	5.760	808	808	7.376

Elaborado por: Los autores

3.3.2.1.3. Comunidad

Analiza la cantidad de wikis, foros y blogs que tiene cada frameworks en caso de estudio. Al igual que en el indicador de la documentación se utilizara la misma técnica de análisis usando los tres motores de búsqueda (Google, Yahoo y Bing) y con la siguiente sintaxis: comunidad + nombre del framework + java. A continuación se ilustra en la tabla III.X el orden de los frameworks de acuerdo al número de resultados arrojados por los buscadores:

Tabla III.X Cantidad de comunidades obtenida por los buscadores

FRAMEWORK	BUSCADOR			Total
	Google	Yahoo	Bing	
Netbeans RCP	247.000	125000	146.000	518.000
Eclipse RCP	92.200	97700	129.000	318.900
OpenSwing	891	17	15	923
JMatter	311	10	13	334
GlueMvc	8	0	0	8

Elaborado por: Los autores

3.3.2.1.4. Resultados

En la tabla III.XI se observa el resultado del análisis del parámetro Facilidades para cada uno de los frameworks que son investigados.

Tabla III.XI Resultado del análisis Facilidades

FRAMEWORK INDICADOR	NetBeans RCP	Eclipse RCP	JMatter	GlueMvc	OpenSwing
Soporte con otras tecnologías y plugins	4	4	1	2	3

Documentación	3	4	2	1	2
Comunidad	4	3	2	1	2
TOTAL/4	3,66	3,66	1,66	1,33	2,33

Elaborado por: Los autores

$$CTP_{Netbeans} = \sum_{i=1}^3 4 + 3 + 4 = \frac{11}{3} = 3,66$$

$$CTP_{Eclipse} = \sum_{i=1}^3 4 + 4 + 3 = \frac{11}{3} = 3,66$$

$$CTP_{JMeter} = \sum_{i=1}^3 1 + 2 + 2 = \frac{5}{3} = 1,66$$

$$CTP_{GlueMvc} = \sum_{i=1}^3 2 + 1 + 1 = \frac{4}{3} = 1,33$$

$$CTP_{OpenSwing} = \sum_{i=1}^3 3 + 2 + 2 = \frac{7}{3} = 2,33$$

3.3.2.1.5. Interpretación

Netbeans RCP y Eclipse RCP poseen una gran cantidad de documentación la misma que servirá de guía al momento de desarrollar un producto software, además posee una buena cantidad de comunidades que apoyan y ayudan en la solución de problemas a los desarrolladores. Ambos frameworks se integran fácilmente con otras tecnologías mejorando así la calidad del producto desarrollado.

OpenSwing y JMeter tienen una cantidad aceptable de documentación y comunidades que dan soporte a los desarrolladores, pero OpenSwing se adapta fácilmente con otras tecnologías lo que no sucede así con JMeter.

GlueMvc carece de información y comunidades de desarrolladores volviéndose esto un problema al momento de desarrollar con este framework.

3.3.2.1.6. Resultados final del análisis realizado

En la tabla III.XII se visualiza el resultado final del análisis para la selección de los frameworks que servirán para realizar el análisis comparativo que es el objetivo principal de este trabajo investigativo.

Tabla III.XII Resultado final para la selección de los frameworks

FRAMEWORK PARAMETROS	NetBeans	Eclipse	JMatter	GlueMvc	OpenSwing
Producto	4	3,5	1,5	2	3
Facilidades	3,66	3,66	1,66	1,33	2,33
Total/2	3,83	3,58	1,58	1,66	2,66

Elaborado por: Los autores

$$PT_{Netbeans} = \frac{3,83 \times 100}{4} = 95,75\%$$

$$PT_{Eclipse} = \frac{3,58 \times 100}{4} = 89,5\%$$

$$PT_{JMatter} = \frac{1,58 \times 100}{4} = 39,58\%$$

$$PT_{GlueMvc} = \frac{1,66 \times 100}{4} = 41,5\%$$

$$PT_{OpenSwing} = \frac{2,66 \times 100}{4} = 66,5\%$$

3.3.3. Diagrama general con porcentajes de los resultados

En la figura III.7 se observa la valoración final de los frameworks con sus respectivos porcentajes ordenados de mayor a menor.

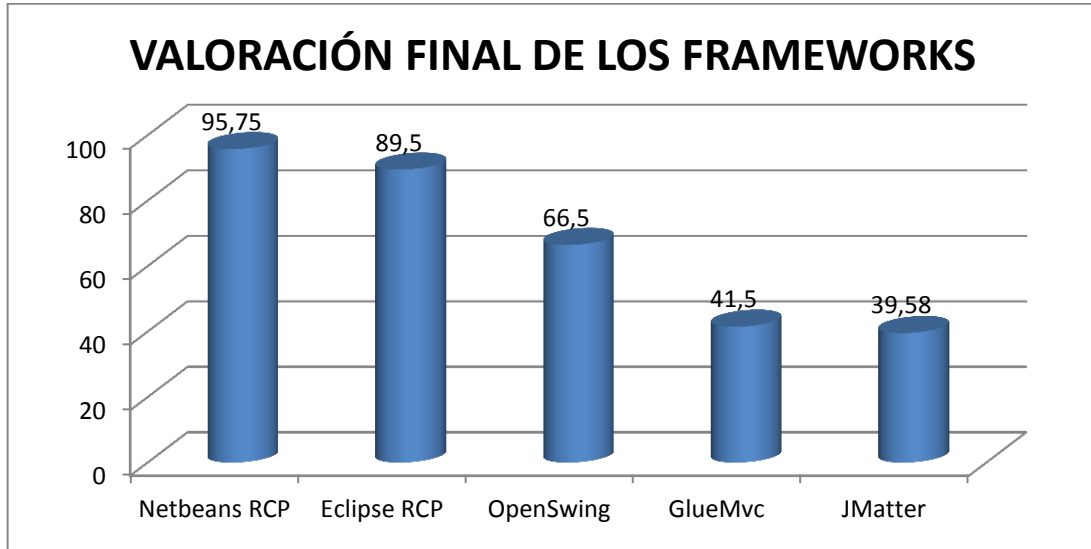


Figura III.7: Valoración final de cada framework

Fuente: Los autores

3.3.3.1. Conclusiones de los resultados obtenidos

Una vez realizado el análisis preliminar se puede decir que de entre estos cinco frameworks estudiados los que representan mejores características desde el punto de vista del desarrollo de aplicaciones de escritorio son: Netbeans RCP, Eclipse RCP y OpenSwing.

Se selecciona a Netbeans RCP y Eclipse RCP para realizar un estudio más extenso y profundo debido a que OpenSwing está cambiando constantemente de versiones y esto se vuelve un problema al momento de desarrollar un producto software de calidad.

3.4. Estudio de frameworks seleccionados

3.4.1. Netbeans (RCP)

En la figura III.8 se observa el logotipo del framework Netbeans RCP.



Figura III.8: Logotipo Netbeans RCP

Fuente: <https://netbeans.org/community/teams/evangelism/collateral.html>

3.4.1.1. Introducción

La Plataforma de NetBeans (RCP) está totalmente basado en Java. El API (AWT y Swing) se basa en el concepto de Java Estándar Edition (JSE). Las plataformas de cliente rico son flexibles y se usan especialmente cuando crecimiento de los requisitos y su arquitectura es contante. Un aspecto importante es el aumento de su productividad y flexibilidad para ser capaz equipar un producto que depende de su uso y adaptarlo a un mercado. Esto es sobre todo importante para las aplicaciones profesionales.

Sin embargo, esta asunción es basado en una condición básica: el conocimiento profesional y uso del los funcionamientos de la plataforma del cliente rica. El diseñador de la aplicación debe saber los principios principales por lo menos; sólo entonces las ventajas reales de productividad aumentada y la calidad del software mayor se logren en la práctica.

Al principio, los diseñadores pueden tener la impresión que hay una pared frente al API y conceptos que dominar. Sin embargo, una vez que usted

empieza a aprender se encuentra que hay inmensas simplificaciones posibles, qué recupera los desafíos de la fase de aprendizaje inicial.

El NetBeans (RCP) simplifica su uso y facilita la reutilización de componentes ya existentes.

3.4.1.2. Definición

Netbeans (RCP) es un framework genérico para el desarrollo de aplicaciones de escritorio basado en Swing independiente de Netbeans IDE y que proporciona la "base" que antes todos los desarrolladores tenían que escribir como la conexión de las acciones de los elementos del menú, los elementos de la barra de herramientas y atajos de teclado, la gestión de ventanas, y así sucesivamente.

3.4.1.3. Arquitectura

En la figura III.9 se observa la arquitectura de la plataforma de desarrollo Netbeans RCP.

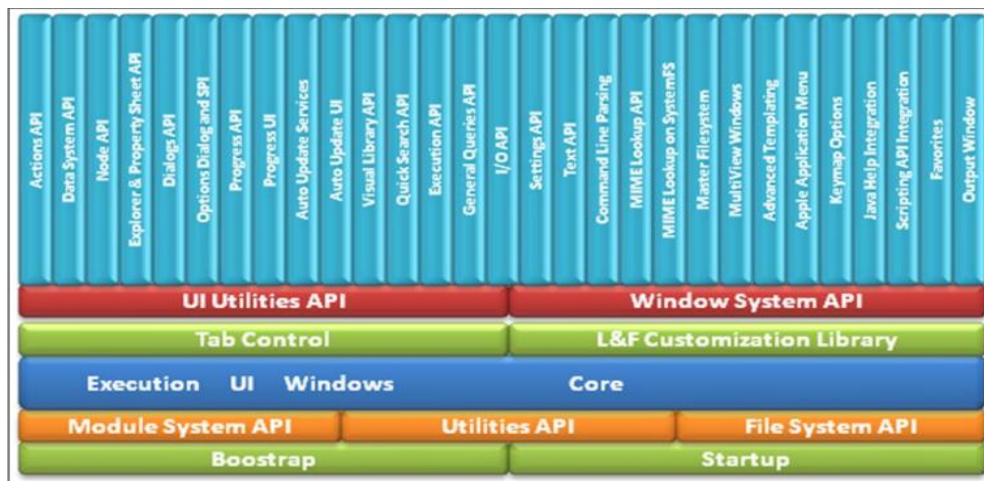


Figura III.9: Arquitectura de la plataforma de desarrollo NetBeans RCP

Fuente: Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO. Meza Ortiz, Carlos Omar

El contenedor base de ejecución “NetBeans runtime container” es el nombre colectivo de los siguientes módulos de la plataforma NetBeans. La figura III.10 muestra los elementos del contenedor de ejecución de Netbeans.

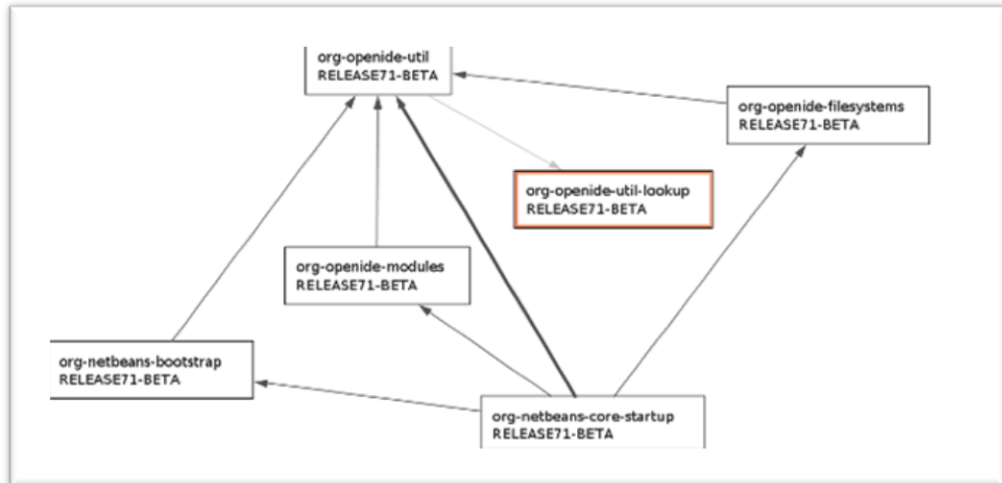


Figura III.10: Contenedor de ejecución NetBeans

Fuente: <http://platform.netbeans.org/tutorials/nbm-runtime-container.html>

En la figura III.11 se observa la iteración de los módulos de una aplicación realizada en Netbeans RCP.

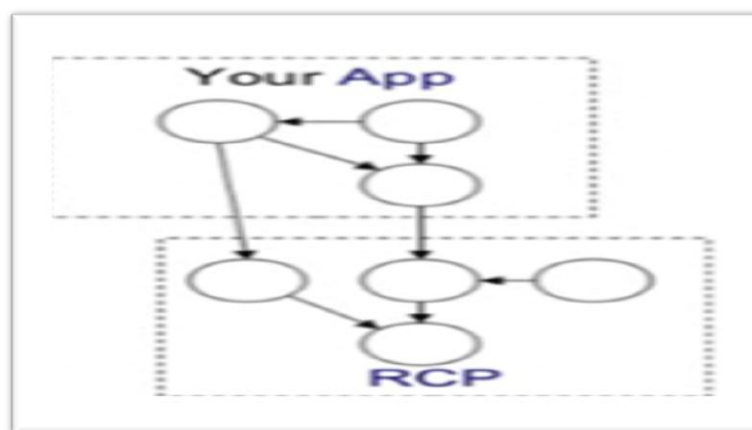


Figura III.11: Interacción de los módulos de una aplicación en NetBeans Platform

Fuente: <http://platform.netbeans.org/tutorials/nbm-10-top-apis.html>

A continuación se realiza una breve descripción de los módulos antes mencionados

Startup (org-netbeans-core-startup) Proporciona el método principal de su aplicación, así como todo el código necesario para la puesta en marcha.

Bootstrap (org-netbeans-bootstrap): permite que el contenedor en tiempo de ejecución entienda que es un módulo y cómo cargar y componerlo en una sola aplicación.

Filesystem API (org-openide-filesystems): Le da a su aplicación un sistema de archivos virtual.

Module System API (org-openide-modules): Le da acceso al ciclo de vida de los módulos de la aplicación.

Lookup API (org-openide-util-lookup): Proporciona un mecanismo de comunicación inter-genérico para modular la interacción.

Utilities API (org-openide-util)- Incluye varias clases de servicios públicos compartidos entre los otros módulos en el contenedor en tiempo de ejecución

3.4.1.4. Características

Características que se mencionan según los editores de la revista de ciencia y tecnología [1].

- Marco de trabajo para la interfaz gráfica basada en el paquete Swing.
- Editor de datos.
- Personalización de pantalla.
- Asistentes.
- Sistema de datos.
- Internacionalización.

- Sistema de ayuda.
- Facilidad de integración.
- Alto grado de cohesión y coherencia.

3.4.2. Eclipse RCP

La figura III.12 muestra el logotipo del framework Eclipse RCP.



Figura III.12: Logotipo Eclipse RCP

Fuente: http://www.rcptraining.de/files/2012/01/eclipse100_v-half.png

3.4.2.1. Introducción

Eclipse se inició originalmente como un sistema modular de aplicación IDE. En el año 2004, se publicó la versión de Eclipse 3.0 que podía ser re-utilizado para construir aplicaciones independientes basadas en la misma tecnología que el IDE de Eclipse [14].

3.4.2.2. Definición

Eclipse RCP es una plataforma para construir y desplegar aplicaciones cliente rico. Incluye Equinox, una estructura de componentes basados en el estándar OSGi, la capacidad de desplegar aplicaciones nativas de interfaz gráfica de usuario a una variedad de sistemas operativos de escritorio, como Windows,

Linux y Mac OSX y un mecanismo de actualización integrado para la implementación de aplicaciones de escritorio desde un servidor central.

La plataforma Eclipse RCP (Rich Client Platform) permite a los desarrolladores utilizar la arquitectura de Eclipse para el diseño de aplicaciones flexibles y extensibles con un montón de funcionalidades ya existentes. Los primeros pasos con Eclipse RCP puede ser lento y difícil.

3.4.2.3. Arquitectura

La base para Eclipse es la Plataforma de cliente enriquecido (del Inglés Rich Client Platform RCP). En la figura III.7 se pueden observar los componentes que constituyen la plataforma de cliente enriquecido:

- ✓ Plataforma principal - inicio de Eclipse, ejecución de plugins
- ✓ OSGi - una plataforma para bundling estándar.
- ✓ El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- ✓ JFace - manejo de archivos, manejo de texto, editores de texto
- ✓ El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

Los widgets de Eclipse están implementados por una herramienta de widget para Java llamada SWT, a diferencia de la mayoría de las aplicaciones Java, que usan las opciones estándar Abstract Window Toolkit (AWT) o Swing. La interfaz de usuario de Eclipse también tiene una capa GUI intermedia llamada JFace, la cual simplifica la construcción de aplicaciones basadas en SWT. En la figura III.13 se observa los elementos básicos del framework Eclipse RCP.

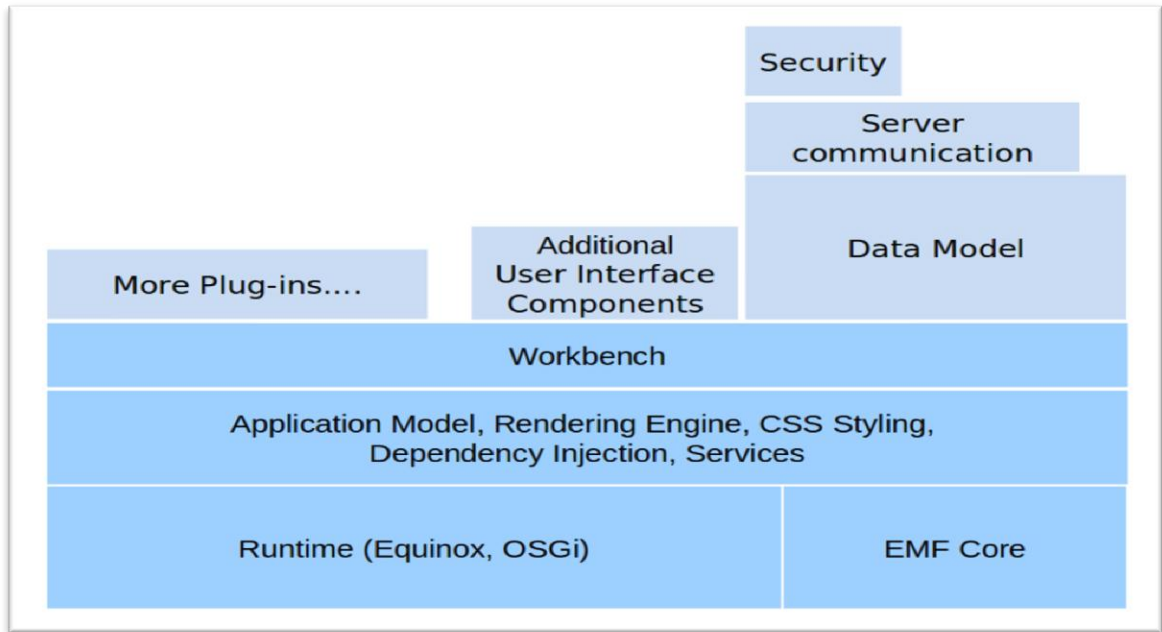


Figura III.13: Elementos básicos de Eclipse RCP

Fuente: http://www.vogella.com/tutorials/EclipseRCP/article.html#architecture_eclipsercp

3.4.2.4. Características

Entres las principales características de Eclipse RCP podemos mencionar las siguientes [22]:

- Eclipse dispone de un Editor de texto con resaltado de sintaxis.
- La compilación es en tiempo real.
- Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.
- Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversión. e integración con Hibérnate.

3.5. Análisis comparativo de los frameworks seleccionados

3.5.1. Determinación de los parámetros de evaluación de los frameworks

Se definen a continuación el modelo de Calidad que permitirá evaluar los frameworks Netbeans RCP y Eclipse RCP, se definen características y subcaracterísticas del Modelo de Calidad prescrito en la Norma ISO/IEC 9126-2 y ISO/IEC 9126-3, las mismas que servirán para evaluar la calidad de los prototipos desarrollados y determinar el framework que permite desarrollar un producto de mayor calidad [17].

3.5.2. Selección de características, subcaracterísticas y métricas para el análisis de los módulos de prueba

La norma NC ISO/IEC 12119 “Paquete de software. Requerimientos de calidad y pruebas” afirma que las tres primeras características de calidad son obligatorias para realizar la medición (funcionalidad, fiabilidad y usabilidad). Y las tres últimas son opcionales (eficiencia, portabilidad y mantenibilidad).

En esta parte del análisis se ha tomado en cuenta medir la eficiencia, cabe mencionar que no todas las subcaracterísticas definidas en esta norma son aplicables por lo que solo se seleccionaron las que se pueden aplicar al módulo de prueba.

En la tabla III.XIII se observa las características y subcaracterísticas para realizar el análisis en los módulos de prueba.

Tabla III.XIII Selección de características, subcaracterísticas y métricas

Característica	Subcaracterística	Peso	Observación
Funcionalidad	Adecuación	A	Ninguna
	Exactitud	A	Ninguna
	Interoperabilidad Seguridad	N/A	El sistema SISCOP no se relaciona con otro sistema por esta razón no se medirá esta subcaracterística. El módulo desarrollado no está implementado con controles de seguridad por lo tanto no se medirá esta subcaracterística.
Confiabilidad	Madurez	A	Ninguna
	Tolerancia ante fallos	A	Ninguna
	Recuperabilidad	A	Ninguna
Usabilidad	Comprensibilidad	A	Ninguna
	Atractibilidad	N/A	Los usuarios que manejan el sistema son pocos y el sistema posee una interfaz que satisface sus necesidades.
	Operabilidad	A	Ninguna
	Facilidad de aprendizaje	N/A	Los usuarios que manejan el sistema son pocos y no existen cambios frecuentes de personal.
Eficiencia	Rendimiento	M	Ninguna
	Utilización de recursos	M	Ninguna

Elaborado por: Los autores

3.5.3. Determinación de la importancia de las características y subcaracterísticas a evaluar

En la tabla III.XIV se visualiza la importancia cada una de las características y subcaracterísticas que se van a evaluar.

Tabla III.XIV Importancia de las características y subcaracterísticas a evaluar

Característica	Subcaracterística	Peso(Alto/Medio/Bajo)
Funcionalidad	Adecuación	A
	Exactitud	A
Confiabilidad	Madurez	A
	Tolerancia a fallos	A
	Recuperabilidad	A
Usabilidad	Comprensibilidad	A
	Operabilidad	A
Eficiencia	Rendimiento	M
	Utilización de Recursos	M

Elaborado por: Los autores

Equivalencia del peso

En la tabla III.XV se visualiza la equivalencia de los pesos para cada una de las de las características y subcaracterísticas que se van a evaluar.

Tabla III.XV Equivalencia del peso

Peso	Valor
A	1
M	0,75
B	0,5

Elaborado por: Los autores

3.5.4. Determinación de los módulos a probar

3.5.4.1. Módulo: Contabilidad

El análisis de este módulo permitirá evaluar la calidad del software final, en función de las características y subcaracterísticas de calidad descritos anteriormente, ya que el éxito del software en el mercado depende de gran parte de la calidad, porque es el usuario final o consumidor final el que finalmente acepta o rechaza el producto.

3.5.4.2. Descripción del módulo de prueba

Antes de iniciar con el desarrollo del módulo de prueba, se precisa crear un proyecto llamado coacChunchiPrueba y además una base de datos llamada COOPERATIVA en el motor de base de datos PostGreSql.

Para seleccionar el módulo de prueba a desarrollar se ha tomado en cuenta la importancia y la frecuencia con la que se usará dicho módulo de la aplicación.

3.5.4.3. Módulo Contabilidad

Este módulo permitirá trabajar con los siguientes objetos contables como son: Cuenta, Registro, Parcial y Registro_Cuenta, las mismas que permitirán

insertar, eliminar y modificar los datos en cada una de estas y generar los respectivos reportes. Estos procedimientos son realizados a través de persistencia con la base de datos, para lo cual se realizo el mapeo utilizando la librería EclipseLink lo cual facilita y ahorra tiempo en el desarrollo e implementación de estos procesos.

El desarrollo detallado de los módulos de prueba en cada uno de los frameworks seleccionados se encuentra en el ANEXO I.

3.5.5. Análisis comparativo de los frameworks Netbeans (RCP) y Eclipse (RCP)

Para realizar el análisis comparativo vamos a usar criterios establecidos por organizaciones dedicadas a estimar la calidad de sistemas software.

En este análisis usaremos la escala de valores del Modelo Sistemico para estimar la calidad de los Sistemas de Software (MOSCA) desarrollado en la Universidad Simón Bolívar por LISI (Laboratorio de Investigación en Sistemas de Información) en Caracas Venezuela, este utiliza el modelo de calidad de la Norma ISO/IEC 9126 y para la valoración de los procesos de software utiliza la Norma ISO/IEC 15504 conocida como SPICE, en el cual determina que los valores aceptables deben ser mayores o iguales al 0.75 (75%) [16].

En la figura III.14 se observa la escala de valores del modelo MOSCA.

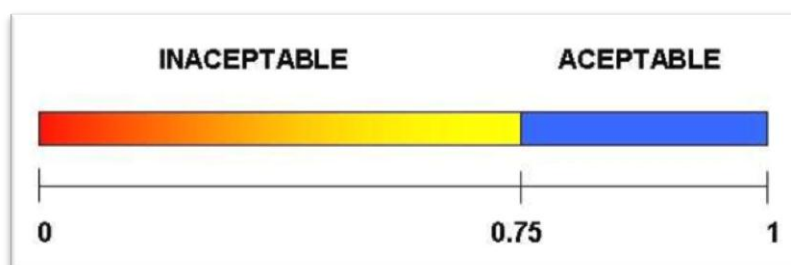


Figura III.14: Modelo MOSCA

Fuente: <http://bibdigital.epn.edu.ec/bitstream/15000/2194/1/CD-2950.pdf>

3.5.6. Funcionalidad

3.5.6.1. Análisis de las subcaracterísticas de Funcionalidad

3.5.6.1.1. Adecuación

Esta es la característica de funcionalidad esencial y se refiere a la adecuación (a la especificación) de las funciones del software [16].

Para medir la métrica Adecuación se utilizará la siguiente fórmula:

$$X_n = 1 - \frac{A}{B}$$

Dónde:

X= Resultado del indicador adecuación

n= nombre del framework

A= Número de funciones en las cuales se detectaron problemas en la evaluación.

B= Número de funciones evaluadas.

Interpretación del valor obtenido: $0 \leq X \leq 1$

A mayor cercanía al 1 resultará más adecuada.

En la tabla III.XVI se muestra los valores obtenidos para la métrica adecuación, los datos para estos cálculos se detalla en el ANEXO II

Tabla III.XVI Valores obtenidos para la métrica adecuación

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=2 B=13	0,846	84,6%	Documento de especificación de requisitos
Módulo E	A=5 B=13	0,615	61,5%	Documento de especificación de requisitos

Elaborado por: Los autores

$$X_{Netbeans RCP} = 1 - \frac{2}{13} = 0,846$$

$$X_{Eclipse RCP} = 1 - \frac{5}{13} = 0,615$$

En la figura III.15 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica Adecuación.

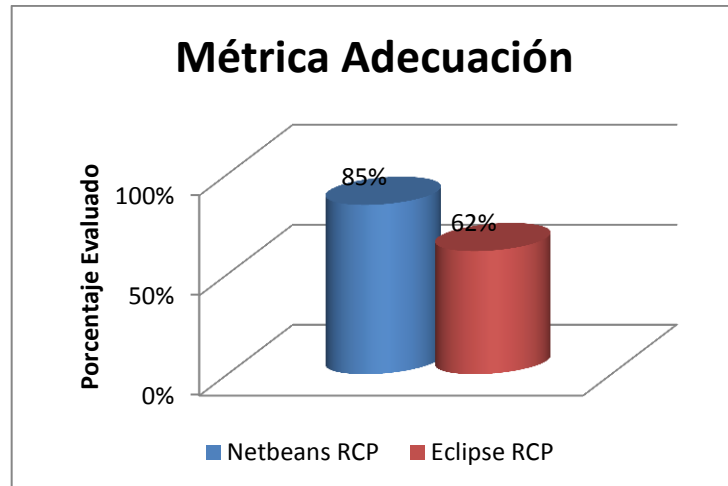


Figura III.15: Métrica Adecuación

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XVI se determina que el prototipo desarrollado con Netbeans RCP presenta un mejor cumplimiento de las especificaciones de requerimientos de software en comparación al prototipo desarrollado con Eclipse RCP.

3.5.6.1.2. Exactitud

Este indicador mide la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado [16].

Para medir la métrica exactitud se utilizará la siguiente fórmula:

$$X_n = \frac{A}{B}$$

Dónde:

X= Resultado del indicador exactitud

n= Nombre del framework

A= Número de datos implementados con los niveles de precisión especificados, confirmado en la evaluación.

B= Número de datos que requirieron los niveles de precisión.

Interpretación del valor obtenido: $0 \leq X \leq 1$

A mayor cercanía al 1 resultará más adecuada.

En la tabla III.XVII se muestra los valores obtenidos para la métrica exactitud, los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XVII Valores obtenidos para la métrica exactitud

Datos Módulo	Total A	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=9 B=9	1	100%	Evaluación GUI con el usuario
Módulo E	A=3 B=9	0,333	33,3%	Evaluación GUI con el usuario

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{9}{9} = 1$$

$$X_{EclipseRCP} = \frac{3}{9} = 0,333$$

En la figura III.16 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica exactitud.

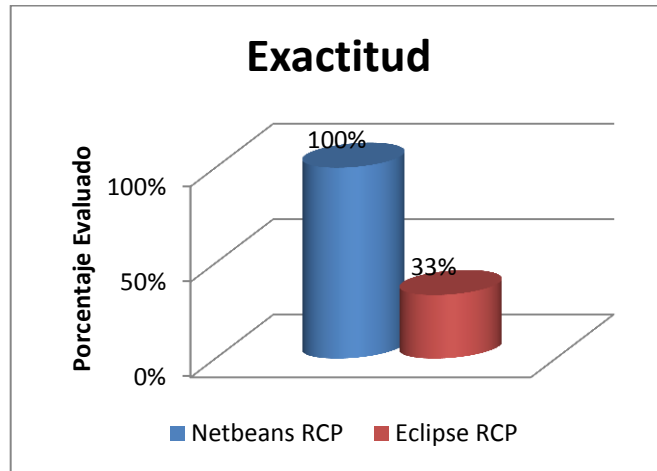


Figura III.16: Métrica exactitud

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XVII se determina que el prototipo desarrollado con Netbeans RCP presenta el resultado correcto y con el grado de precisión que el usuario necesita en los campos numéricos de ingreso y salida en la interfaz de usuario en comparación al prototipo desarrollado con Eclipse RCP.

3.5.6.2. Confiabilidad

3.5.6.2.1. Madurez

Este indicador mide la capacidad del producto software para evitar fallos provocados por errores en el software [16].

Para medir la métrica madurez se utilizará la siguiente fórmula:

$$X_n = \frac{A}{B}$$

Dónde:

X= Resultado del indicador madurez

n= Nombre del framework

A= Número de casos de prueba diseñados en el plan de pruebas y confirmados en la revisión.

B= Número de casos de prueba requeridos.

Interpretación del valor obtenido: $0 \leq X$

Donde X sea mayor será más adecuado.

Los valores obtenidos para la métrica madurez se observa en la tabla III.XVIII.

Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XVIII Valores obtenidos para la métrica madurez

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=23 B=23	1	100%	Plan de pruebas
Módulo E	A=8 B=23	0,347	34,7%	Plan de pruebas

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{23}{23} = 1$$

$$X_{EclipseRCP} = \frac{8}{23} = 0,347$$

En la figura III.17 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica madurez.

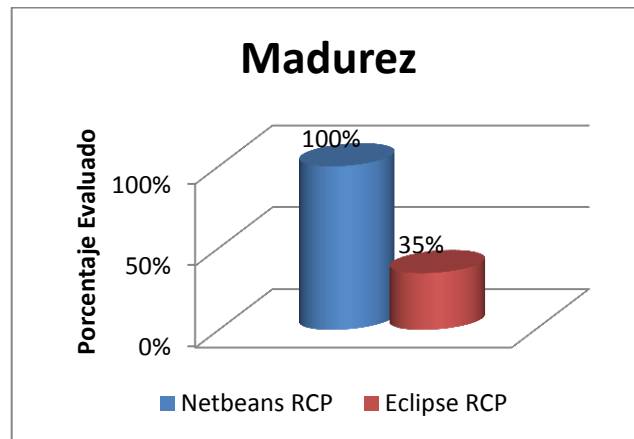


Figura III.17: Métrica Madurez

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XVIII se determina que el prototipo desarrollado con Netbeans RCP superó los casos de prueba de ingreso, actualización y eliminación de los datos sin errores en relación a Eclipse RCP.

3.5.6.2.2. Tolerancia ante fallos

Esta subcaracterística mide la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz [16].

Para medir la métrica tolerancia ante fallos se utilizará la siguiente fórmula:

$$X_n = \frac{A}{B}$$

Dónde:

X= Resultado final tolerancia ante fallos

n= Nombre del framework

A= Número de fallos del sistema

B= Número de fallos totales.

Interpretación del valor obtenido: $0 \leq X \leq 1$

Donde X sea cercano a 1 será más adecuado.

Los valores obtenidos para la métrica evitación de fallos se observa en la tabla III.XIX. Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XIX Valores obtenidos para la métrica tolerancia

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=4 B=4	1	100%	Registro de errores, fallas y caídas del sistema
Módulo E	A=1 B=4	0,25	25%	Registro de errores, fallas y caídas del sistema

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{4}{4} = 1$$

$$X_{EclipseRCP} = \frac{1}{4} = 0,25$$

En la figura III.18 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica no fallos.

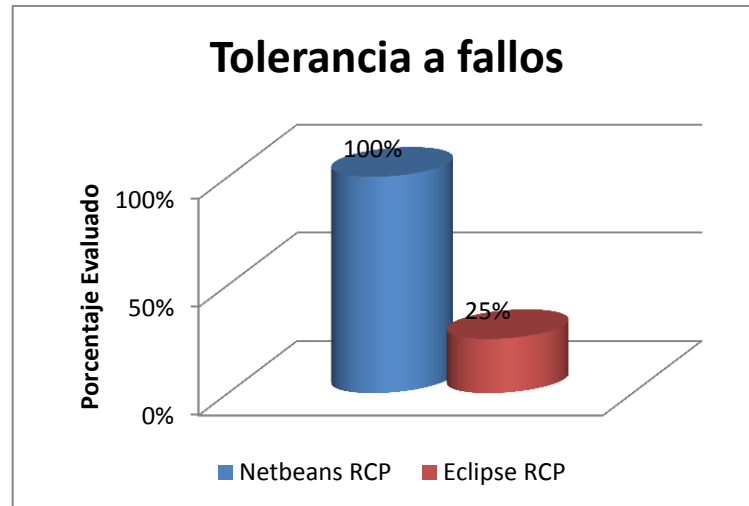


Figura III.18: Métrica No fallos

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XIX se determina que el prototipo desarrollado con Netbeans RCP no presento fallas comunes como falla de la conexión a la base de datos, entre otras evaluadas en comparación con el prototipo desarrollado con Eclipse RCP.

3.5.6.2.3. Recuperabilidad

Esta subcaracterística mide la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo [16].

Para medir la métrica recuperabilidad se utilizará la siguiente fórmula:

$$X_n = \frac{T}{N}$$

Dónde:

X= Tiempo medio de caída

n= Nombre del framework

T= Tiempo total de caída

N= Número de caídas observadas

Interpretación del valor obtenido: $0 < X$

Mientras menor sea X mejor será la recuperabilidad.

En la tabla III.XX se observa los valores obtenidos para la métrica recuperabilidad. El valor estándar definido por la empresa para este parámetro es 30min, los datos para estos cálculos se detallan en el ANEXO II.

Tabla III.XX Valores obtenidos para la métrica recuperabilidad

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	T=0 N=0	0 min	100 %	Registro de errores, fallas y caídas del sistema
Módulo E	T=85min N=7	12,14 min	40,5 %	Registro de errores, fallas y caídas del sistema

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{0}{0} = 0$$

$$X_{EclipseRCP} = \frac{12,14}{30} = 0,405$$

En la figura III.19 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica tiempo promedio de falla para la subcaracterística Recuperabilidad.

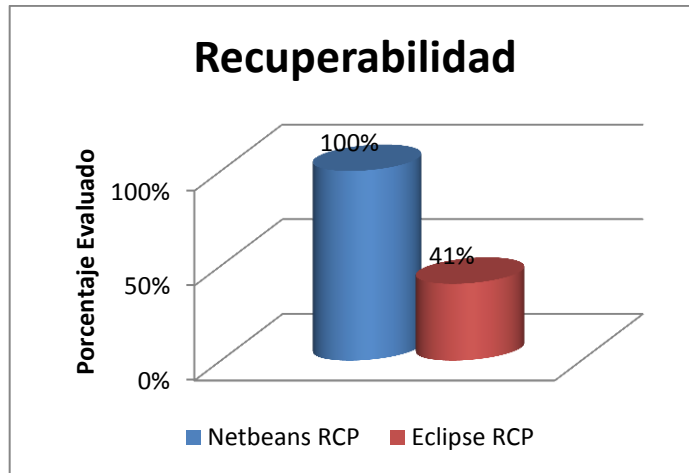


Figura III.19: Métrica tiempo promedio de falla

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XX se determina que el prototipo desarrollado con Netbeans RCP al no presentar fallos durante la evaluación se puede restablecer más rápido en comparación con el prototipo hecho con Eclipse RCP.

3.5.6.3. Usabilidad

3.5.6.3.1. Comprensibilidad

Esta métrica mide la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso [16].

Para medir la métrica comprensibilidad se utilizará la siguiente fórmula:

$$X_n = \frac{A}{B}$$

Dónde:

X= Resultado final comprensibilidad

n= Nombre del framework

A= Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario.

B= Número total de funciones de interfaz de usuario.

Interpretación del valor obtenido: $0 \leq X \leq 1$

Mientras más cercano a 1, mejor.

Los valores obtenidos para la métrica comprensibilidad se observa en la tabla III.XXI.

Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XXI Valores obtenidos para la métrica comprensibilidad

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=9 B=13	0,692	69,2%	Pruebas realizadas con el usuario.
Módulo E	A=8 B=13	0,615	61,5%	Pruebas realizadas con el usuario.

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{9}{13} = 0,692$$

$$X_{EclipseRCP} = \frac{8}{13} = 0,615$$

En la figura III.20 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica comprensibilidad.

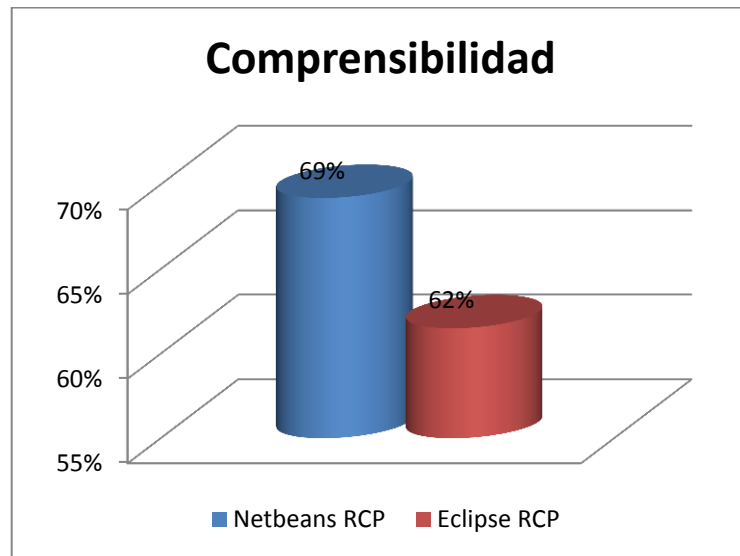


Figura III.20: Métrica comprensibilidad

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XXI se determina que el prototipo desarrollado con Netbeans RCP que las funcionalidades implementadas por éste son más fácil de entender y comprender que el prototipo desarrollado con Eclipse RCP esto significa que la aplicación se puede usar de mejor manera por los usuarios finales.

3.5.6.3.2. Operabilidad

Esta métrica mide la capacidad del producto software para permitir que el usuario lo opere y lo controle [16].

Para medir la métrica operabilidad se utilizará la siguiente fórmula:

$$X_n = \frac{A}{B}$$

Dónde:

X= Resultado final operabilidad

n= Nombre del framework

A= Número de elementos de entrada que chequean validez de los datos.

B= Número de elementos de entrada que pueden chequear la validez de los datos.

Interpretación del valor obtenido: $0 \leq X \leq 1$

Mientras más cercano a 1, mejor.

Los valores obtenidos para la métrica operabilidad se muestran en la tabla III.XXII.

Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XXII Valores obtenidos para la métrica operabilidad

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	A=6 B=7	0,857	85,7%	Pruebas realizadas con el usuario.
Módulo E	A=6 B=7	0,857	85,7%	Pruebas realizadas con el usuario.

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{6}{7} = 0,857$$

$$X_{EclipseRCP} = \frac{6}{7} = 0,857$$

En la figura III.21 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionado para la métrica operabilidad.

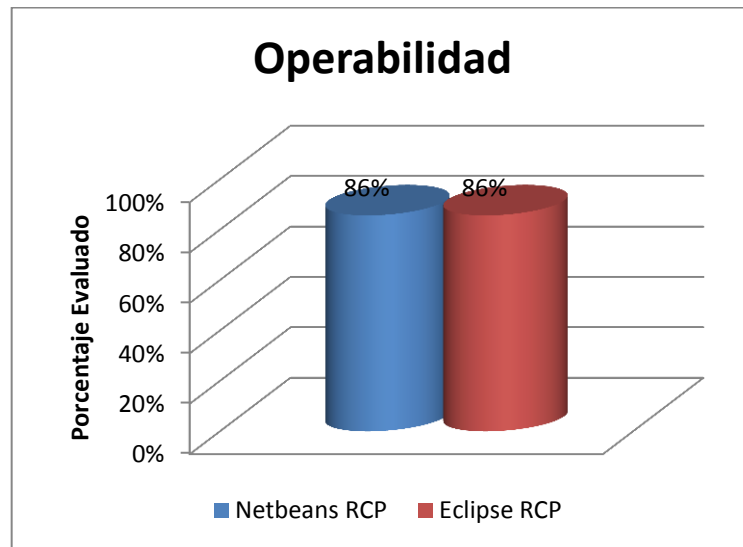


Figura III.21: Métrica operabilidad

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XXII se determina que los prototipos desarrollados con Netbeans y Eclipse RCP son muy fáciles de operar y controlar; ya que en la evaluación se pueden realizar un control efectivo de los datos ingresados.

3.5.6.4. Eficiencia

3.5.6.4.1. Rendimiento

Esta métrica mide la capacidad del producto software para ejecutar una cantidad de tareas en un determinado tiempo [16].

Para medir la métrica rendimiento se calculara el valor de:

$$X_n = \text{Número de tareas por unidad de tiempo}$$

Dónde:

X= Resultado final rendimientoo

n= Nombre del framework

Interpretación del valor obtenido: X Mientras mayor mejor.

En la tabla III.XXIII se visualiza los valores obtenidos para la métrica rendimiento.

Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XXIII Valores obtenidos para la métrica rendimiento

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	X=23/seg	1	100%	Pruebas realizadas con Jconsole
Módulo E	X=18/seg	0,783	78,3%	Pruebas realizadas con Jconsole

Elaborado por: Los autores

$$X_{NetbeansRCP} = \frac{23}{23} = 1$$

$$X_{EclipseRCP} = \frac{18}{23} = 0,783$$

En la figura III.22 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionados para la métrica número de tareas ejecutadas por unidad de tiempo que corresponde a la subcaracterística rendimiento.

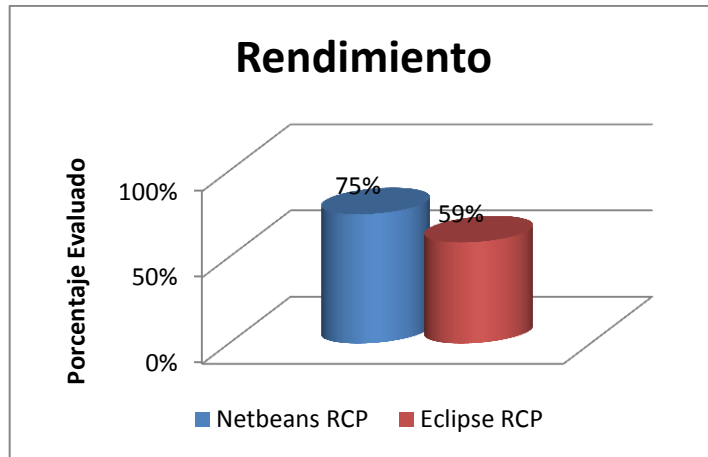


Figura III.22: Subcaracterística rendimiento

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III.XXIII se determina que el prototipo desarrollado con Netbeans RCP tiene mejor rendimiento que el desarrollado con Eclipse RCP; ya que se ejecutan un mayor número de tareas por unidad de tiempo como consecuencia son más rápidas.

3.5.6.4.2. Utilización de recursos

Esta métrica mide la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones [16].

Para medir la métrica utilización de recursos se calculara el valor de:

$$X_n = \text{Tamaño en Mb}(\text{calculado o simulado})$$

Dónde:

X= Resultado final de la utilización de recursos

n= Nombre del framework

Interpretación del valor obtenido: Mientras menor sea, mejor.

Los valores obtenidos para la métrica utilización de recursos se muestran en la tabla III.XXIV.

Los datos para estos cálculos se detallan en el ANEXO II

Tabla III.XXIV Valores obtenidos para la métrica utilización de recursos

Datos Módulo	Valores obtenidos	Resultado de la métrica	Resultado real (%)	Recursos utilizados
Módulo N	X=30,92 Mb	0,323	32,3%	Pruebas realizadas con Jconsole
Módulo E	X=18,44 Mb	1	100%	Pruebas realizadas con Jconsole

Elaborado por: Los autores

$$X_{NetbeansRCP} = 1 - \frac{30,92 - 18,44}{18,44} = 0,323$$

$$X_{EclipseRCP} = \frac{18,44}{18,44} = 1$$

En la figura III.23 se representa los porcentajes de forma gráfica obtenidos en cada una de los frameworks seleccionados para la métrica uso de memoria en Mega bits que corresponde a la subcaracterística utilización de recursos.

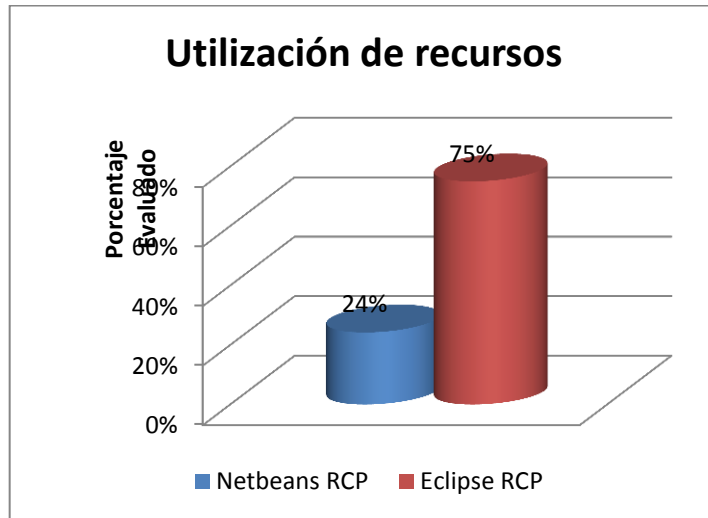


Figura III.23: Subcaracterística utilización de recurso memoria

Fuente: Autores

Interpretación de resultados

De acuerdo con el resultado obtenido en la tabla III. XXIV se determina que el prototipo desarrollado usando Eclipse RCP usa menos memoria en comparación con Netbeans RCP por lo tanto es más eficiente en uso de los recursos computacionales.

3.5.6.5. Tabulación de resultados

En esta investigación, la calidad del producto software fue evaluada a través de las siguientes características del estándar ISO/IEC 1926-2 y ISO/IEC 1926-3 como son: Funcionalidad, Confiabilidad, Usabilidad y Eficiencia.

Estas características fueron seleccionadas de acuerdo a la tabla en donde se registran la importancia de cada una de ellas.

Los promedios de calidad de cada uno de los módulos se detallan en la tabla III.XXV.

Tabla III.XXV Promedio de calidad para los módulos de prueba

Característica	Subcaracterística	Porcentajes Métrica		Peso	Porcentajes Peso	
		Netbeans RCP	Eclipse RCP		Netbeans RCP	Eclipse RCP
Funcionalidad	Adecuación	84,6	61,5	1	84,6	61,5
	Exactitud	100	33,3	1	100	33,3
Confiabilidad	Madurez	100	34,7	1	100	34,7
	Tolerancia a fallos	100	25	1	100	25
	Recuperabilidad	100	40,5	1	100	40,5
Usabilidad	Comprensibilidad	69,2	61,5	1	69,2	61,5
	Operabilidad	85,7	85,7	1	85,7	85,7
Eficiencia	Rendimiento	100	78,2	0,75	75	59
	Utilización de Recursos	32,3	100	0,75	24	75
Total					84,5	51,2

Elaborado por: Los autores

3.5.7. Diagrama general de resultados

En la figura III.24 se muestra un gráfico estadístico con los resultados totales de cada subcaracterística del estándar de calidad usados en la presente investigación.

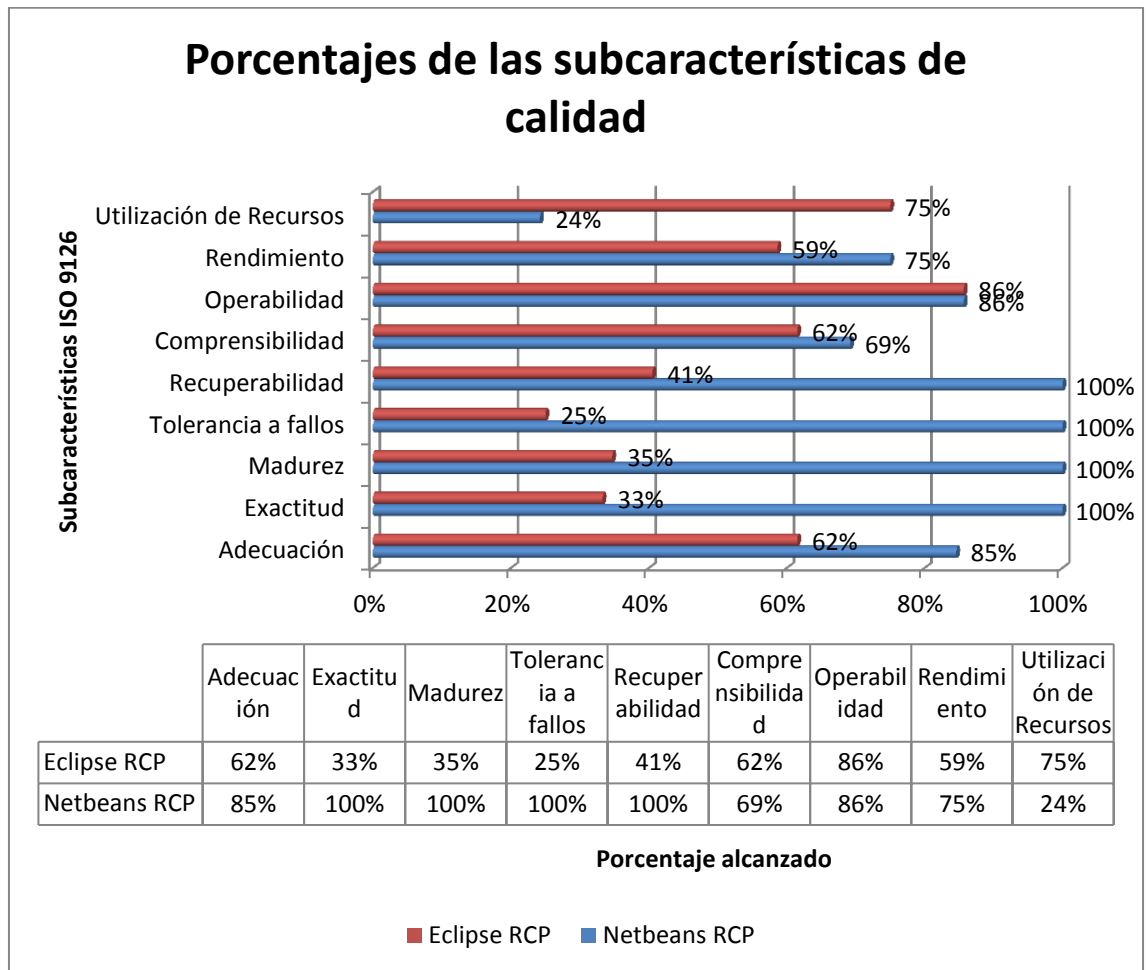


Figura III.24: Resultados totales

Fuente: Autores

En la figura III.25 se muestra un gráfico estadístico con los resultados totales de cada característica del estándar de calidad usados en la presente investigación

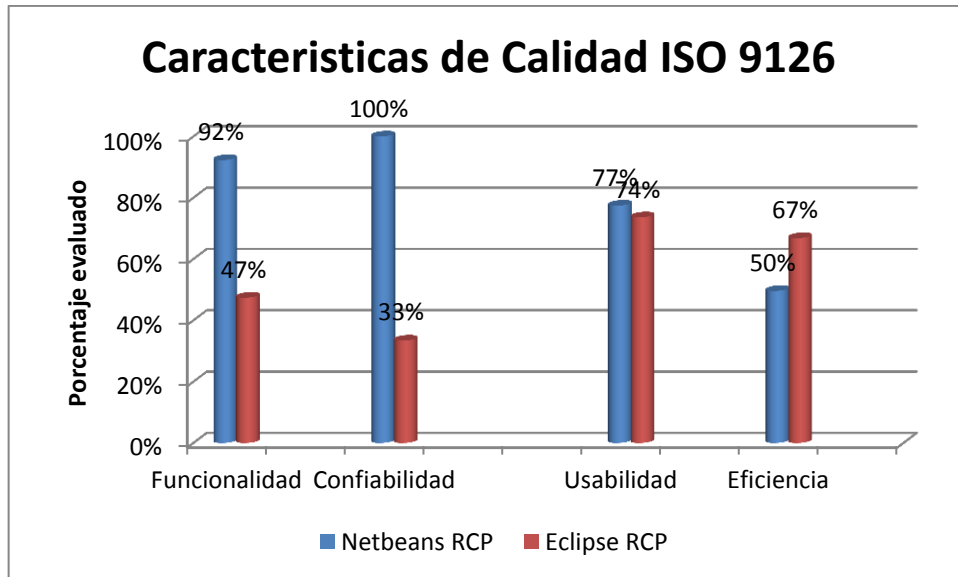


Figura III.25: Características del estándar ISO

Fuente: Autores

Interpretación

- El módulo de prueba desarrollado usando el framework Netbeans RCP alcanzo un 92% de cumplimiento de la funcionalidad requerida por el cliente en comparación con el módulo desarrollado usando Eclipse RCP que obtuvo un porcentaje de 47% de cumplimiento, esto significa que usando el framework Netbeans RCP en gran medida lograremos cumplir lo que pide el cliente.
- El módulo de prueba implementado con el framework Netbeans RCP alcanzó un 100% de confiabilidad que es un excelente nivel para esta característica en las mediciones realizadas esto es tiempos de recuperación cortos dentro de los límites establecidos por la institución en comparación al módulo desarrollado en Eclipse RCP que alcanzó un 33% de cumplimiento de ésta característica de calidad que es baja.

- De acuerdo con el cuadro estadístico los módulos de prueba desarrollados en los frameworks Netbeans RCP y Eclipse RCP presentan un porcentaje similar de 77% y 74% en la característica de usabilidad, lo que significa que el usuario tiene un buen nivel de entendimiento de las funcionalidades plasmadas en los módulos.
- De acuerdo a los porcentajes señalados en el cuadro estadístico el módulo de prueba desarrollado en Eclipse RCP tienen un rendimiento aceptable del 67% esto es un buen uso de los recursos del computador como son el procesador y la memoria en comparación al módulo implementado con Eclipse RCP.

Las características de calidad seleccionadas para la comparación de los frameworks Netbeans RCP y Eclipse RCP de esta investigación, permitió determinar el Framework que mejor calidad brinda a las aplicaciones desarrolladas con éste.

En la figura III.26 se muestra la figura estadística del resultado final de la comparación.

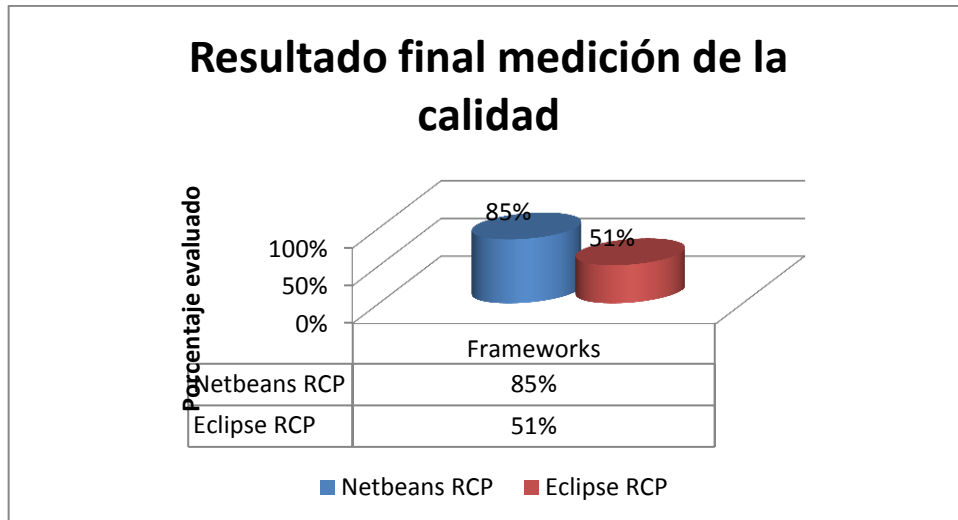


Figura III.26: Resultado final medición

Fuente: Autores

Cálculo de los porcentajes totales

Eclipse RCP

$$PTE = \frac{\sum Pes}{Ns}$$

$$PTE = \frac{62 + 33 + 35 + 25 + 41 + 62 + 86 + 59 + 75}{9}$$

$$PTE = 51\%$$

Netbeans RCP

$$PTN = \frac{\sum Pns}{Ns}$$

$$PTE = \frac{85 + 100 + 100 + 100 + 100 + 69 + 86 + 75 + 24}{9}$$

$$PTN = 85\%$$

Dónde:

PTE: Puntaje total del módulo Eclipse RCP

Pes: Puntaje individual alcanzado por cada sub característica

Ns: Número de subcaracterísticas medidas

PTN: Puntaje total del módulo Netbeans RCP

Pes: Puntaje individual alcanzado por cada sub característica

Interpretación

Una vez realizado el análisis comparativo y con los resultados obtenidos con cada uno de los Frameworks de desarrollo para aplicaciones de escritorio en Java, se concluye que el Framework Netbeans RCP ha obtenido el puntaje más alto con un porcentaje de 85% equivalente a **Aceptable** según la escala de medición **MOSCA**; por el contrario el Framework Eclipse RCP ha obtenido un valor de 51% equivalente a **No Aceptable** según la misma escala.

3.5.8. Análisis de los resultados obtenidos

Después del análisis comparativo realizado a los módulos de prueba desarrollados con los frameworks para aplicaciones de escritorio en Java Netbeans RCP y Eclipse RCP, podemos señalar lo siguiente:

- Usando el framework Netbeans RCP se pueden desarrollar aplicaciones con un alto grado de funcionalidad en comparación al framework Eclipse RCP, esto es que usando el framework Netbeans RCP en gran medida lograremos cumplir lo que pide el cliente.
- Al desarrollar una aplicación usando el framework Netbeans RCP se pueden obtener altos niveles de confiabilidad del producto final en comparación a las aplicaciones desarrolladas con Eclipse RCP, lo que significa que en caso de producirse errores o fallos en el sistema se requieren de tiempos cortos para volver a sus estado de funcionamiento normal.

- Con los dos frameworks analizados se logra un buen nivel de usabilidad esto es que el usuario final con conocimientos intermedios en informática puede entender sin mucho esfuerzo o explicaciones las funcionalidades implementadas en las aplicaciones.
- Al implementar aplicaciones usando el framework Eclipse RCP se obtienen un rendimiento aceptable para las mismas, esto significa que se hace un uso eficiente de los recursos del computador como la memoria y el procesador en relación a las aplicaciones desarrolladas con Netbeans RCP

3.6. COMPROBACION DE LA HIPOTESIS

3.6.1. Hipótesis

El framework de software libre Netbeans RCP para aplicaciones de escritorio en Java permitirá obtener un producto de mayor calidad que el framework Eclipse RCP.

3.6.2. Tipo de hipótesis

La hipótesis de esta investigación es de tipo Causa – Efecto.

3.6.3. Determinación de variables

- **Variable Independiente:** Netbeans RCP.
- **Variable Dependiente:** Calidad en el desarrollo de aplicaciones de escritorio en Java.

3.6.4. Operacionalización Conceptual

Tabla III.XXVI Operacionalización Conceptual

VARIABLE	TIPO	CONCEPTO
Framework Netbeans RCP.	Variable Independiente	Marco genérico para aplicaciones Swing. Que proporciona la "la base" que, antes, todos los desarrolladores tenían que escribir ellos mismos como, la conexión de las acciones de los elementos del menú, los elementos de la barra de herramientas y atajos de teclado, la gestión de ventanas, y así sucesivamente.
Calidad en el desarrollo de aplicaciones de escritorio en Java.	Variable Dependiente	Es un conjunto de normas y directrices que se llevan a cabo dentro de un proceso, utilizando para métricas que permiten medir la calidad de un producto software.

3.6.5. Operacionalización Metodológica

Tabla III.XXVII Operacionalización Metodológica

HIPÓTESIS	VARIABLES	CRITERIO	INDICADORES	TÉCNICAS	FUENTES DE VERIFICACIÓN
El framework software libre para aplicaciones de escritorio en Java Netbeans RCP permitirá obtener un producto de mayor calidad que el framework Eclipse RCP.	Desarrollo de aplicaciones de escritorio en Java	Investigación	Framework de desarrollo Netbeans RCP Framework de desarrollo Eclipse RCP	Revisión de documentos	<ul style="list-style-type: none"> ➤ Internet ➤ Manuales
	Calidad en el desarrollo de aplicaciones de escritorio en Java.	Funcionalidad	Adecuación	Observación	<ul style="list-style-type: none"> ➤ Prototipo con Netbeans RCP y Eclipse RCP
			Exactitud	Observación	<ul style="list-style-type: none"> ➤ Prototipo con Netbeans RCP y Eclipse RCP
		Confiabilidad	Madurez	Observación	<ul style="list-style-type: none"> ➤ Prototipo con Netbeans RCP y Eclipse RCP
			Tolerancia a fallos	Observación	<ul style="list-style-type: none"> ➤ Prototipo con Netbeans RCP y Eclipse RCP
			Recuperabilidad	Observación	<ul style="list-style-type: none"> ➤ Prototipo con Netbeans RCP y Eclipse RCP

Continuación...

HIPÓTESIS	VARIABLES	CRITERIO	INDICADORES	TÉCNICAS	FUENTES DE VERIFICACIÓN
El framework software libre para aplicaciones de escritorio en Java Netbeans RCP permitirá obtener un producto de mayor calidad que el framework Eclipse RCP.	Calidad en el desarrollo de aplicaciones de escritorio en Java.	Usabilidad	Comprensibilidad	Observación	➤ Prototipo con Netbeans RCP y Eclipse RCP
			Operabilidad	Observación	➤ Prototipo con Netbeans RCP y Eclipse RCP
		Eficiencia	Rendimiento	Observación	➤ Prototipo con Netbeans RCP y Eclipse RCP ➤ Jconsole
			Utilización de recursos	Observación	➤ Prototipo con Netbeans RCP y Eclipse RCP ➤ Jconsole

Para la demostración de la hipótesis se utilizó los datos de la tabla III.XXVIII obtenido en el análisis:

Tabla III.XXVIII Promedio de calidad para los módulos de prueba

Característica	Subcaracterística	Porcentajes Métrica		Peso	Porcentajes Peso	
		Netbeans RCP	Eclipse RCP		Netbeans RCP	Eclipse RCP
Funcionalidad	Adecuación	84,6	61,5	1	84,6	61,5
	Exactitud	100	33,3	1	100	33,3
Confiabilidad	Madurez	100	34,7	1	100	34,7
	Tolerancia a fallos	100	25	1	100	25
	Recuperabilidad	100	40,5	1	100	40,5
Usabilidad	Comprensibilidad	69,2	61,5	1	69,2	61,5
	Operabilidad	85,7	85,7	1	85,7	85,7
Eficiencia	Rendimiento	100	78,2	0,75	75	59
	Utilización de Recursos	32,3	100	0,75	24	75
Total					84,5	51,2

Fuente: Los autores

En la figura III.27 se muestra un gráfico estadístico con los resultados totales de cada subcaracterística del estándar de calidad ISO 9126 usados en la presente investigación

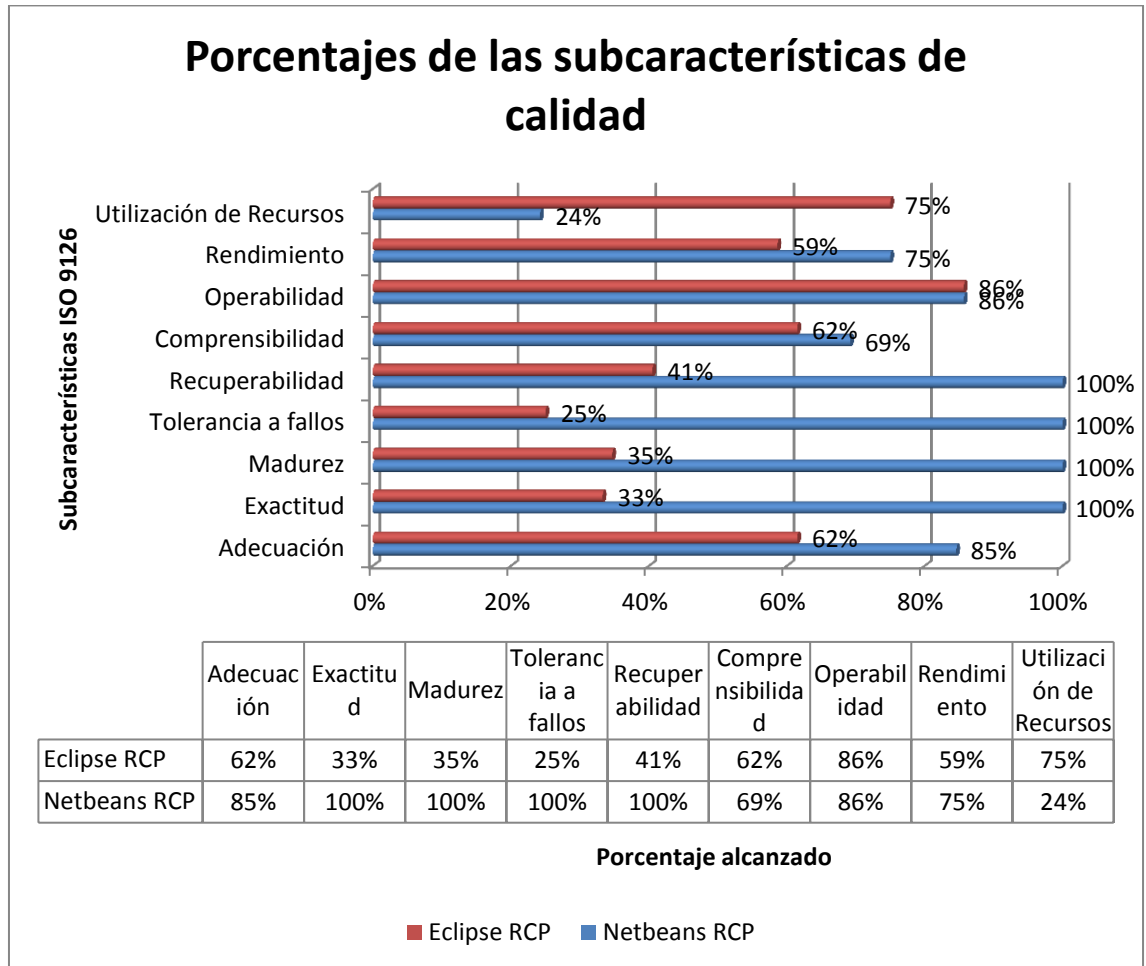


Figura III.27: Resultados totales

Fuente: Autores

En la figura III.28 se muestra un gráfico estadístico con los resultados totales de cada una de las características del estándar de calidad ISO 9126 usadas en la presente investigación

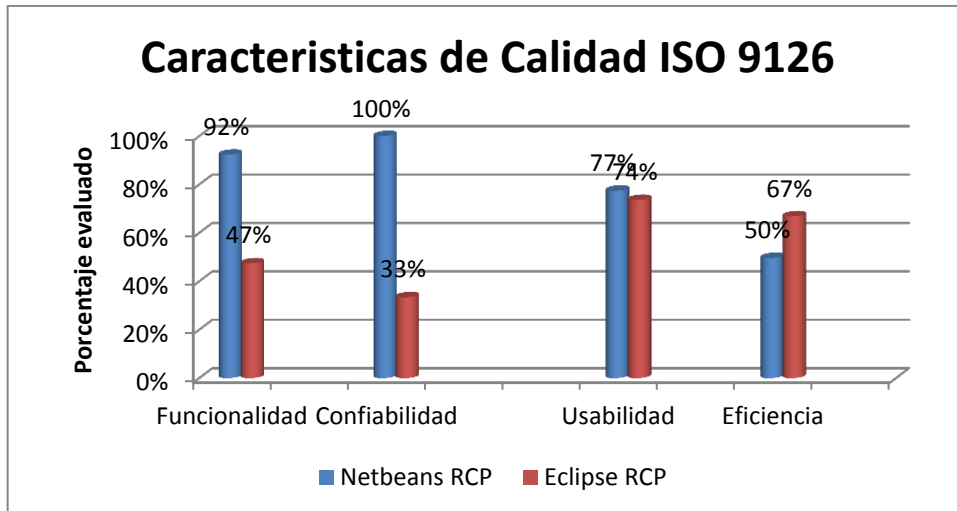


Figura III.28: Características del estándar ISO

Fuente: Autores

Las características de calidad seleccionadas para la comparación de los frameworks Netbeans RCP y Eclipse RCP de esta investigación, permitió determinar el Framework que mejor calidad brinda a las aplicaciones desarrolladas con éste. En la figura III.29 se muestra la figura estadística del resultado final de la comparación.

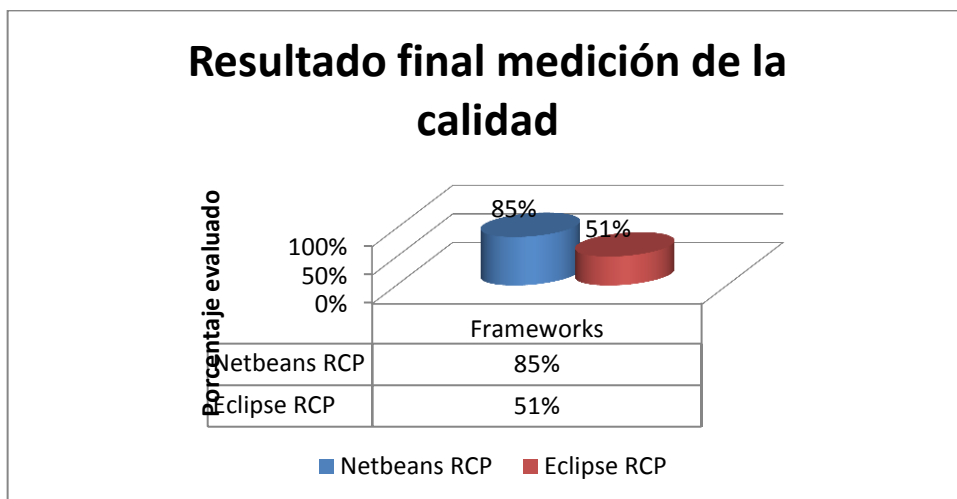


Figura III.29: Resultado final medición

Fuente: Autores

La hipótesis planteada es:

Hi: El framework de software libre Netbeans RCP para aplicaciones de escritorio en Java permitirá obtener un producto de mayor calidad que el framework Eclipse RCP.

La calidad se mide en base a las características y subcaracterísticas del estándar ISO 9126 descritas en la tabla III.XXVIII; haciendo referencia a la figura III.29 en la que se observa que el framework Netbeans RCP obtuvo el porcentaje más alto de 85% en la medición de la calidad de su módulo de prueba; superando a Eclipse RCP con 34% como consecuencia por observación directa se concluye que, la hipótesis planteada **Hi** es verdadera.

3.7. Conclusión

Se concluye que el framework para el desarrollo de aplicaciones de escritorio en Java Netbeans RCP es el más adecuado para el desarrollo de la aplicación "SISCOOP", puesto que ha tenido un mejor puntaje en las características y subcaracterísticas del estándar de calidad ISO 9126 medidas en la presente investigación que el framework Eclipse RCP.

CAPÍTULO IV

IMPLEMENTACION DE LA APLICACIÓN

4.1. Introducción

Para la implementación de la aplicación SISCOOP, caso práctico de esta tesis de grado, se ha seleccionado la metodología SCRUM por ser un proceso ágil para desarrollar software, que centra su atención en las funcionalidades con más prioridad y que pueden ser ejecutadas en un periodo corto de tiempo [15]. Esta metodología se compone de tres fases, los cuales se indica en la figura IV.30.

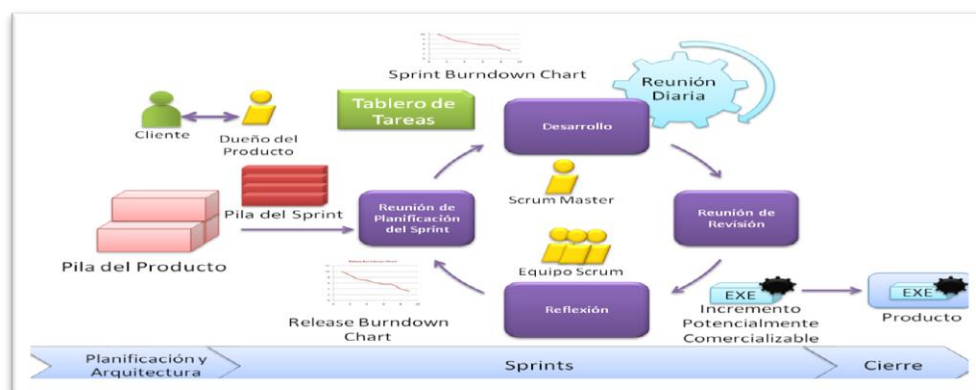


Figura IV.30: Fases de la metodología SCRUM

Fuente: <http://etherpad.proyectolatin.org/up/aaea7098b048d17f4bbe2a2bb599abeb.png>

4.2. Fase Inicial

4.2.1. Destinatarios

Scrum Master (Ing. Raúl Rosero M.)

Product Owner (Jorge Piña - Gerente)

Team (Ángel Cobo, José Manzano)

4.2.2. Definición del problema

La Cooperativa de Ahorro y Crédito Chunchi Ltda. lleva dos años y medio de funcionamiento en el mercado y actualmente no posee un sistema informático propio para sus actividades operativas como gestión de crédito, manejo de las transacciones de caja y manejo de contabilidad, frente a estas dificultades esta institución se ve en la necesidad de implantar un sistema propio de acuerdo a sus necesidades para realizar las actividades anteriormente descritas ya que al momento se encuentran arrendando el sistema CONEXUS Milenium cuyos costos son elevados e influyen considerablemente en los gastos de la institución.

4.2.3. Visión

SISCOOP es una aplicación que se centra en satisfacer la necesidad del cliente y poder contar una aplicación para la Automatización de los procesos realizados en la Cooperativa de Ahorro y Crédito Chunchi Ltda.

Se espera que al finalizar el proyecto, el cliente pueda obtener los beneficios esperados al contar con la aplicación desarrollada en su totalidad, con un nivel de calidad pactado, e implantada de forma tal que sea completamente utilizable. Esta aplicación desea ofrecer un adecuado rendimiento en los

procesos, además garantizar la integridad de los datos provistos y obtenidos; de la misma manera se enfoca en satisfacer las necesidades de los usuarios.

A la finalización del proyecto lo que se pretende es cumplir las siguientes expectativas:

- **Escalable:** El proyecto tenga la capacidad de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes del medio en el que se ve vinculado.
- **Confiable:** La solución sea capaz de proporcionar información fiable e inherentes a lo que se desea obtener, y estos den muestra de información que no de paso a equívocos.
- **Eficiente:** Que los resultados que se pretenden obtener sean accesibles en el menor tiempo requerido.
- **Fácil Uso:** El ambiente sea ágilmente entendible para las personas que se ocupen del manejo del sistema, que los procesos que se realicen sean específicos y claros y no den paso a sobrecarga de contenidos y puedan ser interpretados de distintas maneras.
- **Compatible:** La solución preste las garantías de que pueda funcionar en cualquier ambiente en el cual se le pretenda poner en funcionamiento.
- **Reusable:** La solución se encuentre en la posibilidad de que pueda ser base de la implementación de nuevos proyectos o que forme parte del manejo de algún otro sistema.
- **Portable:** El proyecto en si pueda ser trasladado de cualquier ambiente sin ningún inconveniente y de la mejor manera posible.

4.2.4. Usuarios

En esta sección se identifican los principales perfiles de usuarios del sistema y las principales funciones de cada uno de ellos.

En la tabla IV.XXIX se visualizan los principales usuarios del sistema.

Tabla IV.XXIX Usuarios del sistema

USUARIO	RESPONSABLES	DESCRIPCIÓN
Gerente	Sr. Jorge Piña	Es el encargado de administrar el sistema y la gestión de usuarios con rol autoridad.
Asesor de crédito	Ing. Sidney Carpio	Es el encargado de gestionar todo lo relacionados con los trámites de créditos.
Cajero	Ing. Yurika Calle	Es el encargado de realizar todos los procesos que interviene en caja.

Elaborado por: Los autores

4.2.5. Alcance

El alcance del sistema queda definido en el Product Backlog contenido en la Tabla IV.XXX la cual se encuentra compuesto por los siguientes ítems:

Tabla IV.XXX Product Backlog

CODIGO	REQUERIMIENTOS	PRIORIDAD
REQ1	Como: Gerente Quiero: conocer los lineamientos generales del proyecto. De tal forma que: pueda contar con documentación de respaldo	9
REQ2	Como: Asesor de crédito Quiero: Gestionar los Procesos de Crédito De tal forma que: se pueda imprimir una solicitud de crédito y agilizar los procesos	3
REQ3	Como: Asesor de crédito Quiero: Gestionar Reportes de Crédito De tal forma que: pueda contactar a los socios en caso de mora	4

REQ4	Como: Cajero Quiero: Gestionar los procesos de Caja De tal forma que: permita registrar socios, retiros, abrir libretas y manejar pólizas e inversiones	1
REQ5	Como: Cajero Quiero: Gestionar los reportes de caja De tal forma que: pueda realizar la impresión de libretas, verificar el estado de cuenta y realizar el cierre diario	2
REQ6	Como: Cajero Quiero: Gestionar los procesos de contabilidad De tal forma que: pueda registrar un asiento contable, manejar amortizaciones y depreciaciones	5
REQ7	Como: Cajero Quiero: Gestionar los reportes de contabilidad De tal forma que: pueda realizar el balance general, el estado de resultados, el diario por cuenta.	6
REQ8	Como: Cajero Quiero: Gestionar usuarios De tal forma que: pueda realizar el registro de nuevos usuarios, asignar roles y permisos, autenticar usuarios y realizar auditoria de usuarios.	7
REQ9	Como: Cajero Quiero: Generar reporte de usuarios De tal forma que: pueda realizar el reporte de usuarios registrados, accesos fallidos.	8

Elaborado por: Los autores

4.3. Descripción y características del producto

4.3.1. Descripción general del producto

Se construirá una aplicación de escritorio en Java que permitirá automatizar los procesos que se realizan en la Cooperativa de Ahorro y Crédito Chunchi Ltda.

La aplicación permitirá llevar un seguimiento de las mejoras alcanzadas, registrar resultados del cumplimiento de funcionalidades y a su vez la generación de reportes.

4.3.2. Características del producto

- Aplicación de escritorio de rápido y fácil acceso a través de un acceso directo.
- Diseño simple y efectivo que permita el rápido y eficaz desempeño de los usuarios.
- Sistema de reportes

- Soporte para usuarios concurrentes.
- Plataforma de ejecución : Windows o Linux
- Extensibilidad

4.4. Requerimientos de documentación

4.4.1. Manual de usuario

Se entregará junto con el producto el manual del usuario correspondiente, detallando cada una de las funciones del sistema. El manual de usuario detallado se encuentra en el ANEXO VIII.

4.5. Herramientas

Las herramientas que se utilizará para desarrollar el Sistema SISCOOP, se enumeran a continuación:

- Netbeans 7.4: framework para el desarrollo.
- iReport 5.5: diseñador de reportes
- Java SDK 7: Kit de desarrollo de Java
- PostGresSql 8.4: Motor de base de datos
- Windows 7: Sistema operativo para clientes

4.6. Requerimientos Funcionales

Los siguientes requisitos definen la funcionalidad del sistema, los mismos que tienen que ser implementados en su totalidad.

Req. 1 Como Gerente quiero conocer los lineamientos generales del proyecto de tal forma que pueda contar con documentación de respaldo.

Req. 2 Como asesor de crédito quiero gestionar los procesos de crédito de tal forma que se pueda imprimir una solicitud de crédito y agilizar los procesos.

Req. 3 Como Asesor de crédito quiero gestionar Reportes de Crédito de tal forma que pueda contactar a los socios en caso de mora.

Req. 4 Como cajero quiero gestionar los procesos de Caja de tal forma que permita registrar socios, retiros, abrir libretas y manejar pólizas e inversiones.

Req. 5 Como cajero quiero gestionar los reportes de caja de tal forma que pueda realizar la impresión de libretas, verificar el estado de cuenta y realizar el cierre diario.

Req. 6 Como cajero quiero gestionar los procesos de contabilidad de tal forma que pueda registrar un asiento contable, manejar amortizaciones y depreciaciones.

Req. 7 Como cajero quiero gestionar los reportes de contabilidad de tal forma que pueda realizar el balance general, el estado de resultados, el diario por cuenta.

Req. 8 Como cajero quiero gestionar usuarios de tal forma que pueda realizar el registro de nuevos usuarios, asignar roles y permisos, autenticar usuarios y realizar auditoria de usuarios.

Req. 9 Como: Cajero Quiero: Generar reporte de usuarios De tal forma que: pueda realizar el reporte de usuarios registrados, accesos fallidos.

4.7. Requerimientos No Funcionales

A continuación se muestran los requerimientos no funcionales del sistema.

➤ Rendimiento

- Seguridad
- Mantenibilidad
- Escalabilidad
- Interfaces
- Usabilidad

4.8. Riesgos

El riesgo implica cambios de opinión, funciones, no se puede evitar e implica incertidumbre y pérdida cuando el riesgo se transforma en un problema. Para gestionar el riesgo se debe analizar los riesgos que podrían hacer que nuestro proyecto fracasara, proporcionándole una importancia a cada riesgo tomando en cuenta los tipos de riesgos que se pueden hallar en el desarrollo del sistema.

4.8.1. Identificación del Riesgo

La identificación del riesgo es un proceso sistemático para describir las amenazas dentro del plan del proyecto. Identificado los riesgos conocidos y predecibles, el gestor del proyecto comenzar a trabajar para evitarlos y controlarlos cuando sea posible.

Para la realización de este Proyecto se tomarán en cuenta tres tipos de riesgos:

Riesgo del Proyecto (RP).

Riesgo Técnico (RT).

Riesgo del Negocio (RN).

En la tabla IV.XXXI se muestra la identificación de riesgos.

Tabla IV.XXXI Identificación del Riesgo

ID	DESCRIPCIÓN DEL RIESGO	CATEGORÍA	CONSECUENCIA
R1	Los usuarios no detallaron correctamente los requerimientos.	RP	Retraso en la ejecución del proyecto.
R2	Interfaz de usuario mal definida.	RT	Amenaza de la calidad del software.
R3	Falta de conocimientos por parte de los programadores del lenguaje de programación y proceso de desarrollo de software.	RT	Amenaza de la calidad del producto. Capacitación a los programadores. Retraso del proyecto.
R4	Cambios continuos de los requerimientos por parte del usuario.	RN	Pérdida de recursos económicos. Pérdida de tiempo.
R5	La información no se registra correctamente en la base de datos del sistema.	RN	Inconsistencia de datos.

Elaborado por: Los autores

4.8.2. Análisis de Riesgos

En esta sección se describen las tablas que servirán como referencia para determinar la valoración del riesgo, las probabilidades, y el impacto, de los riesgos que se detallaron en la tabla IV.XXXI, permitiendo establecer la prioridad de cada uno de ellos así establecer la valoración del riesgo mostrados en la tabla IV.XXXII.

Tabla IV.XXXII Valoración del Riesgo

RANGO DE PROBABILIDAD	DESCRIPCIÓN	VALOR
1% - 33%	Baja	1
33% - 66%	Media	2
67% - 99%	Alta	3

Elaborado por: Los autores

La Tabla IV.XXXIII indica la probabilidad de ocurrencia de aparición de cada riesgo en el desarrollo del proyecto.

Tabla IV.XXXIII Probabilidad

ID	PROBABILIDAD		
	%	VALOR	PROBABILIDAD
R1	20	1	Baja
R2	40	2	Media
R3	30	1	Baja
R4	40	2	Media
R5	20	1	Baja

Elaborado por: Los autores

La Tabla IV.XXXIV sirve como referencia para establecer el impacto del riesgo.

Tabla IV.XXXIV Impacto del Riesgo

IMPACTO	RETRASO	IMPACTO TÉCNICO	COSTO	VALOR
Bajo	1 semana	Leve efecto en el desarrollo del proyecto.	< 1%	1
Moderado	2 semanas	Moderado efecto en el desarrollo del proyecto.	< 5%	2
Alto	1 mes	Severo efecto en el desarrollo del proyecto.	< 10%	3
Crítico	> 1 meses	Proyecto no puede ser culminado.	> 20%	4

Elaborado por: Los autores

En la tabla IV.XXXV se describe el valor del impacto que se asigna a cada riesgo.

Tabla IV.XXXV Riesgo – Impacto

ID	IMPACTO	
	VALOR	IMPACTO
R1	2	Moderado
R2	3	Alto
R3	2	Moderado
R4	3	Alto
R5	1	Bajo

Elaborado por: Los autores

4.8.3. Criterios de valoración de la exposición al riesgo

La Tabla IV.XXXVI sirve como referencia para establecer la exposición al riesgo, mediante valores cuantitativos y gráficos.

Tabla IV.XXXVI Exposición al riesgo

EXPOSICIÓN	VALOR	COLOR
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	Mayor a 6	Rojo

Elaborado por: Los autores

La Tabla VI.XXXVII detalla la exposición al riesgo representado por un color dependiendo del valor, este valor se lo obtiene mediante multiplicando el valor del impacto por el valor de la probabilidad.

Tabla IV.XXXVII Impacto – Probabilidad

IMPACTO	BAJO=1	MODERADO=2	ALTO=3	CRÍTICO=4
PROBABILIDAD				
ALTA	3	6	9	12
MEDIA	2	4	6	8
ALTA	2	4	6	8

Elaborado por: Los autores

La tabla IV.XXXVIII fue construida de los valores obtenidos en las tablas anteriores.

Tabla IV.XXXVIII Resumen del Riesgo

ID	PROBABILIDAD			IMPACTO		EXPOSICIÓN AL RIESGO	
	%	Valor	Probabilidad	Valor	Impacto	Valor	Exposición
R1	20	1	Baja	2	Moderado	4	Media
R2	40	2	Media	3	Alto	6	Alta
R3	30	1	Baja	2	Moderado	4	Media
R4	40	2	Media	3	Alto	6	Alta
R5	20	1	Baja	1	Bajo	2	Baja

Elaborado por: Los autores

A continuación se ordenará la tabla IV.XXXIX en forma ascendente de acuerdo a su prioridad e incluiremos el código de colores y la línea de corte para los riesgos de prioridad 1, ya que estos son los que puedes convertirse en problemas.

Tabla IV.XXXIX Prioridades del Riesgo

IDENTIFICACIÓN	PRIORIDAD	EXPOSICIÓN
R2	1	6
R4	1	6
R1	2	4
R3	2	4
R5	3	2

Elaborado por: Los autores

4.8.4. Planeación y Programación del Riesgo

En las siguientes tablas se encuentra la planeación y programación de cada uno de los riesgos identificados.

En la tabla III.XL se encuentra la gestión del riesgo 1.

En la tabla III.XLI se encuentra la gestión del riesgo 2.

En la tabla III.XLII se encuentra la gestión del riesgo 3.

En la tabla III.XLIII se encuentra la gestión del riesgo 4.

En la tabla III.XLIV se encuentra la gestión del riesgo 5.

Tabla IV.XL Gestión del riesgo 1

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R1		FECHA:	
Probabilidad: Baja Valor: 2	Impacto: Moderado Valor: 2	Exposición: Media Valor: 4	Prioridad: 2
DESCRIPCIÓN: Los usuarios no detallaron correctamente los requerimientos.			
REFINAMIENTO: Causas: No existió una buena comunicación entre el usuario y los responsables del proyecto. Consecuencias: Retraso de la ejecución del proyecto.			
REDUCCIÓN: Analizar las necesidades del usuario para establecer correctamente los requerimientos.			
SUPERVISIÓN: Que el ambiente de comunicación sea el más propicio entre el cliente y el responsable del proyecto.			
GESTIÓN: Reuniones para establecer acuerdos que agilicen el éxito del proyecto.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: José Manzano Ángel Cobo			

Elaborado por: Los autores

Tabla IV.XLI Gestión del riesgo 2

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R2		FECHA:	
Probabilidad: Medio Valor: 2	Impacto: Alto Valor: 3	Exposición: Alto Valor: 3	Prioridad: 1
DESCRIPCIÓN: Interfaz de usuario mal definida.			
REFINAMIENTO: Causas: El diseño de las interfaces no se validó entre el diseñador y el cliente. Consecuencias: Amenazan la calidad del software.			
REDUCCIÓN: Diseñar correctamente las interfaces, tomar en cuenta la usabilidad.			
SUPERVISIÓN: Revisar que el diseño de interfaces sea el más óptimo para la gestión.			
GESTIÓN: Exigir al equipo de trabajo que las interfaces sean aprobadas unánimes para garantizar la validación.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: José Manzano Ángel Cobo			

Elaborado por: Los autores

Tabla IV.XLII Gestión del riesgo 3

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R3		FECHA:	
Probabilidad: Baja Valor: 1	Impacto: Moderado Valor: 2	Exposición: Medio Valor: 4	Prioridad: 2
DESCRIPCIÓN: Falta de conocimiento por parte de los programadores del lenguaje de programación y proceso de desarrollo de software.			
REFINAMIENTO: Causas: Especialización en diferentes lenguajes de programación y proceso de desarrollo del software que se utilizara para la elaboración del sistema. Poca investigación del lenguaje de programación a usar. Consecuencias: Amenaza la calidad del software. Capacitación a los programadores. Retraso en desarrollo del proyecto.			
REDUCCIÓN: Capacitar al programador en el lenguaje de programación y desarrollo de software que se usará. Promover la investigación de nuevos lenguajes de programación.			
SUPERVISIÓN: Controlar que la capacitación a los programadores sea la adecuada.			
GESTIÓN: Fomentar la especialización en el lenguaje de programación.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: José Manzano Ángel Cobo			

Elaborado por: Los autores

Tabla IV.XLIII Gestión del riesgo 4

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R4		FECHA:	
Probabilidad: Medio Valor: 2	Impacto: Alto Valor: 3	Exposición: Alto Valor: 6	Prioridad: 1
DESCRIPCIÓN: Cambios continuos de los requerimientos por parte del usuario.			
REFINAMIENTO: Causas: Los procesos del negocio pueden cambiar. Consecuencias: Pérdida de tiempo.			
REDUCCIÓN: Predecir posibles cambios a suscitarse en el negocio.			
SUPERVISIÓN: Estar en constante comunicación con el usuario.			
GESTIÓN: Procurar que el trabajo sea entregado con tiempo, anticipándonos a posibles cambios.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: José Manzano Ángel Cobo			

Elaborado por: Los autores

Tabla IV.XLIV Gestión del riesgo 5

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R5		FECHA:	
Probabilidad: Bajo Valor: 1	Impacto: Bajo Valor: 1	Exposición: Bajo Valor: 2	Prioridad: 3
DESCRIPCIÓN: La información no se registre correctamente en la base de datos del sistema.			
REFINAMIENTO: Causas: No se cuenta con un método apropiado para la actualización automática. Consecuencias: Provoca inconsistencia en los datos.			
REDUCCIÓN: Implementar un método para la actualización de los datos cuando se haga una inserción en la base de datos del sistema.			
SUPERVISIÓN: Revisar que el método implementado cumpla totalmente la tarea para el cual fue creado.			
GESTIÓN: Solicitar al equipo de trabajo que el método implementado sea el adecuado.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: José Manzano Ángel Cobo			

Elaborado por: Los autores

4.9. Recursos Físicos

4.9.1. Hardware

En la tabla IV.XLIII se visualiza el hardware utilizado para desarrollar el sistema.

Tabla IV.XLV Hardware utilizado

Cantidad	Descripción
1 Laptop	S.O Windows 7 Procesador Intel Core i3 Disco 350 Gb Memoria 2GB
1 Laptop	S.O Windows 7 Procesador Intel Core 2 Duo Disco 350 Gb Memoria 2GB

Fuente: Los autores

4.9.2. Software

En la tabla IV.XLVI se observa el software necesario para desarrollar el sistema.

Tabla IV.XLVI Software utilizado

SOFTWARE	DESCRIPCION
NetBeans 7.4	Framework para el desarrollo del sistema
IReport 5.5	Herramienta para el diseño de los reportes
BoUml	Herramienta para diseñar los diagramas
PostgreSql	Motor de Base de Datos

Fuente: Los autores

Con respecto al hardware necesario para la implementación de la aplicación contamos con una portátil para cada desarrollador y una para la base de datos.

4.10. Diagrama Gantt y Presupuesto

El Diagrama Gantt, presupuestos y duración del proyecto se encuentra en el ANEXO IX.

4.11. Planificación

Esta fase se realiza la preparación de la especificación funcional, diseño conceptual. El proceso de planificación juega un papel muy importante al momento de elaborar un proyecto, por lo cual la construcción de un plan ayuda a refinar dichas actividades hasta llegar a la eficiencia requerida.

4.11.1. Diseño Conceptual

El diseño conceptual comprende la abstracción de los requerimientos funcionales, los actores, diagramas de casos de uso, glosario de términos mismo que se detallan a continuación.

4.11.2. Especificación Funcional

En esta sección se define la especificación funcional de los requerimientos definidos. La descripción de la especificación funcional se encuentra en ANEXO III del presente documento.

4.11.3. Casos de Uso

A través de los diagramas de casos de uso se presenta el comportamiento de los requisitos funcionales mediante unos diagramas. Estos diagramas se elaborarán basándose en la definición de los requerimientos previamente establecidos.

Diagrama de Casos de uso - Transacciones Caja

En la figura IV.31 se representa el diagrama de caso de uso de transacciones de caja.

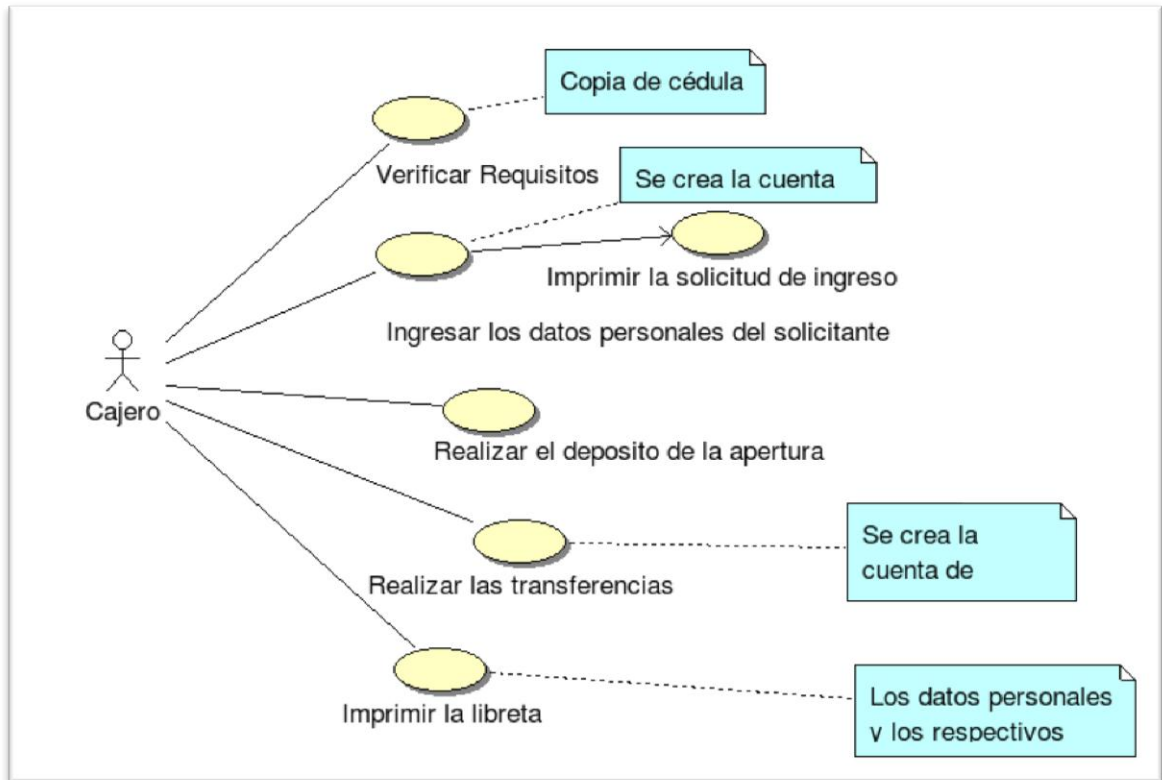


Figura IV.31: Diagrama de caso de uso de transacciones de caja

Caso de uso Esenciales - Transacciones de caja

En la tabla IV.XLVII se observa el caso de uso esencial de las transacciones de caja.

Tabla IV.XLVII Caso de uso esencial de transacciones de caja

Identificador	CU_Transacciones_Caja
Nombre	Transacciones de caja
Actores	Socio, Cajero

Propósito	Realizar el proceso de apertura de libreta
Visión general	El Cajero es la persona responsable de ingresar los datos del socio al sistema cumpliendo con todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio desea abrir una libreta	2. Presenta solicitud de ingreso
3. Cajero solicita datos del socio	4. Valida datos del socio, inserta datos del socio e imprime solicitud de ingreso
5. Socio firma solicitud de ingreso	6. Crea cuenta de ahorros
7. Socio solicita deposito	8. Insertar transacción
9. Cajero inserta asiento y cobra comisión	10. Imprime nota de debito e imprime libreta

Fuente: Los autores

4.11.4. Diseño Lógico

El diseño lógico es el proceso de describir la solución en términos de su organización, su estructura y la interacción de sus partes.

4.11.5. Diagramas de Secuencia

Los diagramas de secuencia permitirán descubrir las clases que se necesitan para el desarrollo del sistema, mostrando una visión dinámica de los procesos de la aplicación ayudando a identificar los métodos que se debe implementar en cada clase.

Diagrama de secuencia – Transacciones Caja

En la figura IV.32 se observa el diagrama de secuencia de transacciones de caja.

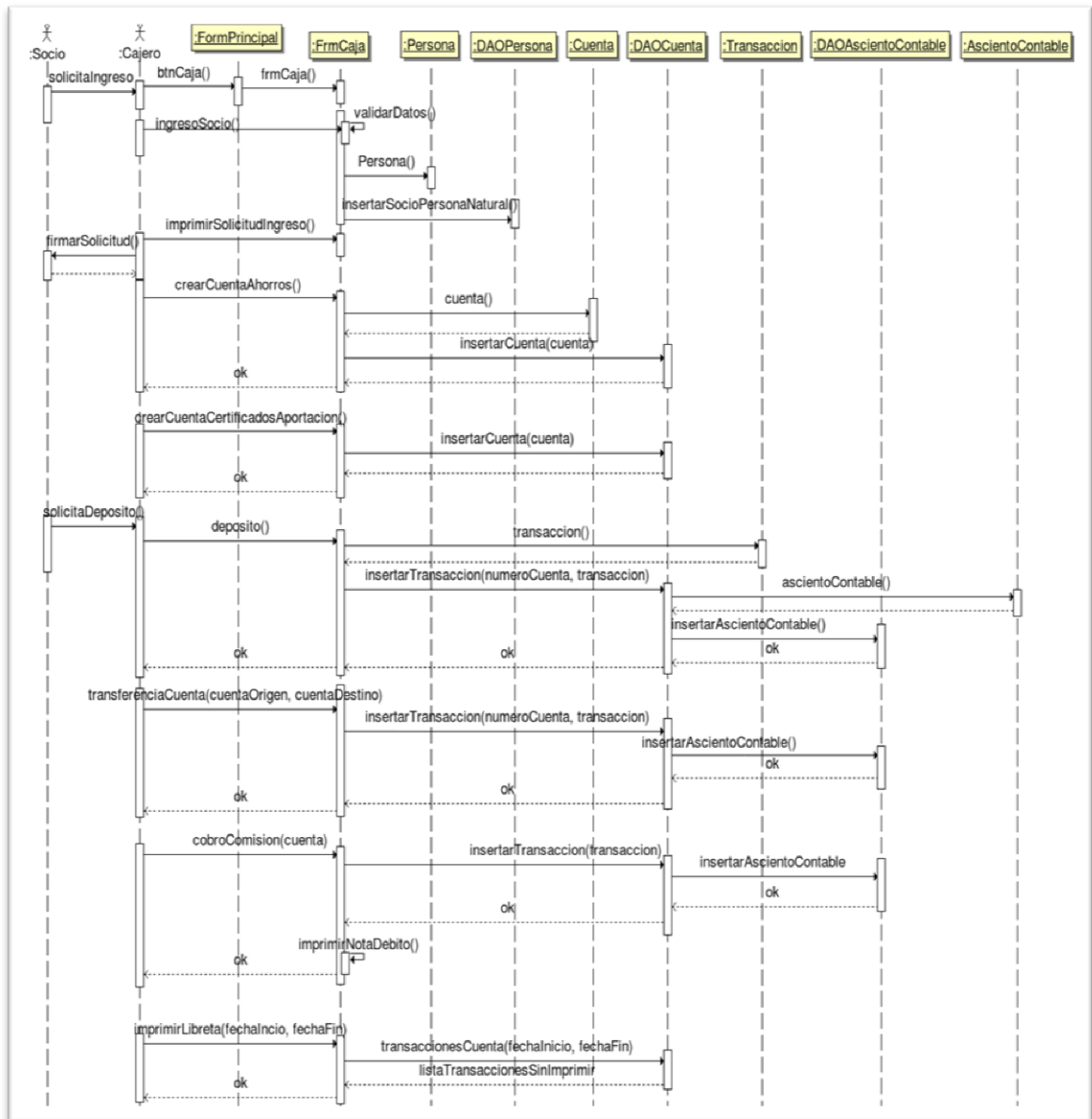


Figura IV.32: Diagrama de secuencia de Transacciones de caja

4.11.6. Diagramas de Clases

A partir de los diagramas de secuencia se especifica el diagrama de clases de diseño.

Paquete en PostgreSQL o también llamado paquetes; son agrupaciones de tablas relacionadas, funciones y otros objetos de una base de datos.

Diagrama de clases – Apertura Libreta

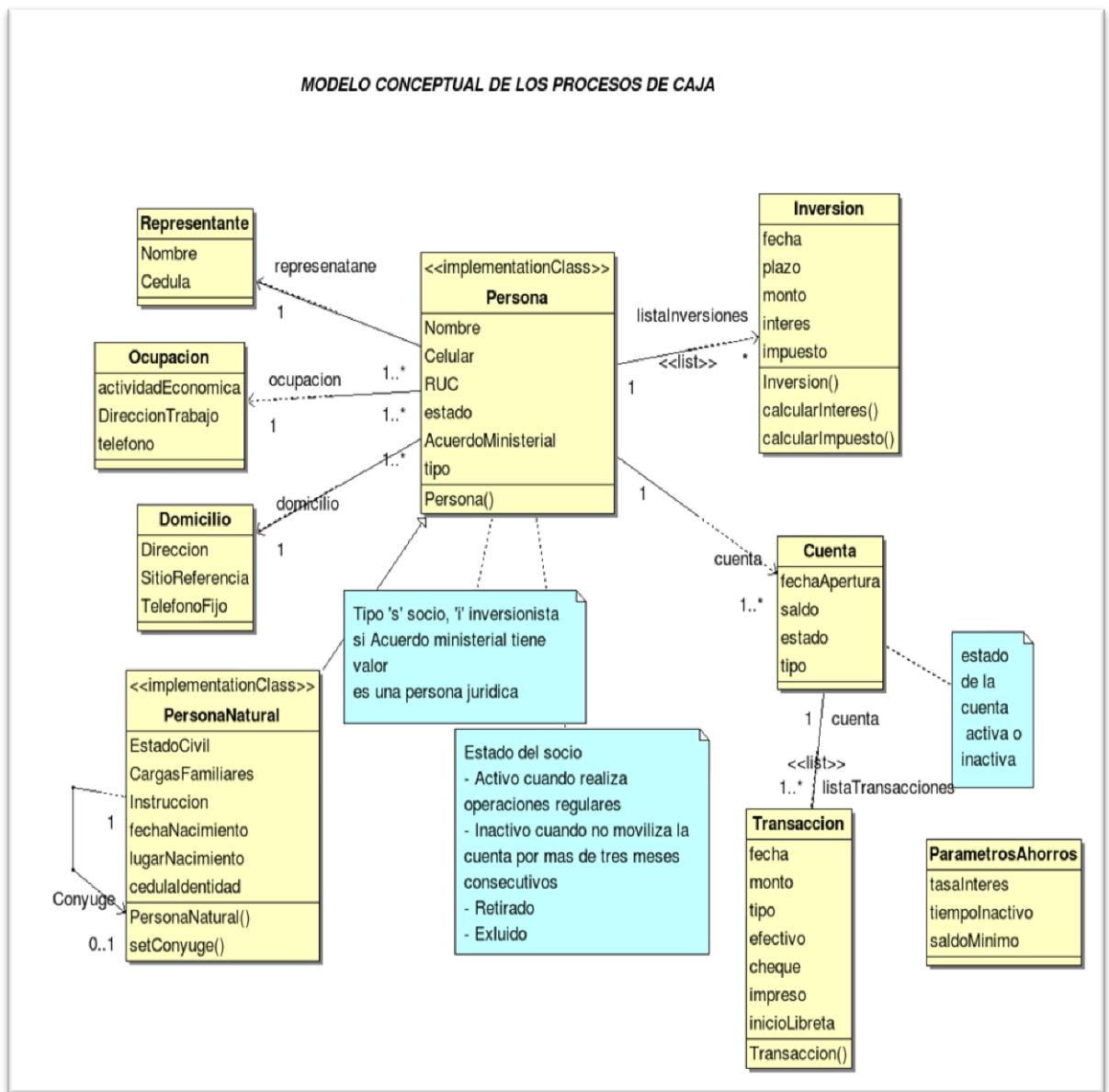


Figura IV.33: Diagrama de clases de transacciones caja

Todos los diagramas de clases, de actividades, de iteración, de despliegue del sistema se encuentran detallados en el ANEXO IV.

4.11.7. Diseño de Interfaces de Usuario

En la figura IV.34 se visualiza la interfaz de login de usuarios, en la cual se debe ingresar el usuario y la contraseña.

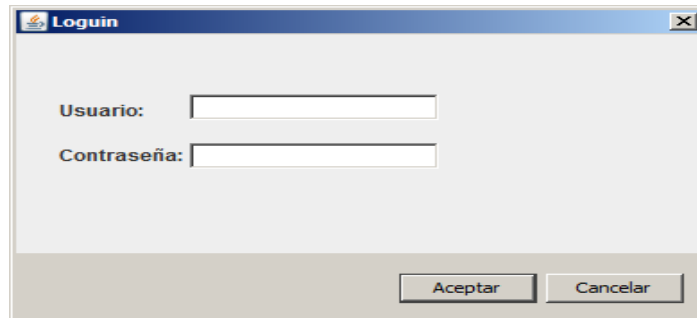


Figura IV.34: Interfaz de login de usuarios

En la figura IV.35 se observa la interfaz inicial del sistema la misma que visualiza todos los socios que se encuentran registrados en el sistema.

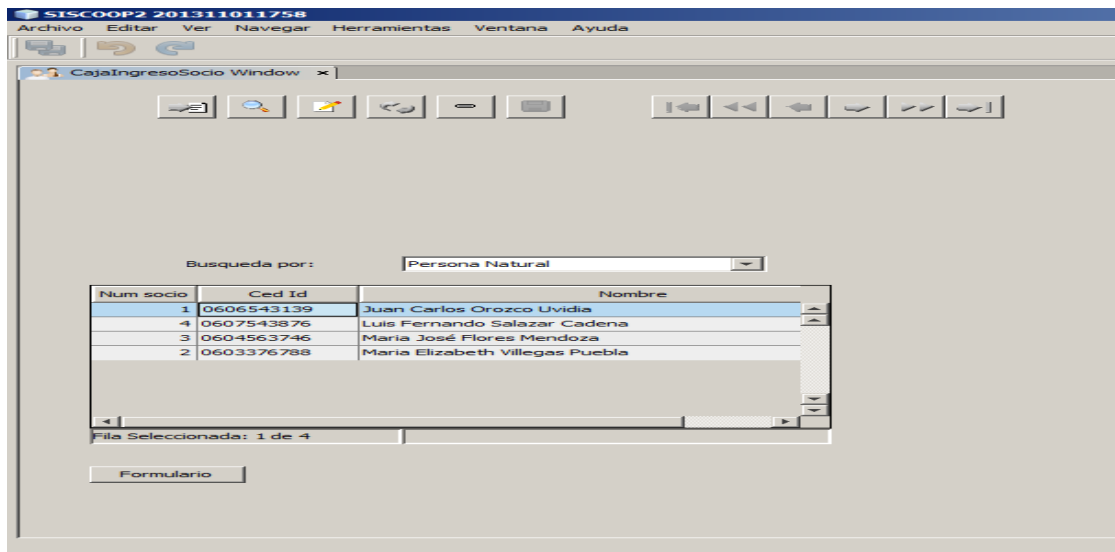


Figura IV.35: Interfaz de inicio con listado de socios

En la figura IV.36 se observa el formulario de ingreso de un nuevo socio al sistema.

Datos Generales | Inf. Representante | Datos Conyuge

Datos Generales

Nombres: Juan Carlos Orozco Uvidia *
Estado Civil: Soltero
Cedula: 0606543139 *
Cargas Famil.: 3
Celular: 0976547865 *
Nivel Instruc.: Basica *
RUC: 0004002 *
Fecha Nac.: 1988/12/13 *
Estado: Activo *
Lugar Nac.: riobamba *

Domicilio

Direccion: espejo y 5 de junio
Sitio referencia: Frente Farmacia
Telefono fijo: 032987665

Trabajo

Actividad: docente
Direccion: Guayaquil y Colon
Telefono: 0987098766

Figura IV.36: Interfaz de ingreso de nuevo socio

En la figura IV.37 se observa el formulario de ingreso de datos del representante del nuevo socio.

Datos Generales | Inf. Representante | Datos Conyuge

Datos generales

Nombres: *
Cedula: *
Celular: *
Estado: *

Domicilio

Direccion: *
Sitio referencia: *
Telefono fijo: *

Figura IV.37: Formulario de ingreso de datos del representante.

En la figura IV.38 se observa el formulario de ingreso de datos del conyugue del nuevo socio.

Datos Generales | Inf. Representante | Datos Conyuge

Datos Generales

Nombres: * Estado Civil:

Cedula: * Cargas Famil.: 0

Celular: * Nivel Instruc.: *

RUC: * Fecha Nac.: / / *

Estado: * Lugar Nac.: *

Domicilio

Direccion:

Sitio referencia:

Telefono fijo:

Trabajo

Actividad:

Direccion:

Telefono:

Figura IV.38: Formulario de ingreso de datos del conyuge.

En la figura IV.39 se observa el formulario de búsqueda de socios en el sistema.

SISCOOP2 201311011758

Archivo Editar Ver Navegar Herramientas Ventana Ayuda

CajaIngresoSocio Window

Busqueda por: Persona Natural

Num socio	Ced Id	Nombre
4	0607543876	Luis Fernando Salazar Cadena

Fila Seleccionada: 1 de 1

Filtering and sorting settings

Filtering conditions | Sorting conditions

Num socio Equals to 4

Figura IV.39: Formulario de búsqueda de socios.

En la figura IV.40 se observa la interfaz de gestión de cuentas.

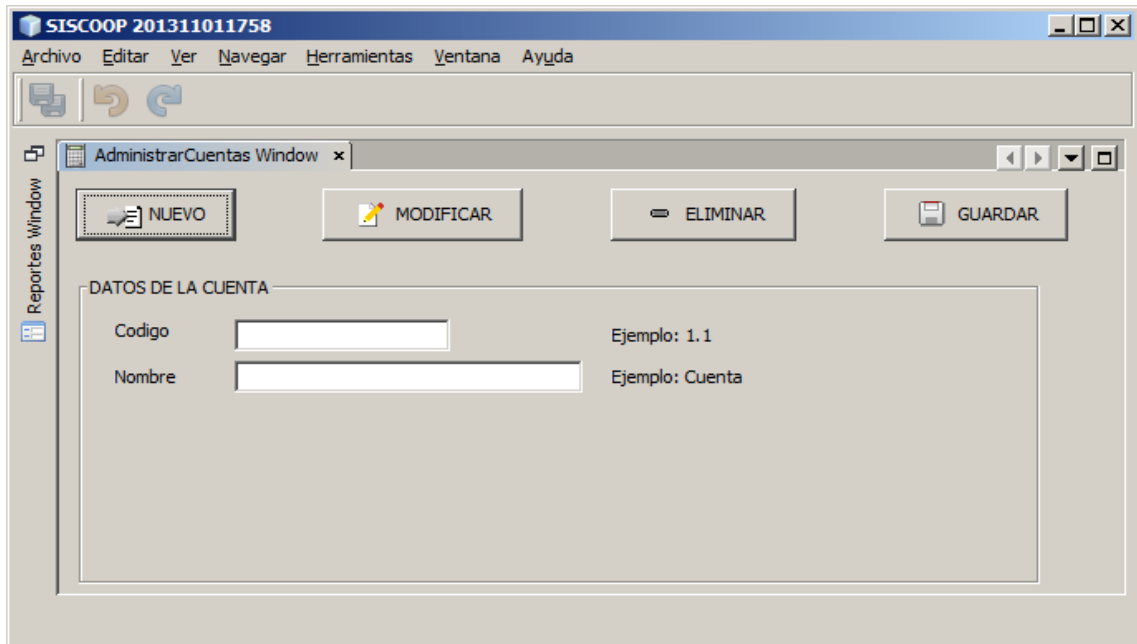


Figura IV.40: Interfaz de gestión de cuentas

En la figura IV.41 se observa la interfaz de búsqueda de cuentas filtradas por número de cuenta, nombre de la cuenta y todos.

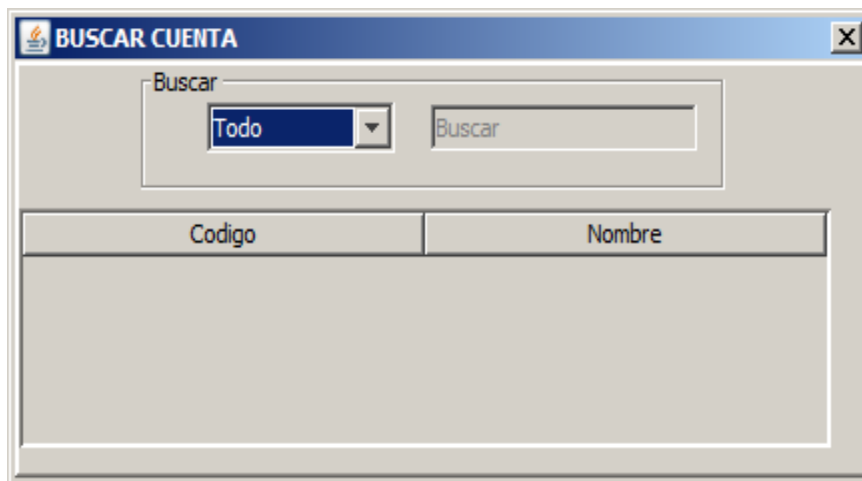


Figura IV.41: Interfaz de búsqueda de cuentas

En la figura IV.42 se observa la interfaz de gestión de registros.

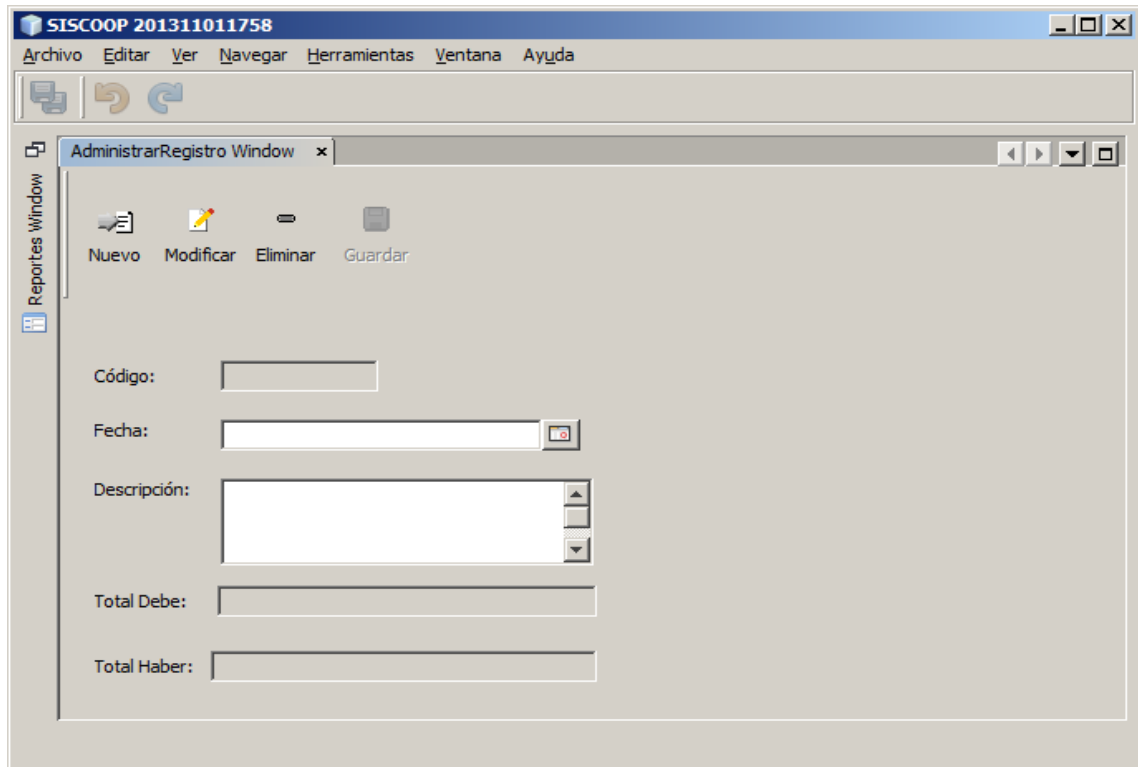


Figura IV.42: Interfaz de gestión de registros

En la figura IV.43 se observa la interfaz de búsqueda de registros, filtrada por número de registro, nombre del registro o todos.

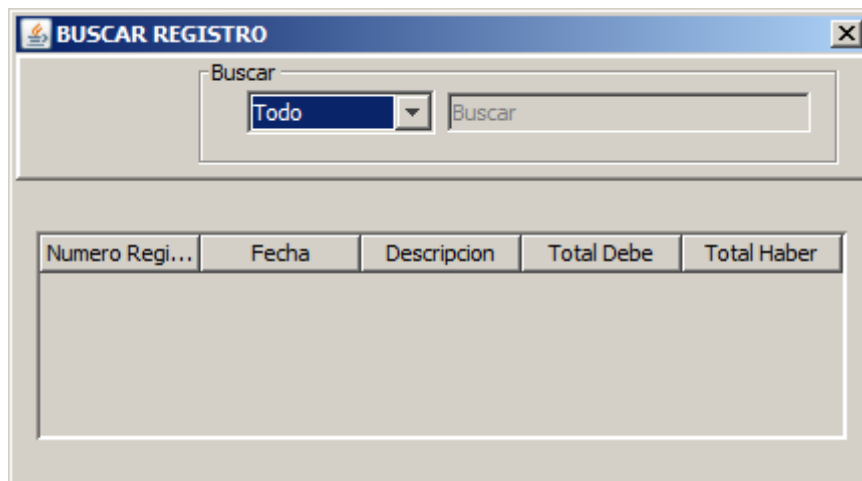


Figura IV.43: Interfaz de búsqueda de registros

En la figura IV.44 se observa la interfaz de gestión de Registros_Cuentas.

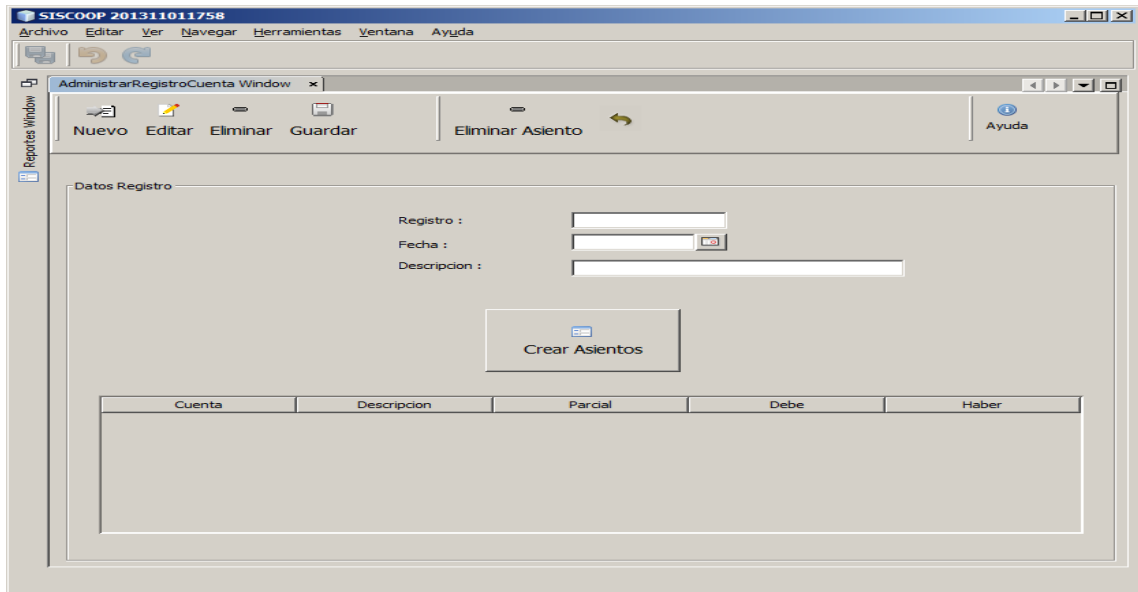


Figura IV.44: Interfaz de gestión de Registros_Cuentas

En la figura IV.45 se observa la interfaz para la creación de asientos contables.

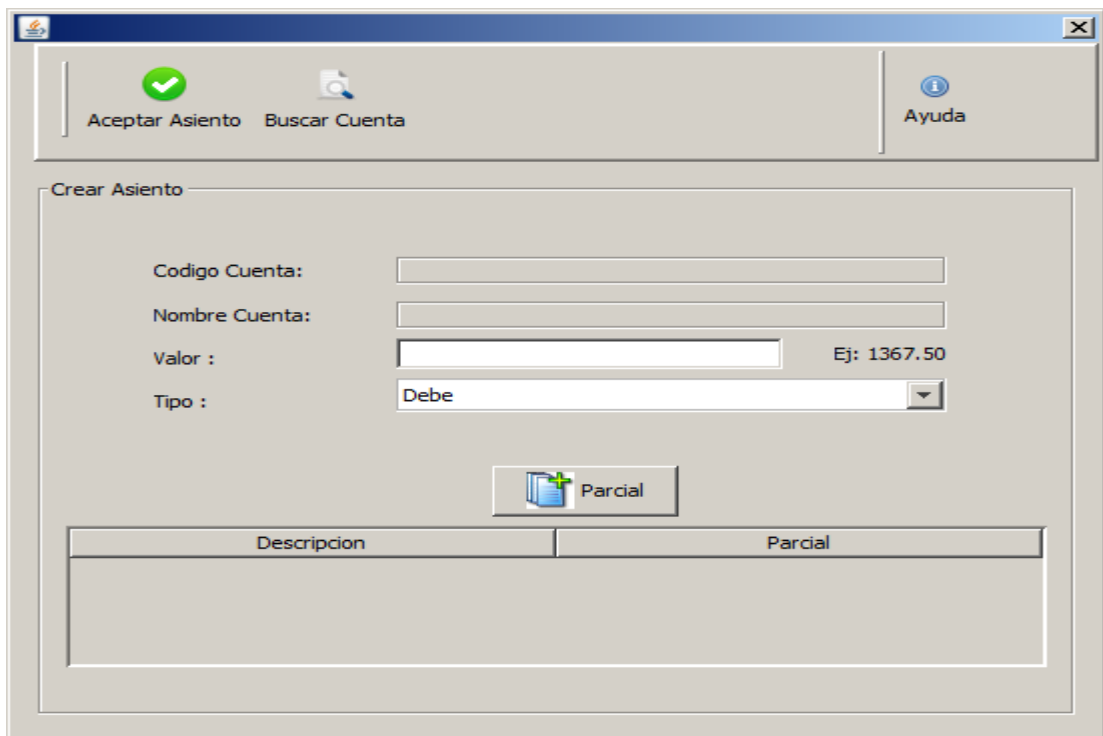


Figura IV.45: interfaz para la creación de asientos contables

En la figura IV.46 se observa la interfaz para el ingreso, modificación y eliminaciones de roles de usuarios del sistema.

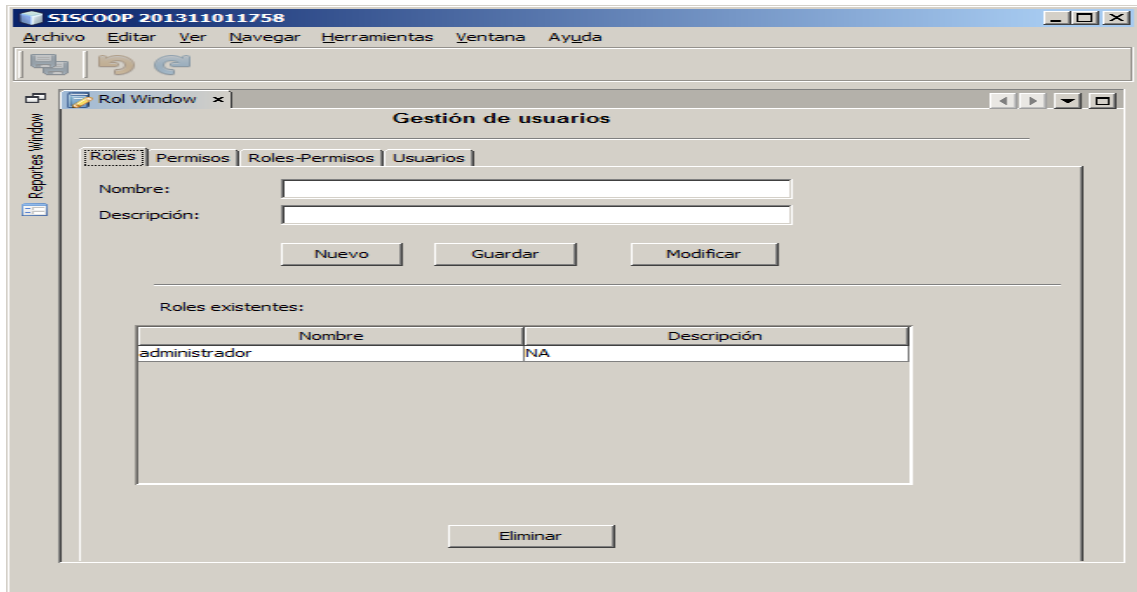


Figura IV.46: Interfaz de gestión de roles de usuario

En la figura IV.47 se observa la interfaz de ingreso, modificación y eliminación de permisos para los usuarios del sistema.

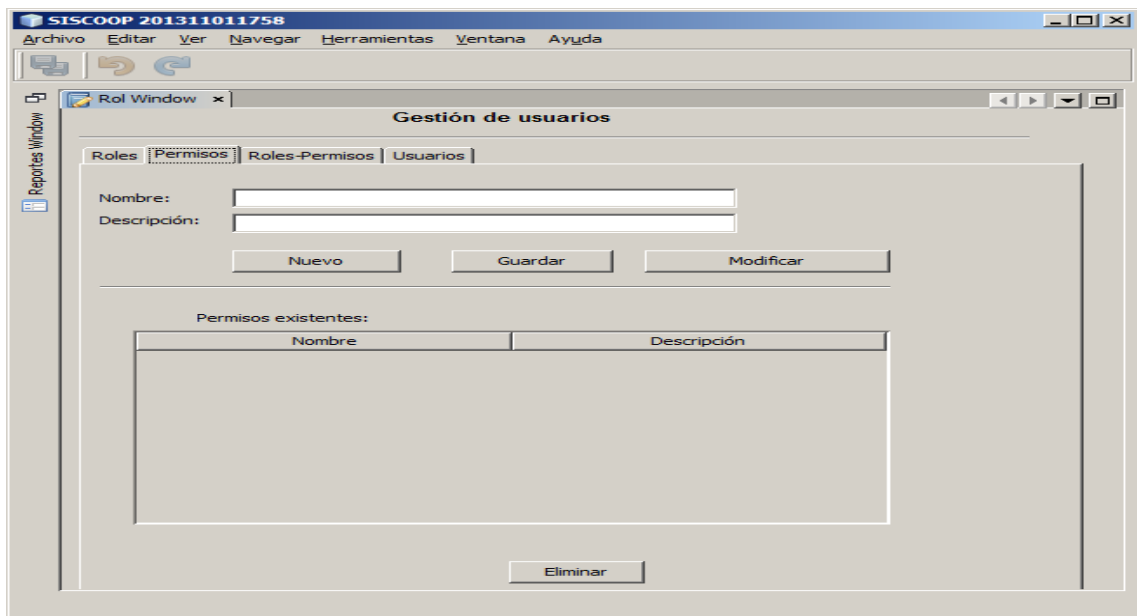


Figura IV.47: Interfaz de gestión de permisos de usuarios

En la figura IV.48 se observa la interfaz donde se asignan los permisos a los roles existentes.

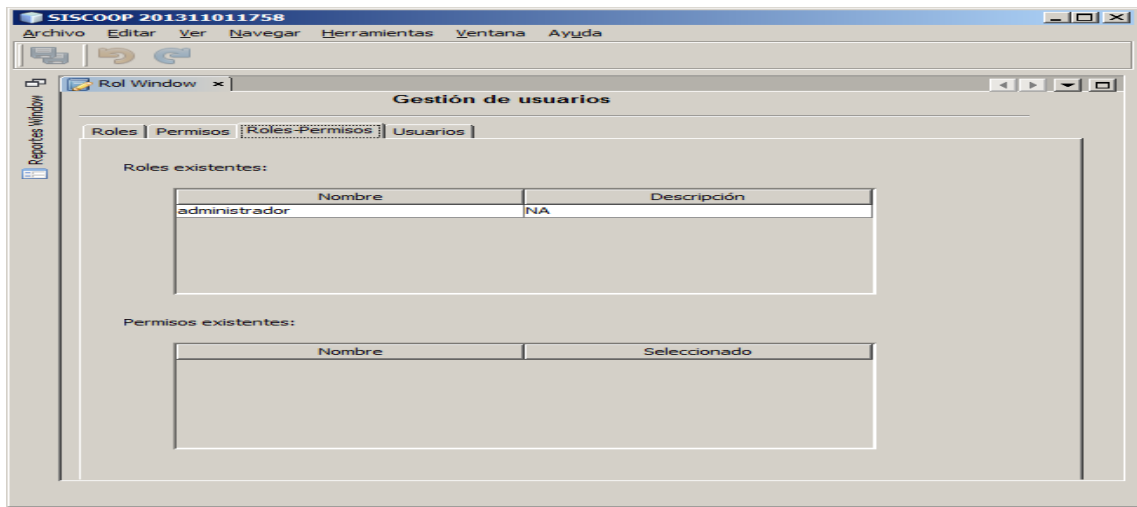


Figura IV.48: Interfaz de asignación de permisos a los roles

En la figura IV.49 se observa la interfaz para el ingreso, modificación y eliminación de usuarios del sistema.

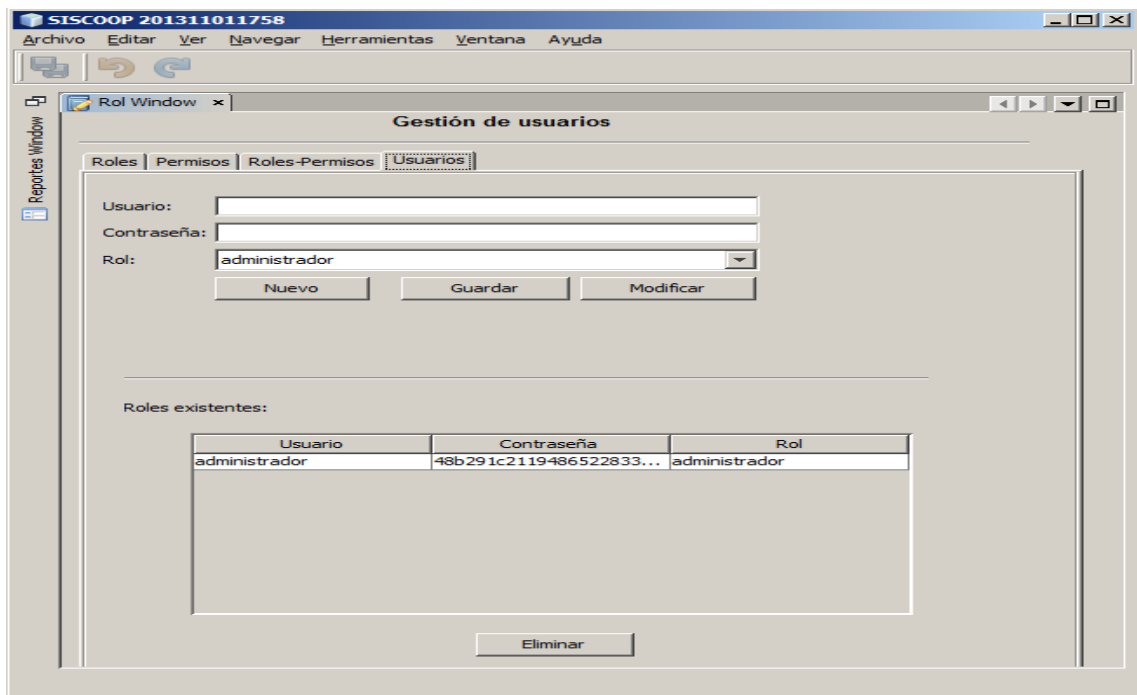


Figura IV.49: Interfaz de gestión de usuarios

En la figura IV.50 se observa la interfaz para generar los reportes de cada módulo.

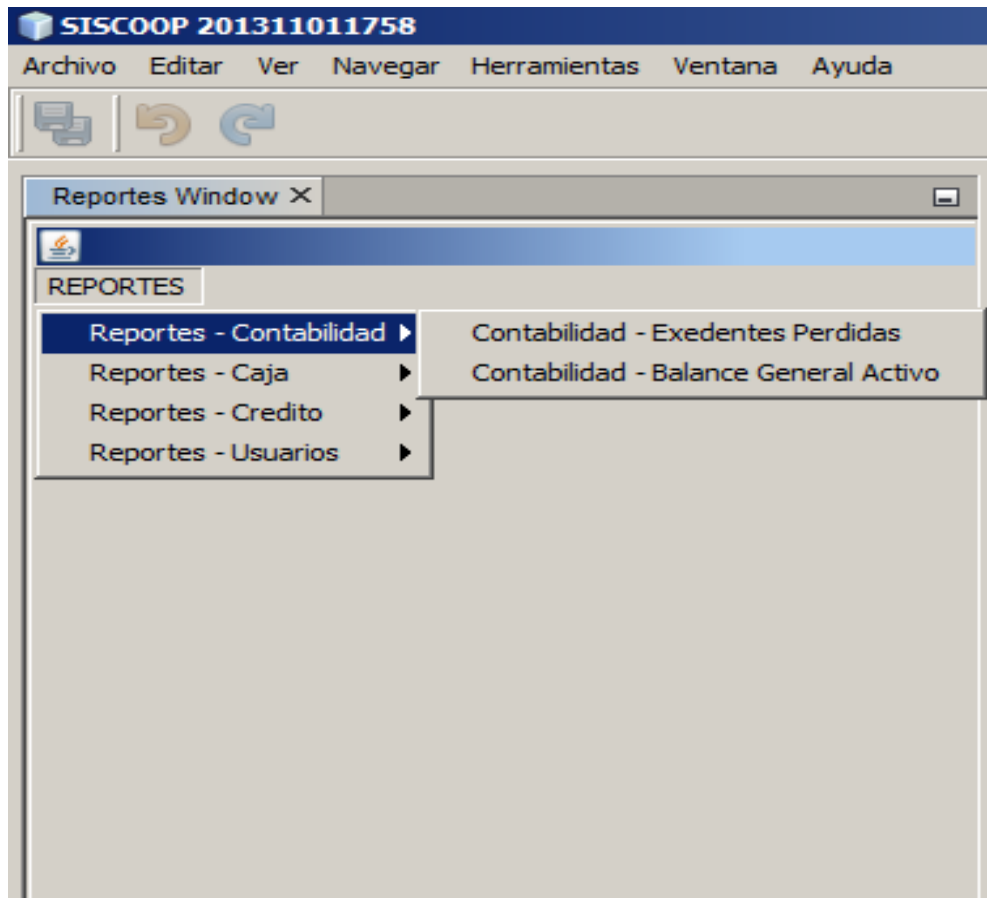


Figura IV.50: Interfaz para generar los reportes de cada módulo

4.11.8. Diseño Físico

Esta es la última etapa de la fase de planeación, en donde se describirá las actividades, componentes, servicios y tecnologías de la solución.

En la figura IV.34 se presenta el diagrama de despliegue que detalla la estructura del sistema.

Diagrama de Despliegue

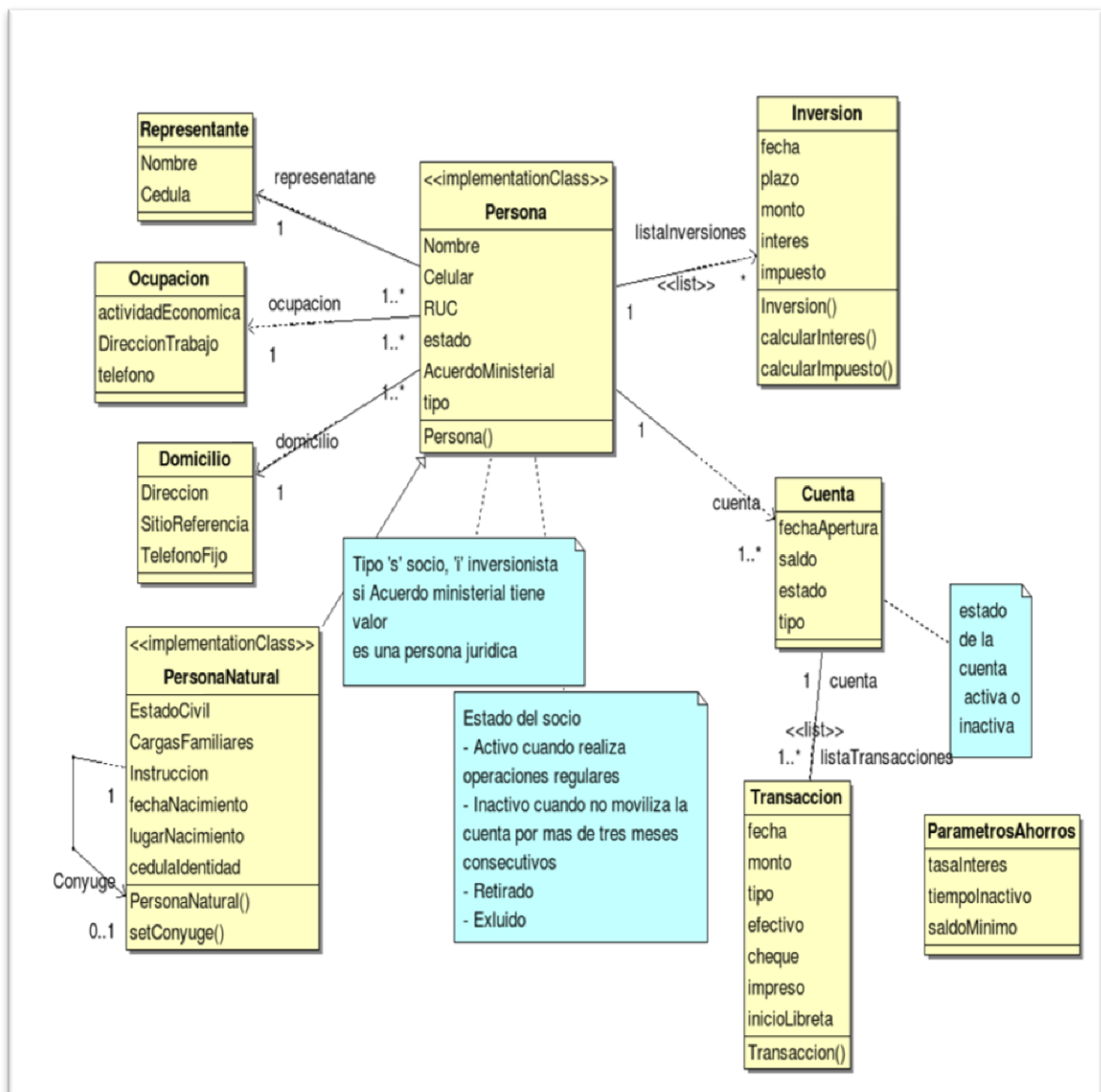


Figura IV.51: Diagrama de despliegue transacciones caja

4.11.10. Diccionario de datos

El diccionario de datos es una descripción lógica y puntual de los datos que se van a utilizar en el sistema. El diccionario de datos de los esquemas utilizados se encuentra, en el ANEXO V.

En el ANEXO VI se encuentra el script de la base de datos de PostgreSQL, motor de base de datos utilizado para el desarrollo.

4.11.11. Arquitectura del sistema

En la figura IV.63 se observa la arquitectura del sistema.

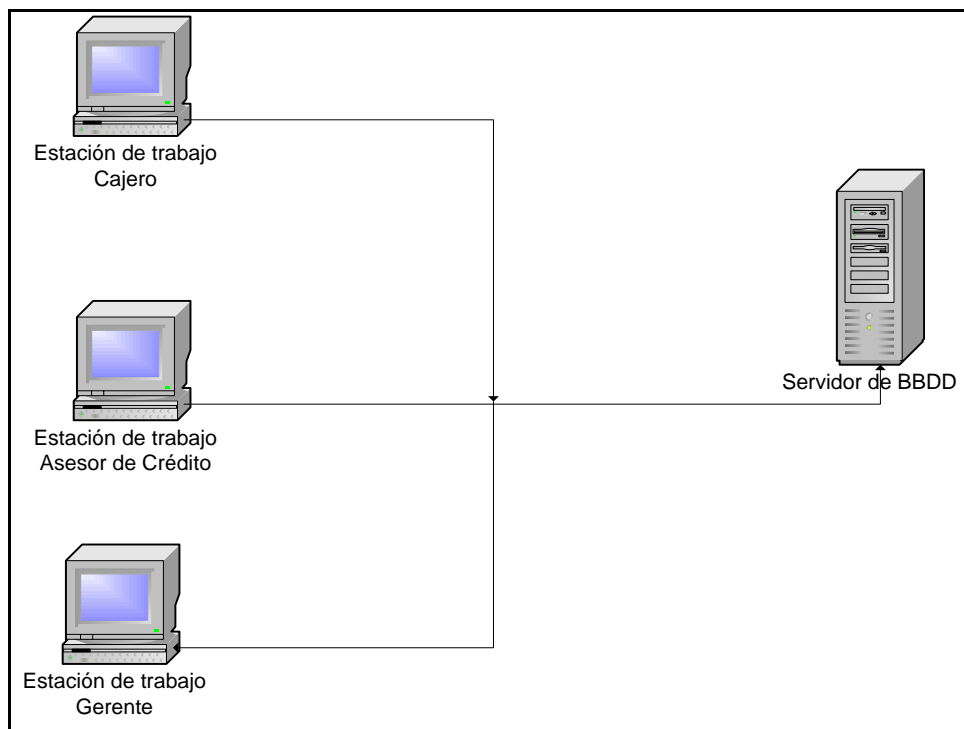


Figura IV.53: Arquitectura del sistema

Fuente: Los autores

4.12. Entregables

En esta sección se detalla las tareas realizadas en cada iteración llamadas por Scrum Sprint, estas tareas son tomadas de la pila Sprint_Backlog que es una lista con requerimientos detallados y priorizados según la importancia del usuario.

En la tabla IV.XLVIII se visualiza los requerimientos de la pila del Spring_Backlog.

Tabla IV.XLVIII Sprint_Backlog

Ítem / Quién	Descripción
1	Como: Gerente Quiero: conocer los lineamientos generales del proyecto. De tal forma que: pueda contar con documentación de respaldo
Equipo	Reunión para definir el alcance del proyecto
Equipo	Requerimientos
Equipo	Cronograma de actividades
Equipo/Product Owner	Armado del product backlog
Equipo	Armado de sprint backlog
Equipo	Armado de burndown charts.
Equipo	Arquitectura del proyecto
2	Como: Asesor de crédito Quiero: Gestionar los Procesos de Crédito De tal forma que: se pueda imprimir una solicitud de crédito y agilizar los procesos
Ángel	Diseño de base de datos
Ángel	Casos de uso
José	Análisis de la capacidad de pago de los clientes
José	Registro de solicitudes de crédito
José	Generación de tablas presuntivas
José	Registro de garantes
Ángel	Liquidación de créditos
Ángel	Cobros de cuotas
Ángel	Cobros de intereses
José	Cobros de multas
José	Cobros de comisiones
José	Generación de notificaciones por atraso
José	Renovación de créditos
Ángel	Seguimiento y cobranza de la cartera de crédito
Ángel	Reclasificación diaria de la cartera
Ángel	Pruebas
Ángel	Manual de usuario, técnico

3	Como: Asesor de crédito Quiero: Gestionar Reportes de Crédito De tal forma que: pueda contactar a los socios en caso de mora
José	Clientes en mora
José	Cartera vencida
José	Cartera que no devenga interés
Ángel	Cartera que devenga interés
Ángel	Cartera contaminada
Ángel	Clasificación del crédito de acuerdo al monto
Ángel	impresión de pagares
José	Generación de tablas de pagos
José	Pruebas
José	Manual de usuario, técnico
4	Como: Cajero Quiero: Gestionar los procesos de Caja De tal forma que: permita registrar socios, retiros, abrir libretas y manejar pólizas e inversiones
José	Registro de nuevos socios
José	Recepción de depósitos
José	Registro de retiros
José	Aperturas de libretas
Ángel	Manejo de inversiones y pólizas
Ángel	Pago de intereses a los ahorros
Ángel	Pago de intereses a las pólizas
Ángel	Pruebas
Ángel	Manual de usuario, técnico
5	Como: Cajero Quiero: Gestionar los reportes de caja De tal forma que: pueda realizar la impresión de libretas, verificar el estado de cuenta y realizar el cierre diario
Ángel	Cierre diario
Ángel	Estado de cuenta
Ángel	Impresión de Libretas
José	Pruebas
José	Manual de usuario, técnico
6	Como: Cajero Quiero: Gestionar los procesos de contabilidad De tal forma que: pueda registrar un asiento contable, manejar amortizaciones y depreciaciones
Ángel	Registro de un asiento contable
Ángel	Manejo de Depreciaciones y Amortizaciones
José	Cierre del año
José	Pruebas
José	Manual de usuario, técnico
7	Como: Cajero Quiero: Gestionar los reportes de contabilidad De tal forma que: pueda realizar el balance general, el estado de resultados, el diario por cuenta.
Ángel	Balance General
Ángel	Estado de resultados
José	Estado de cambio de patrimonio
José	Diario por cuenta
José	Pruebas
José	Manual de usuario, técnico

8	Como: Cajero Quiero: Gestionar usuarios De tal forma que: pueda realizar el registro de nuevos usuarios, asignar roles y permisos, autenticar usuarios y realizar auditoria de usuarios.
José	Registro de nuevos usuarios
José	Asignación de roles y permisos a los usuarios
José	Autenticación de usuarios
Ángel	Auditoria de usuarios
Ángel	Pruebas
Ángel	Manual de usuario, técnico
9	Como: Cajero Quiero: Generar reporte de usuarios De tal forma que: pueda realizar el reporte de usuarios registrados, accesos fallidos.
José	Usuarios registrados
José	Accesos de usuarios en determinada fecha
Ángel	Accesos fallidos
Ángel	Pruebas
Ángel	Manual de usuario, técnico

Fuente: Los autores

2	Como: Cajero Quiero: Gestionar los reportes de caja De tal forma que: pueda realizar la impresión de libretas, verificar el estado de cuenta y realizar el cierre diario																
	Ángel	Diseño de base de datos	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0
	Ángel	Casos de uso	0	0	0	0	0	0	8	8	0	0	0	0	0	0	0
	José	Registro de nuevos socios	0	0	0	0	0	4	4	0	8	4	0	0	0	0	0
	José	Recepción de depósitos	0	0	0	0	0	0	0	4	0	4	8	0	0	0	0
	José	Registro de retiros	0	0	0	0	0	0	0	0	0	0	0	8	8	0	0
	José	Aperturas de libretas	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8
	Ángel	Manejo de inversiones y pólizas	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0
	Ángel	Pago de intereses a los ahorros	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
	Ángel	Pago de intereses a las pólizas	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0
	José	Pruebas	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4
	José	Manual de usuario, técnico	0	0	0	0	8	4	4	4	0	0	0	4	4	0	0
			16	32	16	32	16	16	16	16	16	16	16	16	16	12	12
			264	12	100	200	100	100	100	100	100	100	100	100	100	75	75
		HORAS ESTIMADAS DEL SPRINT	240	224	192	176	144	128	112	96	80	64	48	32	16	0	-12

Fuente: Los autores

4.12.2. BurnDown Chart

Facilita el conocimiento del estado actual del proyecto, basado en la Pila de Producto, es el BurnDown Chart. En la figura IV.64 se visualiza el BurnDown Chart del Sprint 1.

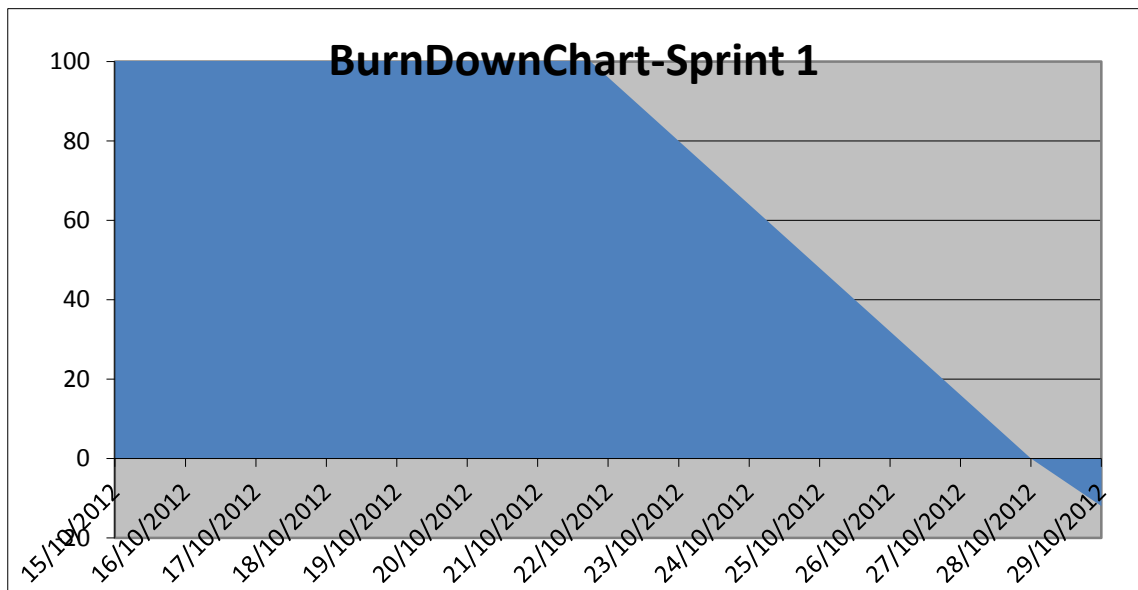


Figura IV.54: BurnDown Chart Sprint 1

Fuente: Los autores

Los Sprint 2, Sprint 3, Sprint 4, Sprint 5 y sus respectivos BurnDown Charts se encuentran en el ANEXO IX.

4.13. Pruebas

Se implementara un plan de pruebas para validar la solución, de este plan depende que la solución sea considerada estable, es necesario tener un ambiente de pruebas. Las pruebas se realizaron con el usuario permitiendo verificar el correcto funcionamiento y detectando ciertos errores que fueron corregidos de manera inmediata. Las pruebas realizadas se detallan en el ANEXO VII.

CONCLUSIONES

- El análisis comparativo de los frameworks Netbeans y Eclipse RCP ha permitido conocer que se puede aplicar un principio básico de Ingeniería de Software como la reutilización de los componentes software de estas herramientas en áreas como la organización de la interfaz de usuario, y en la parte arquitectónica aportaron una adecuada arquitectura modular en el desarrollo de los prototipos y posteriormente en la aplicación cliente enriquecido (aplicación de escritorio) para la Cooperativa de Ahorro y Crédito Chunchi Ltda.
- Se determinó que es importante definir criterios técnicos de preselección previos a la elaboración de los prototipos para los frameworks debido a que algunos no cuentan con documentación suficiente o su desarrollo se encuentra abandonado, consiguiendo así que el análisis final de éstos se lo realice con los que cumplen requisitos mínimos para las aplicaciones de escritorio.
- Se ha determinado que usando Netbeans RCP se pueden desarrollar aplicaciones que permiten cumplir en un alto grado los requisitos determinados en el SRS esto es: elementos de interfaz de usuario que muestran valores correctos y exactos en comparación a las aplicaciones desarrolladas con Eclipse RCP.

- Se ha observado que al desarrollar una aplicación usando Netbeans RCP se pueden obtener altos niveles de confiabilidad del producto final en comparación a las aplicaciones desarrolladas con Eclipse RCP, lo que significa que en caso de producirse errores o fallos en el sistema se requieren de tiempos cortos para volver a su estado de funcionamiento normal.
- Se ha determinado que es importante adaptar el modelo de evaluación de calidad externa del software propuesto en la Norma ISO 9126 ya que es un modelo genérico y determinadas métricas no pueden ser medidas en las aplicaciones cliente enriquecido.
- Se ha determinado que con los dos frameworks analizados se logra un buen nivel de usabilidad esto es que el usuario final con conocimientos intermedios en informática puede entender sin mucho esfuerzo o explicaciones las funcionalidades implementadas, además de que los frameworks con su diseño aportan a la usabilidad de las aplicaciones.
- Se ha observado que al usar Netbeans RCP en el desarrollo de aplicaciones no se obtienen una eficiencia aceptable para las mismas, esto se debe a que requieren de una mayor cantidad de memoria para

su funcionamiento; pero para las aplicaciones de escritorio esto no representa un problema porque las características actuales de los computadores proveen de gran cantidad de procesamiento y memoria, es en ésta característica donde las aplicaciones desarrolladas con Eclipse RCP demuestran un poco de fortaleza.

- Se determinó que una de las razones por las que Eclipse RCP obtiene un bajo puntaje en la calidad del prototipo evaluado se debe a que su API se encuentra en una fase de transición de la versión tres a la cuatro y en este proceso muchas de las funcionalidades están cambiando constantemente.

RECOMENDACIONES

- Se recomienda a los desarrolladores usar estas tecnologías para que las aplicaciones desarrolladas se beneficien de las arquitecturas modulares, componentes ya desarrollados y probados y no empezar desde cero cuando se necesite construir una aplicación cliente enriquecido (aplicación de escritorio).
- Se recomienda a los desarrolladores que al construir aplicaciones de escritorio con requerimientos exigentes en la parte de la interfaz de usuario realizar una combinación de los frameworks Netbeans RCP que provee una arquitectura modular a la aplicación y OpenSwing que contiene un conjunto de librerías especializadas para el desarrollo de interfaz de usuario.
- Se recomienda que al medir calidad del software se debe establecer adecuadamente que parte del proceso de la ingeniería es la que se va a evaluar y en base a eso buscar estándares que permitan lograr este fin.
- Durante la elaboración de prototipos se recomienda que se debe hacer cumpliendo una parte de los requerimientos de la aplicación final para lograr una reducción en el tiempo del desarrollo de la aplicación total.

- La aplicación de los frameworks: Netbeans RCP y Eclipse RCP no sólo puede ser para aplicaciones de escritorio sino también para aplicaciones embebidas debido a la facilidad de integrar funcionalidades ya desarrolladas y probadas

RESUMEN

Análisis comparativo de frameworks software libre para el desarrollo de aplicaciones de escritorio en java. Caso Práctico: Sistema Informático de Servicios Cooperativos "SISCOOP" para la Cooperativa de Ahorro y Crédito Chunchi Ltda. Del cantón Chunchi.

El método utilizado en la investigación fue analítico e investigativo; la parte analítica se desarrolló prototipos con los frameworks seleccionados, luego comparándolos se obtuvo resultados cualitativos y cuantitativos; en la parte investigativa se aplicó el estándar de calidad ISO 9126(Organización Internacional de Estandarización) a los prototipos para determinar el framework que logra un producto de mejor calidad.

Durante el desarrollo se utilizó herramientas software libre: Netbeans RCP (Plataforma de Cliente Enriquecido) para desarrollo, Base de Datos Postgresql 9.1, iReport 5.5 para reportes, UML(Lenguaje de Modelado Unificado) en el modelado y SCRUM (Proceso ágil de desarrollo de software) en la gestión del proyecto.

Las subcaracterísticas del estándar ISO 9126 como: Adecuación, Exactitud, Madurez, Tolerancia a fallos, Recuperabilidad, Comprensibilidad, Operabilidad, Rendimiento y Utilización de Recursos; medidas en los prototipos arrojaron los siguientes resultados: Netbeans RCP con el 85% equivalente a *Aceptable* de acuerdo a la escala MOSCA (Modelo Sistemico para estimar la calidad de los Sistemas de Software) y Eclipse RCP con el 51% equivalente a *No Aceptable*.

Como consecuencia, el framework Netbeans RCP brinda mejor calidad en el desarrollo de aplicaciones de escritorio en Java debido al cumplimiento de los parámetros de calidad, por lo tanto se utiliza en el desarrollo del Sistema "SISCOOP".

Se recomienda a desarrolladores usar Netbeans RCP debido a su facilidad de uso, gran comunidad de desarrolladores y por producir productos software de calidad.

SUMMARY

Comparative analysis of frameworks free software for the development of applications of desktop in Java. Practice case: Informatic System of Cooperative Services "SISCOOP" for de Cooperative of Saving an Credit Chunchi Ltda. Of Chunchi.

The used method was analytic and investigative, the analytic aspect were developed prototypes with the selected framewoks, then compared them and were obtained quantitative and qualitative results; in the investigative aspect was applied the standard of quality ISO 9126(International Organization for Standarization) to the prototypes for determine the framework that achieves a best quality product.

During the development were used tools of free software: Netbeans RCP (Plataatform of rich client) for development, Data Base PostgreSql 9.1, iReport 5.5 for log files, UML (Unifed Model Language) in the modeling and SCRUM (Agile Process of development of Software) in the manament of project.

The subcharacteristics of ISO 9126 like: Adaptation, Exactitud, Maturity, Fault Tolerance, recoverability, comprehensibility, operability, performance and resource utilization, mesures in the prototypes threw the following results: Netbeans RCP with 85% equivalente to Acceptable in accordance with the scale MOSCA (Systemic Model to stimate the quality of Software Systems) and Eclipse RCP with 51% equivalent to Nort Acceptable.

The framework Netbeans RCP provides better quality in the development of desktop application in Java due to the filfullment of the quality parameters; therefore it is used in the development of the "SISCOOP" system.

It is recommended to use Netbeans RCP due to the facility, big community of developers and for offering Software products of quality.

GLOSARIO DE TÉRMINOS

B

Base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente, que permite acceso directo para su posterior uso.

C

Calidad es la totalidad de las características y aspectos de un producto o servicio en los que se basa su aptitud para satisfacer una necesidad dada.

F

Framework está compuesto de varias clases o componentes reutilizables, cada uno de los cuales puede proporcionar una abstracción de algún concepto en particular.

L

Lenguaje Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

M

Métrica es una medida del grado en que un sistema, componente o proceso posee un atributo dado.

O

ORM (Object Record Mapper) es una interfaz que transforma la lógica de objetos a la relacional y viceversa.

P

Patrón de diseño es una base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Plugin es un componente que aporta una función y se puede integrar con otros proyectos

Proceso son actividades software que mayormente conllevan el factor tiempo.

S

Software son aplicaciones que ayudan a crear otras aplicaciones y/o son aplicaciones finales para ser usadas.

BIBLIOGRAFIA

1. **PETRI, J., Y OTROS.,** Guía del desarrollador para la plataforma NetBeans., Birmingham – Reino Unido., Packt Publishing Ltd., 2010., Pp., 11-23.
2. **MEZA ORTIZ., C.,** Revista Internacional de educación en Ingeniería., No. 1., Vol. III., Una Arquitectura Modular Para el desarrollo de un IDE que apoye en la enseñanza de los fundamentos de la POO., 2010., Tepic – México., Academia Journals., Pp., 2-5.
E-BOOK
<http://academiajournals.com/downloads/mezaortiz.pdf>
3. **ANÁLISIS DE PATRONES DE SOFTWARE Y SU APLICACIÓN EN UN FRAMEWORK**
<http://repositorio.espe.edu.ec/handle/21000/2341>
2011-20-12.
4. **ARQUITECTURA DE LA PARTICIPACION: EL MODELO DE DESARROLLO OPEN SOURCE**
<http://www.people.hbs.edu/cbaldwin/DR2/BaldwinArchPartAll.pdf>
2011-20-12.
5. **ARQUITECTURA MODULAR**
<http://www.sevillaweb.es/>
2011-20-12.

6. CARACTERISTICAS DE LA PLATAFORMA NETBEANS

<http://netbeans.org/features/platform/>

2011-21-12.

7. CARACTERISTICAS DE ECLIPSE

[http://es.wikipedia.org/wiki/Eclipse_%28software%29#Caracter.C3.ADsticas.](http://es.wikipedia.org/wiki/Eclipse_%28software%29#Caracter.C3.ADsticas)

2011-17-12.

8. CALIDAD DE SOFTWARE

[http://es.wikipedia.org/wiki/Calidad_de_software#Calidad.](http://es.wikipedia.org/wiki/Calidad_de_software#Calidad)

2013-20-08.

9. CALIDAD DE SOFTWARE

[http://es.wikipedia.org/wiki/Calidad_de_software#Calidad de softwar](http://es.wikipedia.org/wiki/Calidad_de_software#Calidad_de_software)
[e](#)

2013-20-08.

10. COMPONENTES DE OPENSING

[http://oswing.sourceforge.net/doc.html.](http://oswing.sourceforge.net/doc.html)

2013-23-09.

11. CONSTRUYENDO APLICACIONES SOFTWARE CON JMATTER

[http://jmatter.org/documentation/html/index.html.](http://jmatter.org/documentation/html/index.html)

2013-23-08.

12. DEFINICIÓN DE FRAMEWORK

[http://es.wikipedia.org/wiki/Framework.](http://es.wikipedia.org/wiki/Framework)

2011-20-12.

13. DISEÑO E IMPLEMENTACIÓN DE UN MARCO DE TRABAJO (FRAMEWORK) DE PRESENTACIÓN PARA APLICACIONES JEE

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>

2011-20-12.

14. FRAMEWORK GLUE MVC

<http://gluemvc.sourceforge.net/manual/gluemvcmanual-0.1.x.pdf>

2013-16-11.

15. GUIA DE SCRUM

<http://www.scrum.org/Scrum-Guiedes>

2011-17-12.

16. GUIA PARA LA APLICACIÓN DE MÉTRICAS PARA DETERMINAR LA CALIDAD DE UN SISTEMA DE SOFTWARE

<http://bibdigital.epn.edu.ec/bitstream/15000/2194/1/CD-2950.pdf>

2012-10-25.

17. ISO 9126 CARACTERISTICAS DE LA CALIDAD DE SOFTWARE.

<http://www.sqa.net/iso9126.html>

2013-16-05.

18. JAVA PLUG-IN FRAMEWORK (JPF) PROJECT

<http://jpf.sourceforge.net/>

2011-20-12.

19. JSPF JAVA SIMPLE PLUGIN FRAMEWORK. 5 MINUTES AND IT WORKS. NO XML.

<http://code.google.com/p/jspf/>

2011-22-12.

20. NORMA ISO 9126

[http://www.slideshare.net/Kixar/c32cm31-eq2-norma-iso-9126-](http://www.slideshare.net/Kixar/c32cm31-eq2-norma-iso-9126-15470482)

[15470482](http://www.slideshare.net/Kixar/c32cm31-eq2-norma-iso-9126-15470482).

2031-19-09.

21. PROCESO DE DESARROLLO DE SOFTWARE

<http://users.dsic.upv.es/asignaturas/facultad/lsi/doc/IntroduccionProc>

[esoSW.doc](http://users.dsic.upv.es/asignaturas/facultad/lsi/doc/IntroduccionProc)

2011-22-12.

22. TUTORIAL DE ECLIPSE RCP

<http://www.vogella.de/articles/EclipseRCP/article.html>.

2011-17-12.

23. VENTAJAS DE UTILIZAR UN FRAMEWORK

[http://es.scribd.com/doc/55136485/47/Ventajas-de-utilizar-un-](http://es.scribd.com/doc/55136485/47/Ventajas-de-utilizar-un-Framework-de-trabajo)

[Framework-de-trabajo](http://es.scribd.com/doc/55136485/47/Ventajas-de-utilizar-un-Framework-de-trabajo).

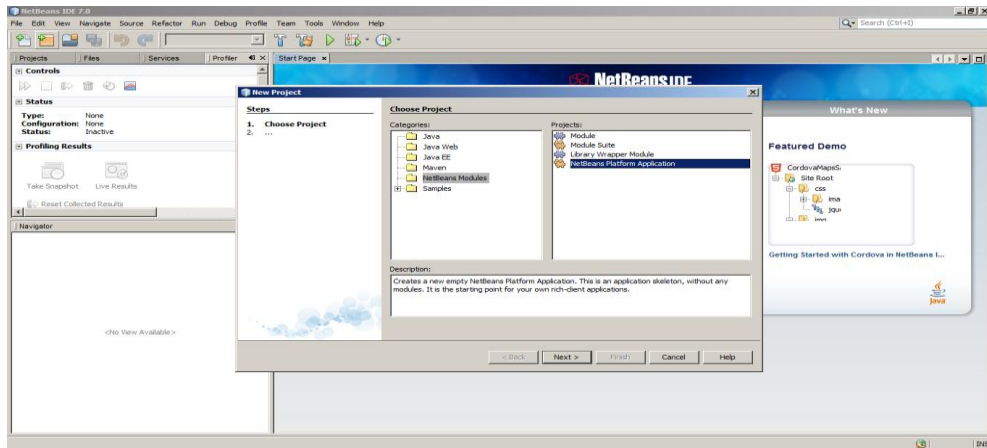
2011-20-12.

ANEXO I

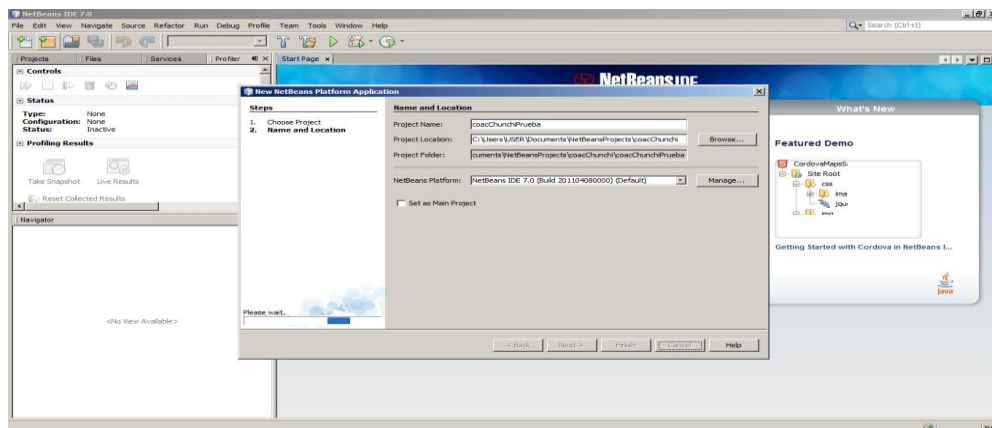
Implementación de los módulos de prueba

IMPLEMENTACIÓN DEL MÓDULO DE PRUEBA EN NETBEANS RCP

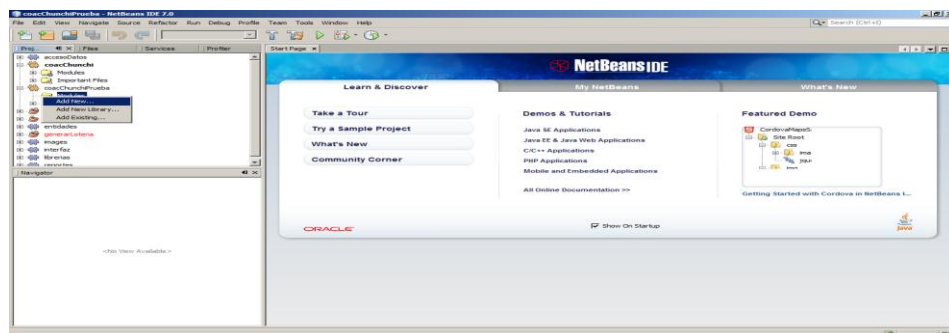
1.- Abrimos el Netbeans 7.4 y damos click en File->nuevo proyecto->Netbeans modules.

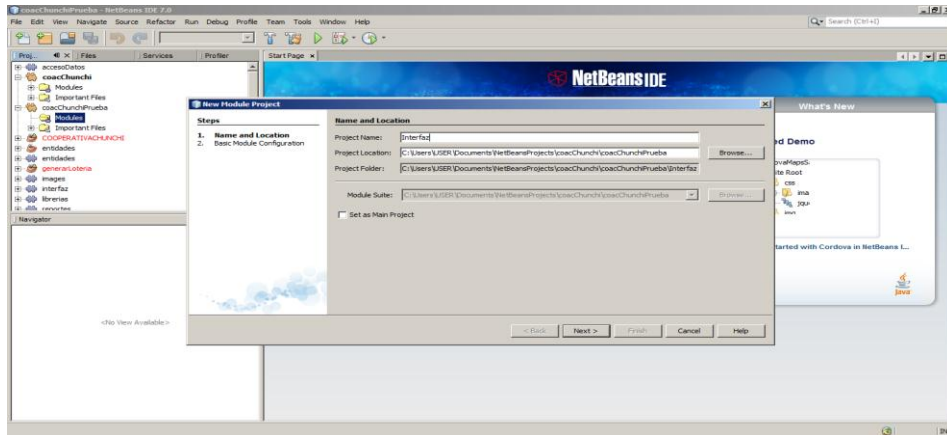


2.- Introducimos el nombre del proyecto y seleccionamos el lugar en donde se guardara el proyecto.

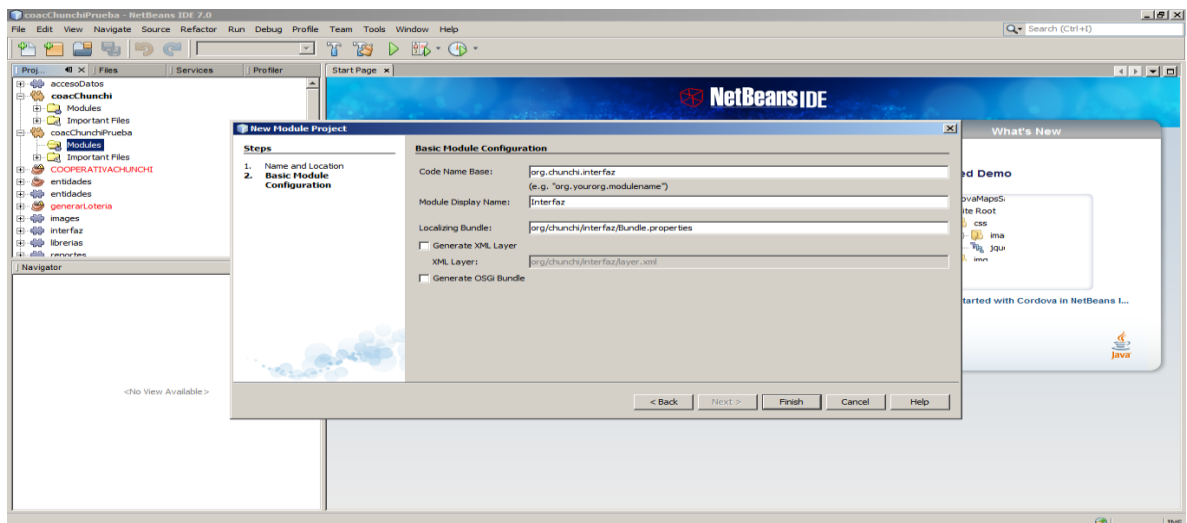


3.- Abrir el proyecto creado anteriormente y hacer click derecho y seleccionar new Module Project e ingresar el nombre del módulo.





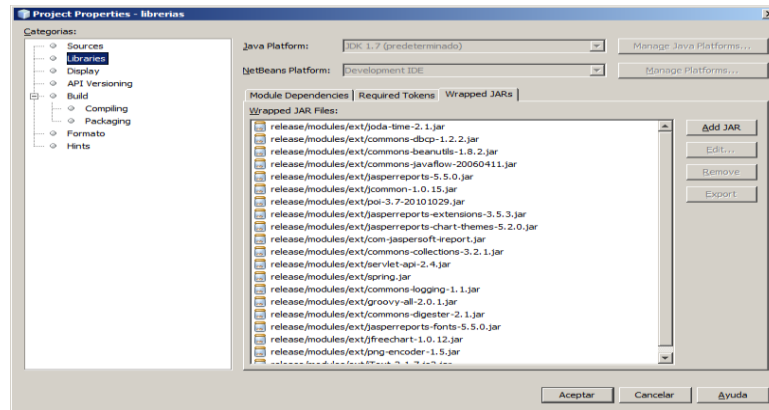
4.- Ingresamos el nombre del paquete del módulo, para nuestro caso tenemos el siguiente nombre: org.chunchi.interfaz, el mismo que contiene todas las interfaces de usuario del módulo de prueba y finalmente dar click en finish.



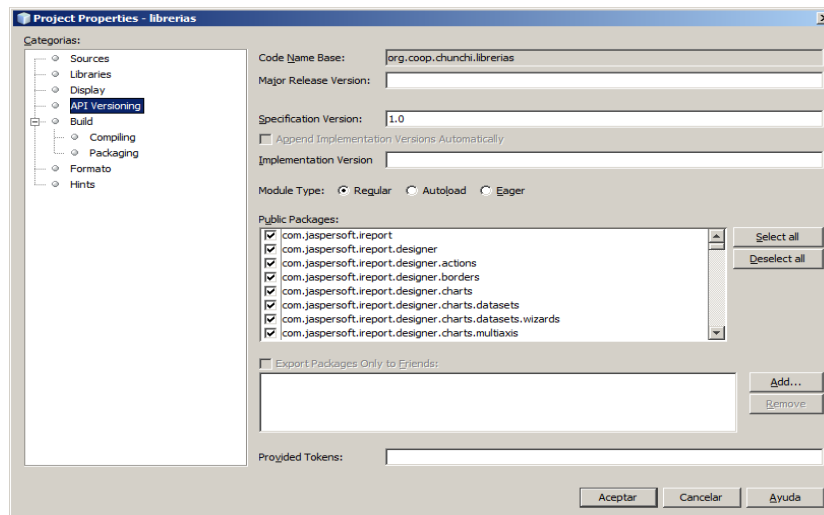
5.- A continuación creamos los módulos: librerías, entidades, accesoDatos y reportes siguiendo los pasos del numeral 3 y 4.

6.- El módulo librerías contiene todas las librerías necesarias para la conexión a la base de datos, creación de reportes y otras librerías que se requieren en este proyecto.

Para añadir las librerías en el módulo hacer click derecho en el módulo librerías->propiedades-> libraries->wrapped JARs-> Add JAR, ubicamos las librerías y damos click en aceptar.



Nota: Para que las librerías añadidas sean reconocidas por los demás módulos ubicamos API Versioning y seleccionamos todos los paquetes para que sean públicos y puedan ser utilizados por otros módulos.

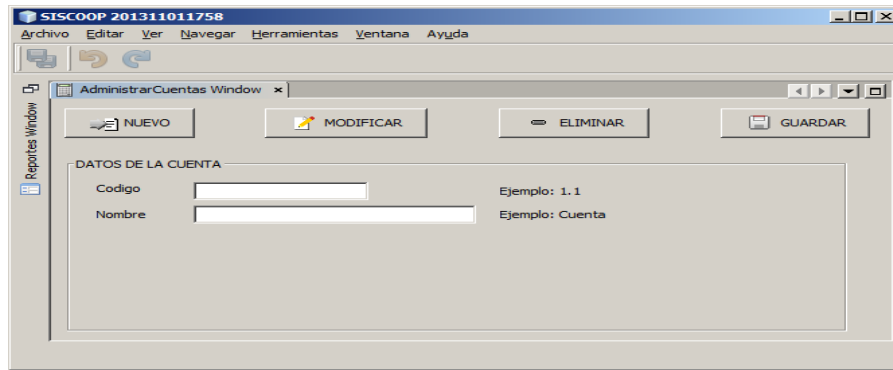


DESARROLLO

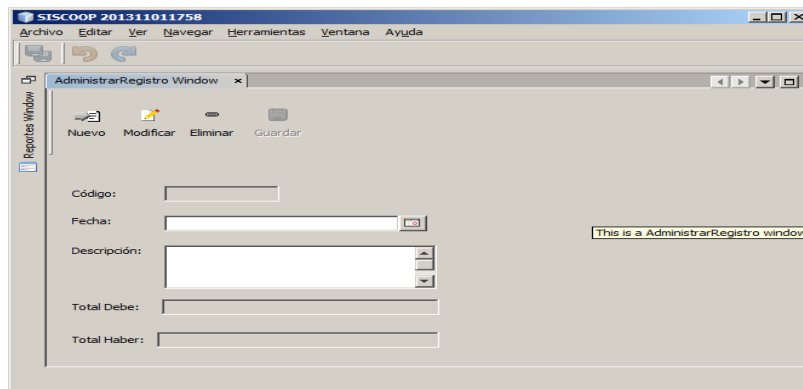
7.- Descripción de módulo interfaz

Este módulo contiene todas las interfaces de usuario necesarias para cumplir con los requerimientos que contiene el mismo.

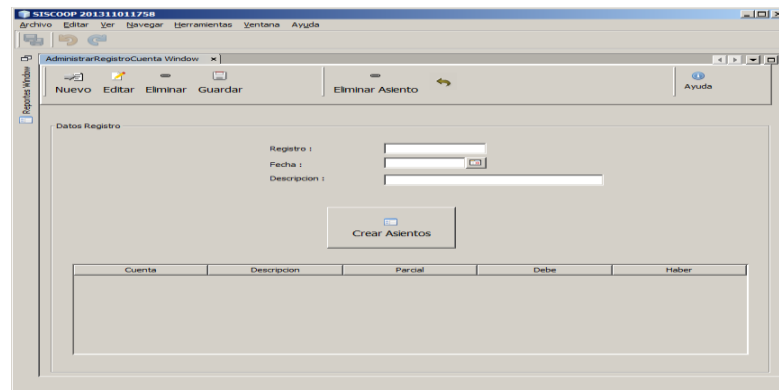
En la figura se observa la interfaz para la gestión de cuentas, la misma permite ingresar, modificar, eliminar y realizar búsquedas filtradas por código y nombre.



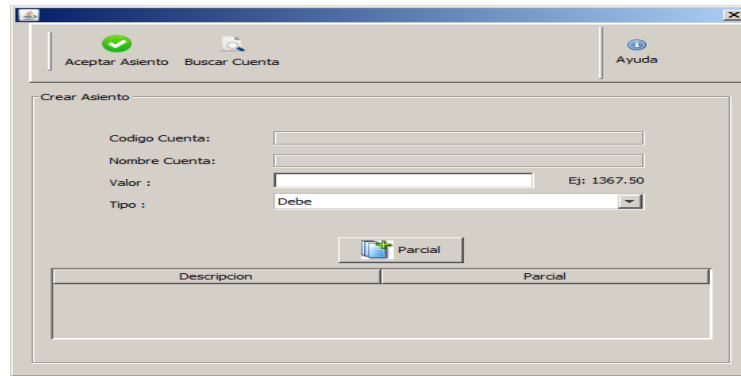
En la figura se observa la interfaz de gestión de Registros, la misma que permite ingresar, modificar, eliminar y realizar búsquedas filtradas por código y nombre.



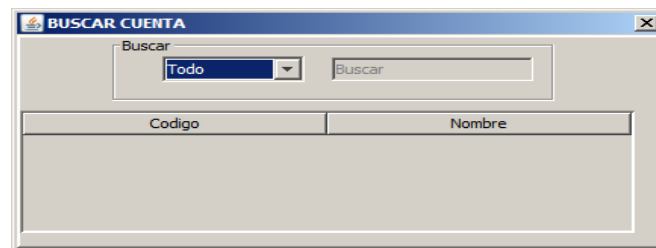
En la figura se observa la interfaz de gestión de Registros_Cuentas, la misma que permite ingresar, modificar, eliminar relaciones entre registros y cuentas.



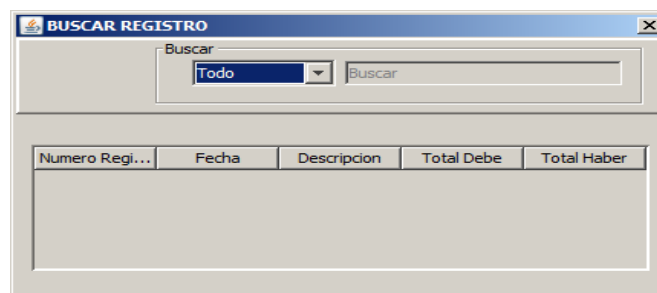
En la figura se observa la interfaz de creación de Asientos contables, dentro de la cual se encuentran las opciones de ingreso, modificación y eliminación de asientos contables.



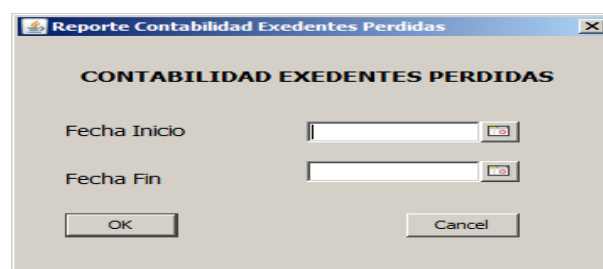
En la figura se observa la interfaz de búsqueda de cuentas.




En la figura se observa la interfaz de búsqueda de registros.



En la figura se observa la interfaz para generar el reporte Exedentes-Perdidas correspondiente al módulo contabilidad.



En la siguiente figura se visualiza el reporte generado desde el módulo reportes.

 ESTADO DE EXCEDENTES O PERDIDAS COOPERATIVA DE AHORRO Y CREDITO CHUNCHI LTDA.					
Trimestre terminado el día		15 febrero 2008	A.E.	14 febrero 2014	
CODIGO	NOMBRE				
4.1.	Ingresos Operacionales				6000.0
4.1.3.	Por intereses y descuentos ganados			6000.0	
4.1.3.01.	Interes en cartes de credito por vencer		2000.0		
4.1.3.02.	Interes en cartes de credito por vencer		2000.0		
5.1.	Gastos operacionales				3000.0
5.1.4.	Gasto de personal			1500.0	
5.1.4.01.	Obligaciones con el publico		500.0		
5.1.4.02.	Obligaciones financieras		500.0		
5.1.4.03.	Otros intereses causados		500.0		
5.1.5.	Gastos generales			1500.0	
5.1.5.01.	Obligaciones financieras		1500.0		
UTILIDAD DE OPERACION					
4.2.	Ingresos no operacionales				3000.0
4.2.1.	Por utilidades financieras			18000.0	18444.0
4.2.1.01.	En venta de acciones		9000.0		
4.2.1.02.	Otros utilidades financieras		9000.0		
4.2.2.	Por servicios no operacionales			444.0	
4.2.2.03.	Arrendamientos		222.0		
4.2.2.04.	Recuperaciones de activos financieros		222.0		
5.2.	Gastos no operacionales				1560.0
5.2.2.	Provisiones			1560.0	
5.2.2.01.	Cartes de credito		777.0		
5.2.2.02.	Cuentas por cobrar		3.0		
5.2.2.03.	Otros		780.0		
EXCEDENTE O PERDIDA DEL EJERCICIO					19884.0

8.- Descripción del módulo AccesoDatos

Este módulo contiene las clases para el acceso a la base de datos y los Dao's que son clases que realizan la operaciones de inserción, modificación y eliminación de objetos desde y hacia la base de datos.

Código de la clase ChunchyEntityManager

```
public class ChunchiEntityManager {
    private static EntityManagerFactory emf = null;
    private static EntityManager em=null;

    private static void init() {
        emf =Persistence.createEntityManagerFactory("entidadesPU");
    }
    /**
     * Creates a new ComicsEntityManager object.
     *
     * @return the entity manager
     */
    public static EntityManager createEntityManager() {
        if (emf == null)
            init();

        return em=emf.createEntityManager();
    }

    public static void iniciarTransaccion(){
```

```

em.getTransaction().begin();
}
    public static void close() {
        em.close();
    }

/**
 * Persist entity.
 *
 * @param entity the entity
 */
public static void persistEntity(Object entity) {
    em.persist(entity);
}

/**
 * Removes the entity.
 *
 * @param entity the entity
 */
public static void removeEntity(Object entity) {
    em.remove(entity);
}

/**
 * Save.
 */
public static void save() {
    em.getTransaction().commit();
    em.getTransaction().begin();
}

/**
 * Checks if is dirty.
 *
 * @return true, if is dirty
 */
public static boolean isDirty() {
    UnitOfWork uow = ((EntityManagerImpl) em).getUnitOfWork();
    return uow.hasChanges();
}
}

```

Ejemplo del código de la clase DaoCuenta

```

public class DaoCuenta extends ChunchiEntityManager{
    public static boolean insertar(Cuenta c)throws Exception {
        boolean estadoOperacion = false;
        try {
            //preparar la llamada al procedimiento almacenado
            if (c.getTxtCodigoCuenta() != null) {

```

```

createEntityManager();
iniciarTransaccion();
    persistEntity(c);
    save();
    estadoOperacion=true;
    close();
    System.out.println("si llevo aca al if");
}else{
    close();
}
}
catch (Exception e) {
}
return estadoOperacion;
}
public static boolean modificar(Cuenta cuenta)throws Exception {
    boolean estadoOperacion = false;
    try {
        //preparar la llamada al procedimiento almacenado
        if(cuenta.getTxtCodigoCuenta()!=null){
            Cuenta toUpdate=createEntityManager().find(Cuenta.class,
cuenta.getTxtCodigoCuenta());
            iniciarTransaccion();
            toUpdate.setTxtNombre(cuenta.getTxtNombre());
            persistEntity(toUpdate);
            save();
            close();
            estadoOperacion=true;
        }
    }
    catch (Exception e) {
        close();
    }
    return estadoOperacion;
}
public static boolean eliminar(Cuenta cuenta)throws Exception {
    boolean estadoOperacion = false;
    try {
        //preparar la llamada al procedimiento almacenado
        if(cuenta.getTxtCodigoCuenta()!=null){
            Cuenta toDelete= createEntityManager().find(Cuenta.class,
cuenta.getTxtCodigoCuenta());
            iniciarTransaccion();
            System.out.println("estoy antes del remove");
            removeEntity(toDelete);
            save();
            close();
            System.out.println("estoy despues del remove");
            estadoOperacion=true;
        }
    }
}
}

```

```

catch (Exception e) {
    close();
}
return estadoOperacion;
}
public List<Cuenta> obtenerCuentasNombre(String nombre)
{
    List<Cuenta> resultList = null;
    try {
        //preparar la llamada al procedimiento almacenado
        createEntityManager();
        iniciarTransaccion();
        Query query = createEntityManager().createQuery("select c from Cuenta c "
            + "where c.txtNombre like '"+nombre+"%");
        resultList=query.getResultList();
        close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,e.getMessage(),"ERROR con la Base Datos" , 0);
    }
    return resultList;
}
public List<Cuenta> obtenerCuentasCodigo(String codigo )
{
    List<Cuenta> resultList = null;
    try {
        //preparar la llamada al procedimiento almacenado
        createEntityManager();
        iniciarTransaccion();
        System.out.println("estoy antes del query codigo");
        Query query = createEntityManager().createQuery("select c from Cuenta c "
            + "where c.txtCodigoCuenta like '"+codigo+"%");
        resultList=query.getResultList();
        System.out.println(resultList.size());
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,e.getMessage(),"ERROR con la Base Datos" , 0);
    }
    return resultList;
}
public List<Cuenta> obtenerCuentas()
{
    List<Cuenta> list = null;
    try {
        //preparar la llamada al procedimiento almacenado
        createEntityManager();
        iniciarTransaccion();
        System.out.println("estoy antes del query");
        Query query = createEntityManager().createNamedQuery("Cuenta.findAll");
        query.setHint(QueryHints.REFRESH, HintValues.TRUE);
        list=query.getResultList();
    } catch (Exception e) {

```



```

JOptionPane.showMessageDialog(null,e.getMessage(),"ERROR con la Base Datos" , 0);
    }
    return list;
}
}

```

9.- Descripción del módulo ReportesConexion.

Este módulo contiene las clases necesarias para la creación de los reportes correspondientes al módulo Contabilidad.

Código de la clase ConexionReportes

```

public class ConexionReportes {
    public static Connection getConnection() throws ClassNotFoundException, SQLException{
        //Configuración de la conexión.
        String driver = "org.postgresql.Driver";
        String connectString = "jdbc:postgresql://localhost:5432/cooperativa";
        String user = "postgres";
        String password = "123456";
        Class.forName(driver);
        Connection conn = DriverManager.getConnection(connectString, user, password);
        //Retornamos la conexión establecida.
        return conn;
    }
}

```

Código de la clase Reportes

```

public class Reportes {
    private JasperReport masterReport;
    public void reporteConParametrosModificado(String urlReporte, Date FECHAINICIO1, Date FECHAFIN1,String IMAGE) throws SQLException, Exception{
        Connection conexion= ConexionReportes.getConnection();
        try{
            System.out.println("Cargando desde: " + urlReporte);
            if(urlReporte == null)
            { System.out.println("No se encuentra el archivo.");
              System.exit(2);
            }

            try {
                masterReport= (JasperReport)JLoader.loadObject(urlReporte);
                System.out.println("archivo cargado correctamente");
            } catch (JRException e)
            {
                System.out.println("Error cargando el reporte maestro: " + e.getMessage());
                System.exit(3);
            }

```

```

    }
    Map parametros=new HashMap();
parametros.put("FECHAINICIO",FECHAINICIO1);//S
    parametros.put("FECHAFIN",FECHAFIN1);
    parametros.put("PIMAGE",IMAGE);
    JasperPrint jasperPrint=
JasperFillManager.fillReport(masterReport,parametros,conexion);
    //Se lanza el Viewerde Jasper, no termina aplicación al salir

    JasperViewer jviewer= new JasperViewer(jasperPrint,false);
jviewer.setTitle("REPORTE GENERAL");
jviewer.setVisible(true);
JasperExportManager.exportReportToPdfFile(urlReporte);
conexion.close();
}catch(SQLException e){
    e.printStackTrace();
}
finally{
    try{
        conexion.close();
    }catch(Exception e){
        e.printStackTrace();
    }
}
}
}
}

```

```

public void reporte(String archivo,String image) throws JRException, SQLException,
ClassNotFoundException{
    try
    {
        String master = archivo;

        JasperReport masterReport = null;
        try
        {
            masterReport = (JasperReport) JRLoader.loadObject(master);
        }
        catch (JRException e)
        {
            System.out.println("Error cargando el reporte maestro: " + e.getMessage());
            //System.exit(3);
        }
        //este es el parametro, se pueden agregar mas parametros
        //basta con poner mas parametro.put
        Map parametro = new HashMap();
        parametro.put("image",image);
    }
}

```

```

        //Reporte diseñado y compilado con iReport
//JasperPrint jasperPrint =
JasperFillManager.fillReport(masterReport,parametro,ConexionReportes.getConnection());
        JasperPrint jasperPrint =
JasperFillManager.fillReport(masterReport,parametro,ConexionReportes.getConnection());

        //Se lanza el Viewer de Jasper, no termina aplicacion al salir
        JasperViewer jviewer = new JasperViewer(jasperPrint,false);
        jviewer.setTitle("REPORTES");
        jviewer.setVisible(true);
    }

    catch (Exception j)
    {
        System.out.println("Mensaje de Error:"+j.getMessage());
    }
}
}
}

```

10.- Descripción de módulo Entidades

Este módulo contiene todas las clases necesarias para crear los objetos cuenta, registro, registro_cuenta, y parcial para ingresar los datos hacia la base de datos.

Ejemplo del código de la clase Cuenta

```

@Entity
@Table(name = "\"Cuenta\"")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Cuenta.findAll", query = "SELECT c FROM Cuenta c"),
    @NamedQuery(name = "Cuenta.findByTxtCodigoCuenta", query = "SELECT c FROM Cuenta c
WHERE c.txtCodigoCuenta = :txtCodigoCuenta"),
    @NamedQuery(name = "Cuenta.findByTxtNombre", query = "SELECT c FROM Cuenta c
WHERE c.txtNombre = :txtNombre")})
public class Cuenta implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "\"txtCodigoCuenta\"")
    private String txtCodigoCuenta;
    @Basic(optional = false)
    @Column(name = "\"txtNombre\"")
    private String txtNombre;
}

```

```

@OneToMany(cascade = CascadeType.ALL, mappedBy = "cuenta")
private List<RegistroCuenta> registroCuentaList;
public Cuenta() { }
public Cuenta(String txtCodigoCuenta) {
    this.txtCodigoCuenta = txtCodigoCuenta;
}
public Cuenta(String txtCodigoCuenta, String txtNombre) {
    this.txtCodigoCuenta = txtCodigoCuenta;
    this.txtNombre = txtNombre;
}
public String getTxtCodigoCuenta() {
    return txtCodigoCuenta;
}
public void setTxtCodigoCuenta(String txtCodigoCuenta) {
    this.txtCodigoCuenta = txtCodigoCuenta;
}
public String getTxtNombre() {
    return txtNombre;
}
public void setTxtNombre(String txtNombre) {
    this.txtNombre = txtNombre;
}
@XmlTransient
public List<RegistroCuenta> getRegistroCuentaList() {
    return registroCuentaList;
}
public void setRegistroCuentaList(List<RegistroCuenta> registroCuentaList) {
    this.registroCuentaList = registroCuentaList;
}
@Override
public int hashCode() {
    int hash = 0;
    hash += (txtCodigoCuenta != null ? txtCodigoCuenta.hashCode() : 0);
    return hash;
}
@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Cuenta)) {
        return false;
    }
    Cuenta other = (Cuenta) object;

```

```
if ((this.txtCodigoCuenta == null && other.txtCodigoCuenta != null) || (this.txtCodigoCuenta !=
null && !this.txtCodigoCuenta.equals(other.txtCodigoCuenta))) {
    return false;
}
return true;
}
@Override
public String toString()
return "org.coop.chunchi.entidades.Cuenta[ txtCodigoCuenta=" + txtCodigoCuenta + " ]";
}}
```

IMPLEMENTACIÓN DEL MÓDULO DE PRUEBA USANDO LA PLATAFORMA ECLIPSE 4 RCP

Esta guía describe la forma de implementar aplicaciones Eclipse RCP usando el nuevo modelo de programación basado en anotaciones e inyección de dependencias propuesto en la plataforma Eclipse 4 RCP

Una aplicación basada en el framework Eclipse RCP está compuesta por componentes específicos individuales llamados plugins además de usar los componentes base de la plataforma Eclipse

Un plug-in tiene dos archivos de configuración importantes:

- MANIFEST.MF, que sirve para definir su API y sus dependencias
- plugin.xml, brinda la posibilidad de crear y contribuir con el API de la plataforma Eclipse

PREREQUISITOS

1. Prerrequisitos

Tener instalado la última versión de Java

2. Descargar e instalar el Eclipse SDK

Para la implementación del módulo se necesita descargar el Eclipse SDK de la siguiente dirección

<http://download.eclipse.org/eclipse/downloads/>

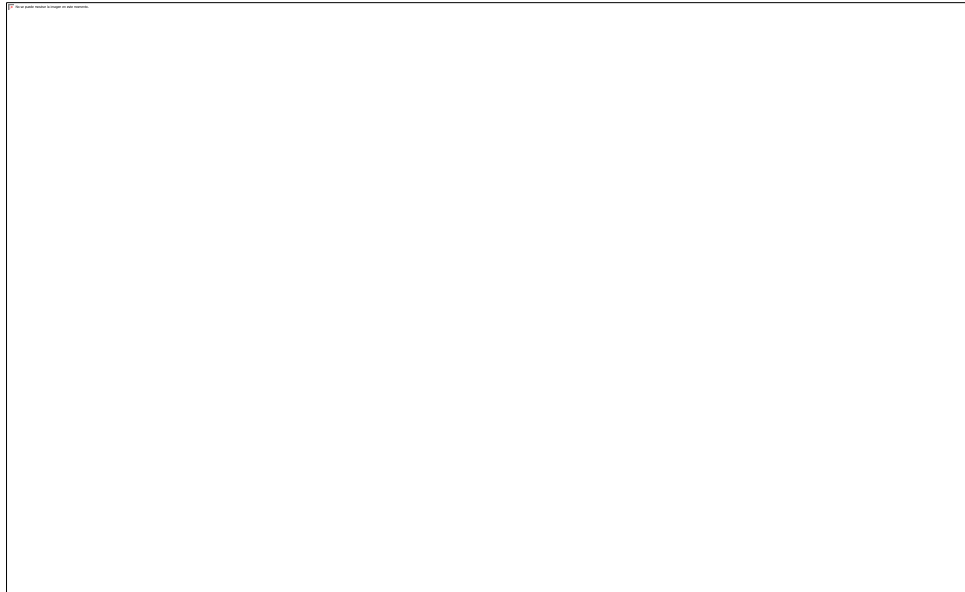
Una vez descargado se debe descomprimir en un directorio cuyo nombre no tenga espacios y ejecutar el archivo *eclipse*

3. Instalar las herramientas e4

La descarga de Eclipse SDK no contiene las herramientas e4, que facilitan el desarrollo de aplicaciones RCP basadas en Eclipse 4

Esto se lo puede realizar a través de *Help -> Install New software* como se indica en la siguiente figura ingresando la siguiente URL:

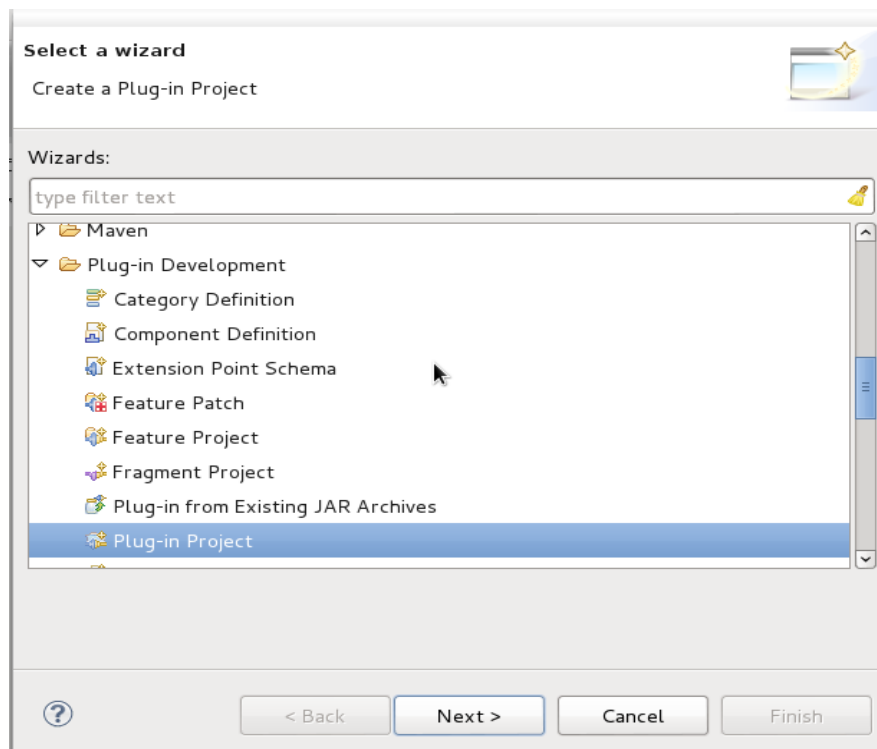
<http://download.vogella.com/kepler/e4tools> de esta dirección únicamente se debe instalar el E4 CSS Spy y el Eclipse e4 Tools.



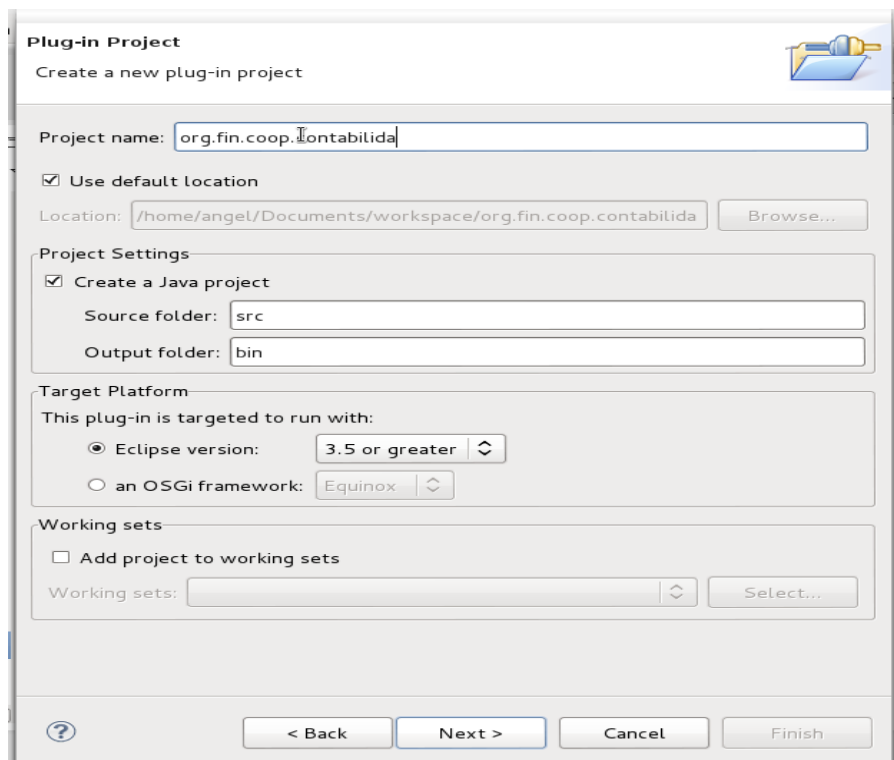
4. DESARROLLO DEL MÓDULO

4.1 Crear el proyecto

Seleccione Archivo -> Nuevo -> Otros-> Plug-in Development-> Plug-in project del menu de Eclipse SDK

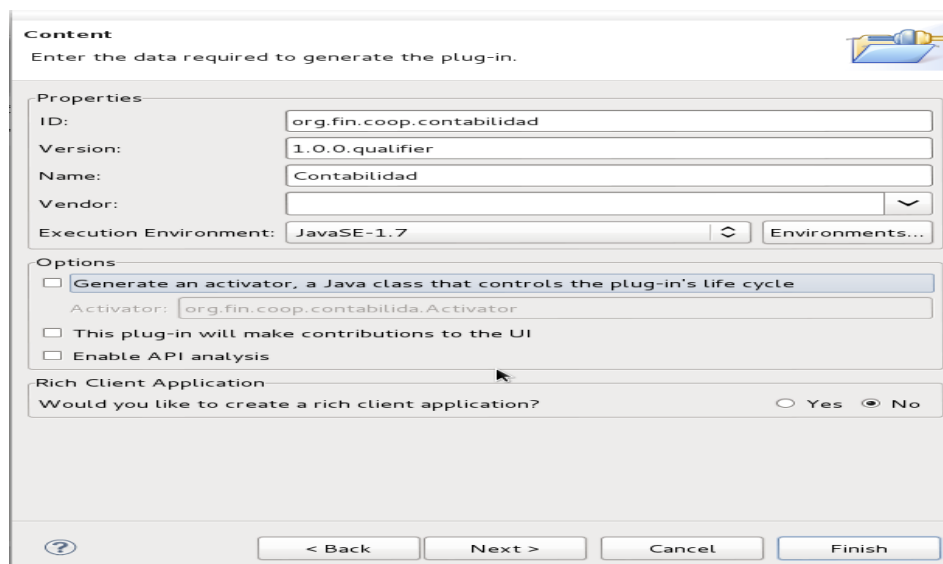


Crear el proyecto *org.fin.coop.contabilidad* y dejar las configuraciones por defecto como se indica en la siguiente figura.



The screenshot shows the 'Plug-in Project' wizard in Eclipse IDE. The title bar reads 'Plug-in Project' and the subtitle is 'Create a new plug-in project'. The 'Project name' field contains 'org.fin.coop.contabilidad'. The 'Use default location' checkbox is checked, and the 'Location' field shows the path '/home/angel/Documents/workspace/org.fin.coop.contabilidad'. Under 'Project Settings', the 'Create a Java project' checkbox is checked, with 'src' as the source folder and 'bin' as the output folder. The 'Target Platform' section has 'Eclipse version: 3.5 or greater' selected. The 'Working sets' section has 'Add project to working sets' unchecked. At the bottom, there are buttons for '< Back', 'Next >', 'Cancel', and 'Finish'.

En la siguiente página del ayudante dejamos las opciones de la siguiente manera



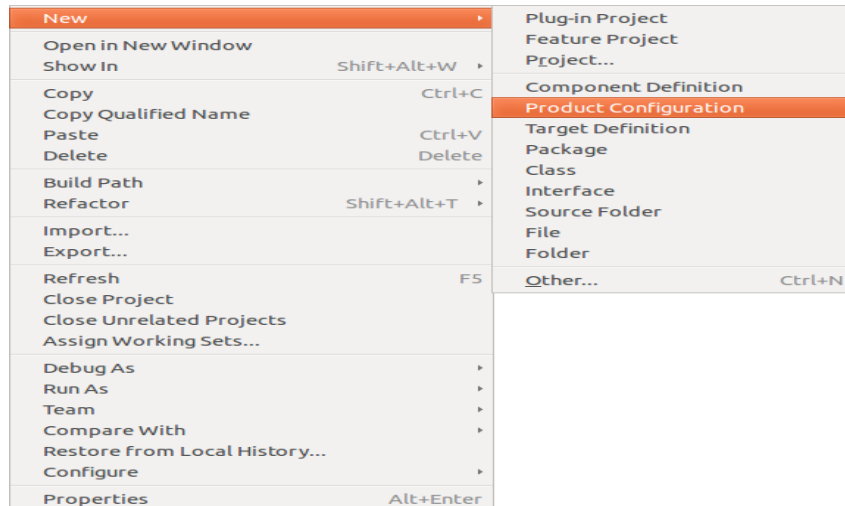
The screenshot shows the 'Content' wizard in Eclipse IDE. The title bar reads 'Content' and the subtitle is 'Enter the data required to generate the plug-in.'. The 'Properties' section has 'ID: org.fin.coop.contabilidad', 'Version: 1.0.0.qualifier', 'Name: Contabilidad', and 'Execution Environment: JavaSE-1.7'. The 'Options' section has 'Generate an activator, a Java class that controls the plug-in's life cycle' checked, with 'Activator: org.fin.coop.contabilidad.Activator' entered. The 'Rich Client Application' section has 'Would you like to create a rich client application?' with 'No' selected. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Cancel', and 'Finish'.

Damos click en el botón finalizar

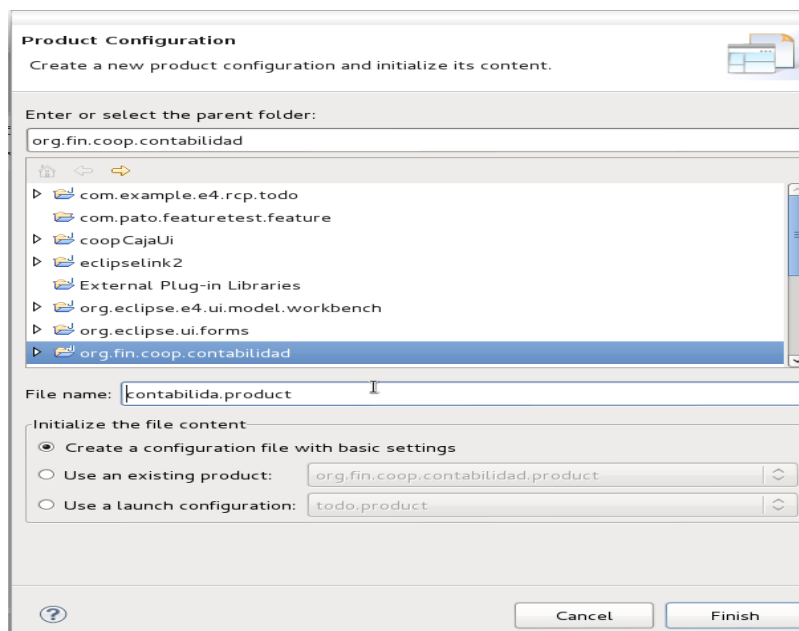
5. Conversión del plug-in en una aplicación Eclipse E4

5.1 Creación del archivo de configuración del producto

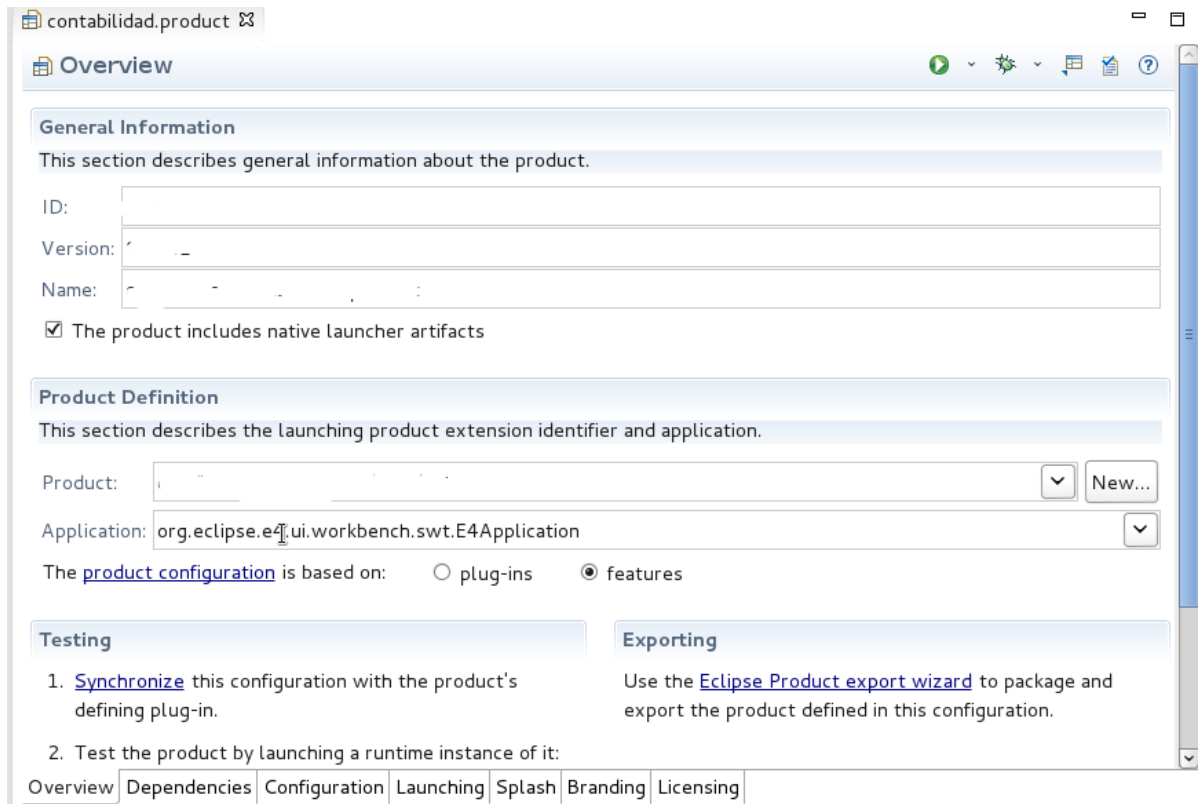
Damos click derecho en el proyecto contabilidad *Nuevo -> Configuración del producto*



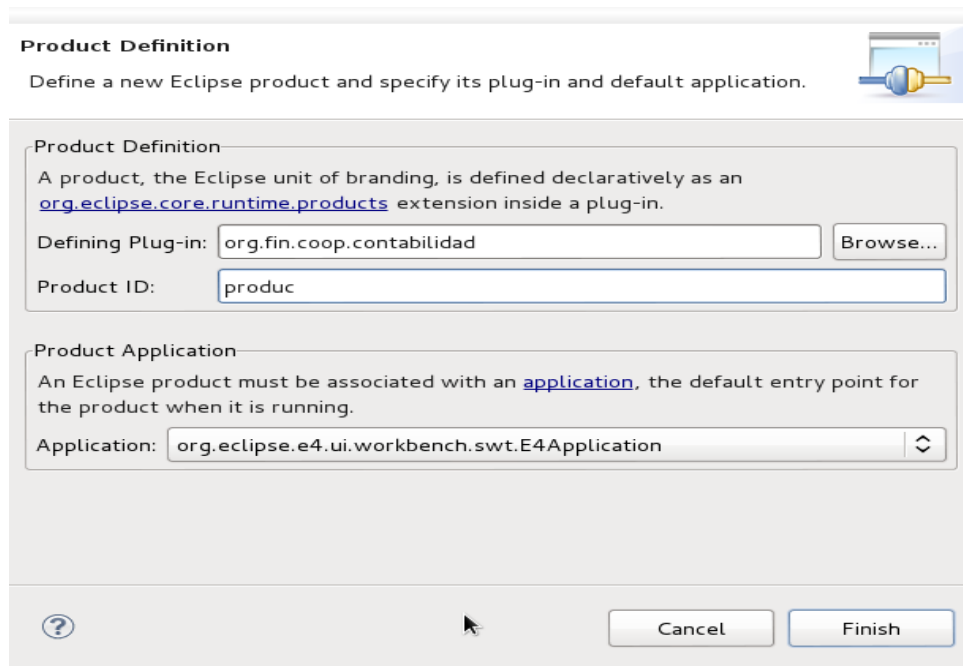
Se ingresa el nombre del archivo de configuración en este caso *contabilidad* como se señala en la siguiente figura.



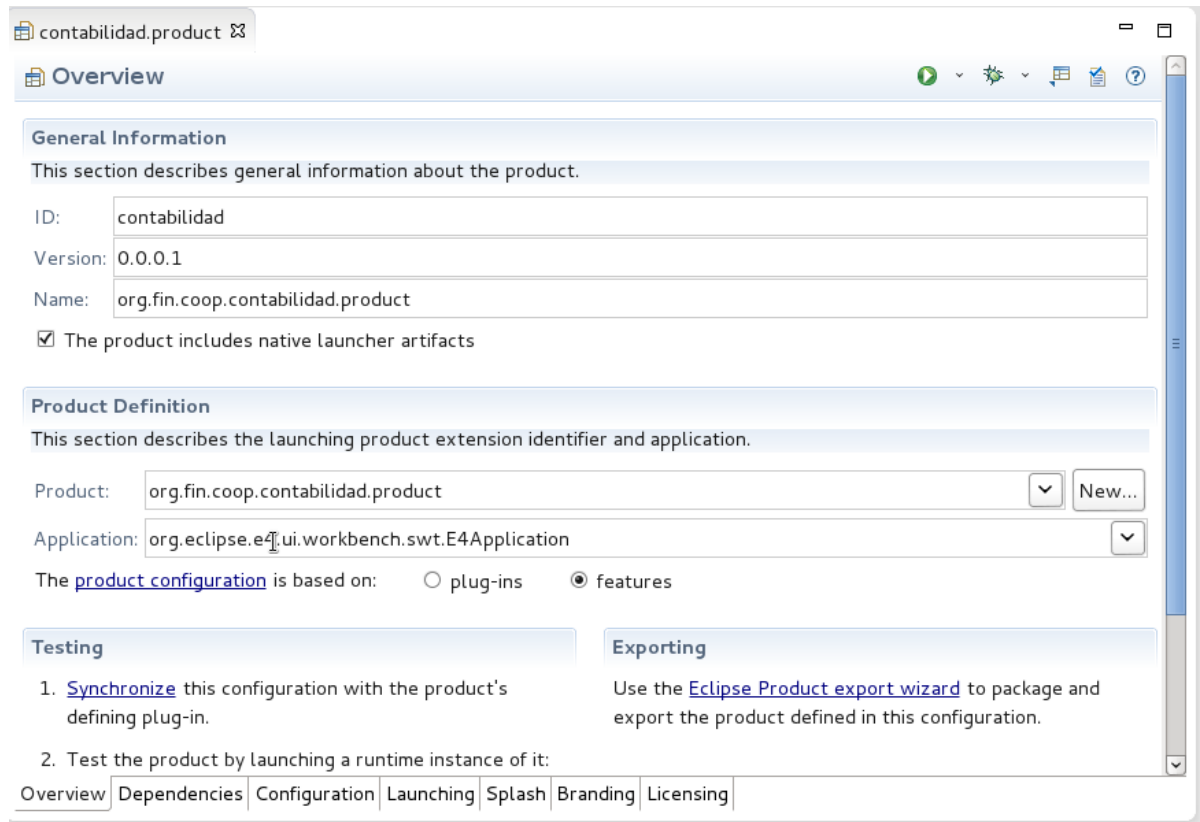
Click en finalizar y este archivo es creado y abierto en el editor como se indica en la siguiente figura e ingresamos el ID: *contabilidad*, Versión 0.0.1 en el Nombre: *org.fin.coop.contabilidad.product*



Damos click en el botón Nuevo y se ingresa el Product ID: como *product*



Al realizar todos estos cambios debe quedar como se indica en la figura

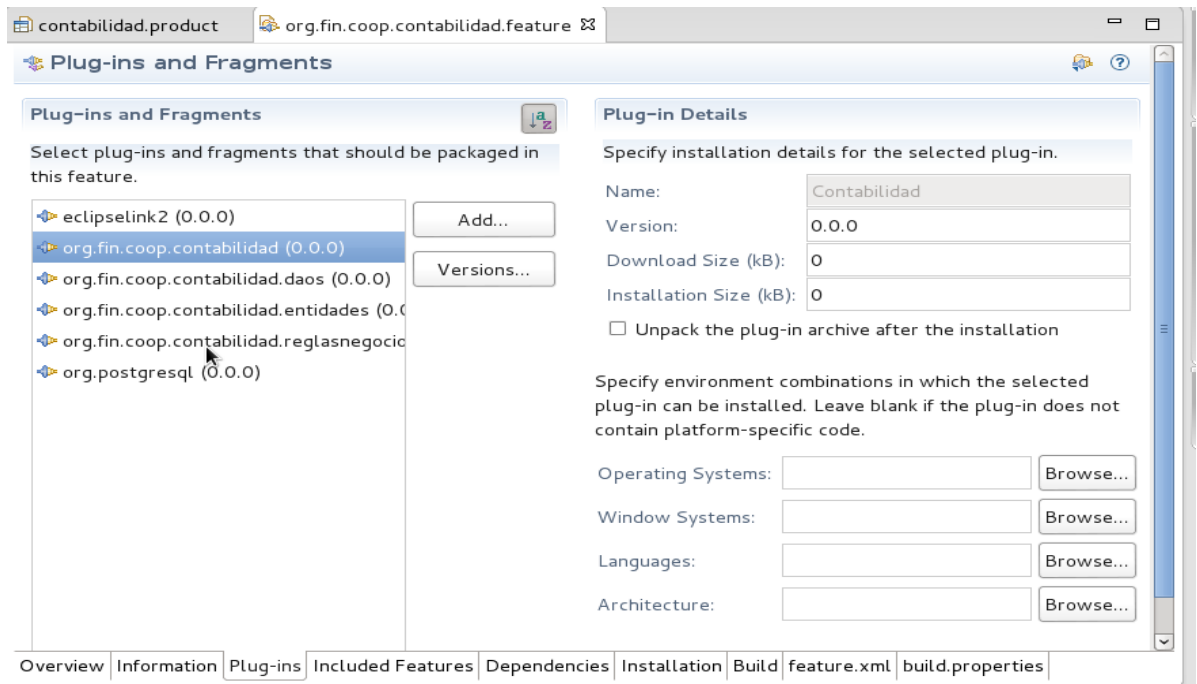


5.2 Creación del Proyecto Feature

Se debe crear un nuevo proyecto llamado *org.fin.coop.contabilidad.feature* a través de *Archivo -> Nuevo -> Otros -> Desarrollo de plug-in -> Proyecto Feature*, el proyecto debe tener la siguiente estructura

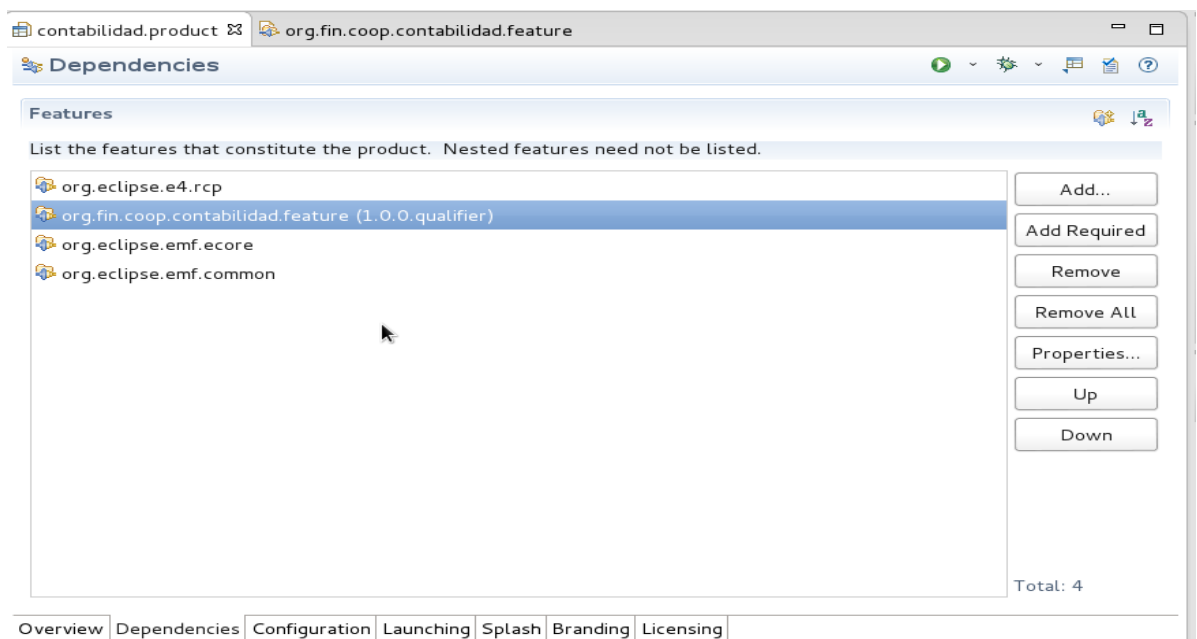


Incluimos el pug-in *org.fin.coop.contabilidad* a través del archivo *feature.xml*



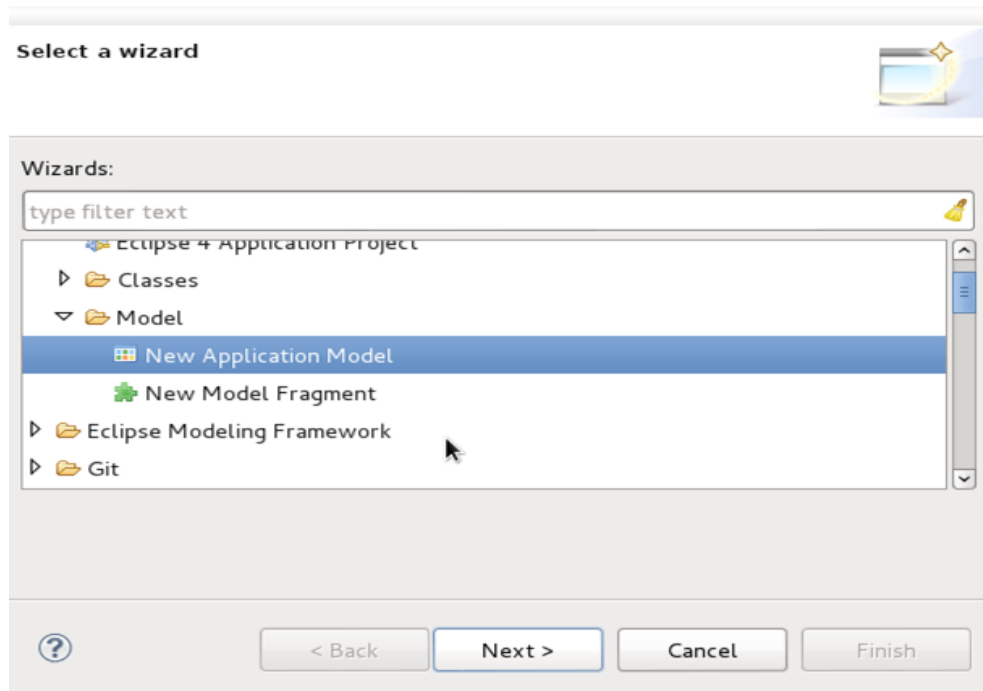
5.3 Ingresar la dependencia características(Features) en el archivo *contabilidad.product*

Seleccionamos la pestaña dependencias y añadimos las siguientes *org.eclipse.e4.rcp*, y la característica *org.fin.coop.contabilidad.feature*, también se debe dar click añadir requeridos y entonces se agregaran las siguientes características *org.eclipse.emf.ecore* , *org.eclipse.emf.ecore.common* como se indica en la figura

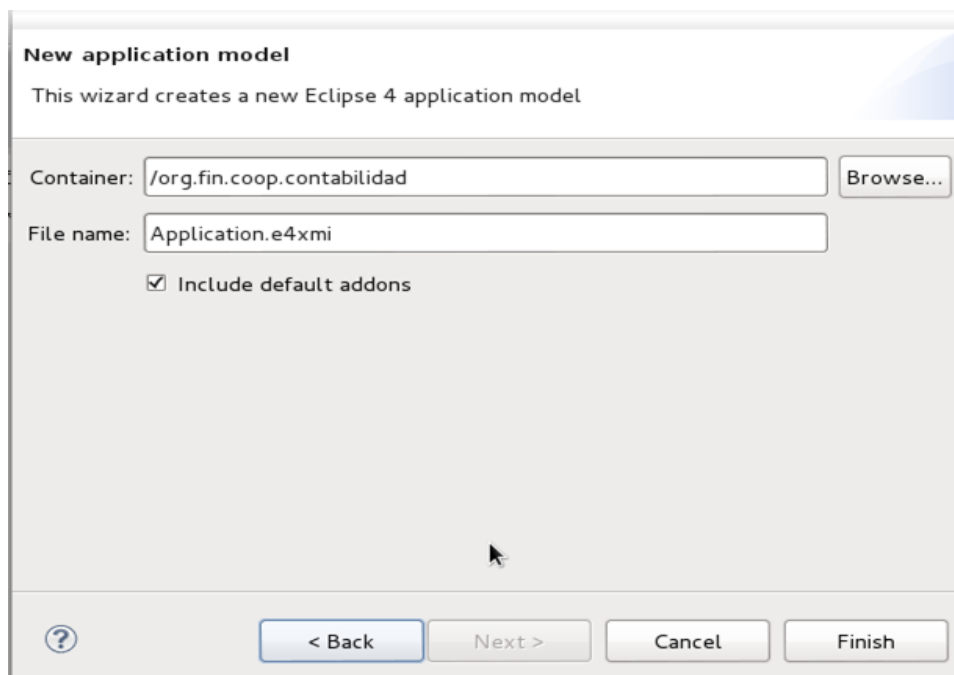


5.4 Creación del modelo de la aplicación

En el proyecto *org.fin.coop.contabilidad* seleccionamos *Archivo -> Nuevo -> Otros -> Eclipse4 -> Modelo -> Nuevo Modelo de Aplicación* para abrir el asistente para la creación del archivo que contiene el modelo de la aplicación como se indica a continuación.



Ingresamos el plug-in *org.fin.coop.contabilidad* como contenedor y aceptamos el nombre por defecto para el archivo que contiene el modelo de la aplicación de acuerdo a la siguiente figura.

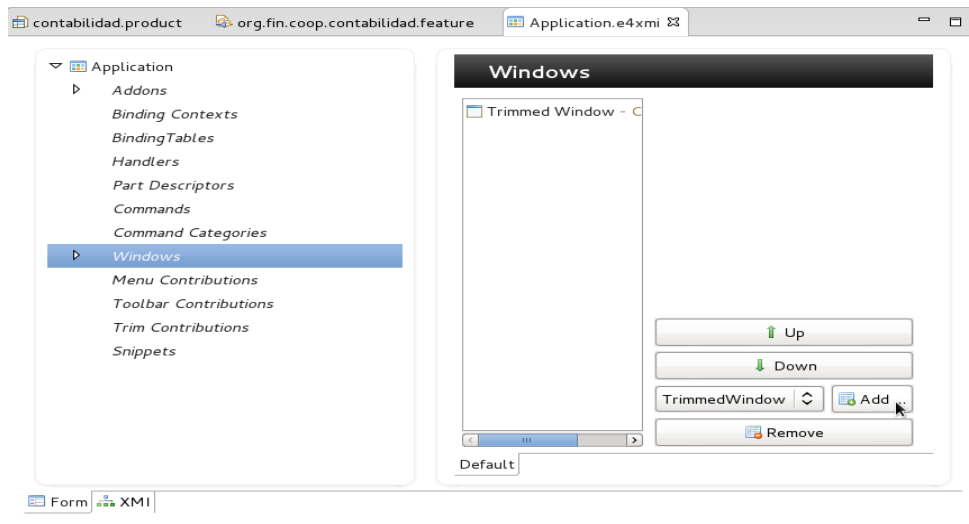


Una vez seleccionado finalizar el archivo creado *Application.e4xmi* será abierto y mostrado en el editor

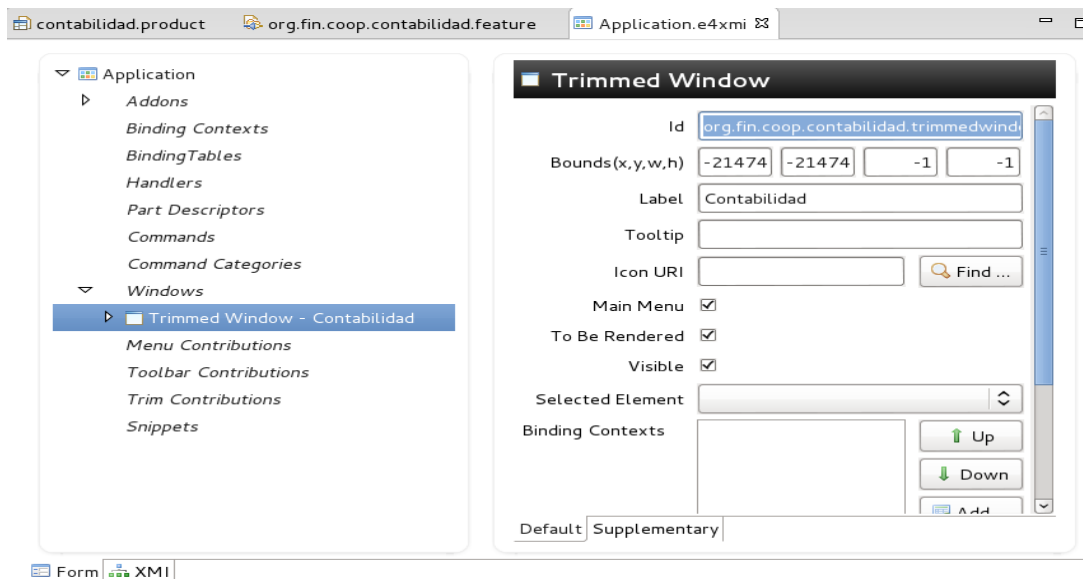
5.5 Agregar elementos pertenecientes al modelo de la aplicación

Agregamos un elemento visual *ventana* al modelo de la aplicación.

Seleccionamos el nodo Ventanas y pulsamos el botón agregar para agregar un *TrimmedWindow*



5.6 Ingresamos el ID *org.fin.coop.contabilidad.trimmedwindow.0* y la posición como se indica a continuación

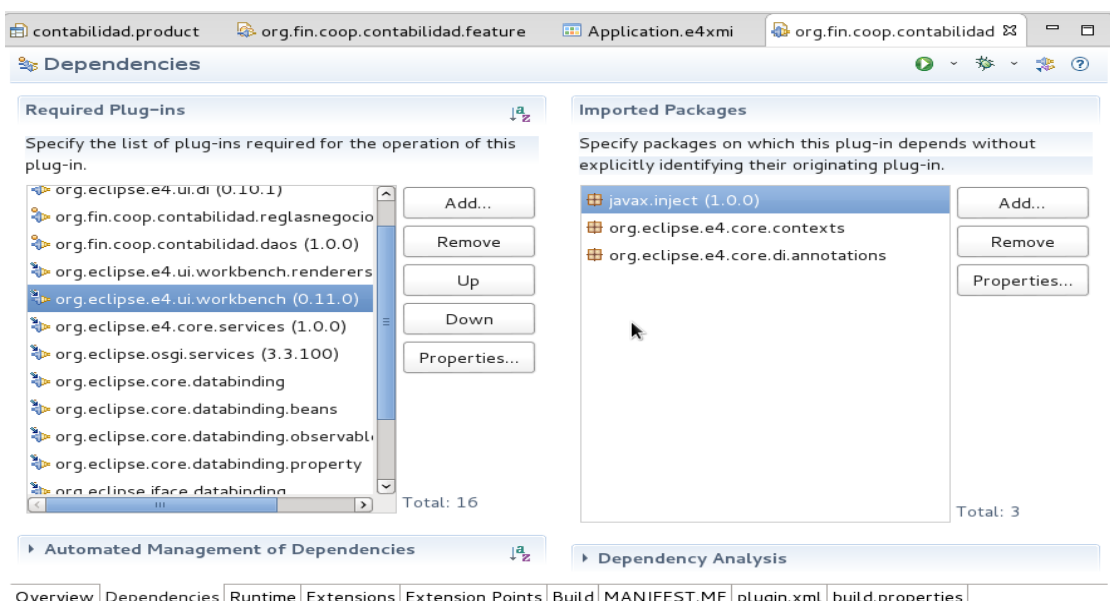


5.7 Agregar dependencias para diseñar las interfaces de usuarios

En el plug-in `org.fin.coop.contabilidad` abrimos el archivo META-INF/MANIFEST.MF seleccionamos la pestaña *dependencias* y *agregamos los siguientes plug-ins* como dependencia (los plug-ins contienen clases para el desarrollo del módulo de prueba desarrollado `org.fin.coop.contabilidad.reglasnegocio`, `org.fin.coop.contabilidad.daos`)

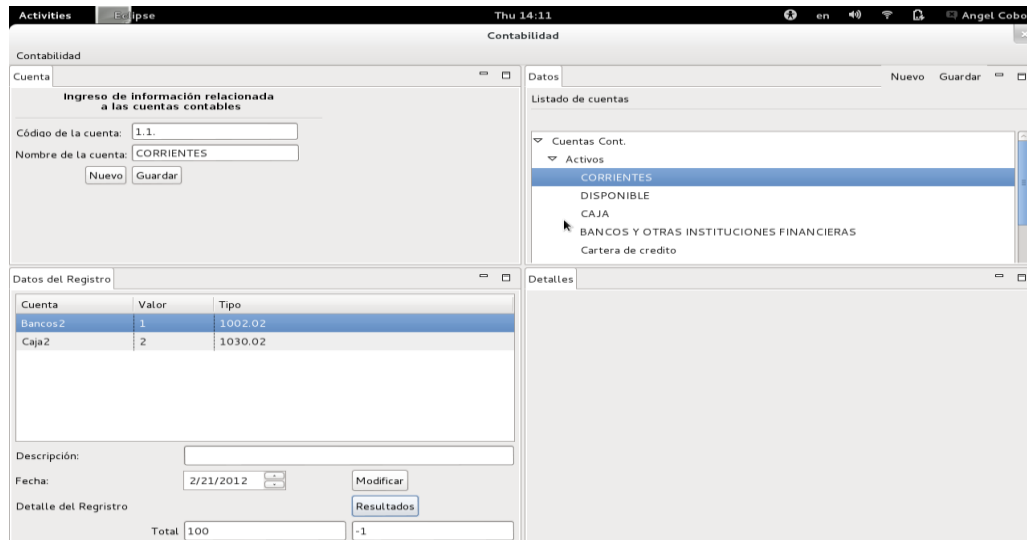
- `org.eclipse.swt;bundle-version="3.100.1"`,
- `org.eclipse.e4.ui.di;bundle-version="0.10.1"`,
- `org.fin.coop.contabilidad.reglasnegocio;bundle-version="1.0.0"`,
- `org.fin.coop.contabilidad.daos;bundle-version="1.0.0"`,
- `org.eclipse.e4.ui.workbench.renderers.swt;bundle-version="0.10.3"`,
- `org.eclipse.e4.ui.workbench;bundle-version="0.11.0"`,
- `org.eclipse.e4.core.services;bundle-version="1.0.0"`,
- `org.eclipse.osgi.services;bundle-version="3.3.100"`,
- `org.eclipse.core.databinding`,
- `org.eclipse.core.databinding.beans`,
- `org.eclipse.core.databinding.observable`,
- `org.eclipse.core.databinding.property`,
- `org.eclipse.jface.databinding`,
- `org.eclipse.equinox.common`,
- `com.ibm.icu`,
- `org.eclipse.jface;bundle-version="3.8.102"`

Como se puede apreciar en la siguiente figura

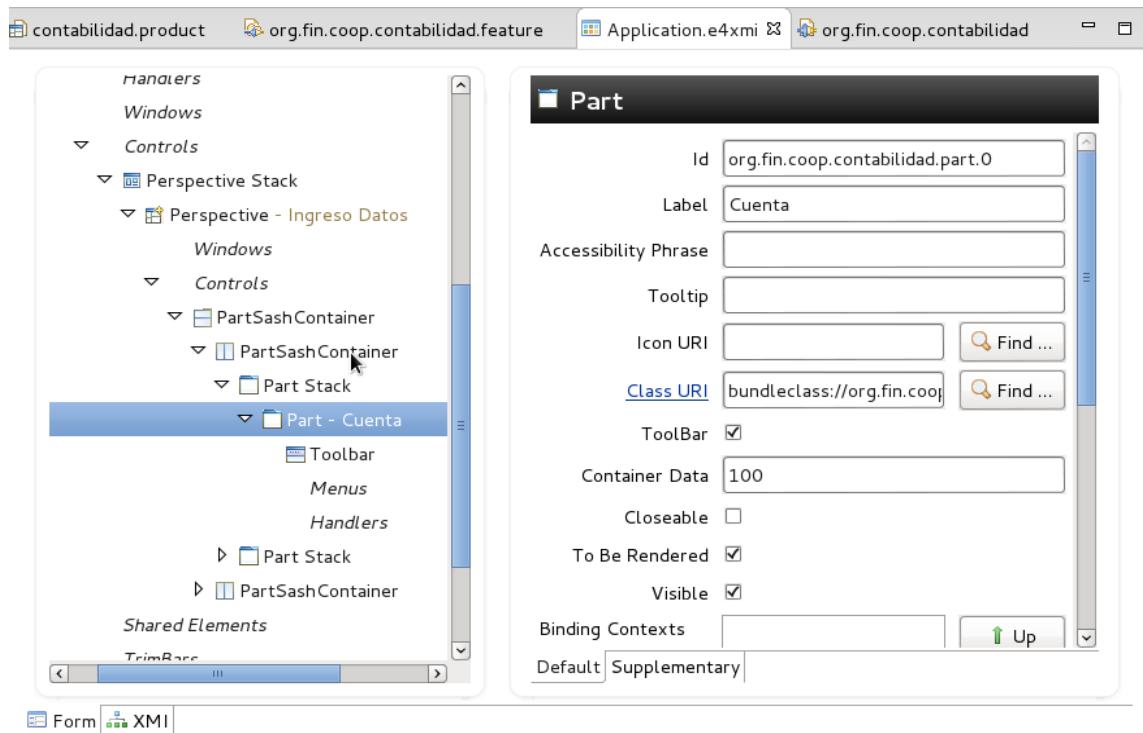


5.8 Modelando la interfaz de usuario

El módulo de prueba tendrá los siguientes elementos y estructura de la interfaz de usuario: Menu, la pantalla dividida en tres partes(búsqueda, visualización de datos y edición de detalles relacionados a las cuentas contables).



Cada parte de la interfaz se crea mediante una perspectiva, un partSashContainer de orientación vertical, y dos de orientación horizontal; dentro de cada uno de estos un partStack y de la misma manera dentro de cada partStack un elemento Part como se indica en la siguiente figura.



La propiedad *Class URI* hace referencia a la clase que implementa las funcionalidades de los elementos de la interfaz de usuario hecha en Java, el valor para esta propiedad es *bundleclass://org.fin.coop.contabilidad/org.fin.coop.contabilidad.parts.UICuenta*.

Esta clase tiene la siguiente estructura y elementos.

```

package org.fin.coop.contabilidad.parts;
import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import javax.inject.Inject;
import org.eclipse.core.databinding.DataBindingContext;
import org.eclipse.core.databinding.beans.PojoProperties;
import org.eclipse.core.databinding.observable.value.IObservableValue;
import org.eclipse.e4.core.di.annotations.Optional;
import org.eclipse.e4.ui.di.Focus;
import org.eclipse.e4.ui.di.UIEventTopic;
import org.eclipse.jface.databinding.swt.WidgetProperties;
import org.eclipse.swt.SWT;
import org.eclipse.swt.events.SelectionAdapter;

```

```

import org.eclipse.swt.events.SelectionEvent;
import org.eclipse.swt.widgets.Button;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.widgets.Text;
import org.eclipse.wb.swt.SWTResourceManager;
import org.fin.coop.contabilidad.daos.DAOCuentaEditor;
import org.fin.coop.contabilidad.reglasnegocio.ContCuenta;
import org.fin.coop.contabilidad.reglasnegocio.NodoCuenta;
/**
 * @author angel
 *
 */
public class UICuenta {
    private DataBindingContext m\_bindingContext;
    private Text txtCodigoCuenta;
    private Text txtNombreCuenta;
    private ContCuenta cuenta;
    private DAOCuentaEditor daoCuentaEditor;
    //private DAOCuentas daoCuenta;
    @PostConstruct
    public void createUi(Composite parent){
        //Incializando las reglas de negocio
        cuenta= new ContCuenta();
        daoCuentaEditor = new DAOCuentaEditor(cuenta);
        //daoCuenta= new DAOCuentas();
        parent.setLayout(null);
        Label lblIngresoDeInformacin = new Label(parent, SWT.NONE);
        lblIngresoDeInformacin.setBounds(65, 5, 262, 34);
        lblIngresoDeInformacin.setAlignment(SWT.CENTER);
        lblIngresoDeInformacin.setFont(SWTResourceManager.getFont("Sans", 10,
SWT.BOLD));
        lblIngresoDeInformacin.setText("Ingreso de información relacionada\n a las cuentas
contables");

```

```

Label separadorTitulo = new Label(parent, SWT.SEPARATOR |
SWT.HORIZONTAL);
separadorTitulo.setBounds(5, 44, 382, 2);
Label lblCodigoDeLa = new Label(parent, SWT.NONE);
lblCodigoDeLa.setBounds(5, 56, 131, 17);
lblCodigoDeLa.setText("Código de la cuenta:");
txtCodigoCuenta = new Text(parent, SWT.BORDER);
txtCodigoCuenta.setBounds(149, 51, 209, 27);
txtCodigoCuenta.setToolTipText("Ejemplo: 1.1.");
Label lblNombreDeLa = new Label(parent, SWT.NONE);
lblNombreDeLa.setBounds(5, 88, 139, 17);
lblNombreDeLa.setText("Nombre de la cuenta:");
txtNombreCuenta = new Text(parent, SWT.BORDER);
txtNombreCuenta.setBounds(149, 83, 210, 27);
txtNombreCuenta.setToolTipText("Ejemplo: Caja");
Button btnNuevo = new Button(parent, SWT.NONE);

btnNuevo.setBounds(91, 115, 53, 29);
btnNuevo.addListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        txtCodigoCuenta.setText("");
        txtNombreCuenta.setText("");
    }
});
btnNuevo.setText("Nuevo");

Button btnGuardar = new Button(parent, SWT.NONE);
btnGuardar.setBounds(149, 115, 64, 29);
btnGuardar.addListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        cuenta.setTxtCodigoCuenta( txtCodigoCuenta.getText());
        cuenta.setTxtNombreCuenta( txtNombreCuenta.getText());
        daoCuentaEditor.setContCuenta(cuenta);
        daoCuentaEditor.persistirContCuenta();
    }
});

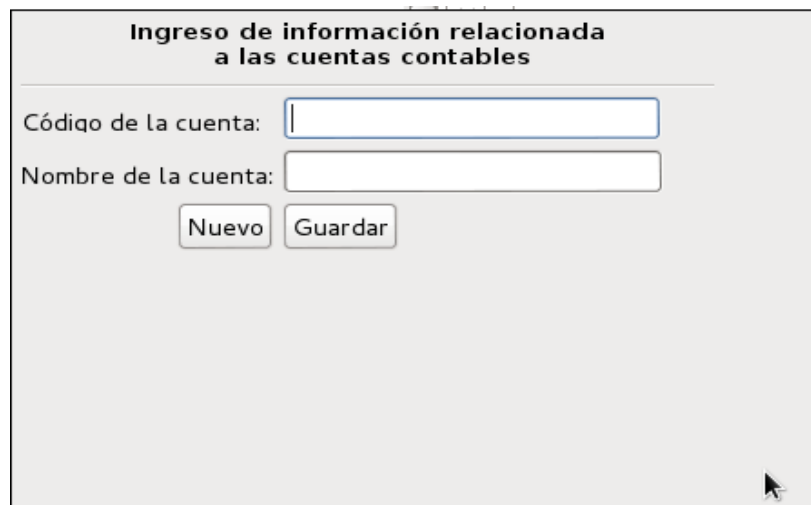
```

```

    }
});
btnGuardar.setText("Guardar");
btnGuardar.setFocus();
m_bindingContext = initDataBindings();
}
@Inject
@Optional
private void getNotified(@UIEventTopic(InterfazEventos.TOPIC_TODO_NEW)
    NodoCuenta nc) {
    if (txtNombreCuenta!=null) {
        txtCodigoCuenta.setText(nc.getCodigo());
        txtNombreCuenta.setText(nc.getNombre());
    }
}
@PreDestroy
public void dispose() {
}
@Focus
private void setFocus(){
    txtCodigoCuenta.setFocus();
}
protected DataBindingContext initDataBindings() {
    DataBindingContext bindingContext = new DataBindingContext();
    //
    ObservableValue observeVisibleTxtCodigoCuentaObserveWidget =
WidgetProperties.visible().observe(txtCodigoCuenta);
    ObservableValue bytesCuentagetTxtCodigoCuentaObserveValue =
PojoProperties.value("bytes").observe(cuenta.getTxtCodigoCuenta());
    bindingContext.bindValue(observeVisibleTxtCodigoCuentaObserveWidget,
bytesCuentagetTxtCodigoCuentaObserveValue, null, null);
    //
    return bindingContext;
}}

```

La interfaz de usuario tiene los siguientes elementos y estructura



Ingreso de información relacionada a las cuentas contables

Código de la cuenta:

Nombre de la cuenta:

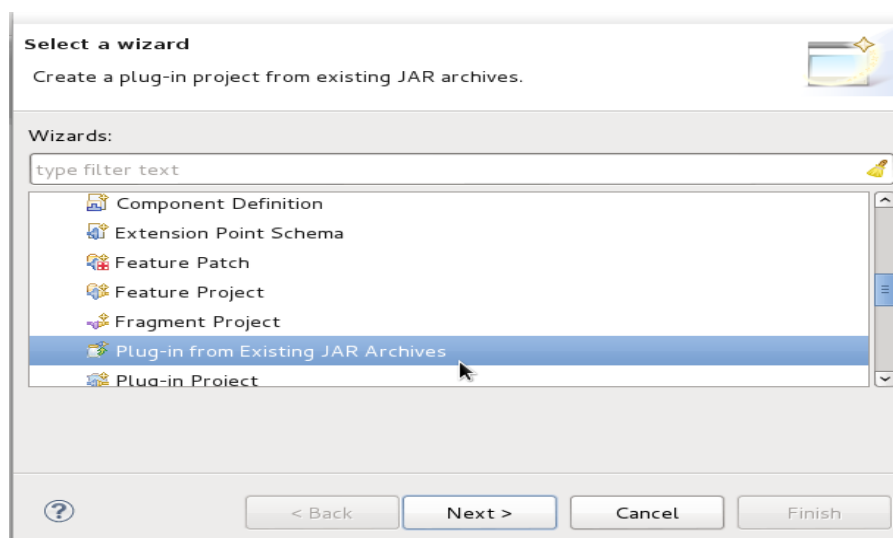
Adicional a esto se han desarrollado las siguientes clases para los demás elementos de interfaz de usuario cuya implementación se detalla en el anexo del código fuente del módulo Eclipse RCP:

- UICuentaEditor
- UIRegistroCuenta

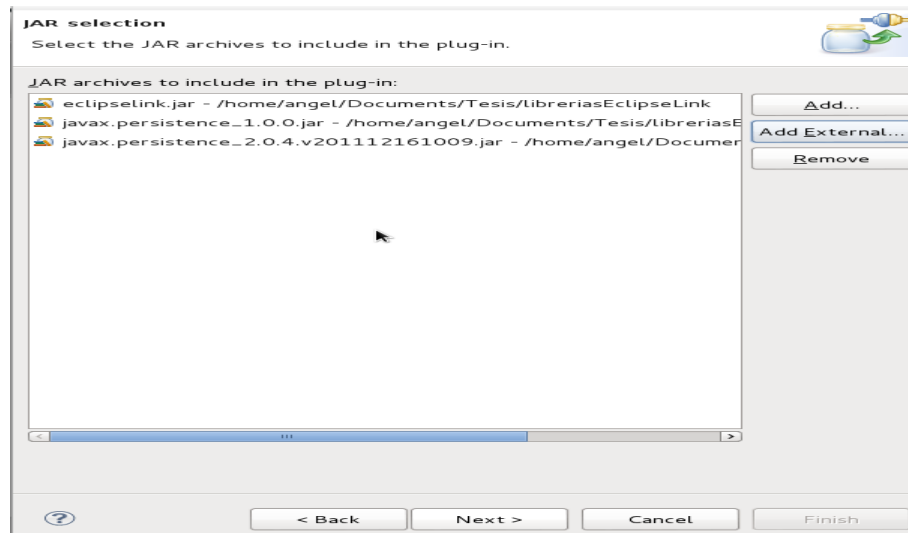
5.9 Plug-ins adicionales desarrollados

- *eclipseLink2*, que contiene las librerías para ejecutar la persistencia

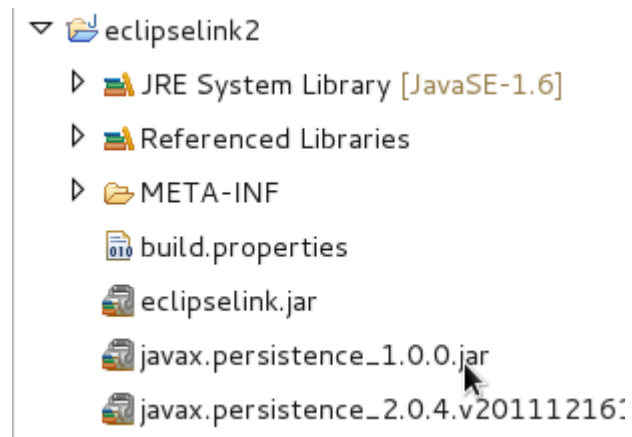
La creación de este plug-in es *Archivo -> Nuevo -> Otros -> Desarrollo de plug-ins -> Plug-in desde archivos Jar existentes* y seleccionados las librerías de eclipseLink para persistencia como se indica a continuación,



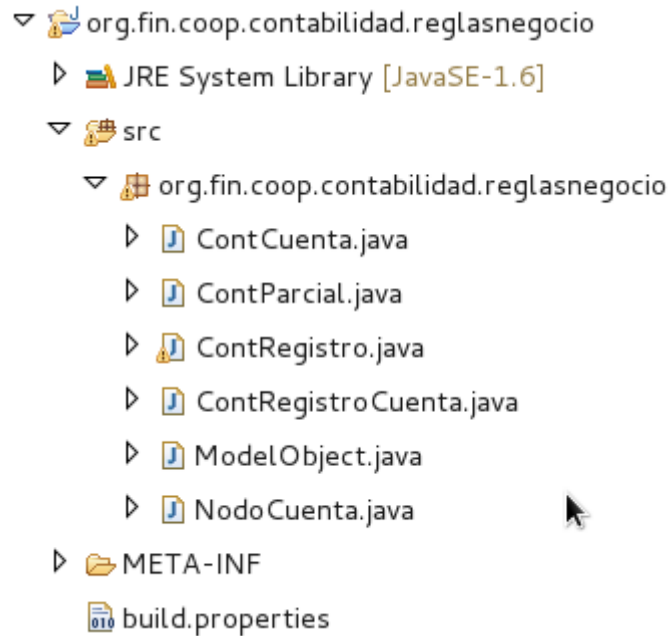
Seleccionamos las librerías como en la figura siguiente.



Y al final queda con la siguiente estructura



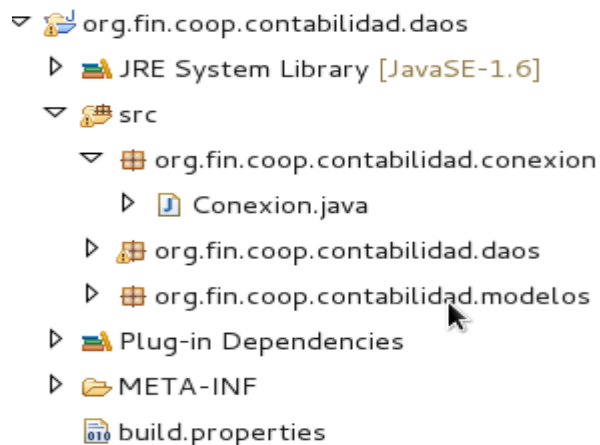
- *Plug-in org.fin.coop.contabilidad.reglasnegocio, que implementa las clases de lógica de negocio para interfaz de interfaz de usuario*



- *plug-in org.postgresql*, que contiene las librerías para la conexión a la base de datos postgresql



- *plug-in org.fin.coop.contabilidad.daos*, contiene las clases para el acceso a datos

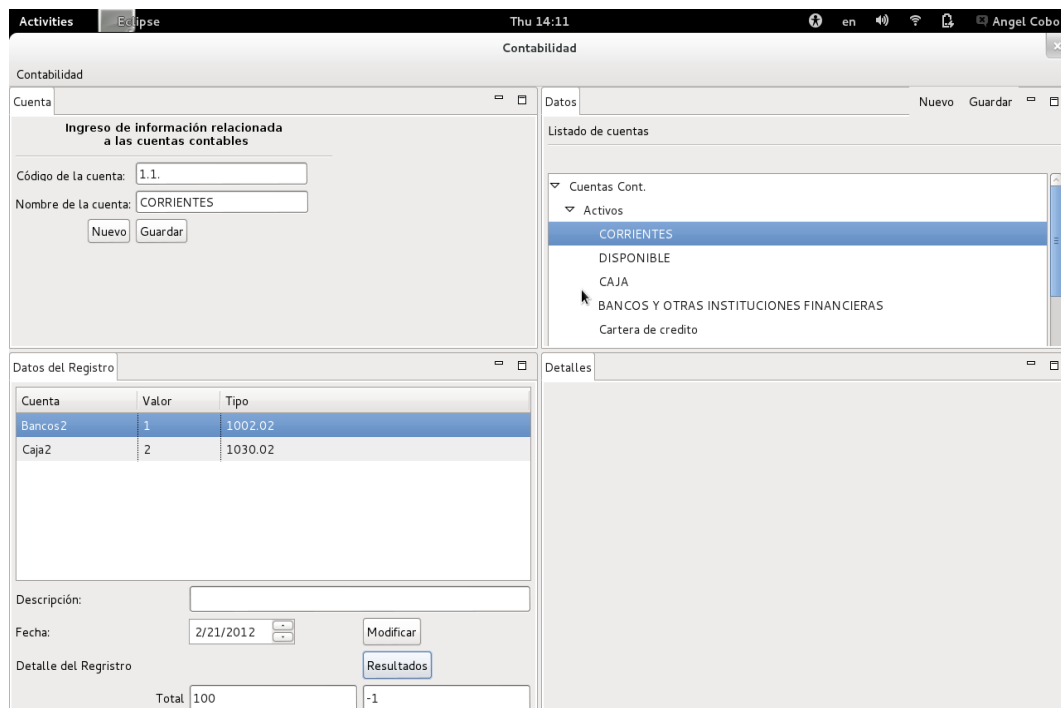


- o *plug-in org.fin.coop.contabilidad.entidades*, contiene las clases de persistencia



5.10 Ejecución de la aplicación

Para la ejecución de la aplicación seleccionamos el archivo *contabilidad.product*, damos click derecho y luego seleccionamos *ejecutar como -> Aplicación Eclipse* y observaremos la siguiente pantalla



ANEXO II

Medición de las métricas de calidad

Evaluación de la Adecuación

Para esta medición se ha tomado los requerimientos del SRS para el módulo de contabilidad y se han evaluado las funcionalidades implementadas por los módulos de prueba. En la siguiente tabla se detallan las funcionalidades evaluadas del módulo de prueba desarrollado en los dos frameworks.

No Requisito	Funcionalidades pedidas en el SRS	Cumplimiento	
		Netbeans RCP	Eclipse RCP
1	Ingresar nueva cuenta	✓	✓
2	Modificar una cuenta	✓	✓
3	Buscar una cuenta filtrado por número, nombre	✓	X
4	Eliminar cuenta	✓	✓
5	Ingresar nuevo asiento contable	✓	✓
6	Modificar un asiento contable	✓	X
7	Eliminar un asiento contable	✓	X
8	Ingresar nuevo registro	✓	✓
9	Modificar un registro	✓	✓
10	Eliminar registro	✓	✓
11	Buscar registro filtrado por código, fecha	✓	✓
12	Visualizar ayuda	✓	✓
13	Generar reportes	✓	✓
Total		11	8

Elaborado por: Los autores

Evaluación de la métrica para calcular la exactitud

Para la evaluación de ésta métrica se han realizado ingresos y visualización de resultados en los campos numéricos de las interfaces de los módulos de prueba. En la siguiente tabla se indican los elementos GUI que fueron evaluados

No Elemento	Descripción	Eclipse RCP	Netbeans RCP
1	Valor Parcial*	✓	✓
2	Valor Total Debe*	✓	✓
3	Valor Total Haber*	✓	✓
4	Valor Total Activos**	✓	X
5	Valor Total Pasivos**	✓	X
6	Valor Total Patrimonio**	✓	X
7	Valor Total Ingresos**	✓	X
8	Valor Total Egresos**	✓	X
9	Resultados del ejercicio**	✓	X
Total		9	3

Elaborado por: Los autores

- ✓ * Campos numéricos de ingreso
- ✓ ** Campos numéricos de salida reportes

Evaluación de la Madurez

Para ésta medición se han hecho ingresos, modificaciones, búsquedas, eliminaciones y generación de reportes. En la siguiente tabla se indican los casos de prueba utilizados para calcular la Madurez del módulo de prueba Netbeans RCP.

No	Descripción	No Pruebas confirmadas en la revisión	No Pruebas Requeridos
1	Ingresar nueva cuenta	2	2
2	Modificar una cuenta	2	2
3	Buscar una cuenta filtrado por número, nombre	2	2
4	Eliminar cuenta	2	2
5	Ingresar nuevo asiento contable	2	2
6	Modificar un asiento contable	2	2
7	Eliminar un asiento contable	2	2
8	Ingresar nuevo registro	2	2
9	Modificar un registro	2	2
10	Eliminar registro	2	2
11	Buscar registro filtrado por código, fecha	1	1
12	Visualizar ayuda	1	1
13	Generar reportes	1	1
Total		23	23

Elaborado por: Los autores

En la siguiente tabla se indican los casos de prueba utilizadas para calcular la Madurez del Módulo de prueba Eclipse RCP

No	Descripción	No Pruebas confirmadas en la revisión	No Pruebas Requeridos
1	Ingresar nueva cuenta	1	2
2	Modificar una cuenta	1	2
3	Buscar una cuenta filtrado por número, nombre	1	2
4	Eliminar cuenta	1	2
5	Ingresar nuevo asiento contable	1	2
6	Modificar un asiento contable	0	2
7	Eliminar un asiento contable	0	2
8	Ingresar nuevo registro	0	2
9	Modificar un registro	0	2
10	Eliminar registro	0	2
11	Buscar registro filtrado por código, fecha	1	1
12	Visualizar ayuda	1	1
13	Generar reportes	1	2
Total		8	23

Elaborado por: Los autores

Evaluación de la tolerancia a fallos

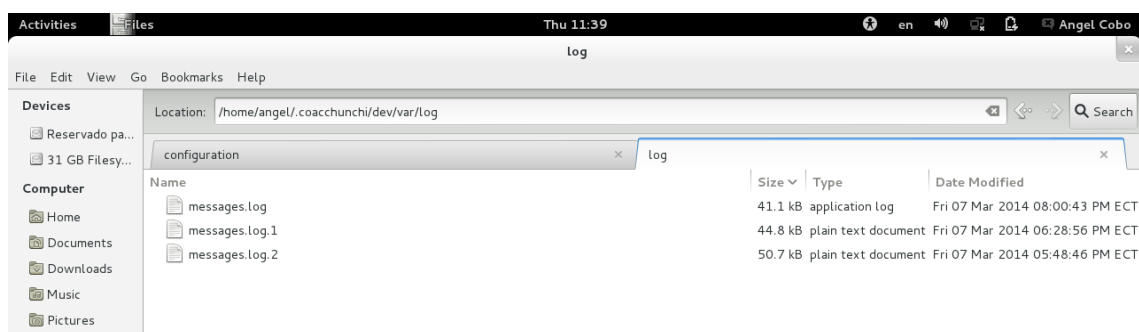
En esta evaluación se ha tomado como métrica las no fallas durante un periodo de tiempo de dos semanas. La siguiente tabla indica las posibles fallas que se pueden dar dentro de una aplicación y cuáles de los dos módulos desarrollados no las han presentado.

No	Descripción	Eclipse RCP	Netbeans RCP
1	Fallas en la visualización de la interfaz de usuario	✓	X
2	Falla en la conexión a la base de datos	✓	X
3	Falla en la persistencia de la base de datos	✓	X
4	Falla en la generación de los reportes	X	X
Total		1	4

Elaborado por: Los autores

- X No tiene falla
- ✓ Tiene falla

El registro de fallos y errores del módulo de prueba hecho con Netbeans RCP se encuentra en el siguiente directorio: directorio de instalación/dev/var/log/ en donde podremos observar los siguientes archivos



El contenido del archivo **messages.log** se detalla a continuación:

```

-----
-----
>Log Session: Friday, decemb 7, 2013 6:36:44 PM ECT
>System Info:
  Product Version      = coacChunchi 201311011758
  Operating System     = Linux version 3.2.0-4-amd64 running on
amd64
  Java; VM; Vendor     = 1.7.0_51; Java HotSpot(TM) 64-Bit Server
VM 24.51-b03; Oracle Corporation
  Runtime              = Java(TM) SE Runtime Environment 1.7.0_51-
b13
  Java Home            =
/home/angel/Documents/descargas/jdk1.7.0_51/jre
  System Locale; Encoding = en_US (coacchunchi); UTF-8

```

```
Home Directory          = /home/angel
Current Directory       =
/home/angel/Documents/coacchunchi/app/coacchunchi/bin
User Directory         = /home/angel/.coacchunchi/dev
Cache Directory        = /home/angel/.coacchunchi/dev/var/cache
Installation           =
/home/angel/Documents/coacchunchi/app/coacchunchi/coacchunchi

/home/angel/Documents/coacchunchi/app/coacchunchi/platform
Boot & Ext. Classpath  =
INFO [org.netbeans.core.startup.NbEvents]: Turning on modules:
org.openide.util.lookup [8.22.1 201311011758]
org.openide.util [8.33.2 201312091611]
org.openide.modules [7.39.1 201311011758]
org.refich.librerias [1.0 140307]
org.openide.filesystems [8.8.1 201311011758]
org.netbeans.api.annotations.common/1 [1.21.1 201311011758]
org.openide.awt [7.59.1 201311011758]
org.netbeans.api.progress/1 [1.35.1 201311011758]
org.openide.dialogs [7.32.1 201311011758]
org.openide.nodes [7.36.1 201311011758]
org.openide.windows [6.65.1 201311011758]
org.netbeans.modules.editor.mimelookup/1 [1.33.1 201311011758]
org.openide.text [6.58.2 201312091611]
org.netbeans.swing.tabcontrol [1.46.1 201311011758]
org.netbeans.swing.outline [1.27.1 201311011758]
org.openide.explorer [6.54.1 201312091611]
org.openide.actions [6.32.1 201311011758]
org.netbeans.modules.queries/1 [1.36.1 201311011758]
org.openide.loaders [7.51.1 201311011758]
org.openide.io [1.42.1 201311011758]
org.netbeans.swing.plaf [1.34.1 201311011758]
org.netbeans.spi.quicksearch [1.20.1 201311011758]
org.netbeans.bootstrap/1 [2.63.1 201311011758]
org.netbeans.core.startup/1 [1.51.1 201311011758]
org.netbeans.modules.settings/1 [1.42.1 201311011758]
org.netbeans.modules.sampler [1.7.1 201311011758]
org.netbeans.modules.progress.ui [1.26.1 201311011758]
org.netbeans.modules.print [7.19.1 201311011758]
org.netbeans.modules.keyring [1.17.1 201311011758]
org.netbeans.core/2 [3.43.1 201311011758]
org.netbeans.modules.options.api/1 [1.36.1 201311011758]
org.netbeans.modules.options.keymap [1.28.2 201312091611]
org.netbeans.modules.masterfs/2 [2.44.3 201312091611]
org.netbeans.libs.jna/1 [1.31.1 201311011758]
org.netbeans.modules.masterfs.linux [1.7.1 201311011758]
org.netbeans.modules.keyring.fallback [1.1.1 201311011758]
org.netbeans.modules.keyring.impl [1.14.1 201311011758]
org.netbeans.modules.editor.mimelookup.impl/1 [1.24.1
201311011758]
org.netbeans.libs.osgi [1.15.1 201311011758]
org.netbeans.libs.jna.platform/1 [1.1.1 201311011758]
org.netbeans.libs.felix [2.9.1 201311011758]
org.netbeans.core.windows/2 [2.66.1 201311011758]
org.netbeans.core.ui/1 [1.38.1 201311011758]
```

```
org.netbeans.core.output2/1 [1.34.3 201312091611]
org.netbeans.core.network [1.1.2 201312091611]
org.netbeans.core.netigso [1.23.1 201311011758]
org.netbeans.core.nativeaccess/1 [1.23.1 201311011758]
org.netbeans.core.multitabs [1.3.3.1 1 201311011758]
org.netbeans.core.io.ui/1 [1.23.1 201311011758]
org.chunchi.postgres [1.0 140307]
org.chunchi.eclipselink [1.0 140307]
org.chunchi.entidades [1.0 140307]
org.chunchi.accesoDatos [1.0 140307]
org.chunchi.reportes [1.0 140307]
org.chunchi.imagenes [1.0 140307]
org.chunchi.contabilidad [1.0 140307]
INFO [org.netbeans.core.network.proxy.NetworkProxyReloader]:
System network proxy resolver: Gnome
INFO
[org.netbeans.core.network.proxy.gnome.GsettingsNetworkProxy]:
GSettings system proxy resolver: direct connection
INFO [org.netbeans.core.network.proxy.NetworkProxyReloader]:
System network proxy reloading succeeded.
INFO [org.netbeans.core.network.proxy.NetworkProxyReloader]:
System network proxy - mode: direct
INFO [org.netbeans.core.network.proxy.NetworkProxyReloader]:
System network proxy: failed to default (corect if direct mode
went before)
INFO [org.netbeans.ui.metrics.laf]: USG_LOOK_AND_FEEL
Diagnostic information
Input arguments:
-Djdk.home=/home/angel/Documents/descargas/jdk1.7.0_51
-Dnetbeans.running.environment=gnome
-
Dnetbeans.dirs=/home/angel/Documents/coacchunchi/app/coacchunchi
/coacchunchi:./../
:
-
Dnetbeans.home=/home/angel/Documents/coacchunchi/app/coacchunchi
/platform
-Xms24m
-Xmx64m
-XX:+HeapDumpOnOutOfMemoryError
-
XX:HeapDumpPath=/home/angel/.coacchunchi/dev/var/log/heapdump.hp
rof
Compiler: HotSpot 64-Bit Tiered Compilers
Heap memory usage: initial 24.0MB maximum 57.0MB
Non heap memory usage: initial 23.4MB maximum 130.0MB
Garbage collector: PS Scavenge (Collections=12 Total time
spent=0s)
Garbage collector: PS MarkSweep (Collections=0 Total time
spent=0s)
Classes: loaded=4340 total loaded=4340 unloaded 0
INFO [org.netbeans.core.ui.warmup.DiagnosticTask]: Total memory
4,091,637,760
```

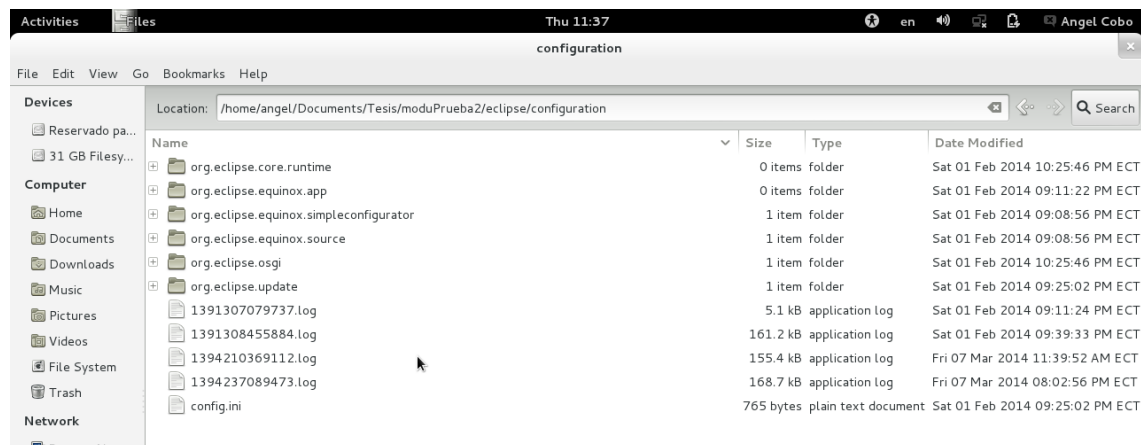


```

INFO [null]: Total physical memory 4,091,637,760
WARNING [org.openide.filesystems.Ordering]: Not all children in
Menu/Window/ marked with the position attribute: [org-chunchi-
contabilidad-AdministrarCuentasTopComponent.shadow, org-chunchi-
contabilidad-AdministrarRegistroCuentaTopComponent.shadow, org-
chunchi-contabilidad-AdministrarRegistroTopComponent.shadow,
org-chunchi-contabilidad-
ContabilidadExplorerTopComponent.shadow, org-chunchi-reportes-
ReportesTopComponent.shadow], but some are: [org-netbeans-core-
io-ui-IOWindowAction.shadow,
SwitchToRecentDocumentAction.shadow, Web, Tools,
Separator3.instance, ConfigureWindow, org-netbeans-core-windows-
actions-ResetWindowsAction.shadow, Separator4.instance,
CloseWindowAction.shadow, CloseAllDocumentsAction.shadow,
CloseAllButThisAction.shadow, DocumentsAction.shadow]
WARNING [org.netbeans.TopSecurityManager]: use of system
property netbeans.home has been obsoleted in favor of
InstalledFileLocator/Places at
org.netbeans.Clusters.relativeDirsWithHome (Clusters.java:137)
...

```

El registro de fallos y errores del módulo de prueba hecho con Eclipse RCP se encuentra en el siguiente directorio: `directorio de instalación/clipse/configuration/` en donde podremos observar los siguientes archivos



El contenido del archivo **log** se detalla a continuación:

```

!SESSION 2014-02-01 21:34:15.141 -----
-----
eclipse.buildId=unknown
java.version=1.7.0_51
java.vendor=Oracle Corporation
BootLoader constants: OS=linux, ARCH=x86_64, WS=gtk, NL=en_US

```

Framework arguments: -clearPersistedState
Command-line arguments: -os linux -ws gtk -arch x86_64 -
clearPersistedState

```
!ENTRY org.eclipse.e4.ui.workbench 4 0 2014-02-01 21:34:29.071
!MESSAGE Unable to create class
'org.fin.coop.contabilidad.parts.UICuenta' from bundle '68'
!STACK 0
org.eclipse.e4.core.di.InjectionException:
java.lang.StackOverflowError
  at
org.eclipse.e4.core.internal.di.MethodRequestor.execute(MethodRe
questor.java:63)
  at
org.eclipse.e4.core.internal.di.InjectorImpl.processAnnotated(In
jectorImpl.java:861)
  at
org.eclipse.e4.core.internal.di.InjectorImpl.inject(InjectorImpl
.java:113)
  at
org.eclipse.e4.core.internal.di.InjectorImpl.internalMake(Inject
orImpl.java:321)
  at
org.eclipse.e4.core.internal.di.InjectorImpl.make(InjectorImpl.j
ava:242)
  at
org.eclipse.e4.core.contexts.ContextInjectionFactory.make(Context
InjectionFactory.java:161)
  at
org.eclipse.e4.ui.internal.workbench.ReflectionContributionFacto
ry.createFromBundle(ReflectionContributionFactory.java:102)
  at
org.eclipse.e4.ui.internal.workbench.ReflectionContributionFacto
ry.doCreate(ReflectionContributionFactory.java:71)
  at
org.eclipse.e4.ui.internal.workbench.ReflectionContributionFacto
ry.create(ReflectionContributionFactory.java:53)
  at
org.eclipse.e4.ui.workbench.renderers.swt.ContributedPartRender
er.createWidget(ContributedPartRenderer.java:141)
  at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.cre
ateWidget(PartRenderingEngine.java:894)
  at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.saf
eCreateGui(PartRenderingEngine.java:629)
  at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.saf
eCreateGui(PartRenderingEngine.java:731)
  at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.acc
ess$2(PartRenderingEngine.java:702)
```

```
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine$7.run(PartRenderingEngine.java:696)
  at org.eclipse.core.runtime.SafeRunner.run(SafeRunner.java:42)
  at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.createGui(PartRenderingEngine.java:681)
  at
org.eclipse.e4.ui.workbench.renderers.swt.StackRenderer.showTab(StackRenderer.java:1115)
  at
org.eclipse.e4.ui.workbench.renderers.swt.LazyStackRenderer$1.handleEvent(LazyStackRenderer.java:67)
  at
org.eclipse.e4.ui.services.internal.events.UIEventHandler$1.run(UIEventHandler.java:41)
  at
org.eclipse.swt.widgets.Synchronizer.syncExec(Synchronizer.java:180)
  at org.eclipse.swt.widgets.Display.syncExec(Display.java:4299)
  at
org.eclipse.e4.ui.internal.workbench.swt.E4Application$1.syncExec(E4Application.java:187)
  at
org.eclipse.e4.ui.services.internal.events.UIEventHandler.handleEvent(UIEventHandler.java:38)
  at
org.eclipse.equinox.internal.event.EventHandlerWrapper.handleEvent(EventHandlerWrapper.java:197)
  at
org.eclipse.equinox.internal.event.EventHandlerTracker.dispatchEvent(EventHandlerTracker.java:197)
  at
org.eclipse.equinox.internal.event.EventHandlerTracker.dispatchEvent(EventHandlerTracker.java:1)
  at
org.eclipse.osgi.framework.eventmgr.EventManager.dispatchEvent(EventManager.java:230)
  at
org.eclipse.osgi.framework.eventmgr.ListenerQueue.dispatchEventSynchronous(ListenerQueue.java:148)
  at
org.eclipse.equinox.internal.event.EventAdminImpl.dispatchEvent(EventAdminImpl.java:135)
  at
org.eclipse.equinox.internal.event.EventAdminImpl.sendEvent(EventAdminImpl.java:78)
  at
org.eclipse.equinox.internal.event.EventComponent.sendEvent(EventComponent.java:39)
  at
org.eclipse.e4.ui.services.internal.events.EventBroker.send(EventBroker.java:80)
```

```
at
org.eclipse.e4.ui.internal.workbench.UIEventPublisher.notifyChan
ged(UIEventPublisher.java:58)
at
org.eclipse.emf.common.notify.impl.BasicNotifierImpl.eNotify(Bas
icNotifierImpl.java:374)
at
org.eclipse.e4.ui.model.application.ui.impl.ElementContainerImpl
.setSelectedElement(ElementContainerImpl.java:171)
at
org.eclipse.e4.ui.workbench.renderers.swt.LazyStackRenderer.post
Process(LazyStackRenderer.java:103)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.saf
eCreateGui(PartRenderingEngine.java:645)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.saf
eCreateGui(PartRenderingEngine.java:731)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.acc
ess$2(PartRenderingEngine.java:702)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine$7.r
un(PartRenderingEngine.java:696)
at org.eclipse.core.runtime.SafeRunner.run(SafeRunner.java:42)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.cre
ateGui(PartRenderingEngine.java:681)
at
org.eclipse.e4.ui.workbench.renderers.swt.SWTPartRenderer.proces
sContents(SWTPartRenderer.java:59)
at
org.eclipse.e4.ui.internal.workbench.swt.PartRenderingEngine.saf
eCreateGui(PartRenderingEngine.java:641)
at
```

Evaluación de la recuperabilidad

Para esta evaluación se ha tomado como métrica el tiempo total en minutos de la caída del sistema hasta que regrese a las condiciones normales de funcionamiento del módulo de prueba y el número de veces que se produjo la caída del mismo.

En la siguiente tabla se indica las mediciones realizadas.

No	Descripción	Eclipse RCP		Netbeans RCP	
		T(min)	N	T(min)	N
1	Fallas en la visualización de la interfaz de usuario	10	2	0	0
2	Falla en la conexión a la base de datos	0	0	0	0
3	Falla en la persistencia de la base de datos	30	2	0	0
4	Falla en la generación de los reportes	45	3	0	0
Total		85	7	0	0

Elaborado por: Los autores

Donde:

T: Tiempo total de falla

N: Número de fallas

Evaluación de la comprensibilidad

En la evaluación de ésta subcaracterística se ha tomado las funcionalidades implementadas en el módulo de prueba y cuáles de éstas son o no entendidas por el usuario en un escenario de prueba. En la siguiente tabla se indica las funcionalidades evaluadas de la interfaz de usuario que fueron entendidas

No Requisito	Descripción	Cumplimiento	
		Netbeans RCP	Eclipse RCP
1	Ingresar nueva cuenta	✓	✓
2	Modificar una cuenta	✓	✓
3	Buscar una cuenta filtrado por número, nombre	✓	X
4	Eliminar cuenta	✓	✓
5	Ingresar nuevo asiento contable	X	✓
6	Modificar un asiento contable	✓	X

7	Eliminar un asiento contable	✓	X
8	Ingresar nuevo registro	X	✓
9	Modificar un registro	✓	✓
10	Eliminar registro	✓	✓
11	Buscar registro filtrado por código, fecha	✓	✓
12	Visualizar ayuda	X	X
13	Generar reportes	X	X
Total		9	8

Elaborado por: Los autores

- ✓ Función de la interfaz de usuario entendida
- X Función de la interfaz de usuario no entendida

Evaluación de la operabilidad

Para la evaluación de ésta subcaracterística se ha usado elementos de ingreso de datos que realizan chequeo de datos o validaciones. En la siguiente tabla se indica los elementos evaluados para cada módulo.

No	Descripción	Cumplimiento	
		Netbeans RCP	Eclipse RCP
1	Validación del campo numérico debe	✓	X
2	Validación del campo numérico haber	✓	X
3	Validación del campo descripción	X	X
4	Validación del campo código cuenta	✓	X
5	Validación del campo descripción	✓	X
6	Validación del campo fecha registro	X	X
7	Validación del campo fecha de corte balance general	✓	X
Total		6	6

Elaborado por: Los autores

- ✓ Función de la interfaz de usuario entendida
- X Función de la interfaz de usuario no entendida

Evaluación de rendimiento

Software usado

En esta evaluación se usó la herramienta Jconsole (Consola de administración y monitoreo de Java), esta herramienta se encuentra en el directorio de instalación del JDK y como métrica el número de tareas ejecutadas.

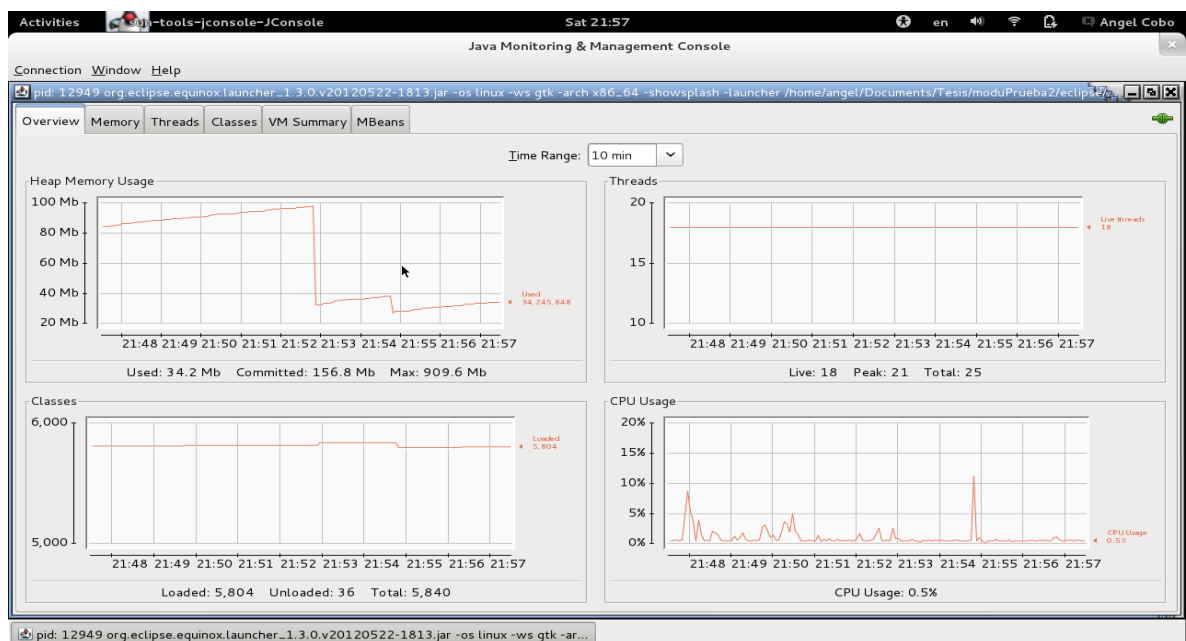
Hardware usado

Para ésta evaluación se usó un computador portátil HP, procesador AMD TurionX2 64 bits, memoria RAM 4GB, sistema operativo GNU/Linux Distribución Debian 7.0

Periodo de tiempo

El periodo de tiempo usado para la medición fue de una hora.

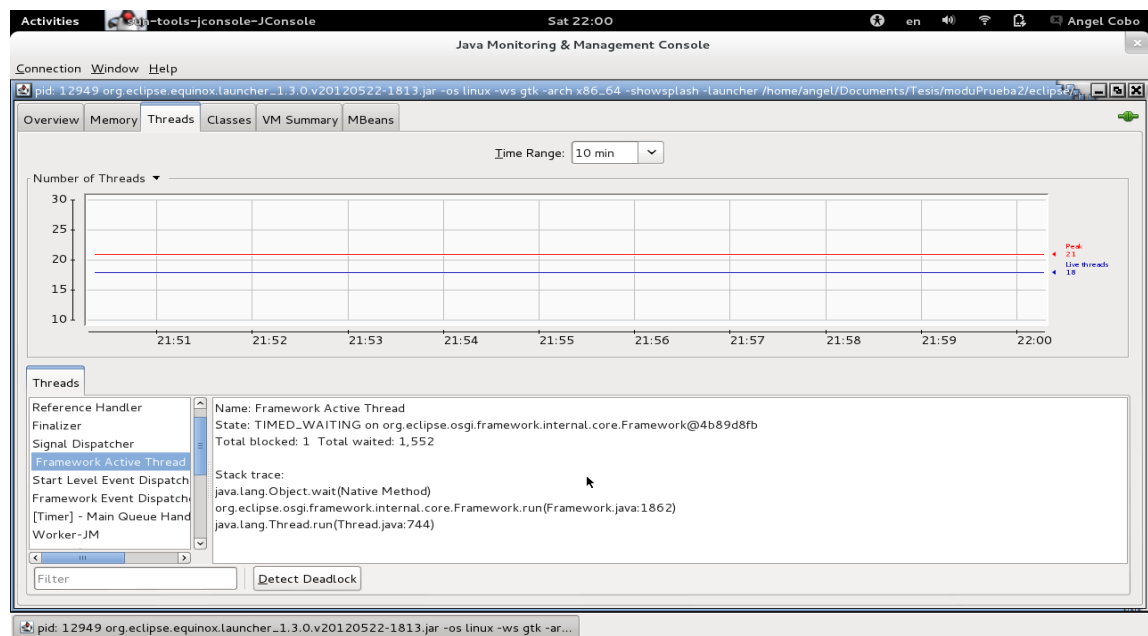
En la siguiente figura se indica un resumen de los recursos usados por el módulo de prueba en Eclipse RCP y generado por JConsole.



Fuente: Los autores

Métrica tareas ejecutadas (Módulo Eclipse RCP)

Se observó 18 tareas ejecutándose continuamente como se indica en la figura.

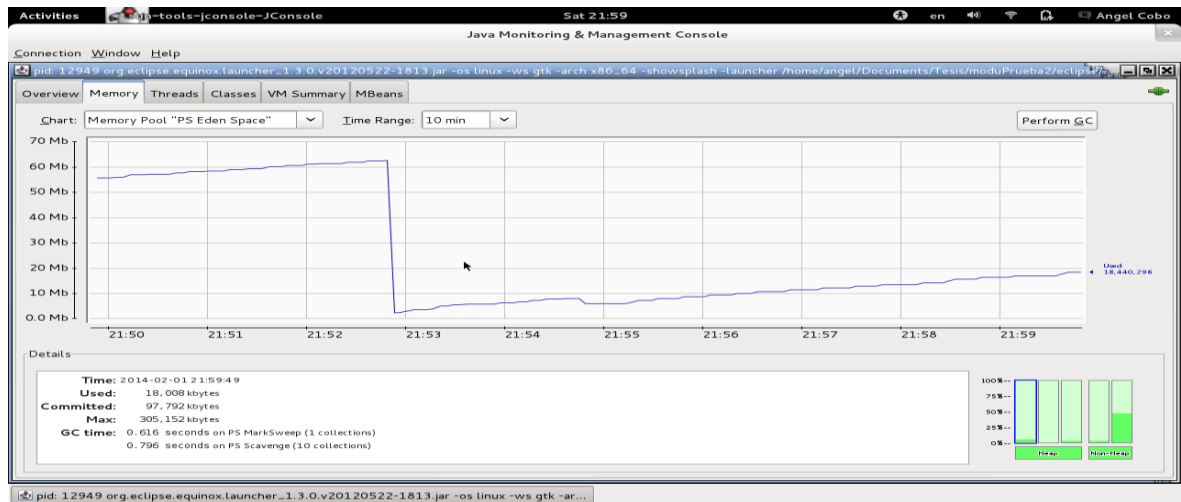


Fuente: Los autores

Evaluación utilización de recursos

Métrica uso de memoria (Módulo Eclipse RCP)

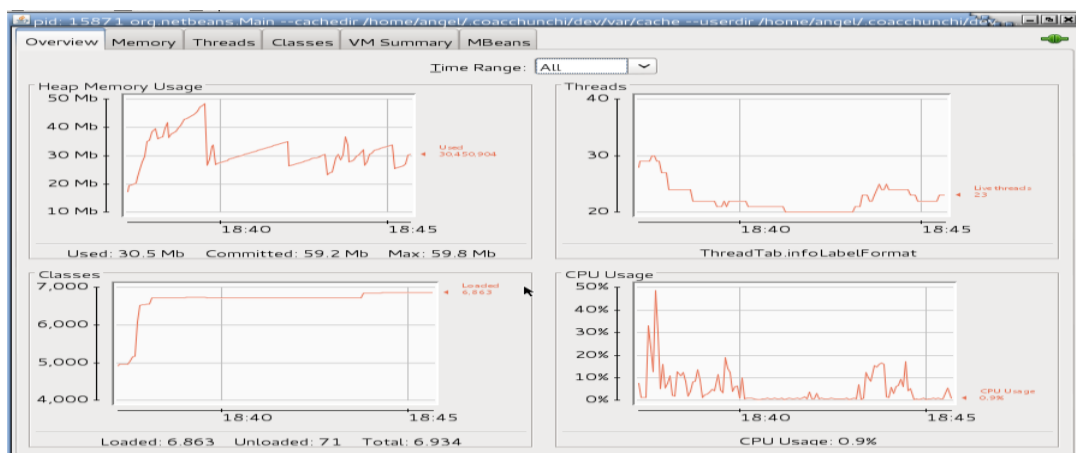
Se observó 18 Mb en promedio de memoria usada continuamente durante un periodo de una hora de funcionamiento como se muestra en siguiente la figura.



Fuente: Los autores

Evaluación de Rendimiento módulo de prueba Netbeans RCP

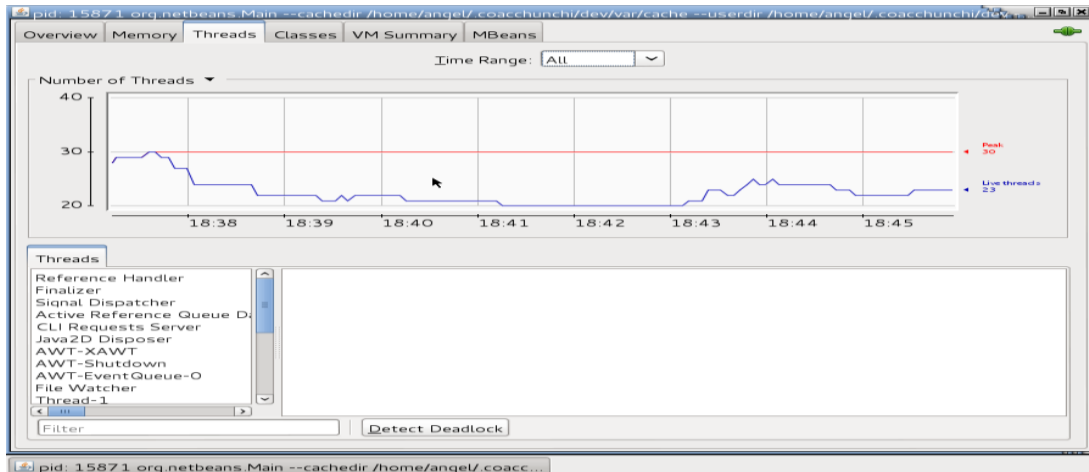
En esta evaluación se usó la herramienta Jconsole, esta herramienta se encuentra en el directorio de instalación del JDK. En la siguiente figura se indica un resumen de los recursos usados por el módulo de prueba desarrollado usando Netbeans RCP.



Fuente: Los autores

Métrica Tareas ejecutadas (Netbeans RCP)

Se observó 23 tareas ejecutándose continuamente como se muestra en la siguiente figura.

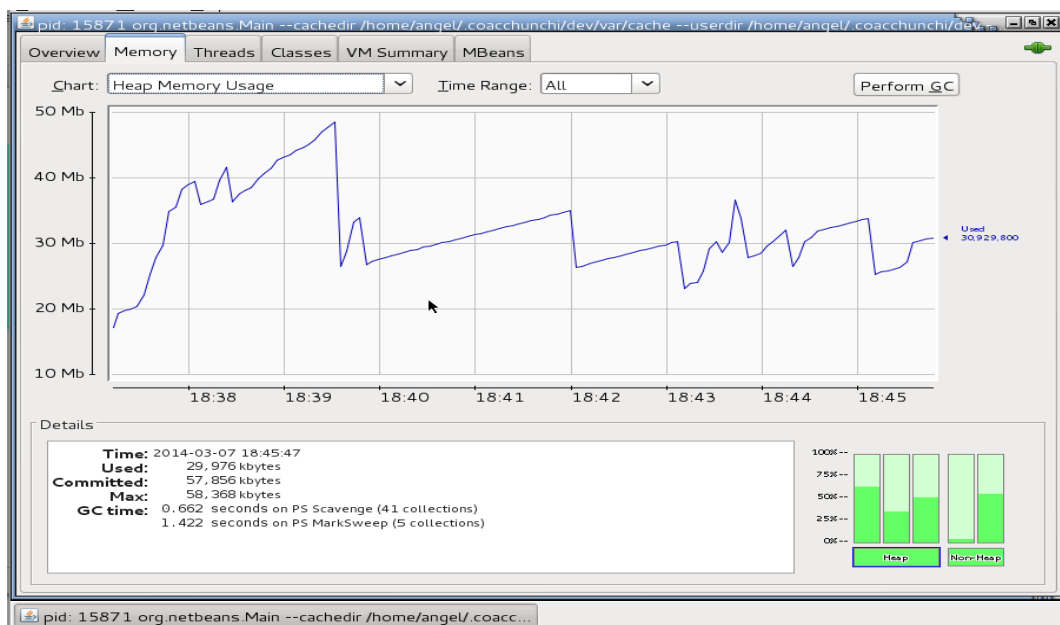


Fuente: Los autores

Evaluación utilización de recursos

Métrica uso de memoria (Módulo Netbeans RCP)

Se observó 30.92 Mb de memoria usada continuamente como se muestra en la siguiente figura.



Fuente: Los autores

ANEXO III

Especificación Funcional de Requerimientos

1. INTRODUCCIÓN

La presente Especificación de requerimientos de software (SRS) del sistema a construir surge para ser un conjunto de información necesaria que ayuda a los desarrolladores del software a analizar y entender todos los requisitos y requerimientos que nuestro cliente desea, de la misma forma como este constituye un informe útil para que el cliente del producto final describa lo que el realmente desea obtener, y de esta manera lograr tener un documento necesario cuya información en el futuro servirá para el desarrollo del software, es decir en la codificación correcta del mismo.

Se describirá en forma detallada las interface de usuario, de software, del hardware y comunicaciones, así como de los requerimientos del cliente, atributos del sistema entre otros.

1.1. Propósito

- Permitir establecer las bases de acuerdo entre usuarios en lo que al proyecto de software se refiere.
- Ayudar a los usuarios finales del software a entender exactamente que es lo que el cliente de software desea.

1.2. Alcance

Identificación del producto de software
"SISCOOP"

1.3. Objetivos del Sistema

- Permitir la gestión de usuarios, clientes y socios
- Apertura de libretas de ahorro
- Realizar créditos y depósitos ahorros
- Emitir informes cuando los usuarios o directivos lo necesiten (diarios, semanales, quincenales, etc.)

1.4. Personal involucrado

En las siguientes tablas se indican el personal involucrado en el desarrollo del sistema.

Nombre	José Manzano
Rol	Programador
Categoría Profesional	Analista
Responsabilidad Es	Programar los módulos del sistema
Información de Contacto	manzanoj@yao.es
Aprobación	

Nombre	Ángel Cobo
Rol	Programador
Categoría Profesional	Analista
Responsabilidad Es	Diseño de la arquitectura del sistema
Información de Contacto	coboangel210@gmail.com
Aprobación	

1.5. Definiciones, acrónimos y abreviaturas

DEFINICIONES

Actualización.- Insertar, eliminar, modificar los registros de los usuarios (Profesores).

Almacenamiento.- En relación con ordenadores o computadoras, cualquier dispositivo capaz de almacenar información procedente de un sistema informático.

Backup.- Las copias de seguridad en un sistema informático tienen por objetivo el mantener cierta capacidad de recuperación de la información ante posibles pérdidas. Esta capacidad puede llegar a ser algo muy importante, incluso crítico, para las empresas. Se han dado casos de empresas que han llegado a desaparecer ante la imposibilidad de recuperar sus sistemas al estado anterior a que se produjese un incidente de seguridad grave.

Base de Datos.- Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo.

Botón.- Es un objeto tangible que realiza un evento tras su activación.

Conexión.- Comunicación entre dos entes que tienen características similares de comunicación.

Interfaz.- Medio que permite la comunicación entre el usuario y el sistema.

Login.- Nombre o alias que se le da a una persona para permitirle el acceso al sistema siempre y cuando estén registrados.

PASSWORD.- Contraseña o clave para autenticar el ingreso a un lugar o sitio.

Periférico.- En informática, término utilizado para dispositivos , como unidades de disco, impresoras, módem que están conectados a un ordenador o computadora y son controlados por su microprocesador.

Protocolo.- Señal mediante la cual se reconoce que puede tener lugar la comunicación o la transferencia de información.

Servidor.- Computadora conectada a una red que pone sus recursos a disposición del resto de los integrantes de la red. Suele utilizarse para mantener datos centralizados o para gestionar recursos compartidos.

Servidor de archivos.- Dispositivo de almacenamiento de archivos en una red de área local, o en Internet, al que los distintos usuarios de la red pueden acceder, en función de los privilegios que les hayan sido dados por parte del administrador.

Sistema Operativo.- Software básico que controla una computadora. El sistema operativo tiene tres grandes funciones: coordina y manipula el hardware del ordenador o computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el mouse; organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas, y gestiona los errores de hardware y la pérdida de datos.

SQL- Lenguaje de consulta estructurado, en informática, un lenguaje utilizado en bases de datos para consultar, actualizar y manejar bases de datos relacionales.

PostgreSQL.- DBMS relacional de licencia GNU GPL para la implementación de la base de datos del sistema propuesto

Tabla.- Entidad que posee campos físicos primarios secundarios

ACRÓNIMOS

DBA.- Es aquella persona que tiene el control central del sistema de base de datos.

DBMS.- Sistema de Administración de base de datos, El software que permite la creación, administración e implementación de la base de datos.

GUI o acrónimo de Graphical User Interface.- En informática, tipo de entorno que permite al usuario elegir comandos, iniciar programas, ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón.

SRS.- Acrónimo de Software Requirements Specifications (Especificación de Requerimientos de Software)

TCP/IP.- Acrónimo de Transmisión Control Protocol/Internet Protocol (protocolo de control de transmisiones/protocolo de Internet), protocolos usados para el control de la transmisión en Internet. Permite que diferentes tipos de ordenadores o computadoras se comuniquen a través de redes heterogéneas.

URL.- Un URL es una cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

ABREVIATURAS

HW: Hardware

SW: Software

Sr. Señor

Sra. Señora

Ing. Ingeniero(a)

Srs. Señores

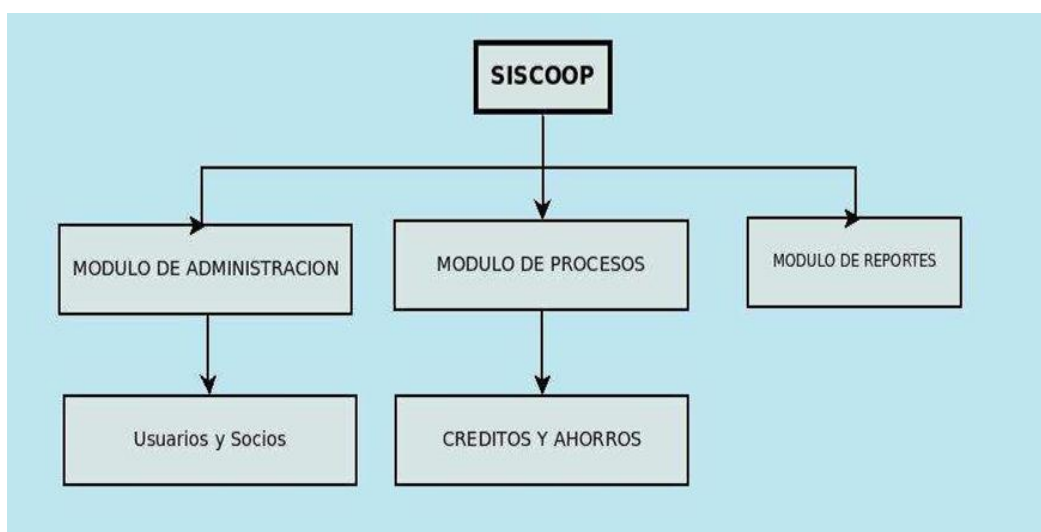
1.6. Resumen

El SRS está compuesto de la siguiente manera

- **Introducción:** En ésta sección se detalla los objetivos que tiene el SRS y de nuestro sistema en forma general.
- **Descripción General:** Describe una perspectiva general del producto a desarrollarse, como también las características del usuario y las limitaciones que podría tener.
- **Requerimientos Específicos:** Muestra paso a paso todos los requerimientos que el usuario desea en el producto final. Para el cual se ha utilizado el “Prototipo 2 del Estándar IEEE 380”.

2. Descripción general

2.1. Perspectiva del producto



El sistema que se va a desarrollar es independiente, y tendrá un diseño modular para gestionar las diferentes áreas dentro de una cooperativa

2.2. Funcionalidad del producto

2.3. Características de los usuarios

Tipo de usuario	Cajera
Formación	Conocimientos básicos de contabilidad
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Realizar operaciones de depósitos y retiros

Tipo de usuario	Asesor de crédito
Formación	Conocimientos sólidos de contabilidad (tercer nivel)
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Realizar los análisis de los clientes para los créditos, liquida Créditos

Tipo de usuario	Gerente
Formación	Conocimientos sólidos de contabilidad (tercer nivel)
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Generar, analizar y tomar decisiones en función de los reportes que genere el sistema

Tipo de usuario	Técnico en sistemas
Formación	Analista en sistemas Informáticos
Habilidades	Conocimientos básicos de bases de datos , hardware de
	computador, redes LAN
Actividades	Mantenimiento de los equipos, revisar y mantener la red

2.4. Restricciones

El sistema será desarrollado en JAVA un lenguaje con una orientación a objetos del 97%, el motor de la base de datos es PostgreSQL 8.4 de tecnología Objeto-Relacional, la metodología para el desarrollo se basará en las mejores características de las metodologías tradicionales (evolutivas) y ágiles como MSF.

2.5. Suposiciones y dependencias

Ninguno.

2.6. Evolución previsible del sistema

Trabajar con base de datos distribuidas, Inteligencia de negocios.

3. Requisitos específicos

Los requisitos se los ha ordenado por módulos debidamente priorizados en conjunto con el personal designado para colaborar con el desarrollo del proyecto.

3.1. Requisitos comunes de los interfaces

3.1.1. Interfaces de usuario

Las interfaces de usuario están relacionadas con las pantallas, ventanas (formularios) que debe manipular el usuario para realizar una operación determinada. Dicha manipulación el usuario la realizará por medio del teclado y el Mouse (ratón).

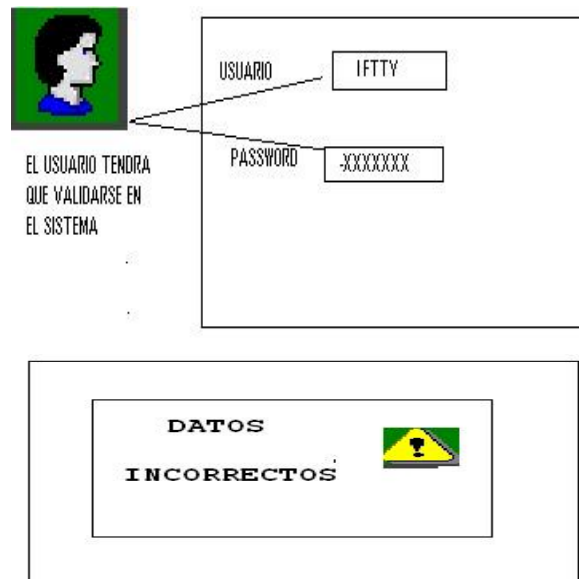
Es importante mencionar que las interfaces de usuario también abarcan las ayudas correspondientes en cada uno de los procesos que realice el sistema.

Las interfaces de usuario ayudarán al usuario final trabajando en un ambiente Form, por lo que se dichas interfaces incluirán:

- Botones
- Menús despegables
- Mensajes informativos
- Mensajes de error
- Cuadros de diálogo
- Formularios para el ingreso, modificación, actualización y eliminación de datos. Así como para las operaciones y las ayudas que se mencionó anteriormente.
- Otros

A continuación se muestra una previa de lo que será las interfaces de usuario.

El usuario previamente debe tener su cuenta de usuario en el sistema para poder acceder.



En caso de que no ingrese correctamente el USUARIO o el PASSWORD se desplegara un mensaje de datos incorrectos. Como a continuación se muestra.

3.1.2. Interfaces de hardware

La pantalla del monitor.- el software deberá mostrar información al usuario a través de la pantalla del monitor.

Ratón.- el software debe interactuar con el movimiento del ratón y los botones del ratón. El ratón se activan las zonas de entrada de datos, botones de comando y seleccione las opciones de los menús.

Teclado.- el software deberán interactuar con las pulsaciones del teclado. El teclado de entrada de datos en el área activa de la base de datos.

Impresora.- el software imprimirá los reportes en la impresora instalada.

3.1.3. Interfaces de software

Ninguno.

3.1.4. Interfaces de comunicación

La interfaz de comunicación entre el servidor de base de datos PostgreSQL y la aplicación desarrollada en JAVA se lo realiza mediante un ORM.

3.2. Requisitos funcionales

3.2.1. Requisito funcional 1

Número de requisito	REQ22
Nombre de requisito	Registro de nuevos socios
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir el registro de un nuevo socio, como son sus datos personales, actividad económica a la que se dedica y dos referencias del lugar donde vive en caso de personas naturales datos del conyuge, en caso de personas jurídicas registrar datos del representante

ENTRADAS

Número de Cédula, Nombres, Apellidos, estado civil, fecha de nacimiento, nivel e instrucción, lugar de residencia y trabajo(dirección, el nombre de un lugar de referencia)

PROCESOS

El sistema verificará si el usuario que intenta realizar esta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos

Registro Socio, el sistema pedirá los correspondientes datos del nuevo socio luego verificará que no haya espacios en blanco, en el caso de ningún error guardara los datos del nuevo socio.

SALIDAS

Las salidas van dirigidas a: Administrador (Gerente), asesor crédito, cajero.

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un numero de cédula ya existente en la base de datos

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.2. Requisito funcional 2

Número de requisito	REQ23		
Nombre de requisito	Recepción de depósitos		
Tipo	X Requisito	Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir el registro de un depósito, mediante el número de socio o cédula el sistema permitirá registrar el depósito en la cuenta respectiva (ahorros, certificados de aportación o inversion) correspondiente al socio

ENTRADAS

Número de Cédula o socio, Nombres, Apellidos, el valor a depositar, la fecha actual, la papeleta de depósito

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción depósitos, el sistema pedirá los correspondientes datos del depósito y luego verificará que la cuenta esté registrada y activa, en el caso de que no exista ningún error registra el depósito

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de estar inactiva la cuenta.

Mensaje de error en el caso de ingresar un número de cédula no existente en la base de datos o número de cédula incorrecto o número de socio incorrecto

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.3. Requisito funcional 3

Número de requisito	REQ24
Nombre de requisito	Registro de retiros
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir el registro de un retiro de dinero, mediante el número de socio o cédula el sistema permitirá registrar el retiro de la cuenta respectiva (ahorros o certificados de aportación) correspondiente al socio

ENTRADAS

Número de Cédula o socio, Nombres, Apellidos, el valor a retirar, la fecha actual, papeleta de retiro

PROCESOS

El sistema verificará si el usuario que intenta realizar esta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción retiros, el sistema pedirá los correspondientes datos del retiro y luego verificará que la cuenta esté registrada, activa y el valor del retiro no supere el saldo de la cuenta, en el caso de que no exista ningún error registra el retiro

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de estar inactiva la cuenta.

Mensaje de error en el caso de que el valor a retirar sea superior al saldo existente en la cuenta.

Mensaje de error en el caso de ingresar un numero de cédula no existente en la base de datos o número de cedula incorrecto o numero de socio incorrecto

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.4. Requisito funcional 4

Número de requisito	REQ25
Nombre de requisito	Apertura de una libreta de ahorros
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Eencial o Opcional

INTRODUCCION

El sistema debe permitir el registro de una apertura de libreta de ahorros, mediante el número de socio o cédula el sistema imprimira la libreta de ahorros con los registros de las transacciones realizadas en la cuenta(depósitos, retiros, pago de intereses, transferencias, notas de débito por cobro de comisiones o multas y crédito) correspondiente al socio.

ENTRADAS

Número de Cédula o socio, Nombres, Apellidos, el rango de registros que desea imprimir de acuerdo a la fecha

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción imprimir libreta, el sistema pedirá que seleccione el rango de transacciones de la cuenta a imprimir en la libreta

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un numero de cédula no existente en la base de datos o número de cedula incorrecto o numero de socio incorrecto

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.5. Requisito funcional 5

Número de requisito	REQ25		
Nombre de requisito	Manejo de inversiones		
Tipo	X Requisito	Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir el registro de una cuenta como inversionista, como son sus datos personales, actividad económica a la que se dedica y dos referencias del lugar donde vive

ENTRADAS

Número de Cédula del inversionista o socio, Nombres, Apellidos, el monto de la inversión el plazo en días y el interés anual

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción inversiones, el sistema pedirá el número de socio o Cédula y los datos personales en caso de no encontrarse registrado, el ingreso del monto de la inversión, el plazo en días e imprimira el certificado de depósito correspondiente a la inversión realizada

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un numero de cédula no existente en la base de datos o número de cedula incorrecto o numero de socio incorrecto

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.6. Requisito funcional 6

Número de requisito	REQ25
Nombre de requisito	Pago de intereses a los ahorros
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir el pago de los intereses a los ahorros mensualmente de acuerdo a la tasa establecida

ENTRADAS

Tasa de interés pasiva anual

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción procesos de fin de mes, y pago de intereses a ahorros

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje con el número de cuentas pagadas los intereses correspondientes

3.2.7. Requisito funcional 7

Número de requisito	REQ25
Nombre de requisito	Pago de intereses a las inversiones
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir el pago de los intereses a las inversiones mensualmente de acuerdo a la tasa establecida

ENTRADAS

Tasa de interés pasiva anual para las inversiones

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción procesos de fin de mes, y pago de intereses a inversiones.

SALIDAS

Las salidas van dirigidas a: asesor crédito, cajero.

Mensaje con el número de inversiones pagadas los intereses correspondientes

3.2.8. Requisito funcional 8

Módulo 2 Reportes de caja

Número de requisito	REQ31
Nombre de requisito	Registrar los ingresos y egresos de dinero de caja e imprimir a fin de día el cuadro de caja
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe registrar los movimientos de dinero sucitados en caja como depositos, retiros, cobro de comisiones, inversiones, distribución de fondos, devolucion de fondos e imprimir el respectivo cuadro de caja con las operaciones antes mencionadas

ENTRADAS

Los depósitos, retiros echos por los socios, inversiones las distribuciones de fondos y las respectivas devoluciones a fin de día

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción procesos de fin de día y escogemos resumen diario de movimientos de caja

SALIDAS

Las salidas van dirigidas a: cajero, asesor crédito, administrador.

Reporte con el resumen diario de movimientos de caja

3.2.9. Requisito funcional 9

Número de requisito	REQ32
Nombre de requisito	Imprimir el estado de cuenta de un socio
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir imprimir el estado de cuenta de un socio mediante el número de socio o cédula de identidad

ENTRADAS

El número de socio o cédula

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Caja del menú principal y escogemos la opción estado de cuenta e ingresamos el número de socio o cédula

SALIDAS

Las salidas van dirigidas a: cajero, asesor crédito, administrador
Reporte con el estado de cuenta del socio

3.2.10. Requisito funcional 10

Módulo 3 Procesos de Crédito

Número de requisito	REQ2
Nombre de requisito	Registro de la solicitud de crédito de un socio
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema

Prioridad del requisito	X Alta/Esencial	Media/Deseado	Baja/Opcional
-------------------------	-----------------	---------------	---------------

INTRODUCCION

El sistema debe permitir registrar los datos adicionales como conyuge, direccion del lugar de trabajo, sitios de referencia, información económica tanto del socio como del conyuge que no son ingresados cuando se apertura la cuenta de ahorros

ENTRADAS

Datos de la solicitud como: fecha de la solicitud, oficial responsable, el número de la solicitud y el número de socio. Datos del crédito como: monto solicitado, recuperación propuesta, número de cuatos, detalle del destino del crédito. Datos del solicitante tales como: nombre, identificación, nivel de instrucción, domicilio, sitios de referencia, edad, estado civil, teléfono, número de cargas familiares. Datos de la ocupación como: Nombre de la empresa, dirección, tiempo en el trabajo, sitios de referencia, sueldo mensual, teléfono. Si tiene vivienda propia, Datos del conyuge como: Nombre, cédula, si es socio el número de socio, ocupación, cargo, empresa donde trabaja, dirección del trabajo, teléfono, tiempo en el trabajo y sueldo mensual

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción solicitud de crédito e ingresamos los datos del socio como: personales, económicos, del conyuge que se mencionan anteriormente

SALIDAS

Las salidas van dirigidas a: asesor crédito, administrador

Reporte con la solicitud de crédito para que pueda firmar el solicitante, su conyuge, el garante y su conyuge.

3.2.11. Requisito funcional 11

Número de requisito	REQ3
Nombre de requisito	Generar la tabla presuntiva de cuotas del crédito
Tipo	X Requisito Restricción

Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir generar la tabla presuntiva para que pueda el oficial de crédito informarle sobre el valor de las cuotas que tiene que cancelar para el monto solicitado en el plazo solicitado al socio

ENTRADAS

El tipo cuotas (fija o variable), el monto del crédito, la tasa de interés, plazo en días, valores adicionales como cuota de ahorro obligatorio si aplica, valor del seguro de desgrabamen

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción tabla presuntiva e ingresamos los datos antes mencionados del crédito solicitado

SALIDAS

Las salidas van dirigidas a: asesor crédito, administrador

Reporte con la tabla presuntiva de pagos del crédito solicitado

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un número de cuenta ya existente o mal ingresada en la base de datos.

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.12. Requisito funcional 12

Número de requisito	REQ4
Nombre de requisito	Registro de garantes
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema

Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional
-------------------------	--------------------	-------------------	-------------------

INTRODUCCION

El sistema debe permitir registrar los datos personales de un garante y su conyuge para el otorgamiento del credito a un socio

ENTRADAS

Datos del garante como: nombre, identificación, nivel de instrucción, domicilio, sitios de referencia, edad, estado civil, teléfono, número de cargas familiares.

Datos de la ocupación como: Nombre de la empresa, dirección, tiempo en el trabajo, sitios de referencia, sueldo mensual, teléfono. Si tiene vivienda propia,

Datos del conyuge como: Nombre, cédula, si es socio el número de socio, ocupación, cargo, empresa donde trabaja, dirección del trabajo, teléfono, tiempo en el trabajo y sueldo mensual

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción registro de garantes e ingresamos los datos del garante como: personales, económicos, del conyuge que se mencionan anteriormente.

SALIDAS

Las salidas van dirigidas a: asesor crédito, administrador

Reporte registro exitoso de garantes

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un número de cédula ya existente

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.2.13. Requisito funcional 13

Número de requisito	REQ5
Nombre de requisito	Liquidación del crédito
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema

Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional
-------------------------	--------------------	-------------------	-------------------

INTRODUCCION

El sistema debe permitir registrar la liquidación de un crédito esto es, imprimir el pagaré y acreditar a la cuenta de ahorros a la vista el valor del crédito solicitado e imprimir el comprobante de liquidación del crédito

ENTRADAS

Cuenta origen de la que provienen los fondos para transferir , el monto y la cuenta de ahorros del socio

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción *liquidación de crédito* se imprime el pagaré e ingresamos la cuenta origen de los fondos, el monto y la cuenta destino del socio. Al momento de la acreditación a la cuenta de ahorros se generan los respectivos asientos contables y el socio puede retirar el dinero de su cuenta de ahorros

SALIDAS

Las salidas van dirigidas a: asesor crédito, administrador
Pagaré para que firme el socio su conyuge y los garantes
Comprobante de liquidación del crédito para que firme el socio

3.2.14. Requisito funcional 14

Número de requisito	REQ6,7,8,9		
Nombre de requisito	Cobro de una cuota del crédito		
Tipo	X Requisito	Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir registrar el cobro de una cuota del crédito su capital, interés, seguro de desgravamen y otros aportes

ENTRADAS

Número del socio o cédula de identidad

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción *cobro de cuota de crédito* e ingresamos el número de socio o su cédula y buscamos el crédito que tiene vigente, el valor de la cuota a pagar, se verifican si tiene multas u otros valores pendientes de pago; se procede a realizar el depósito por ese valor en la cuenta de ahorros y luego se lo debita para el cobro de la cuota. De igual manera se generan los respectivos asientos contables y se imprime el comprobante de pago de cuota del crédito

SALIDAS

Las salidas van dirigidas a: asesor crédito, administrador

Comprobante de pago de cuota del crédito

3.2.15. Requisito funcional 15

Número de requisito	REQ10
Nombre de requisito	Generación de notificaciones por mora en el pago de cuotas
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir generar una notificación de aviso de mora en el crédito concedido al socio

ENTRADAS

Número del socio o cédula de identidad

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción *notificación crédito* e ingresamos el número de socio o su cédula y buscamos el crédito que tiene vigente, el valor de la cuota a pagar, se verifican si tiene multas u otros valores pendientes de pago; se procede a imprimir la notificación de mora para el socio y el garante

SALIDAS

Las salidas van dirigidas al: asesor crédito, administrador

Notificación de mora en el pago de cuota del crédito para que firme el oficial de crédito y el socio que está en mora

3.2.16. Requisito funcional 16

Número de requisito	REQ13, 15, 16, 17, 18, 19
Nombre de requisito	Reclasificación diaria de la cartera
Tipo	X Requisito Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir reclasificar diariamente la cartera de crédito para observar su evolución.

ENTRADAS

Ninguna

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Crédito del menú principal y escogemos la opción *reclasificación de cartera* y se actualiza la clasificación de la cartera con los nuevos pagos realizados durante el día

SALIDAS

Las salidas van dirigidas al: asesor crédito, administrador
Clasificación de la cartera actualizada

3.2.17. Requisito funcional 17

Módulo 5 Proceso de Contabilidad

Número de requisito	REQ34,35
Nombre de requisito	Registro de un asiento contable
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Tabla requerimientos del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Desead <input type="checkbox"/> Baja/ <input type="checkbox"/> Opcional

INTRODUCCION

El sistema debe permitir registrar los asientos contables de las operaciones que genere la cooperativa con su entorno como pago de arriendos, gastos por utiles de oficina, limpieza, seguros, servicios básicos

ENTRADAS

Datos del asiento contable como: nombre de la cuenta, el valor de la transacción el tipo(debe o haber), fecha y una descripción

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Contabilidad del menú principal y escogemos la opción *registros contables* e ingresamos los datos del respectivo asiento contable además permitirá actualizar y eliminar.

SALIDAS

Las salidas van dirigidas a: administrador, contador
Comprobante del registro contable

3.2.18. Requisito funcional 18

Módulo 6 Reportes de Contabilidad

Número de requisito	REQ36		
Nombre de requisito	Emitir los estados financieros		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Desead	<input type="checkbox"/> Baja/ Opcional

INTRODUCCION

El sistema debe permitir emitir los estados financieros correspondientes a un período de tiempo(mensuales, anuales) como son: balance general, estado de resultados, estado de cambio de patrimonio

ENTRADAS

Periodo de tiempo de los estados financieros(mensuales o anuales)

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Contabilidad del menú principal y escogemos la opción *estados financieros* e ingresamos el periodo de tiempo para el que deseamos el estado financiero

SALIDAS

Las salidas van dirigidas a: administrador, contador

Estados financieros para que firme el contador, gerente y presidente del Consejo de Administración y Vigilancia

3.2.19. Requisito funcional 19

Módulo 6 Reportes de Contabilidad

Número de requisito	REQ36		
Nombre de requisito	Imprimir un reporte de diario por cuenta		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Desead	<input type="checkbox"/> Baja/ Opcional

INTRODUCCION

El sistema debe permitir imprimir un reporte de los ingresos y egresos diarios por cuenta

ENTRADAS

Número de la cuenta y el día

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Nos ubicamos en la parte de Contabilidad del menú principal y escogemos la opción *diario por cuenta* e ingresamos el número de cuenta y seleccionamos el día

SALIDAS

Las salidas van dirigidas a: administrador, contador

Informe del movimiento diario de la cuenta

3.2.20. Requisito funcional 20

Módulo 7 Gestion de usuarios

Número de requisito	REQ42, 43		
Nombre de requisito	Permitir la gestión (crear, modificar, eliminar) de usuarios		
Tipo	X Requisito	Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir gestionar es decir crear, modificar y eliminar las cuentas de usuarios

ENTRADAS

Cédula, Nombres, Apellidos, Tipo de Usuario (Contador, Cajero, Gerente),

Dirección, Teléfono.

PROCESOS

El sistema verificará si el usuario que intenta realizar ésta operación tiene el permiso para realizar esto. Para cumplir con este requerimiento se le presentara una sola pantalla donde el sistema pedirá la correspondiente identificación como administrador. Nos ubicamos en la parte de Administrar del menú principal y

escogemos Usuario El sistema pedirá los correspondientes datos del nuevo usuario, cliente o socio luego verificara que no haya espacios en blanco, en el caso de ningún error guardara los datos del nuevo usuario. En este mismo formulario podrá crear, modificar y guardar.

SALIDAS

Las salidas van dirigidas a: Administrador (Gerente).

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un numero de cedula ya existente o mal ingresada en la base de datos.

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

Mensaje de error al eliminar una cuenta, ya que no se permite eliminaciones en cascada.

3.2.21. Requisito funcional 21

Número de requisito	REQ45, 456 47, 48		
Nombre de requisito	Auditar las acciones de los usuarios en el sistema		
Tipo	X Requisito	Restricción	
Fuente del requisito	Tabla requerimientos del sistema		
Prioridad del requisito	X Alta/Esencial	Media/Desead o	Baja/ Opcional

INTRODUCCION

El sistema debe permitir registrar todas las acciones que realizan los usuarios en el sistema, como por ejemplo que módulos utilizan y que hacen con ellos, hora de entrada y salida

ENTRADAS

Acciones que realizan los usuarios en los distintos módulos del sistema

PROCESOS

El sistema llevará un registro automático en la base de datos de las distintas acciones que realizan los usuarios en el sistema y presentara un reporte de las acciones hechas por usuario, con fecha

SALIDAS

Las salidas van dirigidas a: Administrador (Gerente).

3.2.22. Requisito funcional 22

Número de requisito	REQ44
Nombre de requisito	Permitir la autenticación de los usuarios.
Tipo	X Requisito Restricción
Fuente del requisito	Tabla de requerimientos del sistema
Prioridad del requisito	X Media/Desead Baja/ Alta/Esencial o Opcional

INTRODUCCION

El sistema debe permitir el ingreso del nombre y password del usuario para realizar las diferentes funciones que tendrá cada uno.

ENTRADAS

Cedula, Nombres, Apellidos, Contraseña, Tipo de Usuario (Técnico, Cajero, Gerente).

PROCESOS

El sistema pedirá la correspondiente identificación como administrador. Nos ubicamos en la parte de Administrar del menú principal y escogemos Usuario El sistema pedirá los correspondientes datos del nuevo usuario luego verificara que no haya espacios en blanco, en el caso de ningún error guardara los datos del nuevo usuario.

SALIDAS

Las salidas van dirigidas a: Administrador (Gerente).

Mensaje de error en el caso de no haber llenado algún campo.

Mensaje de error en el caso de ingresar un numero de cédula ya existente en la base de datos

Mensaje de error en casos de ingresar incorrectamente los datos es decir que el formato de los datos sea incorrecto.

3.3. Requisitos no funcionales

3.3.1. Requisitos de rendimiento

La infraestructura de red, así como sus terminales deben cumplir con normas según la IEEE en la forma de conexión a los equipos, para tener tiempos de respuesta mínimos.

Numero de terminales a manejar:

Se contará con un servidor de base de datos en la matriz de la cooperativa.

Número de usuarios simultáneos:

El número de usuarios que interactuarán simultáneamente con nuestro sistema es de 3 usuarios.

Numero de transacciones a manejar dentro de ciertos periodos de tiempo:

Se estima que se manejará alrededor 30 transacciones durante el día, tomando en cuenta que la cajera realiza aproximadamente 20 operaciones diarias, como ahorros y retiros y el resto lo hace el gerente y la secretaria tareas propias.

El servidor de base de datos, deberá tener un respaldo apropiado, así como personal técnico listo para cualquier eventualidad.

3.3.2. Seguridad

La seguridad del sistema es por:

Uso de contraseñas para cada usuario (administrador, cajera, oficial de crédito). Esto permitirá que tengan acceso al sistema solo las personas que tienen autorización.

Registros de ingreso al sistema.

Creación de roles y asignarlos a cada usuario dependiendo su funcionalidad.

3.3.3. Fiabilidad

Es uno de los factores que dará confianza al cliente, para lo cual el sistema está controlando todo tipo de transacción y está apto a responder todo tipo de incidente.

3.3.4. Disponibilidad

El sistema ha sido desarrollado tomando en cuenta las necesidades, requerimientos, reglas, política, misión, objetivos etc. De la cooperativa, por lo que se

encuentra disponible el 80% del tiempo del día tomando en cuenta que el día tiene 24 horas; mientras que el 20% del tiempo es para tareas administrativas sobre el sistema.

3.3.5. Mantenibilidad

El sistema cuenta con características parametrizables lo que permitirá futuros mantenimientos. Es decir cada tres meses se va a realizar un mantenimiento preventivo, encargado de hacerlo están los desarrolladores.

Se realizara el mantenimiento dos veces sin ningún recargo económico, pasados estas dos revisiones tendrán costos adicionales.

3.3.6. Portabilidad

Una de las ventajas de utilizar herramientas y lenguajes basados en sw libre estamos garantizando la portabilidad. De esta manera: 99.9% es portable la aplicación por el simple hecho de utilizar el lenguaje y plataforma JAVA. 99% es portable la base de datos, PostgreSQL es decir puedo tenerlo en Windows o Linux.

3.3.7. Otros requisitos

Ninguno

ANEXO IV

Diagramas de Modelado UML

DIAGRAMAS DE CASOS DE USO

Diagrama de caso de uso – Imprimir Libreta

En la siguiente figura se observa el diagrama de caso de uso imprimir libreta.

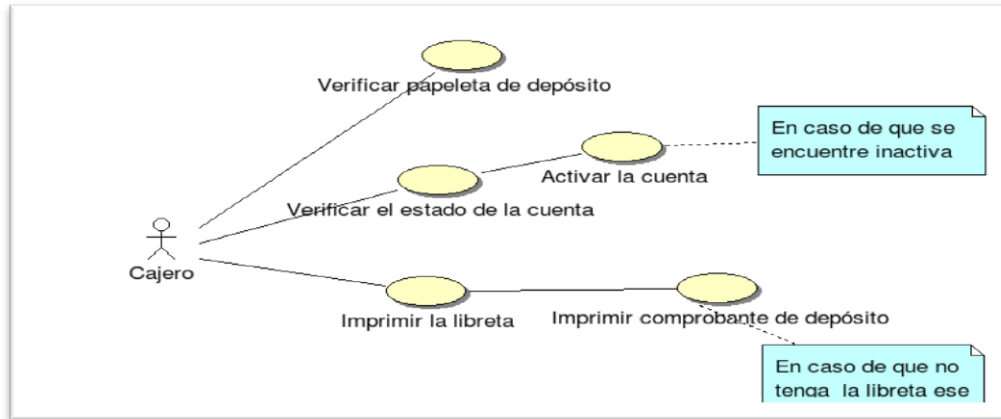


Diagrama de caso de uso – Imprimir Comprobante Depósito

En la siguiente figura se observa el diagrama de caso de uso imprimir comprobante de depósito.

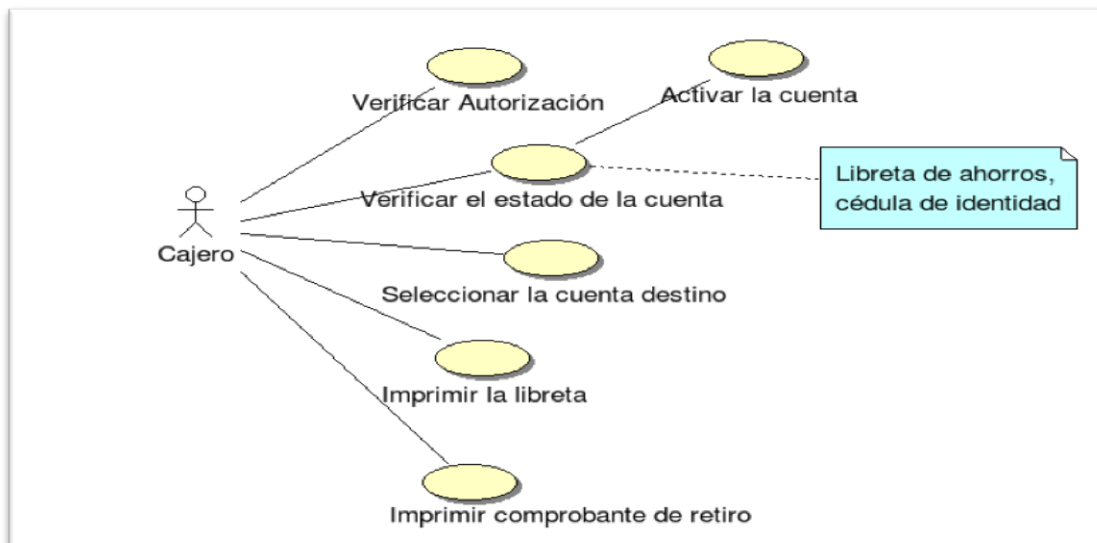


Diagrama de caso de uso – Seleccionar cuenta destino

En la siguiente figura se observa el diagrama de caso de uso seleccionar cuneta destino.

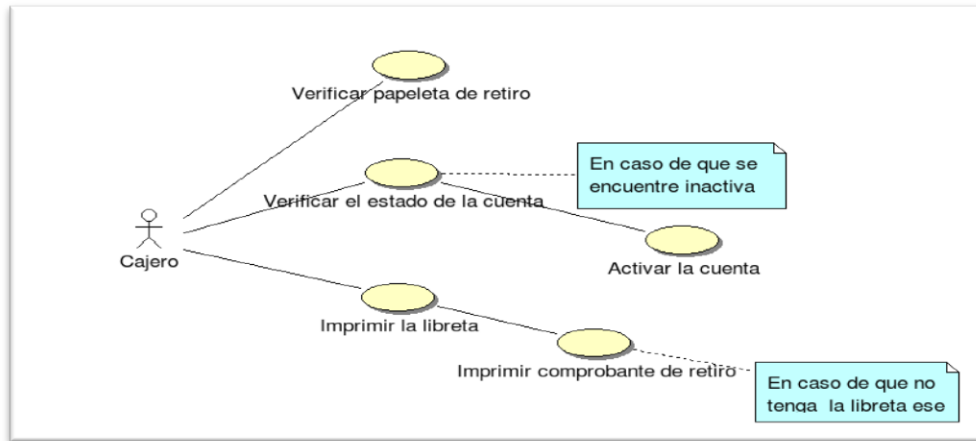


Diagrama de caso de uso – Registrar datos de Inversionista

En la siguiente figura se observa el diagrama de caso de uso registrar datos del inversionista.

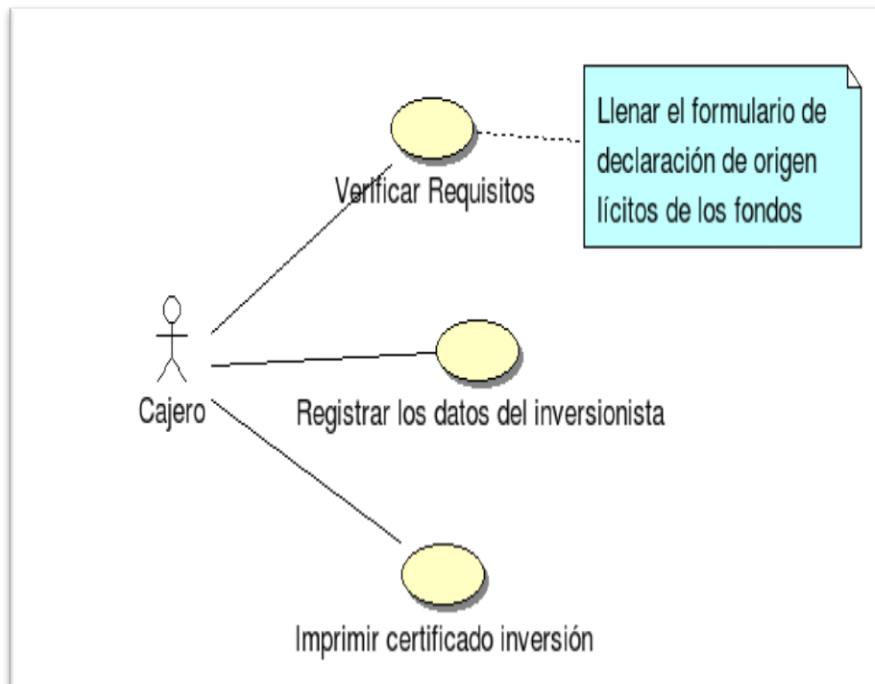


Diagrama de caso de uso – Ingreso de usuarios

En la siguiente figura se observa el diagrama de caso de uso ingreso de usuario al sistema.

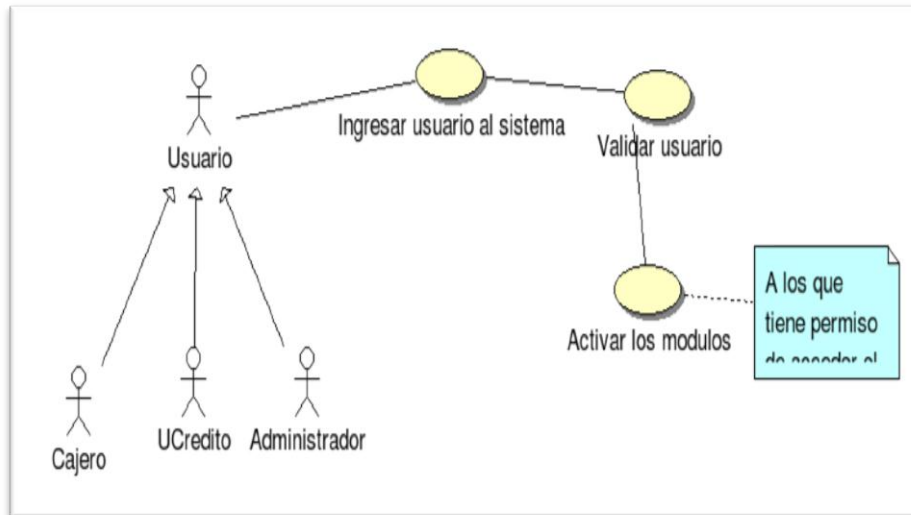


Diagrama de caso de uso – Solicitar crédito

En la siguiente figura se observa el diagrama de caso de uso solicitar crédito.

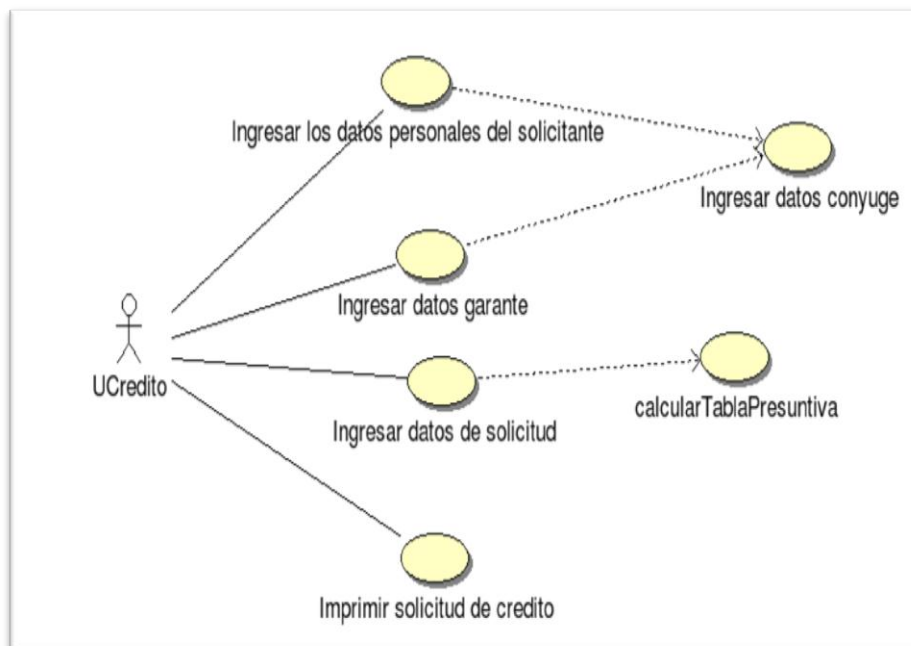


Diagrama de caso de uso – Otorgar crédito

En la siguiente figura se observa el diagrama de caso de uso otorgar crédito.

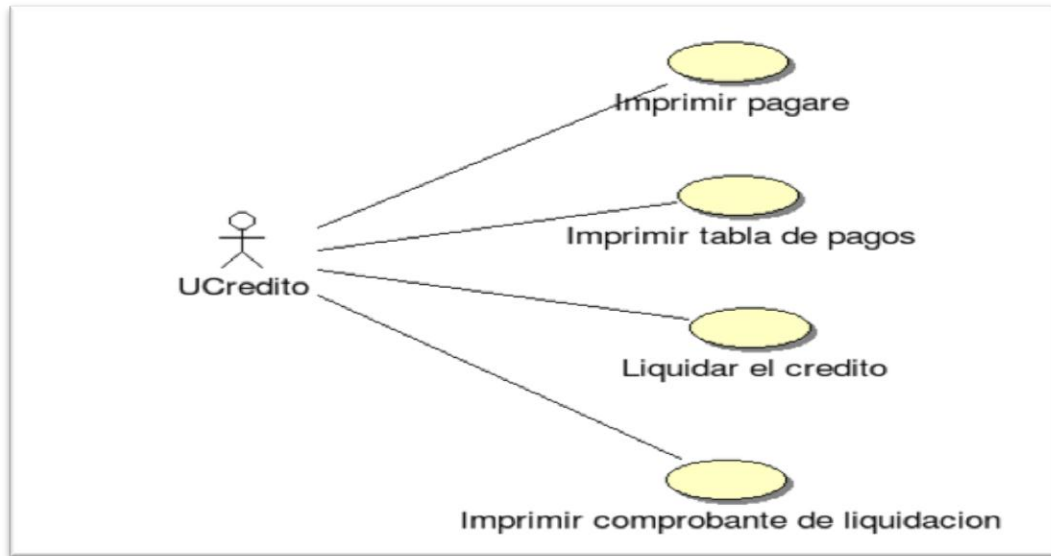
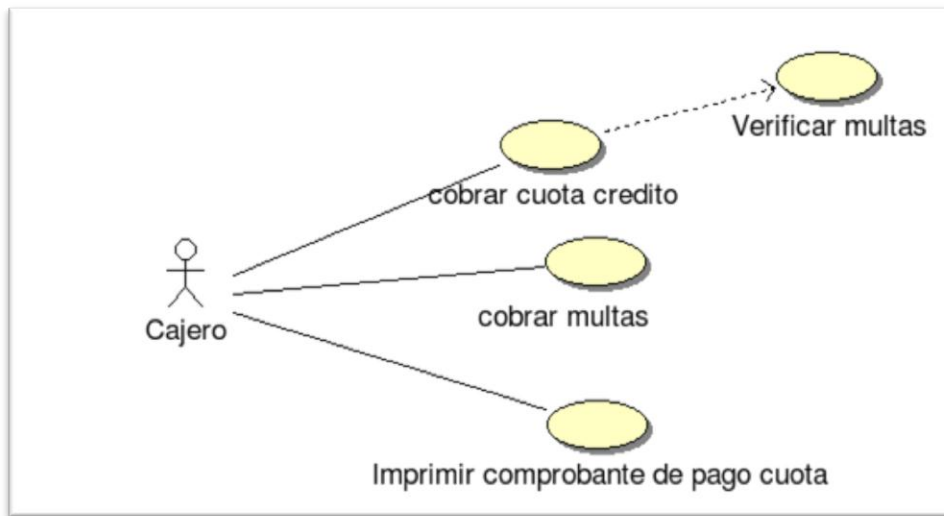


Diagrama de caso de uso – Cobrar cuota de crédito

En la siguiente figura se observa el diagrama de caso de uso cobrar cuota de crédito.



CASOS DE USO ESENCIALES

Caso de uso esencial – Apertura de Libreta

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial apertura de libreta.

Identificador	CU_Apertura_Libreta
Nombre	Apertura Libreta
Actores	Socio, Cajero
Propósito	Realizar el proceso de apertura de libreta
Visión general	El Cajero es la persona responsable de ingresar los datos del socio al sistema cumpliendo con todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio desea abrir una libreta	2. Presenta solicitud de ingreso
3. Cajero solicita datos del socio	4. Valida datos del socio, inserta datos del socio e imprime solicitud de ingreso
5. Socio firma solicitud de ingreso	6. Crea cuenta de ahorros
7. Socio solicita deposito	8. Insertar transacción
9. Cajero inserta asiento y cobra comisión	10. Imprime nota de debito e imprime libreta

Caso de uso esencial – Retiro de ahorros

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial retiro de ahorros.

Identificador	CU_Retiro
Nombre	Retiro de Ahorros
Actores	Socio, Cajero
Propósito	Realizar el proceso de retiro de ahorros
Visión general	El Cajero es la persona responsable de realizar las transacciones del socio en el sistema cumpliendo con

	todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio desea realizar un retiro de dinero de su cuenta.	2. Presenta formulario de búsqueda de socio
3. Cajero ingresa datos del socio	4. Valida datos del socio y busca socio en el sistema
5. Cajero activa cuenta e inserta transacción	6. Inserta asiento contable
7. Imprime libreta	8. Imprime comprobante de retiro sin libreta
9. Cajero entrega dinero al socio	10. Cierra transacción

Caso de uso esencial – Realizar inversión

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial realizar inversión.

Identificador	CU_Inversion
Nombre	Realizar Inversion
Actores	Socio, Cajero
Propósito	Realizar el proceso de inversión
Visión general	El Cajero es la persona responsable de realizar las transacciones del socio en el sistema cumpliendo con todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio desea	2. Presenta formulario de inversión

realizar una inversión	
3. Cajero solicita e ingresa datos de socio en el formulario de inversión	4. Valida datos del socio e inserta nuevo inversionista
5. Cajero imprime formulario de origen licito de fondos 6. Crea cuenta Inversión	7. Calcula interés 8. Crea inversión 9. Inserta Inversión 10. Inserta asiento contable
11. Cajero imprime certificado de deposito	12. Cierra transacción
13. Socio firma certificado de deposito 14. Cajero archiva Certificado de deposito	N/A

Caso de uso esencial – Distribución de fondos

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial Distribución de fondos.

Identificador	CU_Distribucion_Fondos
Nombre	Realizar distribución de fondos
Actores	Cajero
Propósito	Realizar el proceso de distribución de fondos
Visión general	El Cajero es la persona responsable de realizar las transacciones diarias en el sistema cumpliendo con todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el cajero debe realizar la distribución de fondos	2. Presenta formulario caja para realizar la distribución de fondos
3. Cajero realiza distribución de fondos en caja	4. Busca caja, busca origen y destino e inserta movimiento

5. Cajero acepta movimiento	6. Devuelve mensaje movimiento exitoso
-----------------------------	--

Caso de uso esencial – Cobro Crédito

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial Cobro de créditos.

Identificador	CU_Cobro_Credito
Nombre	Realizar cobro de créditos
Actores	Cajero, Socio
Propósito	Realizar el proceso de cobro de créditos
Visión general	El Cajero es la persona responsable de realizar las transacciones diarias en el sistema cumpliendo con todas las funcionalidades descritas en los requisitos
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio solicita pago de cuota. 2. Cajero inicia transacción cobro cuota	3. Sistema presenta formulario caja para realizar búsqueda del valor de la cuota a pagar
4. Cajero verifica si el socio tiene cuotas pendientes o multas	5. Sistema calcula el valor total a pagar
6. Socio solicita depósito	7. Sistema presenta formulario para buscar No de cuenta
8. Cajero inicia transacción	9. Inserta transacción, debita cuenta, inserta asiento contable e inserta el pago
10. Cajero imprime comprobante de pago de la cuota	11. Cierra transacción.

Caso de uso esencial – Solicitar Crédito

En la siguiente tablase visualiza el curso típico de eventos del caso de uso esencial Solicitar créditos.

Identificador	CU_Solicitar_Credito
Nombre	Solicitar créditos
Actores	UCrédito, Socio
Propósito	Realizar el proceso de solicitud de crédito
Visión general	El UCrédito es la persona responsable de realizar las transacciones diarias en lo que se refiere a la aceptación y entrega de créditos.
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el socio solicita un crédito 2. UCrédito inicia transacción asignar crédito	3. Sistema presenta formulario crédito
4. UCrédito solicita datos de solicitante	5. Sistema busca socio por el numero de cedula, ingresa datos del conyugue e inserta socio persona natural
6. UCrédito ingresa datos de solicitud de crédito	7. Sistema calcula tabla presuntiva, genera solicitud de crédito e inserta solicitud de crédito
8. UCrédito solicita impresión solicitud de crédito	9. Sistema imprime solicitud de crédito
10. Socio firma solicitud de crédito	11. Cierra transacción.

Caso de uso esencial – Otorgar Crédito

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial Otorgar créditos.

Identificador	CU_Otorgar_Credito
Nombre	Otorgar créditos
Actores	UCrédito, Socio
Propósito	Realizar el proceso de otorgación de crédito
Visión general	El UCrédito es la persona responsable de realizar las transacciones diarias en lo que se refiere a la aceptación y entrega de créditos.
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el UCrédito solicita información del garante al socio	2. Sistema presenta formulario crédito
3. UCrédito ingresa datos de garante	4. Sistema valida datos del garante, conyugue e inserta garante
5. UCrédito ingresa datos del crédito	6. Sistema calcula tabla de cuotas e inserta crédito
7. UCrédito solicita impresión tabla de cuotas y pagare	8. Sistema imprime tabla de cuotas y pagare
9. Socio firma pagare	10. Cierra transacción.

Caso de uso esencial – Liquidar Crédito

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial Liquidar créditos.

Identificador	CU_Liquidar_Credito
Nombre	Liquidar créditos
Actores	UCrédito, Socio
Propósito	Realizar el proceso de liquidación de crédito
Visión general	El UCrédito es la persona responsable de realizar las transacciones diarias en lo que se refiere a la aceptación y entrega de créditos.

Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el UCrédito necesita realizar una liquidación de crédito	2. Sistema presenta formulario crédito
3. UCrédito ingresa datos para liquidación de crédito	4. Sistema busca numero cuenta, inserta transacción e inserta asiento contable
5. UCrédito solicita impresión de liquidación de crédito	6. Sistema imprime documento liquidación de crédito
7. UCrédito solicita impresión tabla de cuotas y pagare	8. Sistema imprime tabla de cuotas y pagare
9. Socio firma pagare	10. Cierra transacción.

Caso de uso esencial – Administra Cartera

En la siguiente tabla se visualiza el curso típico de eventos del caso de uso esencial Administrar cartera.

Identificador	CU_Administrar_Cartera
Nombre	Administrar cartera
Actores	UCrédito
Propósito	Realizar la correcta administración de la cartera
Visión general	El UCrédito es la persona responsable de administrar y actualizar la cartera diariamente.
Tipo	Primario y esencial.
Curso típico de eventos	
Acción del actor	Respuesta del sistema
1. Este caso de uso inicia cuando el UCrédito necesita revisar la cartera	2. Sistema presenta formulario de crédito
3. UCrédito solicita reclasificar cartera	4. Sistema reclasifica cartera y genera listado de cartera

5. UCrédito solicita impresión de clasificación de cartera	6. Sistema imprime documento de clasificación cartera
7. UCrédito solicita impresión de notificación socio	8. Sistema imprime notificación socio

DIAGRAMAS DE ITERACIÓN

Diagrama de Iteración – Apertura Libreta

En la siguiente figura se observa el diagrama de iteración de Apertura de libreta.

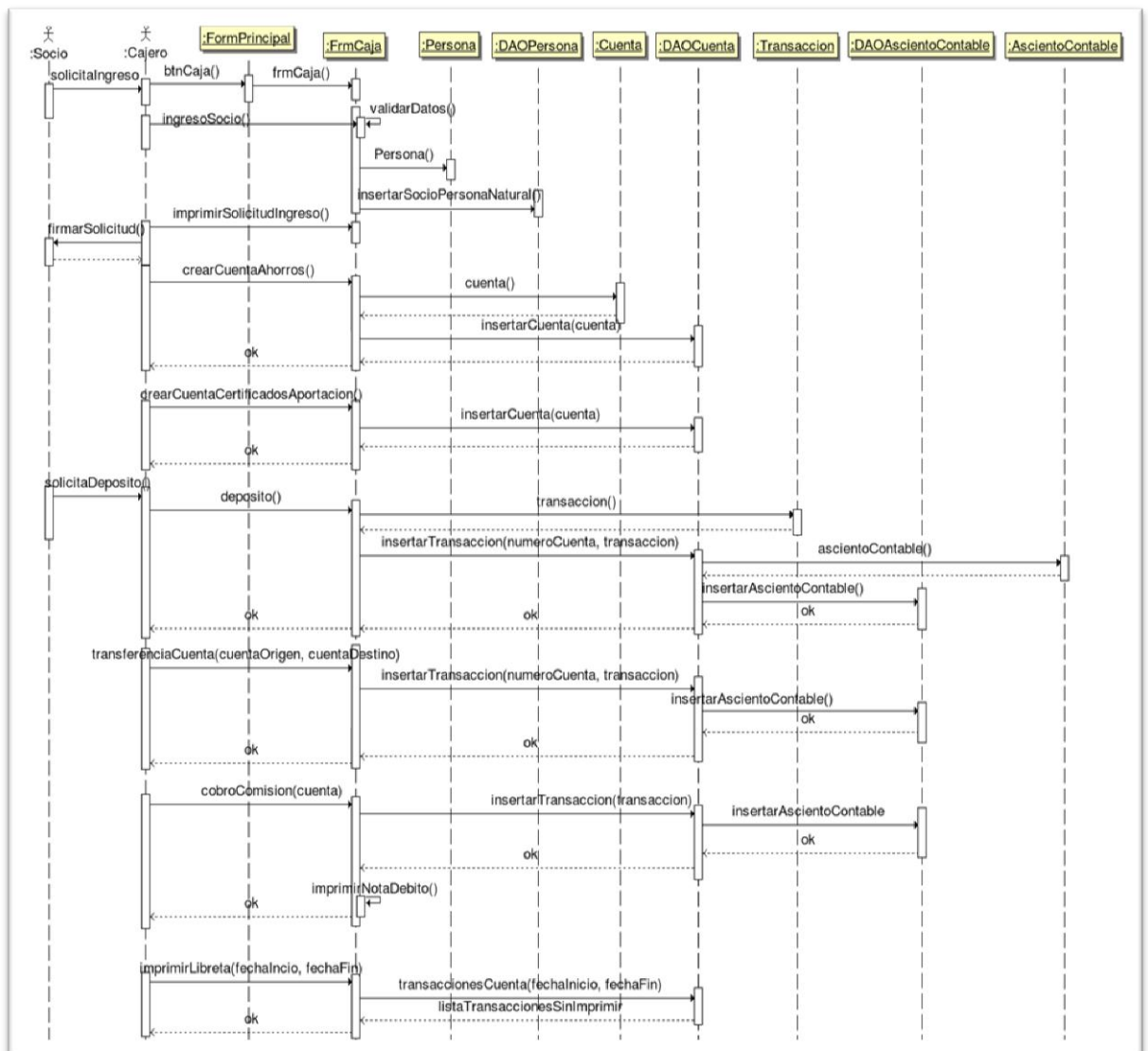


Diagrama de Iteración – Retiro de ahorros

En la siguiente figura se observa el diagrama de iteración de Retiro de ahorros.

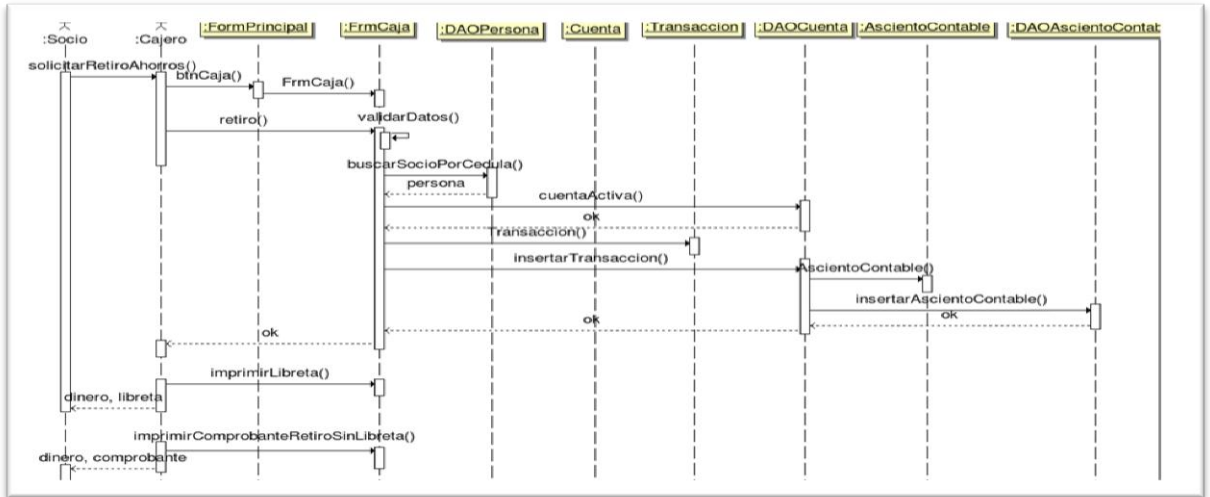


Diagrama de Iteración – Realizar inversión

En la siguiente figura se observa el diagrama de iteración de Realizar inversión.

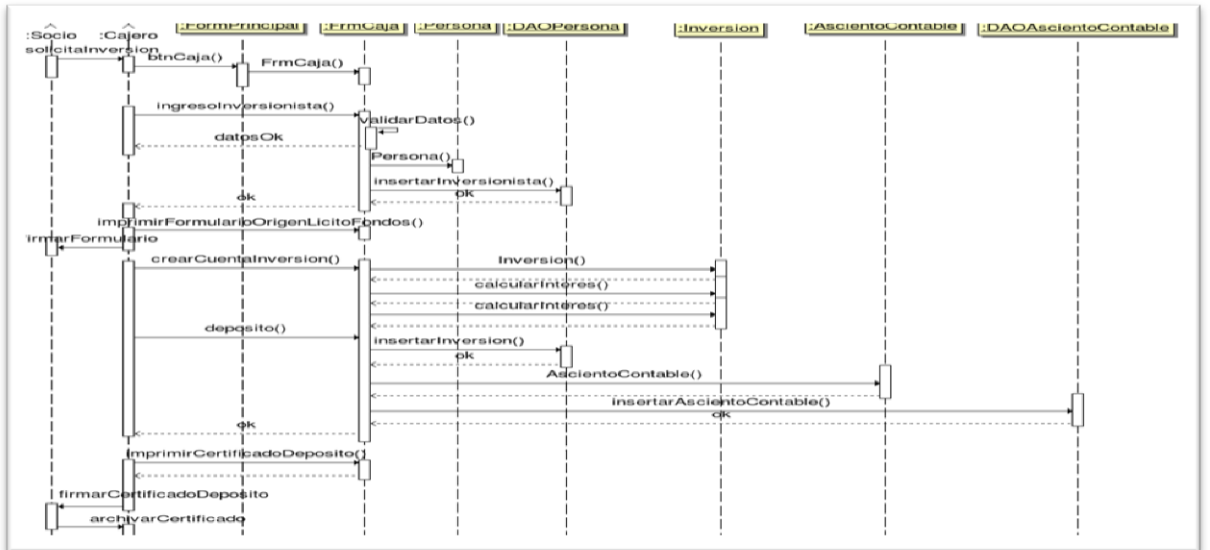


Diagrama de Iteración – Distribución de fondos

En la siguiente figura se observa el diagrama de iteración de Distribución de fondos.

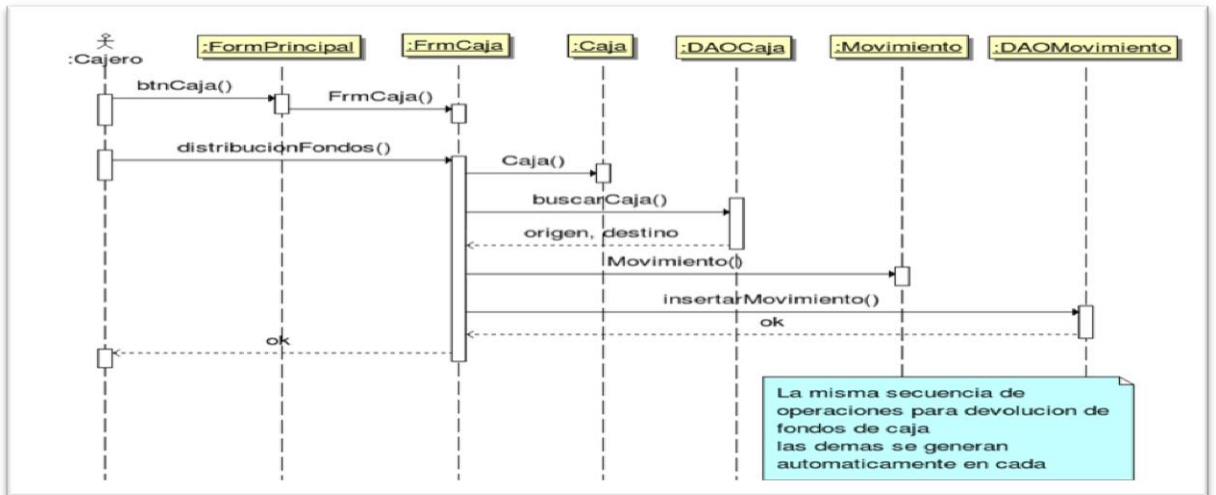


Diagrama de Iteración – Cobro de crédito

En la siguiente figura se observa el diagrama de iteración de Cobro de crédito.

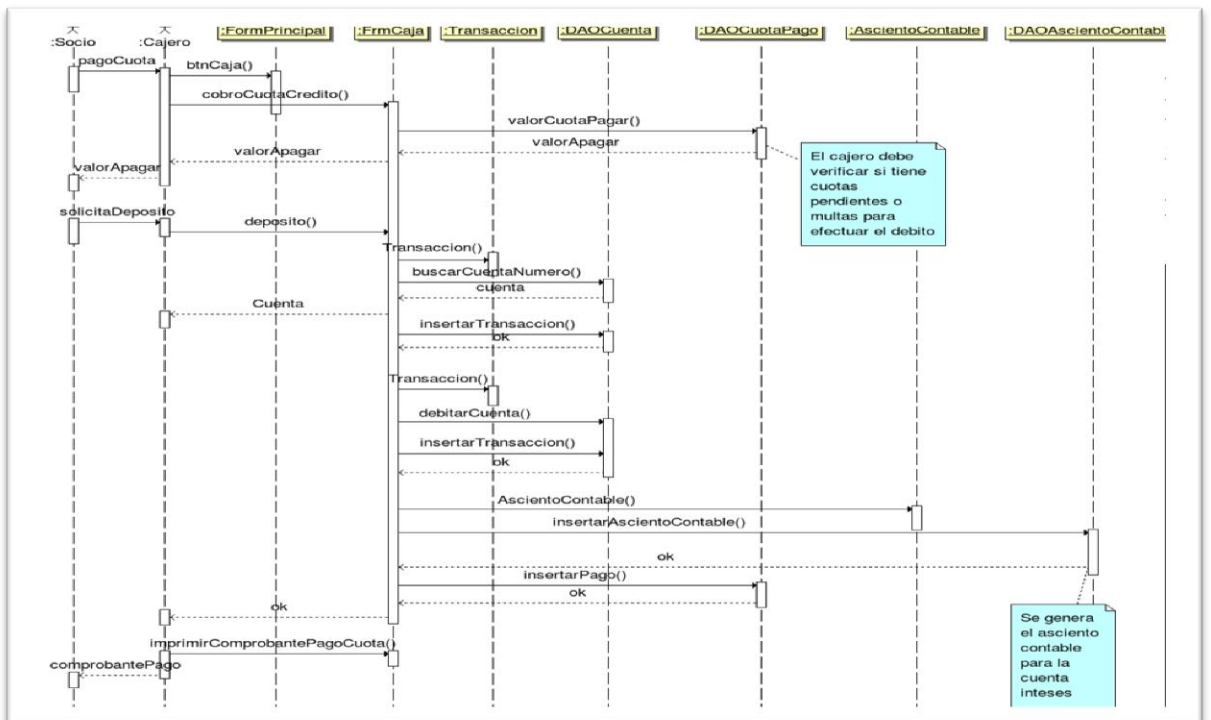


Diagrama de Iteración – Solicitar crédito

En la siguiente figura se observa el diagrama de iteración de Solicitar crédito.

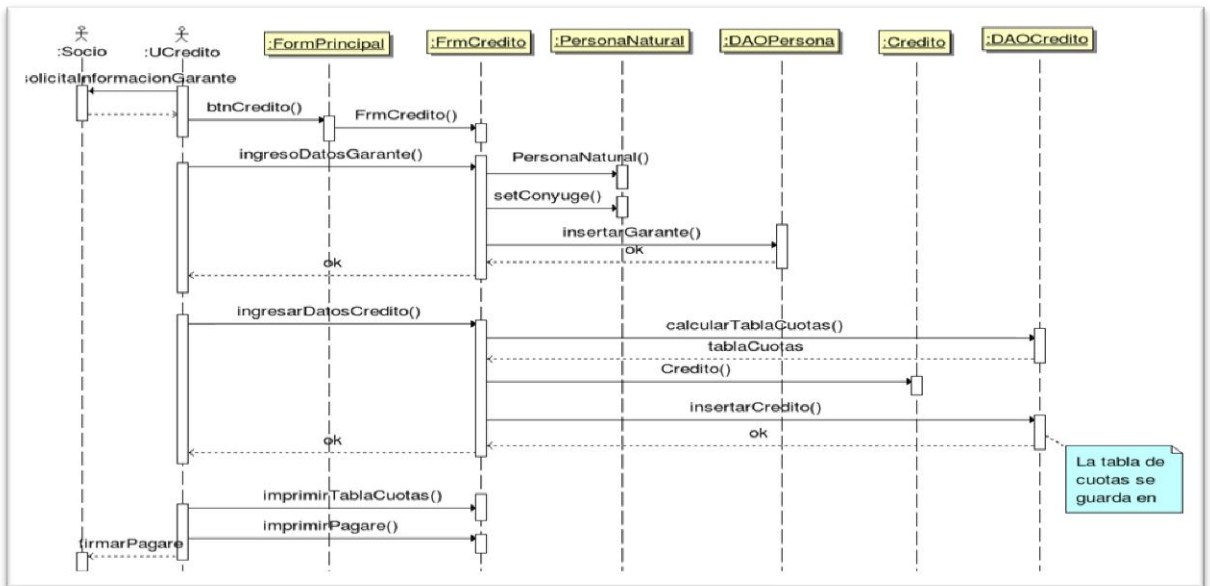


Diagrama de Iteración – Otorgar crédito

En la siguiente figura se observa el diagrama de iteración de Otorgar crédito.

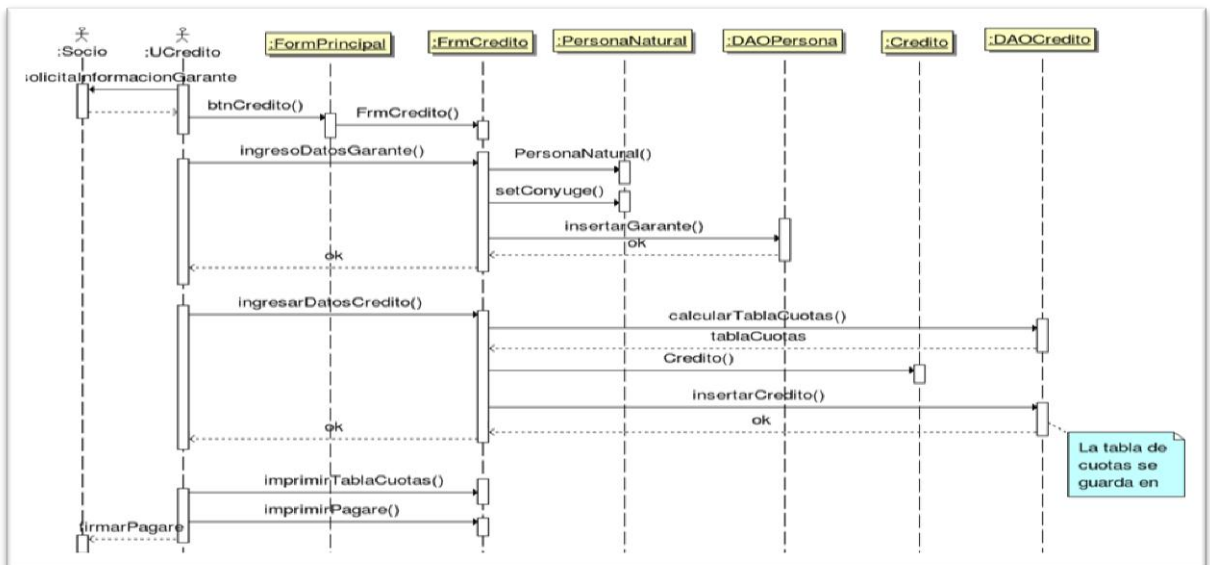


Diagrama de Iteración – Liquidar crédito

En la figura 16 se observa el diagrama de iteración de Liquidar crédito.

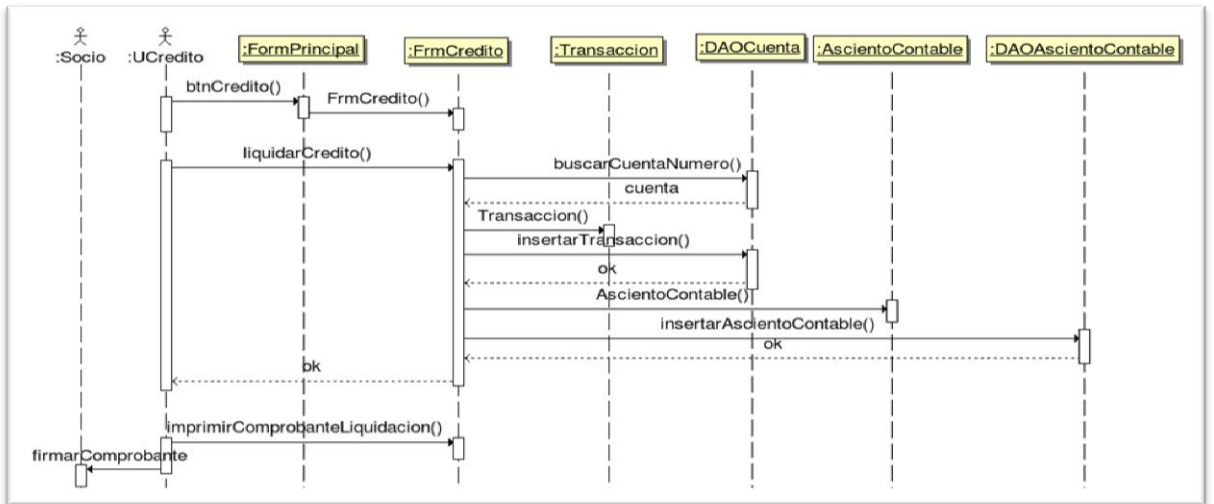


Diagrama de Iteración – Administrar cartera

En la siguiente figura se observa el diagrama iteración de Administrar cartera.

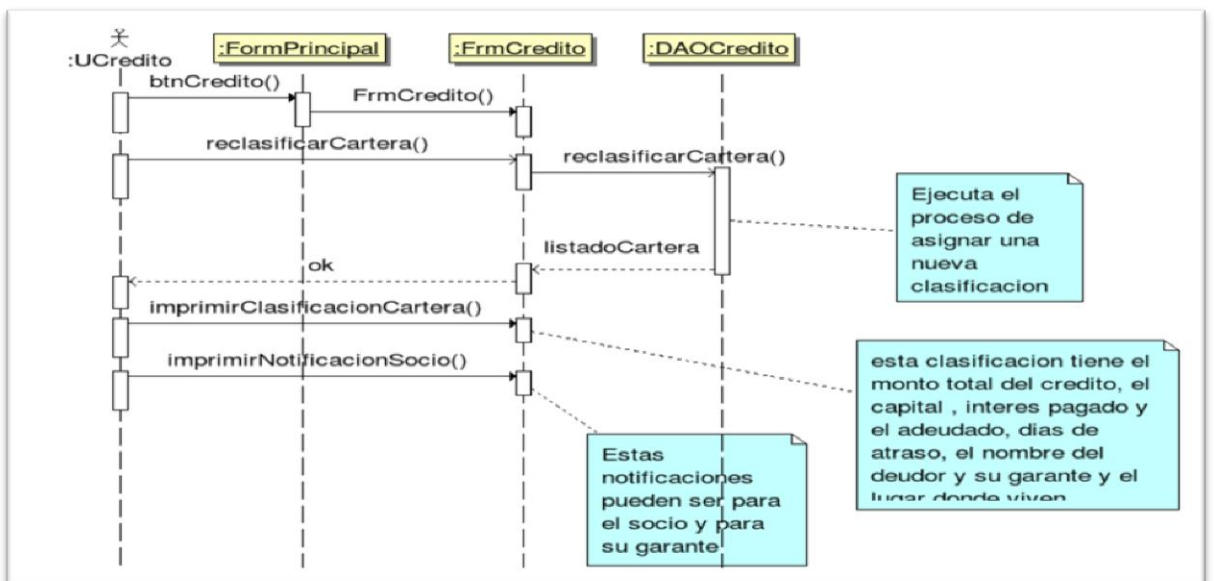


DIAGRAMA DE CLASES

Diagrama de Clases – Apertura Cuenta

En la siguiente figura se observa el diagrama clases de Apertura cuenta.

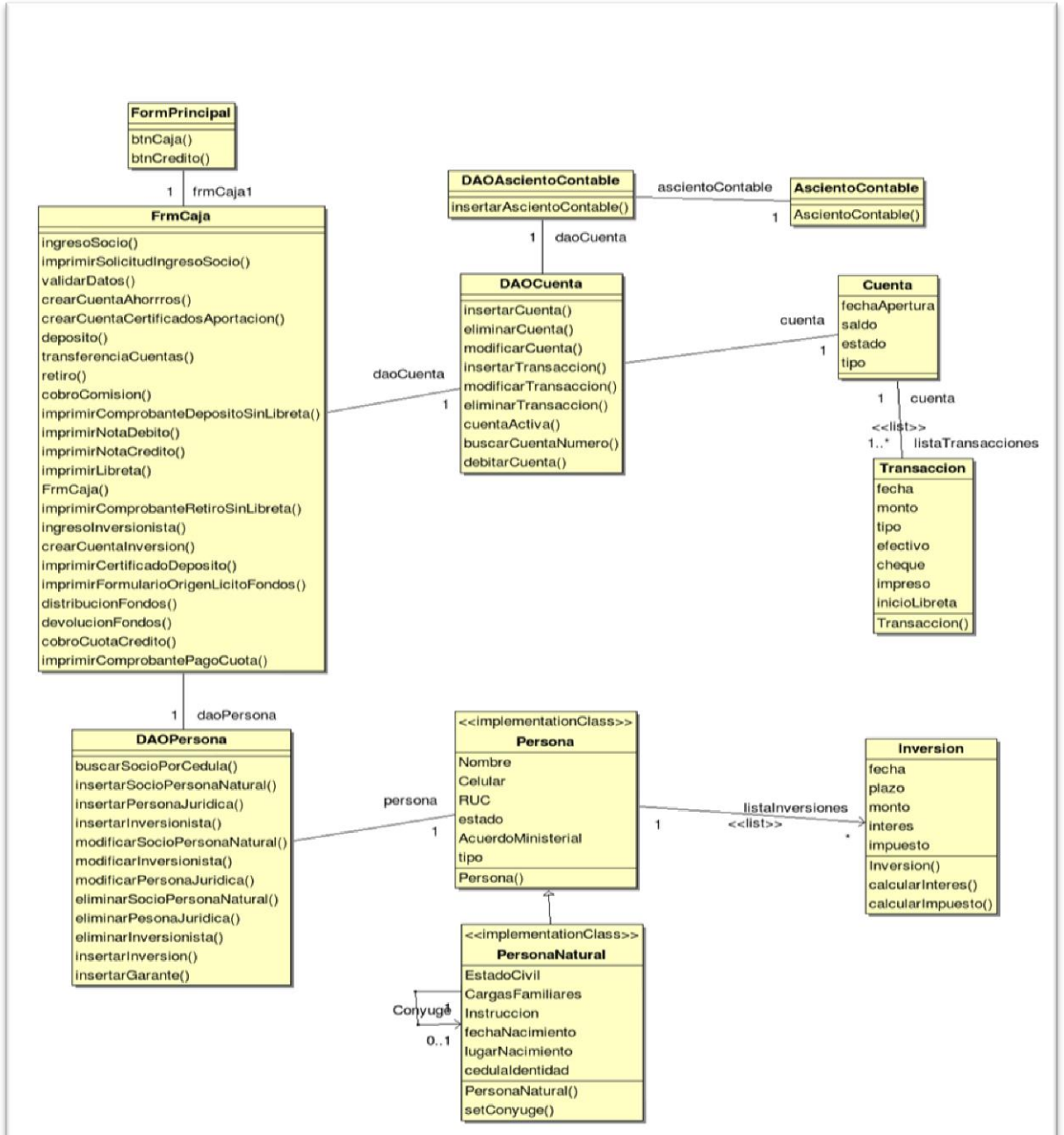


Diagrama de Clases – Movimientos caja

En la siguiente figura se observa el diagrama clases de Movimiento de caja.



DIAGRAMAS DE DESPLIEGUE

Diagrama de Despliegue – Caja

En la siguiente figura se observa el diagrama de despliegue de Caja.

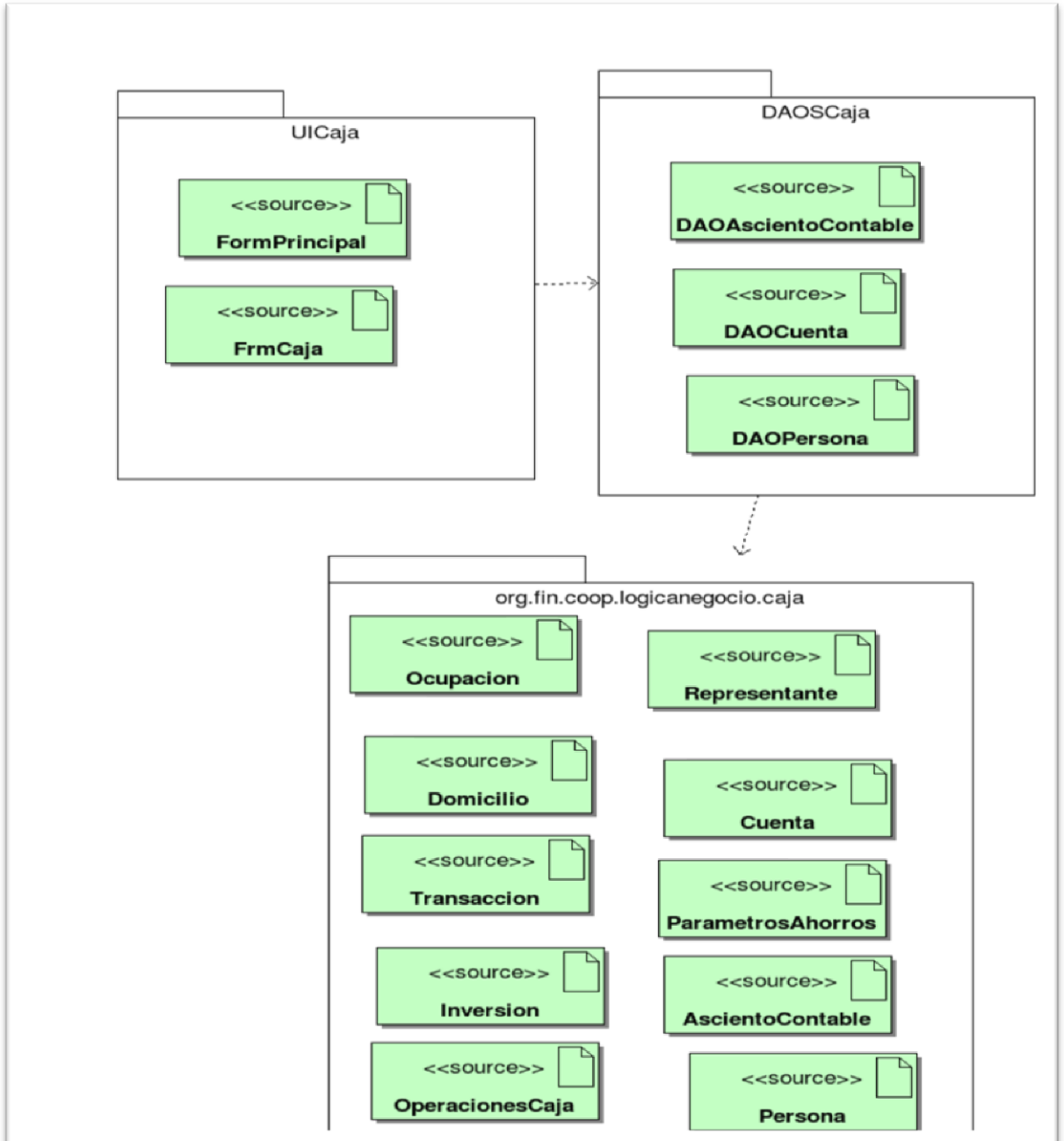
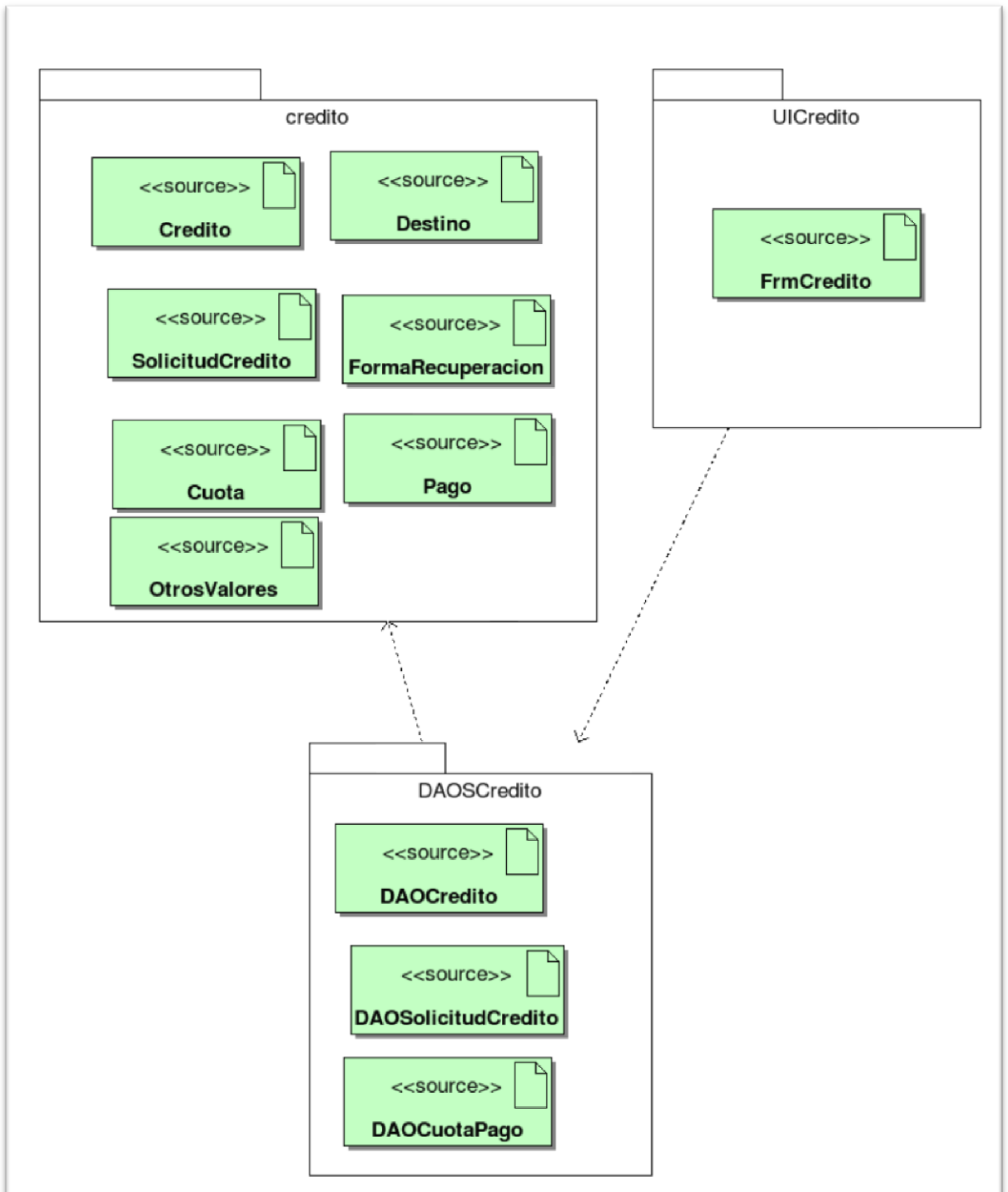


Diagrama de Despliegue – Crédito

En la siguiente figura se observa el diagrama de despliegue de Crédito.



ANEXO V

Diccionario de datos

DICCIONARIO DE DATOS DEL ESQUEMA CAJA

caja.caja

F-Key	Nombre	Tipo	Descripción
	Srlcodcaja	Serial	<i>PRIMARY KEY</i>
	Txtresponsable	character varying(20)	<i>NOT NULL</i>
	Dblsaldo	double precisión	<i>NOT NULL</i>
	dblfondocambio	double precisión	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.movimiento

Index - Esquema caja

caja.conyuge

F-Key	Nombre	Tipo	Descripción
caja.persona_natural.srlcodpersona_natural	Intcodconyuge	integer	<i>PRIMARY KEY</i>
caja.persona_natural.srlcodpersona_natural	intpersona_natural	integer	<i>PRIMARY KEY</i>

Index - Esquema caja

caja.cuenta

F-Key	Name	Type	Description
	Srlnumcuenta	serial	<i>PRIMARY KEY</i>
	Datfechaapertura	date	<i>NOT NULL DEFAULT now()</i>
	Dblsaldo	double precision	<i>NOT NULL</i>
	Chrestado	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
	Chrtipocuenta	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
caja.persona.srlnumsocio	Intnumsocio	integer	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.transaccion

Index - Esquema caja

caja.domicilio

F-Key	Nombre	Tipo	Descripción
	Srlcoddomicilio	Serial	<i>PRIMARY KEY</i>
	txtdireccion	character varying(50)	<i>NOT NULL</i>
	txtsitioreferencia	character varying(50)	<i>NOT NULL</i>
	txttelefonofijo	character varying(10)	<i>NOT NULL DEFAULT '000000'::character varying</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.persona

Index - Esquema caja

caja.inversion

F-Key	Nombre	Tipo	Descripción
	Srlcodinversion	serial	<i>PRIMARY KEY</i>
	Datfecha	date	<i>NOT NULL DEFAULT now()</i>
	Intplaso	integer	<i>NOT NULL</i>
	Dblmonto	double precision	<i>NOT NULL</i>
	Dblinteres	double precision	<i>NOT NULL</i>
	Dblimpuesto	double precision	<i>NOT NULL</i>
caja.persona.srlnum socio	Intcodinversionista	integer	<i>NOT NULL</i>

Index - Esquema caja

caja.movimiento

F-Key	Nombre	Tipo	Descripción
caja.caja.srlcodcaja	Srlcodmovimiento	serial	<i>PRIMARY KEY</i>
	Datfechamovimiento	date	<i>NOT NULL DEFAULT now()</i>
	Chrtipomovimiento	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
	Dblvalormovimiento	double precision	<i>NOT NULL</i>
	Intcodcaja	integer	<i>NOT NULL</i>

Index - Esquema caja

caja.ocupacion

F-Key	Nombre	Tipo	Descripción
	Srlcodocupacion	Serial	<i>PRIMARY KEY</i>
	txtactividadeconomica	character varying(50)	<i>NOT NULL</i>
	Txtdirecciontrabajo	character varying(50)	<i>NOT NULL</i>
	Txttelefono	character varying(10)	<i>NOT NULL DEFAULT '000000'::character varying</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.persona

Index - Esquema caja

caja.parametrosahorros

F-Key	Nombre	Tipo	Descripción
	Srlcodparametro	Serial	<i>PRIMARY KEY</i>
	Dbltasainterres	double precision	<i>NOT NULL</i>
	Inttiempoinactivo	Integer	<i>NOT NULL</i>
	dblsaldominimo	double precision	<i>NOT NULL</i>

Index - Esquema caja

caja.persona

F-Key	Nombre	Tipo	Descripción
	Srlnumsocio	Serial	<i>PRIMARY KEY</i>
	Txtcedula	character varying(10)	<i>NOT NULL</i>
	Txtnombre	character varying(50)	<i>NOT NULL</i>
	Txtcelular	character varying(10)	<i>NOT NULL DEFAULT '000000'::character varying</i>
	Txtruc	character varying(13)	<i>NOT NULL DEFAULT '000000'::character varying</i>
	txtacuerdoministerial	character varying(20)	<i>NOT NULL DEFAULT '000000'::character varying</i>
	Chrestado	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>

	Chrtipo	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
caja.ocupacion.srlcodocupacion	intocupacion	integer	<i>NOT NULL</i>
caja.domicilio.srlcodomicilio	intdomicilio	integer	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.cuenta
- caja.inversion
- caja.persona_natural
- caja.representante
- credito.solicitudcredito

Index - Esquema caja

caja.persona_natural

F-Key	Nombre	Tipo	Descripción
	srlcodpersona_natural	serial	<i>PRIMARY KEY</i>
	chrestadocivil	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
	intcargasfamiliares	integer	<i>NOT NULL</i>
	chrinstruccion	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
	datfechanacimiento	date	<i>NOT NULL DEFAULT now()</i>
	txtlugarnacimiento	character varying(50)	<i>NOT NULL</i>
caja.persona.srlnumsocio	Intpersona	integer	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- caja.conyuge

Index - Esquema caja

caja.representante

F-Key	Nombre	Tipo	Descripción

caja.persona.srlnumsocio	intnumrepresentante	integer	PRIMARY KEY
caja.persona.srlnumsocio	intnumrepresentado	integer	PRIMARY KEY

Index - Esquema caja

caja.transaccion

F-Key	Name	Type	Description
	Srlcodtransaccion	serial	PRIMARY KEY
	Datfechatransaccion	date	NOT NULL DEFAULT now()
	Dbلمonto	double precision	NOT NULL
	Chrtipotransaccion	character(1)	NOT NULL DEFAULT '0'::bpchar
	Chrformatraccion	character(1)	NOT NULL DEFAULT '0'::bpchar
caja.cuenta.srlnumcuenta	Intcodcuenta	integer	NOT NULL
	Chrimpreso	character(1)	NOT NULL DEFAULT '0'::bpchar
	Chriniciolibreta	character(1)	NOT NULL DEFAULT '0'::bpchar

DICCIONARIO DE DATOS DEL ESQUEMA CREDITO

credito.credito

F-Key	Name	Type	Description
	srlcodcredito	serial	PRIMARY KEY
	dbltasainteres	double precision	NOT NULL
	dblinteresmora	double precision	NOT NULL
	dblsegurodesgravamen	double precision	NOT NULL
	Datfechaliquidacion	date	NOT NULL DEFAULT now()
	Datfechavencimiento	date	NOT NULL

			<i>DEFAULT now()</i>
	Dblmontocredito	double precision	<i>NOT NULL</i>
	Chrtipoamortizacion	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
credito.solicitudcredito.srlcodsolicitudcredito	Intcodsolicitud	integer	<i>NOT NULL</i>

Index - Esquema credito

credito.cuota

F-Key	Nombre	Tipo	Descripción
	srlcodcuota	Serial	<i>PRIMARY KEY</i>
	datfechavencimiento	Date	<i>NOT NULL DEFAULT now()</i>
	dblvalorcapital	double precision	<i>NOT NULL</i>
	dblvalorinteres	double precision	<i>NOT NULL</i>
	chrclasificacion	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
credito.otros_valores.srlcodotrosvalores	intotrosvalores	Integer	<i>NOT NULL</i>
credito.pago.srlcodpago	Intcodpago	Integer	<i>NOT NULL</i>

Index - Esquema credito

credito.destino

F-Key	Nombre	Tipo	Descripción
	Srlcoddestino	Serial	<i>PRIMARY KEY</i>
	Txtdescripcion	character varying(50)	<i>NOT NULL</i>
	chrareaproductiva	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>

Tablas referencing this one via Foreign Key Constraints:

- credito.solicitudcredito

Index - Esquema credito

credito.forma_recuperacion

F-Key	Nombre	Tipo	Descripción
	Srlcodformarecuperacion	serial	PRIMARY KEY
	Txtdescripcion	character varying(50)	NOT NULL
	Intnumdias	integer	NOT NULL

Tablas referencing this one via Foreign Key Constraints:

- credito.solicitudcredito

Index - Esquema credito

credito.otros_valores

F-Key	Name	Type	Description
	srlcodotrosvalores	serial	PRIMARY KEY
	txtnombre	character varying(50)	NOT NULL
	dblvalor	double precision	NOT NULL

Tablas referencing this one via Foreign Key Constraints:

- credito.cuota

Index - Esquema credito

credito.pago

F-Key	Nombre	Tipo	Descripción
	srlcodpago	Serial	PRIMARY KEY
	datfechapago	Date	NOT NULL DEFAULT now()
	chrformapago	character(1)	NOT NULL DEFAULT '0'::bpchar
	dblvalorpago	double precision	NOT NULL

Tablas referencing this one via Foreign Key Constraints:

- credito.cuota

Index - Esquema credito

credito.solicitudcredito

F-Key	Nombre	Tipo	Descripción
	srlcodsolicitudcredito	serial	PRIMARY KEY
	datfechaelaboracion	date	NOT NULL DEFAULT now()
	datfechaaprobacion	date	NOT NULL DEFAULT

			<i>now()</i>
	chrestadosolicitud	character(1)	<i>NOT NULL DEFAULT '0'::bpchar</i>
	txtdescripcion	character varying(50)	<i>NOT NULL</i>
	dblmontosolicitado	double precision	<i>NOT NULL</i>
	innumerocotas	integer	<i>NOT NULL</i>
credito.destino.srlcoddestino	intdestino	integer	<i>NOT NULL</i>
credito.forma_recuperacion.srlcodformare cuperacion	intformarecuperacion	integer	<i>NOT NULL</i>
caja.persona.srlnumsocio	intcodsolicitante	integer	<i>NOT NULL</i>
caja.persona.srlnumsocio	intcodgarante	integer	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- credito.credito

Index - Esquema credito

DICCIONARIO DE DATOS DEL ESQUEMA PUBLIC

public.Cuenta

F-Key	Nombre	Tipo	Descripción
	txtCodigoCuenta	character varying(50)	<i>PRIMARY KEY</i>
	txtNombre	character varying(50)	<i>NOT NULL</i>

Tablas referencing this one via Foreign Key Constraints:

- public.Registro_Cuenta

Index - Esquema public

public.Parcial

F-Key	Nombre	Tipo	Descripción
	srlCodigoParcial	serial	<i>PRIMARY KEY</i>
	txtDescripcion	Text	<i>NOT NULL</i>
	dblDouble	double precision	

public.Registro_Cuenta.intNumeroRegistro#1	intNumeroRegistro	integer	NOT NULL
public.Registro_Cuenta.txtCodigoCuenta#1	txtCodigoCuenta	character varying(50)	NOT NULL

Index - Esquema public

public.Registro

F-Key	Nombre	Tipo	Descripción
	intNumeroRegistro	Integer	PRIMARY KEY
	fchFechaRegistro	Date	NOT NULL DEFAULT now()
	txtDescripcion	Text	
	dblTotalDebe	double precisión	NOT NULL
	dblTotalHaber	double precisión	

public.Registro Constraints

Nombre	Constraint
checkdebe	CHECK ("dblTotalDebe" >= (0)::double precision)
checkhaber	CHECK ("dblTotalHaber" >= (0)::double precision)

Tablas referencing this one via Foreign Key Constraints:

- public.Registro_Cuenta

Index - Esquema public

public.Registro_Cuenta

F-Key	Nombre	Tipo	Descripción
public.Registro.intNumeroRegistro	intNumeroRegistro	Integer	PRIMARY KEY
public.Cuenta.txtCodigoCuenta	txtCodigoCuenta	character varying(50)	PRIMARY KEY
	dblValor	double precision	NOT NULL
	chrTipo	character(1)	NOT NULL

public.Registro_Cuenta Constraints

Nombre	Constraint
checkDebeHaberDH	CHECK (((chrTipo = 'D'::bpchar) OR (chrTipo = 'H'::bpchar)))
Checkvalor	CHECK ("dblValor" >= (0)::double precision)

Tablas referencing this one via Foreign Key Constraints:

- public.Parcial

Index - Esquema public

ANEXO VI

Script de la base de datos

SCRIPT DEL ESQUEMA CAJA

Creación de la tabla ocupacion

```
CREATE TABLE ocupacion (  
  srlcodocupacion serial not null,  
  txtactividadeconomica character varying(50) NOT NULL,  
  txtdirecciontrabajo character varying(50) NOT NULL,  
  txttelefono character varying(10) default '000000' NOT NULL ,  
  CONSTRAINT pk_ocupacion PRIMARY KEY (srlcodocupacion )  
);
```

Creación de la Tabla domicilio

```
CREATE TABLE domicilio (  
  srlcodomicilio serial not null,  
  txtdireccion character varying(50) NOT NULL,  
  txtsitioreferencia character varying(50) NOT NULL,  
  txttelefonofijo character varying(10) default '000000' NOT NULL,  
  CONSTRAINT pk_domicilio PRIMARY KEY (srlcodomicilio )  
);
```

Creación de la tabla persona

```
CREATE TABLE persona (  
  srlnumsocio serial not null,  
  txtcedula character varying(10) NOT NULL,  
  txtnombre character varying(50) NOT NULL,  
  txtcelular character varying(10) default '000000' NOT NULL,  
  txtruc character varying(13) default '000000' NOT NULL,  
  -- El campo estado puede tener los siguientes valores  
  -- '0' que significa Activo  
  -- '1' que significa Inactivo  
  -- '2' que significa Retirado  
  -- '3' que significa excluido  
  --  
  txtacuerdoministerial character varying(20) default '000000' not null,  
  chrestado char DEFAULT '0' NOT NULL,  
  -- El campo tipo puede tener los siguientes valores  
  -- '0' que significa socio  
  -- '1' solo inversionista  
  -- '2' solo representante  
  --  
  chrtipo char DEFAULT '0' NOT NULL,  
  intocupacion INTEGER NOT NULL,  
  intdomicilio INTEGER NOT NULL,  
  CONSTRAINT persona_pk PRIMARY KEY (srlnumsocio ),  
  CONSTRAINT fkey_persona_ocupacion FOREIGN KEY (intocupacion)  
    REFERENCES ocupacion(srlcodocupacion) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT fkey_persona_domicilio FOREIGN KEY (intdomicilio)
```

```
REFERENCES domicilio(srlcodomicilio) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

Creación de la tabla representante

```
CREATE TABLE representante (
intnumrepresentante INTEGER not null,
intnumrepresentado INTEGER NOT NULL,
CONSTRAINT pk_representante PRIMARY KEY (intnumrepresentante, intnumrepresentado ),
CONSTRAINT fkey_representate_persona FOREIGN KEY (intnumrepresentante)
REFERENCES persona(srlnumsocio) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkey_representate_representado FOREIGN KEY (intnumrepresentado)
REFERENCES persona(srlnumsocio) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

Creación de la tabla persona_natural

```
CREATE TABLE persona_natural (
srlcodpersona_natural serial not null,
-- El campo estadocivil puede tener los siguientes valores
-- '0' que significa soltero
-- '1' casado
-- '2' divorciado
-- '3' viudo
--
chrestadocivil char DEFAULT '0' NOT NULL,
intcargasfamiliares INTEGER DEFAULT 0 NOT NULL,
-- El campo instruccion puede tener los siguientes valores
-- '0' que significa basica
-- '1' secundaria
-- '2' tercer nivel
-- '3' que no tiene
chrinstruccion char DEFAULT '0' NOT NULL,
datfechanacimiento date DEFAULT now() NOT NULL,
txtlugarnacimiento character varying(50) NOT NULL,
intpersona INTEGER DEFAULT 0 NOT NULL,
CONSTRAINT pk_persona_natural PRIMARY KEY (srlcodpersona_natural ),
CONSTRAINT "fkey_personaNatural_persona" FOREIGN KEY (intpersona)
REFERENCES persona(srlnumsocio) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

Creación de la tabla conyuge

```
CREATE TABLE conyuge (
intcodconyuge INTEGER NOT NULL,
intpersona_natural INTEGER NOT NULL,
CONSTRAINT pk_conyuge PRIMARY KEY (intcodconyuge, intpersona_natural ),
```

```

CONSTRAINT fkey_conyuge1 FOREIGN KEY (intcodconyuge)
  REFERENCES persona_natural(srlcodpersona_natural) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkey_conyuge2 FOREIGN KEY (intpersona_natural)
  REFERENCES persona_natural(srlcodpersona_natural) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

Creación de la tabla inversion

```

CREATE TABLE inversion (
  srlcodinversion serial not null,
  datfecha date DEFAULT now() NOT NULL,
  intplaso INTEGER DEFAULT 0 NOT NULL,
  dblmonto double precision DEFAULT 0 NOT NULL,
  dblinteres double precision DEFAULT 0 NOT NULL,
  dblimpuesto double precision DEFAULT 0 NOT NULL,
  intcodinversionista INTEGER DEFAULT 0 NOT NULL,
  CONSTRAINT pk_inversion PRIMARY KEY (srlcodinversion ),
  CONSTRAINT fkey_inversion_persona FOREIGN KEY (intcodinversionista)
    REFERENCES persona(srlnumsocio) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

Creación de la tabla cuenta_socio

```

CREATE TABLE cuenta_socio (
  -- el codigo contable de la cuenta_socio se concatena este numero + el codigo del catalogo de
  -- cuentas para el ahorro
  srlnumcuentasocio serial not null,
  datfechaapertura date DEFAULT now() NOT NULL,
  dblsaldo double precision DEFAULT 0 NOT NULL,
  -- El campo estado puede tener los siguientes valores
  -- '0' que significa activa
  -- '1' inactiva
  --
  chrestado char DEFAULT '0' NOT NULL,
  -- El campo tipocuenta puede tener los siguientes valores
  -- '0' que significa ahoros
  -- '1' certificados aportacion
  -- '2' ahorro programado
  chrtipocuentasocio char DEFAULT '0' NOT NULL,
  intnumsocio INTEGER DEFAULT 0 NOT NULL,
  CONSTRAINT pk_cuentasocio PRIMARY KEY (srlnumcuentasocio ),
  CONSTRAINT fkey_cuentasocio_persona FOREIGN KEY (intnumsocio)
    REFERENCES persona(srlnumsocio) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

Creación de la tabla transaccion

```

CREATE TABLE transaccion (
  srlcodtransaccion serial not null,

```

```

datfechatransaccion date DEFAULT now() NOT NULL,
dblmonto double precision DEFAULT 0 NOT NULL,
-- El campo tipotransaccion puede tener los siguientes valores
-- '0' que significa deposito
-- '1' retiro
-- '2' credito
-- '3' debito
-- '4' transferencia
chrtipotransaccion char DEFAULT '0' NOT NULL,
-- El campo formatraccion puede tener los siguientes valores
-- '0' que significa efectivo
-- '1' cheque
--
chrformatraccion char DEFAULT '0' NOT NULL,
intcodcuentasocio INTEGER DEFAULT 0 NOT NULL,
-- El campo impreso puede tener los siguientes valores
-- '0' que significa no impreso
-- '1' impreso
--
chrimpreso char DEFAULT '0' NOT NULL,
-- El campo iniciolibreta puede tener los siguientes valores
-- '0' que significa no es inicio de una nueva libreta
-- '1' es inicio de una nueva libreta
--
chriniciolibreta char DEFAULT '0' NOT NULL,
CONSTRAINT pk_transaccion PRIMARY KEY (srlnumtransaccion ),
CONSTRAINT fkey_transaccion_cuenta_socio FOREIGN KEY (intcodcuentasocio)
REFERENCES cuenta_socio(srlnumcuentasocio) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

Creación de la tabla parametrosahorros

```

CREATE TABLE parametrosahorros (
srlnumparametro serial not null,
dbltasainteres double precision DEFAULT 0 NOT NULL,
inttiempoinactivo INTEGER DEFAULT 0 NOT NULL,
dblsaldominimo double precision DEFAULT 0 NOT NULL,
CONSTRAINT pk_parametrosahorros PRIMARY KEY (srlnumparametro )
);

```

-- MANEJO DE CAJA

Creación de la tabla caja

```

CREATE TABLE caja (
srlnumcaja serial not null,
txtresponsable character varying(20) NOT NULL,
dblsaldo double precision DEFAULT 0 NOT NULL,
dblfondocambio double precision DEFAULT 0 NOT NULL,
CONSTRAINT pk_caja PRIMARY KEY (srlnumcaja )
);

```

Creación de la tabla movimiento

```
CREATE TABLE movimiento (  
  srlcodmovimiento serial not null,  
  datfechamovimiento date DEFAULT now() NOT NULL,  
  -- El campo tipomovimiento puede tener los siguientes valores  
  -- '0' que significa devoluciones  
  -- '1' distribuciones  
  -- '2' depositos  
  -- '3' retiros  
  -- '4' inversiones  
  -- '5' comisiones  
  -- '6' ingresos  
  -- '7' egresos  
  -- '8' cartera  
  --  
  chrtipomovimiento char DEFAULT '0' NOT NULL,  
  dblvalormovimiento double precision DEFAULT 0 NOT NULL,  
  intcodcaja INTEGER DEFAULT 0 NOT NULL,  
  CONSTRAINT pk_movimiento PRIMARY KEY (srlcodmovimiento ),  
  CONSTRAINT fkey_movimiento_caja FOREIGN KEY (srlcodmovimiento)  
    REFERENCES caja(srlcodcaja) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION  
);
```

SCRIPT DEL ESQUEMA CREDITO

Creación de la tabla destino

```
CREATE TABLE destino (  
  srlcoddestino serial not null,  
  txtdescripcion character varying(50) NOT NULL,  
  -- El campo areaproductiva puede tener los siguientes valores  
  -- '0' que significa produccion  
  -- '1' servicios  
  --  
  chrareaproductiva char DEFAULT '0' NOT NULL,  
  CONSTRAINT pk_destino PRIMARY KEY (srlcoddestino )  
);
```

Creación de la tabla forma recuperacion

```
CREATE TABLE forma_recuperacion (  
  srlcodformarecuperacion serial not null,  
  txtdescripcion character varying(50) NOT NULL,  
  intnumdias INTEGER DEFAULT 0 NOT NULL,  
  CONSTRAINT pk_forma_recuperacion PRIMARY KEY (srlcodformarecuperacion )  
);
```

Creación de la tabla solicitudcredito

```
CREATE TABLE solicitudcredito (  
  srlcodsolicitudcredito serial not null,  
  datfechaelaboracion date DEFAULT now() NOT NULL,  
  datfechaaprobacion date DEFAULT now() NOT NULL,  
  -- El campo estadossolicitud puede tener los siguientes valores  
  -- '0' que significa en proceso de aprobacion  
  -- '1' aprobada  
  -- '2' no aprobada  
  --  
  chrestadosolicitud char DEFAULT '0' NOT NULL,  
  txtdescripcion character varying(50) NOT NULL,  
  dblmontosolicitado double precision DEFAULT 0 NOT NULL,  
  intnumerocotas INTEGER DEFAULT 0 NOT NULL,  
  intdestino INTEGER DEFAULT 0 NOT NULL,  
  intformarecuperacion INTEGER DEFAULT 0 NOT NULL,  
  intcodsolicitante INTEGER DEFAULT 0 NOT NULL,  
  intcodgarante INTEGER DEFAULT 0 NOT NULL,  
  
  CONSTRAINT pk_solicitudcredito PRIMARY KEY (srlcodsolicitudcredito ),  
  CONSTRAINT fkey_solicitudcredito_destino FOREIGN KEY (intdestino)  
    REFERENCES destino(srlcoddestino) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT fkey_solicitudcredito_formarecuperacion FOREIGN KEY (intformarecuperacion)  
    REFERENCES forma_recuperacion(srlcodformarecuperacion) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT fkey_solicitudcredito_solicitante FOREIGN KEY (intcodsolicitante)  
    REFERENCES caja.persona(srlnumsocio) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT fkey_solicitudcredito_garante FOREIGN KEY (intcodgarante)  
    REFERENCES caja.persona(srlnumsocio) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION  
);
```

Creación de la tabla credito

```
CREATE TABLE credito (  
  srlcodcredito serial not null,  
  dbltasainterres double precision DEFAULT 0 NOT NULL,  
  dblinteresmora double precision DEFAULT 0 NOT NULL,  
  dblsegurodesgravamen double precision DEFAULT 0 NOT NULL,  
  datfechaliquidacion date DEFAULT now() NOT NULL,  
  datfechavencimiento date DEFAULT now() NOT NULL,  
  dblmontocredito double precision DEFAULT 0 NOT NULL,  
  -- El campo tipoamortizacion puede tener los siguientes valores  
  -- '0' que significa fija  
  -- '1' variable  
  chrtipoamortizacion char DEFAULT '0' NOT NULL,  
  intcodsolicitud INTEGER DEFAULT 0 NOT NULL,  
  CONSTRAINT pk_credito PRIMARY KEY (srlcodcredito ),
```

```
CONSTRAINT fkey_credito_solicitudcredito FOREIGN KEY (intcodsolicitud)
REFERENCES solicitudcredito(srlcodsolicitudcredito) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

Creación de la tabla pago

```
CREATE TABLE pago (
srlcodpago serial not null,
datfechapago date DEFAULT now() NOT NULL,
-- El campo formapago puede tener los siguientes valores
-- '0' que significa efectivo
-- '1' cheque
chrformapago char DEFAULT '0' NOT NULL,
dblvalorpago double precision DEFAULT 0 NOT NULL,
CONSTRAINT pk_pago PRIMARY KEY (srlcodpago )
);
```

Creación de la tabla otros valores

```
CREATE TABLE otros_valores (
srlcodotrosvalores serial not null,
txtnombre character varying(50) NOT NULL,
dblvalor double precision DEFAULT 0 NOT NULL,
CONSTRAINT pk_otros_valores PRIMARY KEY (srlcodotrosvalores )
);
```

Creación de la tabla cuota

```
CREATE TABLE cuota (
srlcodcuota serial not null,
datfechavencimiento date DEFAULT now() NOT NULL,
dblvalorcapital double precision DEFAULT 0 NOT NULL,
dblvalorinteres double precision DEFAULT 0 NOT NULL,
-- El campo clasificacion puede tener los siguientes valores
-- 'a' que significa pago puntual
-- 'b' , etc de acuerdo a la clasificacion establecida por la
-- superintendencia
--
chrclasificacion char DEFAULT '0' NOT NULL,
intotrosvalores INTEGER DEFAULT 0 NOT NULL,
intcodpago INTEGER DEFAULT 0 NOT NULL,
CONSTRAINT pk_cuota PRIMARY KEY (srlcodcuota ),
CONSTRAINT fkey_cuota_otrosvalores FOREIGN KEY (intotrosvalores)
REFERENCES otros_valores(srlcodotrosvalores) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkey_cuota_pago FOREIGN KEY (intcodpago)
REFERENCES pago(srlcodpago) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```


SCRIPT DEL ESQUEMA PUBLIC

Creación de la tabla Cuenta

```
CREATE TABLE public."Cuenta"  
(  
  "txtCodigoCuenta" character varying(50) NOT NULL,  
  "txtNombre" character varying(50) NOT NULL,  
  CONSTRAINT "Cuenta_pkey" PRIMARY KEY ("txtCodigoCuenta")  
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE public."Cuenta" OWNER TO postgres;
```

Creación de la tabla Parcial

```
CREATE TABLE public."Parcial"  
(  
  "srlCodigoParcial" integer NOT NULL DEFAULT  
nextval("Parcial_srlCodigoParcial_seq"::regclass),  
  "txtDescripcion" text NOT NULL,  
  "dblDouble" double precision DEFAULT 0,  
  "intNumeroRegistro" integer NOT NULL,  
  "txtCodigoCuenta" character varying(50) NOT NULL,  
  CONSTRAINT "Parcial_pkey" PRIMARY KEY ("srlCodigoParcial"),  
  CONSTRAINT "Parcial_intNumeroRegistro_fkey" FOREIGN KEY ("intNumeroRegistro",  
"txtCodigoCuenta")  
REFERENCES public."Registro_Cuenta" ("intNumeroRegistro", "txtCodigoCuenta") MATCH  
SIMPLE  
ON UPDATE NO ACTION ON DELETE NO ACTION  
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE public."Parcial" OWNER TO postgres;
```

Creación de la tabla Registro

```
CREATE TABLE public."Registro"  
(  
  "intNumeroRegistro" integer NOT NULL,  
  "fchFechaRegistro" date NOT NULL DEFAULT now(),  
  "txtDescripcion" text,  
  "dblTotalDebe" double precision NOT NULL DEFAULT 0,  
  "dblTotalHaber" double precision DEFAULT 0,  
  CONSTRAINT "Registro_pkey" PRIMARY KEY ("intNumeroRegistro"),  
  CONSTRAINT checkdebe CHECK ("dblTotalDebe" >= 0::double precision),  
  CONSTRAINT checkhaber CHECK ("dblTotalHaber" >= 0::double precision)  
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE public."Registro" OWNER TO postgres;  
GRANT ALL ON TABLE public."Registro" TO postgres;
```

Creación de la tabla Registro_Cuenta

```
CREATE TABLE public."Registro_Cuenta"  
(  
  "intNumeroRegistro" integer NOT NULL,  
  "txtCodigoCuenta" character varying(50) NOT NULL,  
  "dblValor" double precision NOT NULL DEFAULT 0,  
  "chrTipo" character(1) NOT NULL DEFAULT 0,  
  CONSTRAINT "Registro_Cuenta_pkey" PRIMARY KEY ("intNumeroRegistro",  
"txtCodigoCuenta"),  
  CONSTRAINT "FKREGISTROCuenta1" FOREIGN KEY ("txtCodigoCuenta")  
    REFERENCES public."Cuenta" ("txtCodigoCuenta") MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT "FKREGISTROCuenta2" FOREIGN KEY ("intNumeroRegistro")  
    REFERENCES public."Registro" ("intNumeroRegistro") MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION,  
  CONSTRAINT "checkDebeHaberDH" CHECK ("chrTipo" = 'D'::bpchar OR "chrTipo" =  
'H'::bpchar),  
  CONSTRAINT checkvalor CHECK ("dblValor" >= 0::double precision)  
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE public."Registro_Cuenta" OWNER TO postgres;  
  
-- Trigger: actualizartotaldebehaberregistro on public."Registro_Cuenta"  
  
-- DROP TRIGGER actualizartotaldebehaberregistro ON public."Registro_Cuenta";  
  
CREATE TRIGGER actualizartotaldebehaberregistro  
  AFTER INSERT OR UPDATE OR DELETE  
  ON public."Registro_Cuenta"  
  FOR EACH ROW  
  EXECUTE PROCEDURE public.actualizartotaldebehaber();
```

ANEXO VII

Plan de Pruebas

Plan de Pruebas

1. Introducción

1.1. Propósito

Definir las metas y los objetivos para las pruebas realizadas en el sistema SISCOOP, para obtener la aceptación y aprobación de las partes interesadas.

Este Plan de Pruebas para el sistema SISCOOP tiene los siguientes objetivos:

Demostrar la funcionalidad total del sistema.

- Identificar los elementos que pueden ser objetivo de las pruebas.
- Describir el tipo de pruebas que serán usadas.
- Identificar los recursos requeridos para realizar las pruebas.

1.2. Alcance

Este documento aplica en el desarrollo del sistema SISCOOP.

1.3. Referencias

- Diagramas de Casos de uso
- Especificación de requerimientos
- Gestión de Riesgos

2. Requerimientos para las Pruebas

2.1. Evaluación funcional

MÓDULO CONTABILIDAD

- ✚ Verificar Ingreso al sistema
- ✚ Verificar Ingreso de Cuentas
- ✚ Verificar Modificación de Cuentas
- ✚ Verificar Eliminación de Cuentas
- ✚ Verificar Ingreso de Registros

- ✚ Verificar Eliminación de Registros
- ✚ Verificar Modificación de Registros
- ✚ Verificar Ingreso de Cuentas Registro
- ✚ Verificar Modificación de Cuentas Registro
- ✚ Verificar Eliminación de Cuentas Registro
- ✚ Verificar Ingreso de Asientos Contables
- ✚ Verificar Modificación de Asientos Contables
- ✚ Verificar Eliminación de Asientos Contables
- ✚ Verificar Ingreso de Fechas para generar Reportes
- ✚ Verificar Creación de Reportes

3. Desarrollo de Pruebas

Pruebas Funcionales

Estas pruebas se centran principalmente en evaluar lo que se espera de un módulo o sección específica de un software, es decir es una manera de encontrar casos específicos en ese módulo. Estas pruebas solamente se limitan en probar con datos de entrada y estudiar como salen, sin preocuparse de lo que ocurre en el interior.

Se realizarán pruebas para la autenticación del administrador y el ingreso de fecha de inicio y fin para generar los reportes.

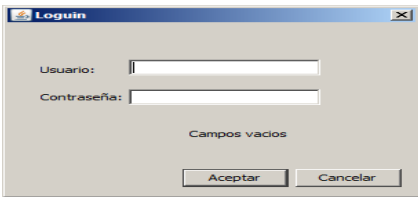
Caso de Prueba 1: Autenticación de un usuario administrador

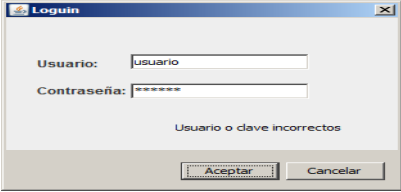
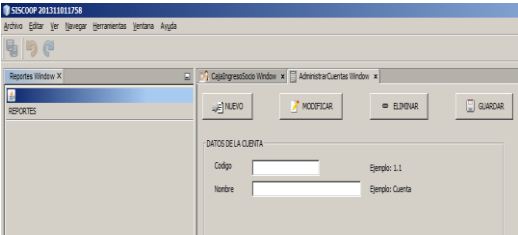
En la siguiente tabla se muestra el proceso para la autenticación del administrador.

Referencia	Ingreso al sistema	
Objetivo	Controlar el ingreso al sistema con las credenciales del administrador.	
Descripción	El sistema mediante la interfaz de autenticación permitirá o denegara el acceso al mismo mediante la introducción de usuario y clave.	
Procesos	No	Acción
	1	Ejecutar la aplicación
	2	En la sección de autenticación ingresar el usuario y contraseña.
	3	Al seleccionar "Aceptar", verifica la existencia del usuario mediante el uso de la base de datos.
	4	En caso de existir el usuario, la aplicación estará disponible para el usuario con sus respectivos permisos.
Excepciones	No	Acción
	1	Si el usuario y contraseña no corresponde con los datos de la tabla usuarios en la BD, muestra un

		mensaje de error, solicitando de nuevo el ingreso de usuario y contraseña.
	2	Antes de enviar los datos, la interfaz evalúa que los campos usuario y contraseña no estén vacíos en caso de existir solicita con un mensaje el ingreso de los datos.

En la siguiente tabla se describen las pruebas funcionales que se realizaron. Para esta prueba se lo realizó con los datos del administrador. Por seguridad no se muestra los valores para esta prueba.

Credenciales de Prueba	Usuario: usuario
	Contraseña: contraseña
Mensajes Obtenidos	
Completa este campo.- cuando el usuario no ingresa el dato en un campo.	

<p>Datos Incorrectos.- cuando el usuario y la contraseña no corresponde con la BD, es decir el usuario no existe.</p>	
<p>Resultados Obtenidos</p>	
<p>Si las credenciales corresponden con los datos de la BD, se muestra la interfaz principal de la aplicación.</p>	

En la siguiente tabla se muestra el resultado del caso de prueba 1.

Referencia	Resultado
Fecha y Hora	07/01/2014 12:30
Conclusión	El sistema cuenta con todos los controles y mensajes claros para la validación de un usuario.

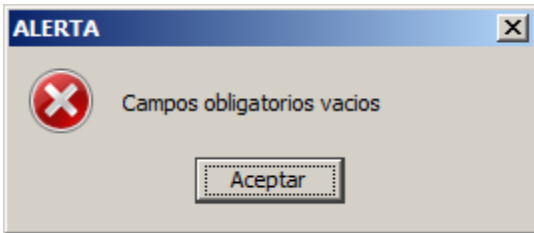
Caso de Prueba 2: Verificar el ingreso de Cuentas

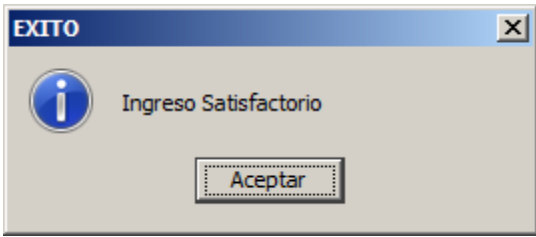
En la siguiente tabla se muestra el proceso para el ingreso de cuentas.

Referencia	Ingreso de Cuentas	
Objetivo	Controlar el ingreso correcto de cuentas al sistema	
Descripción	El sistema mediante la interfaz de ingreso de cuentas permitirá ingresar el código y el nombre de la cuenta.	
Procesos	No	Acción
	1	Ejecutar la aplicación
	2	En la sección de autenticación ingresar el usuario y contraseña.
	3	Al seleccionar "Aceptar", verifica la existencia del usuario mediante el uso de la base de datos.
	4	En caso de existir el usuario, la aplicación estará disponible para el usuario con sus respectivos permisos.
	5	Seleccionar en el menú principal la interfaz Admionistrar cuentas.
	6	Seleccionar el botón Nuevo e ingresar los valores en los campos correspondientes

	No	Acción
Excepciones	1	Si el valor del campo código ya existe se produce un error de clave duplicada.
	2	Antes de guardar los datos, la interfaz evalúa que los campos código y nombre no estén vacíos en caso de existir solicita con un mensaje de ingreso de los datos.

En la siguiente tabla se describen las pruebas funcionales que se realizaron. Para esta prueba se lo realizó con los datos del administrador. Por seguridad no se muestra los valores para esta prueba.

Credenciales de Prueba	Usuario: usuario
	Contraseña: contraseña
Mensajes Obtenidos	
Campos vacíos.- cuando el usuario no ingresa el dato en un campo.	

Resultados Obtenidos	
Si los datos son ingresados correctamente se emite un mensaje de verificación.	

En la siguiente tabla se muestra el resultado del caso de prueba 1.

Referencia	Resultado
Fecha y Hora	07/01/2014 14:30
Conclusión	El sistema cuenta con todos los controles y mensajes claros para el ingreso de cuentas al sistema.

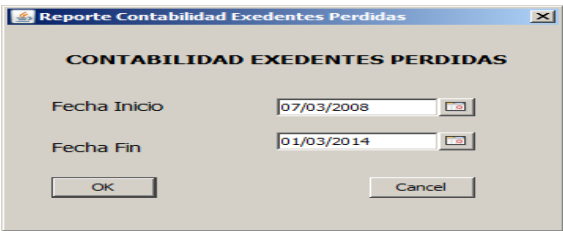
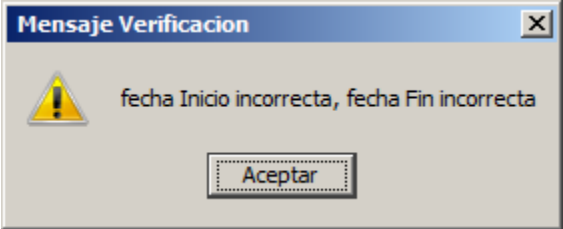
Caso de Prueba 3: Generacion de reportes

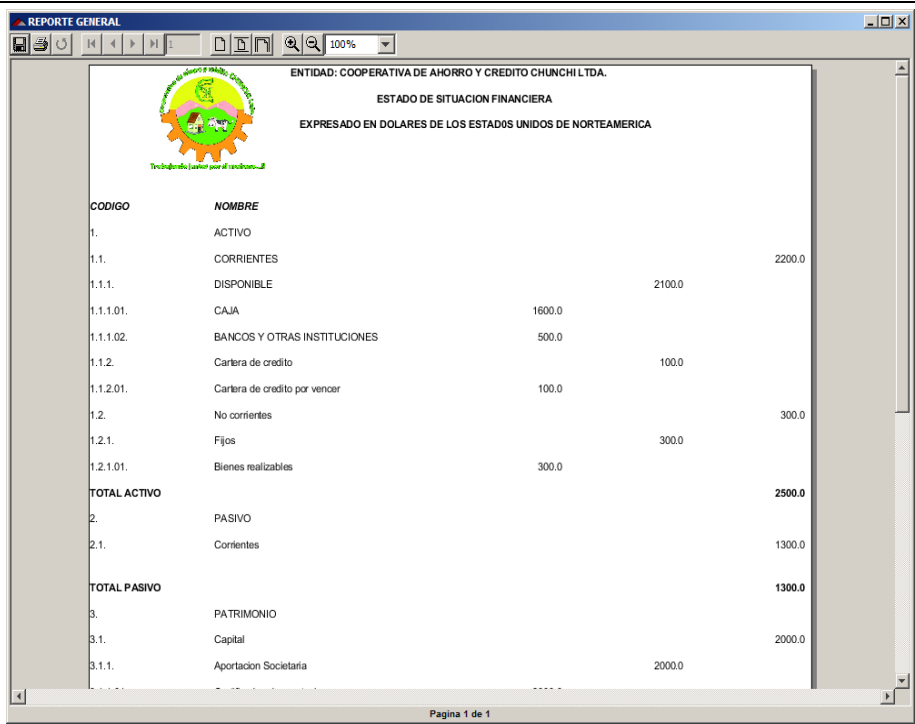
En la siguiente tabla se muestra el proceso para generar los reportes entre una fecha de inicio y una fecha final.

Referencia	Ingreso de fechas para generar reportes
Objetivo	Controlar el ingreso de fechas para generar los reportes.

Descripción	El sistema verifica que el valor de la fecha de inicio no sea mayor a la fecha final ó mayor a la fecha actual; además controla que la fecha final no sea menor que la fecha de inicio ó mayor que la fecha actual.	
Procesos	No	Acción
	1	Ejecutar la aplicación.
	2	Ingresar al sistema con el usuario respectivo.
	3	Seleccionar la opción "Reportes".
	4	Seleccionar el nombre del reporte.
	5	Ingresar la fecha de inicio y la fecha final.
	6	Aceptar la operación.
Excepciones	No	Acción
	1	Si la fecha de inicio es mayor a la fecha final o a la fecha actual se muestra un mensaje de error.
	2	Si la fecha final es mayor que la fecha de inicio o a la fecha actual, se muestra un mensaje de error.

En la siguiente tabla se describen las pruebas funcionales que se realizaron al caso de prueba 2. Para esta prueba se lo realizó con los datos del Director de la tesis. Por seguridad no se muestran los valores para realizar esta prueba.

<p>Credenciales de Prueba</p>	<p>Usuario: usuario</p> <hr/> <p>Contraseña: contraseña</p>
<p>Mensajes Obtenidos</p>	
<p>Los valores deben estar en el rango de fechas válido.</p>	
<p>Usted no puede ingresar fechas que no estén dentro de un rango válido.</p>	
<p>Resultados Obtenidos</p>	

<p>Si los valores de las fechas están dentro del rango válido, se genera el reporte seleccionado</p>	 <p>The screenshot shows a financial statement window. At the top, it says 'REPORTE GENERAL' and 'ENTIDAD: COOPERATIVA DE AHORRO Y CREDITO CHUNCHI LTDA.'. Below that, it says 'ESTADO DE SITUACION FINANCIERA' and 'EXPRESADO EN DOLARES DE LOS ESTADOS UNIDOS DE NORTEAMERICA'. The main content is a table with columns 'CODIGO' and 'NOMBRE'. The table lists various financial items and their values. At the bottom, it says 'Pagina 1 de 1'.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>CODIGO</th> <th>NOMBRE</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>ACTIVO</td> <td></td> </tr> <tr> <td>1.1.</td> <td>CORRIENTES</td> <td>2200.0</td> </tr> <tr> <td>1.1.1.</td> <td>DISPONIBLE</td> <td>2100.0</td> </tr> <tr> <td>1.1.1.01.</td> <td>CAJA</td> <td>1600.0</td> </tr> <tr> <td>1.1.1.02.</td> <td>BANCOS Y OTRAS INSTITUCIONES</td> <td>500.0</td> </tr> <tr> <td>1.1.2.</td> <td>Cartera de credito</td> <td>100.0</td> </tr> <tr> <td>1.1.2.01.</td> <td>Cartera de credito por vencer</td> <td>100.0</td> </tr> <tr> <td>1.2.</td> <td>No corrientes</td> <td>300.0</td> </tr> <tr> <td>1.2.1.</td> <td>Fijos</td> <td>300.0</td> </tr> <tr> <td>1.2.1.01.</td> <td>Bienes realizables</td> <td>300.0</td> </tr> <tr> <td colspan="2">TOTAL ACTIVO</td> <td>2500.0</td> </tr> <tr> <td>2.</td> <td>PASIVO</td> <td></td> </tr> <tr> <td>2.1.</td> <td>Corrientes</td> <td>1300.0</td> </tr> <tr> <td colspan="2">TOTAL PASIVO</td> <td>1300.0</td> </tr> <tr> <td>3.</td> <td>PATRIMONIO</td> <td></td> </tr> <tr> <td>3.1.</td> <td>Capital</td> <td>2000.0</td> </tr> <tr> <td>3.1.1.</td> <td>Aportacion Societaria</td> <td>2000.0</td> </tr> </tbody> </table>	CODIGO	NOMBRE	Valor	1.	ACTIVO		1.1.	CORRIENTES	2200.0	1.1.1.	DISPONIBLE	2100.0	1.1.1.01.	CAJA	1600.0	1.1.1.02.	BANCOS Y OTRAS INSTITUCIONES	500.0	1.1.2.	Cartera de credito	100.0	1.1.2.01.	Cartera de credito por vencer	100.0	1.2.	No corrientes	300.0	1.2.1.	Fijos	300.0	1.2.1.01.	Bienes realizables	300.0	TOTAL ACTIVO		2500.0	2.	PASIVO		2.1.	Corrientes	1300.0	TOTAL PASIVO		1300.0	3.	PATRIMONIO		3.1.	Capital	2000.0	3.1.1.	Aportacion Societaria	2000.0
CODIGO	NOMBRE	Valor																																																					
1.	ACTIVO																																																						
1.1.	CORRIENTES	2200.0																																																					
1.1.1.	DISPONIBLE	2100.0																																																					
1.1.1.01.	CAJA	1600.0																																																					
1.1.1.02.	BANCOS Y OTRAS INSTITUCIONES	500.0																																																					
1.1.2.	Cartera de credito	100.0																																																					
1.1.2.01.	Cartera de credito por vencer	100.0																																																					
1.2.	No corrientes	300.0																																																					
1.2.1.	Fijos	300.0																																																					
1.2.1.01.	Bienes realizables	300.0																																																					
TOTAL ACTIVO		2500.0																																																					
2.	PASIVO																																																						
2.1.	Corrientes	1300.0																																																					
TOTAL PASIVO		1300.0																																																					
3.	PATRIMONIO																																																						
3.1.	Capital	2000.0																																																					
3.1.1.	Aportacion Societaria	2000.0																																																					

En la siguiente tabla se muestra el resultado del caso de prueba 2.

Referencia	Resultado
Fecha y Hora	20/01/2014 14:00
Conclusión	El sistema valida el ingreso de fechas de inicio y fecha final para generar los distintos reportes en los que se solicita ingresar dichos valores, cumpliéndose así en su totalidad con el requerimiento.

4. Herramientas

Microsoft Word

Microsoft Excel

5. Recursos

En esta sección se presenta los recursos principales para las pruebas:

Recursos Humanos		
Rol	Recursos necesarios	Responsabilidades específicas
Administrador de las Pruebas	José Manzano	Proporcionar la dirección técnica. Adquirir los recursos adecuados. Presentación de informes de gestión.
Diseñador de las Pruebas	Ángel Cobo	Identificar, priorizar, y ejecutar los casos de prueba, generar el plan de pruebas.
Evaluador del Sistema SISCOOP	Ángel Cobo José Manzano	Ejecutar las pruebas. Registrar los resultados. Recuperarse de los errores. Documento defectos.

ANEXO VIII

Manual de Usuario



ESPOCH
FACULTAD DE INFORMÁTICA
Y
ELE

TRÓNICA



MANUAL
DE
USUARIO

ÍNDICE

1. INTRODUCCIÓN	315 -
2. REQUISITOS.....	315 -
2.1. Requisitos del sistema	315 -
2.2. Pasos de la Instalación	315 -
3. EJECUCIÓN	316 -
3.1. Ejecución desde el CD	316 -
3.2. Ejecución desde la Instalación Previa.	316 -
4. INSTRUCCIONES	316 -
4.1. Guía de instrucciones	316 -
4.2. Navegación del Software	316 -
4.2.1. Presentación	316 -
5. CONTENIDO.....	317 -
5.1. Menú Principal de Opciones.....	317 -
6. REPORTES	327 -
7. RECOMENDACIONES DE USO.....	329 -
8. SERVICIOS AL CLIENTE.....	330 -

1. INTRODUCCIÓN

Este sistema es un software elaborado para promocionar ayuda a los usuarios del producto y servicios que provee el sistema "SISCOOP", diseñado para el fácil manejo del cliente. Este documento facilitará al usuario el uso de las herramientas necesarias; con el fin de manejar correctamente el software que tengan a su disposición. Aquí aprenderá como comunicarse con las interfaces que se encuentran a su disposición. Les ofrecemos la forma más fácil e intuitiva para solucionar con eficacia y eficiencia cualquier problema que se les presente al usar este software.

2. REQUISITOS

2.1. Requisitos del sistema

Los siguientes requisitos son considerados como mínimos:

Requisitos recomendados:

HARDWARE:

- Procesador Intel Core i7
- Memoria RAM: 4GB
- Espacio libre en disco: 50 GB
- Configuración colores del monitor: 24 bits

SOFTWARE

- Los requisitos de software necesarios para la aplicación son:
- Sistemas Operativos Windows XP SP2, Windows Vista, Windows 7.
- Video: mínimo: 32 MB
- Sonido: Tarjeta compatible
- Hardware Adicional: Lector de Discos digitales CD-ROM/CD-RW, RW/DVD-ROM/DVD-RW, Parlantes.

2.2. Pasos de la Instalación

Para un apropiado y rápido proceso de instalación se recomienda siga cuidadosamente los siguientes pasos:

- ✓ Coloque el CD en la bandeja de CD-ROM/CD-RW/DVD-ROM/DVD-RW
- ✓ Explore MI-PC y abrir la unidad especificada en paso 1.
- ✓ Ejecutar el archivo siscoop.exe de la Carpeta Siscoop

3. EJECUCIÓN

3.1. Ejecución desde el CD

Poner el CD en la bandeja de CD-ROM/CD-RW

Seleccionar el ejecutable que es un icono con una letra f de color celeste

3.2. Ejecución desde la Instalación Previa.

Poner el CD en la bandeja de CD-ROM/CD-RW

Acceder al escritorio

Hacer clic en el Acceso directo "SISCOOP"

4. INSTRUCCIONES

4.1. Guía de instrucciones

Para iniciar la aplicación el usuario debe dar doble clic sobre el ejecutable.

Para dirigirse a la página que etiqueta el botón debe dar clic sobre el mismo.

La pantalla principal del sistema aparecerá automáticamente. Una vez iniciado se presentará la interfaz principal, donde encontraremos el menú.

4.2. Navegación del Software

4.2.1. Presentación

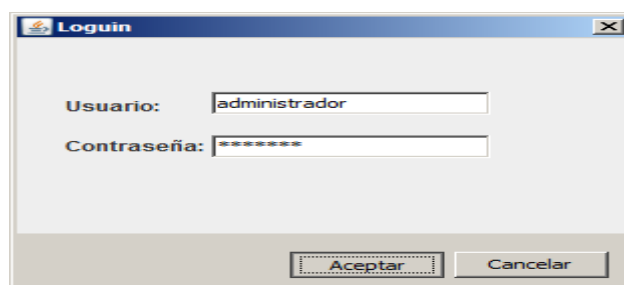
Muestra la representación de la aplicación de **SISCOOP** Versión 1.0, Aparecerá la siguiente pantalla la cual al momento de cargarse presentara la interfaz inicial de nuestro sistema.



5. CONTENIDO

5.1. Pantalla de Inicio de Sesión

A continuación se observara la pantalla de inicio de sesión, en la misma usted debe introducir el usuario y la contraseña para poder acceder al sistema.



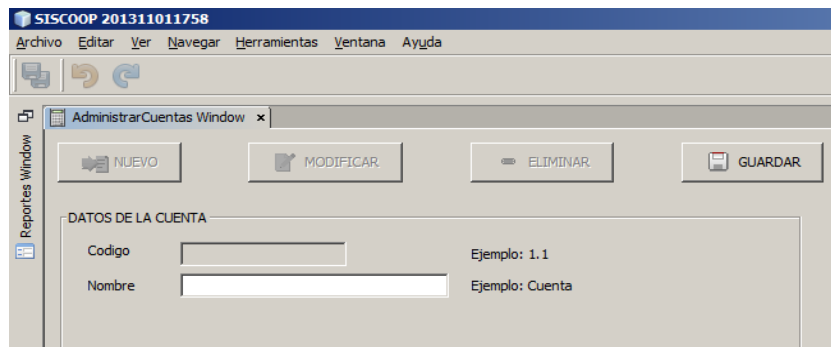
5.2. Menú Principal de Opciones

El menú muestra las siguientes opciones principales como:

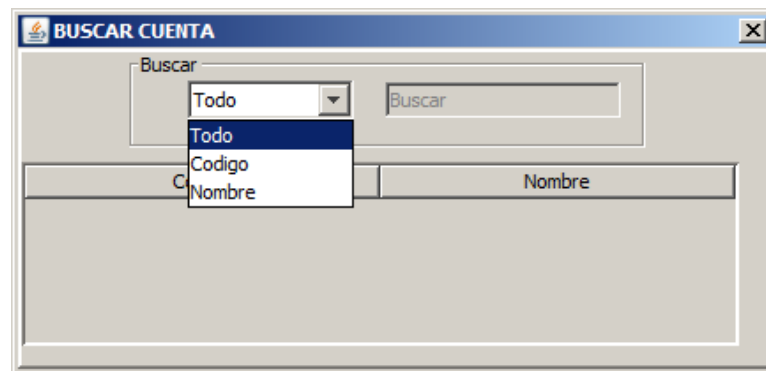
- Administrar Cuentas
- Administrar Registros
- Administrar Cuentas Registros
- Caja Ingreso Socio
- Cuenta Socio
- Usuarios
- Reportes

Una vez que haya ingresado a la interfaz de **INICIO** usted puede escoger la opción que desee. Si usted escogió la opción de **Administra Cuentas** se le presentara la siguiente pantalla con los siguientes campos:

- ✚ Código
- ✚ Nombre

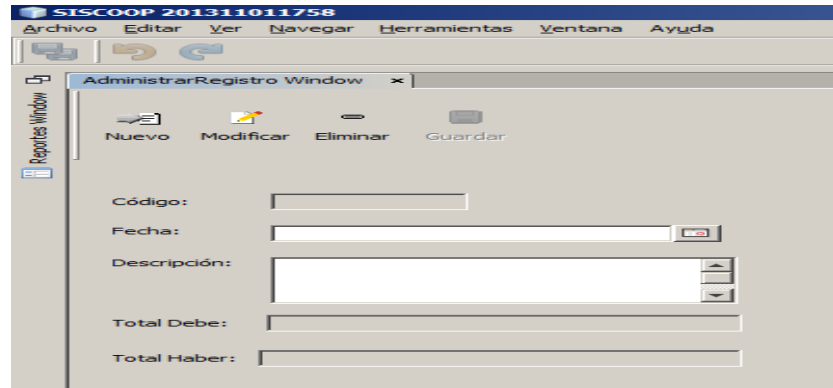


Dentro de esta interfaz existen las opciones de **Nuevo, Modificar, Eliminar y Guardar**. Para las opciones de Modificar y Eliminar usted cuenta con la opción de búsqueda la misma que puede hacer de tres formas: filtradas por código, filtradas por nombre ó visualizar todas. En la siguiente figura se observa la interfaz de búsqueda de cuentas.



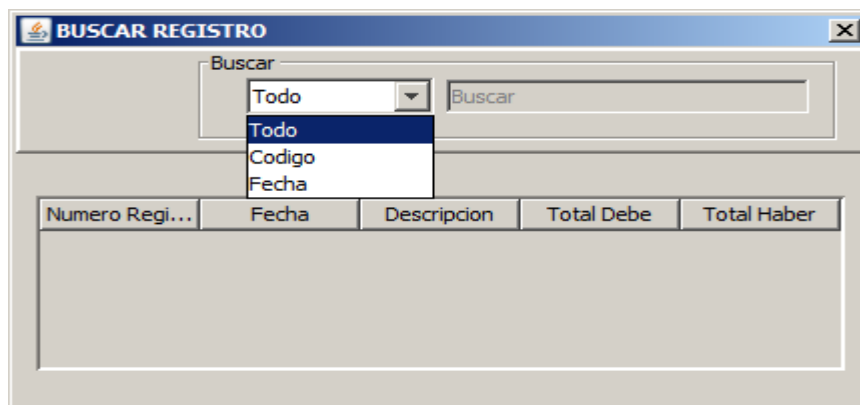
Si usted escogió la opción de **Administrar Registros** se le mostrara la siguiente pantalla la misma que consta con los siguientes campos:

- ✚ Código
- ✚ Fecha
- ✚ Descripción
- ✚ Total Debe
- ✚ Total Haber






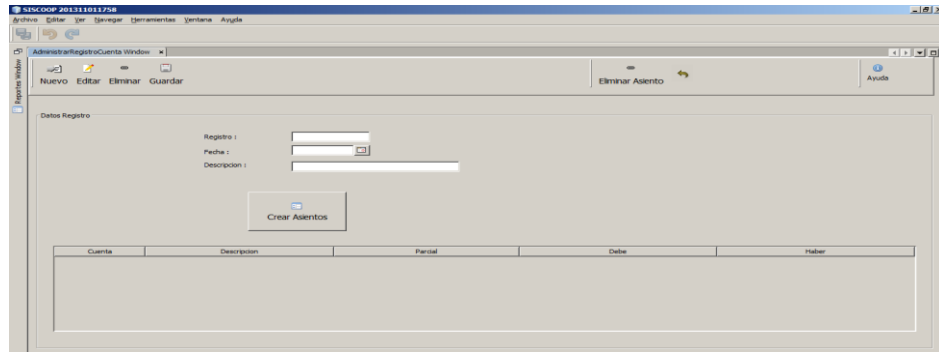
Dentro de esta interfaz se encuentran las opciones de **Nuevo, Modificar, Eliminar y Guardar**. Para las opciones de Modificar y Eliminar usted podrá seleccionar el registro buscándolo de tres formas diferentes: filtrado por código, filtrado por fecha y visualizar todo.

En la siguiente figura se observa la interfaz de búsqueda de registros.

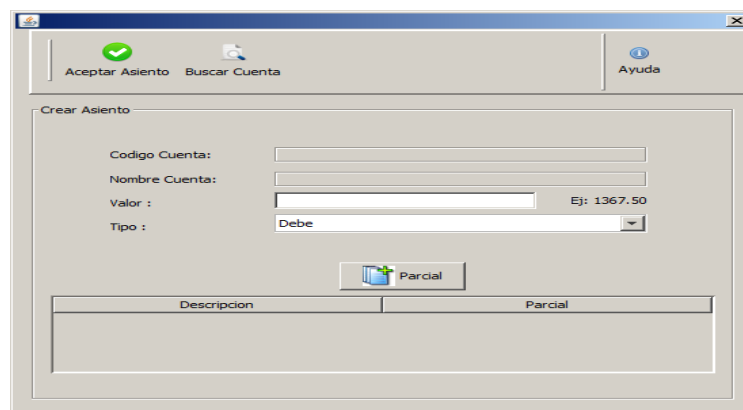


Si la opción escogida fue **Administrar Cuenta Registro** se le mostrara la siguiente pantalla y sus campos son:

-  Registro
-  Fecha
-  Descripción



Dentro de esta interfaz se encuentra la opción de crear asientos contables en la siguiente figura se visualiza la interfaz de dicha opción.

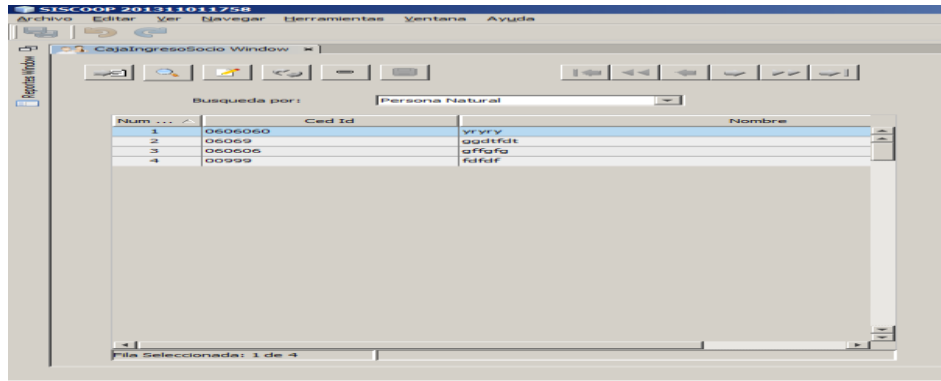


Dentro de la interfaz de creación de asientos contables se cuenta con los siguientes campos:

- ✚ Valor
- ✚ Tipo

Si escoge la opción **Caja Ingreso Socio** se mostrara la siguiente pantalla con los siguientes campos.

- ✚ Num Socio
- ✚ Ced ID
- ✚ Nombre



Dentro de esta interfaz al dar doble click sobre cualquier fila de la tabla se presenta la interfaz que servirá para edita o ingresar nuevos socios. En la siguiente figura se visualiza la interfaz llamada Datos Generales que realiza esta operación.

The screenshot shows the 'Datos Generales' form with the following fields and values:

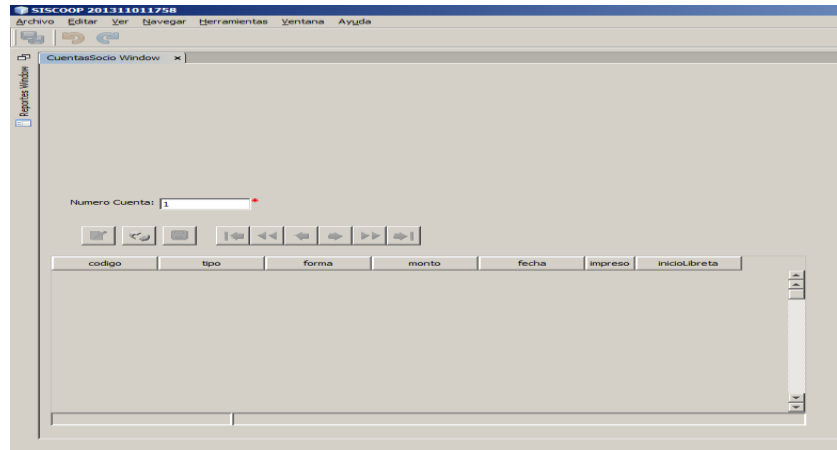
- Names:** yryy
- Cedula:** 0606060
- Celular:** 0987575
- RUC:** 0004002
- Estado:** Activo
- Estado Civil:** Soltero
- Cargas Famil.** 3
- Nivel Instruc.** Basica
- Fecha Nac.** 1988/12/13
- Lugar Nac.** riobamba
- Domicilio:**
 - Direccion:** espejo
 - Sitio referencia:** fdfd
 - Telefono fijo:** 0329987
- Trabajo:**
 - Actividad:** profe
 - Direccion:** guayas
 - Telefono:** 098747

La siguiente interfaz llamada Inf. Representante contiene todos los campos necesarios para ingresar los datos del representante de un socio. En la siguiente figura se visualiza la interfaz Inf. Representante.

La siguiente interfaz llamada Datos del Conyugue contiene todos los campos necesarios para ingresar la información del Conyugue de un socio en particular. En la siguiente figura se visualiza la interfaz Datos Conyugue.

Si escoge la opción **Cuenta Socio** se mostrara la siguiente pantalla con los siguientes campos:




- ✚ Código
- ✚ Tipo
- ✚ Forma
- ✚ Monto
- ✚ Fecha
- ✚ Impreso
- ✚ Inicio Libreta






MODULO DE GESTIÓN USUARIOS

Interfaces a utilizarse en el modulo de Gestión de Usuarios

Interfaz grafica de **Registro de Usuario** es la nos permite el registro de usuarios además elegir el rol que puede ser:

-  Administrador
-  Usuario
-  Invitado

Y dependiendo del Rol que escoja obtiene los respectivos permisos como;

-  Lectura
-  Escritura
-  Control Total

REGISTRO DE USUARIOS

Nombre de Usuario

Contraseña

Seleccione el Rol

Nombre de Rol

Descripcion

Seleccione los Permisos

Nombre de Rol Lectura Escritura Control Total

Descripcion

Luego de haberse registrado se inicia la sesión como se muestra en la siguiente pantalla.

INICIAR SESIÓN

Nombre de Usuario

Contraseña

NOTA: Se debe recalcar que el menú de opciones de la aplicación que se encontraba en Inglés, ahora se la cambio a español para facilitar al usuario la navegación en el sistema

Roles-Permisos | Permisos | Roles-Permisos | Usuarios

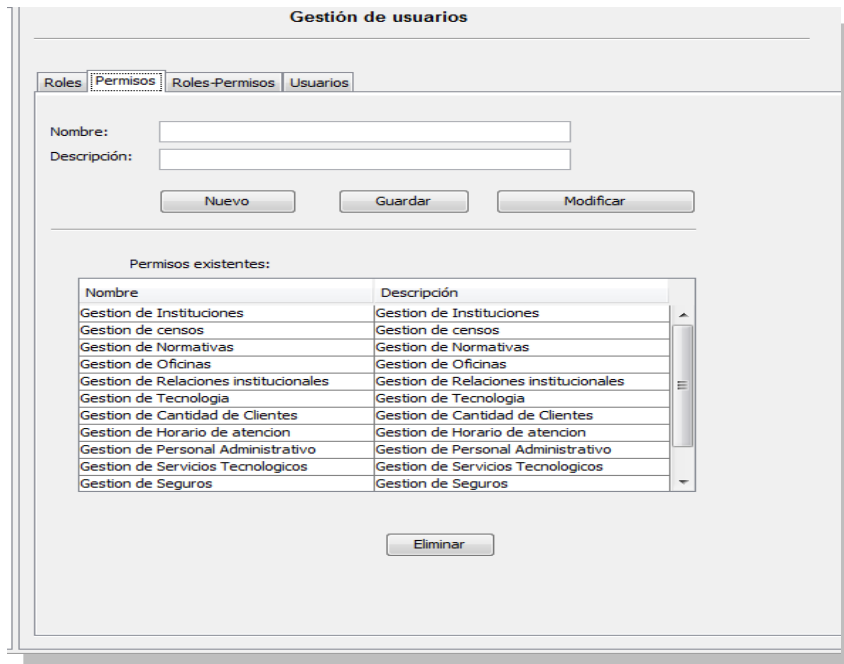
Nombre:

Descripción:

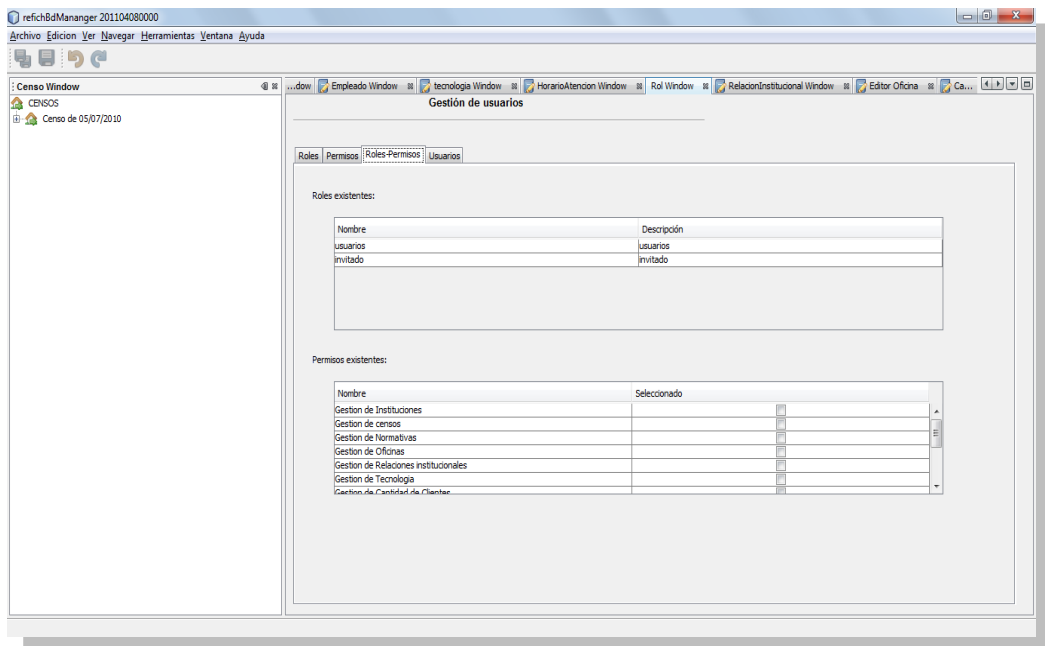
Roles existentes:

Nombre	Descripción
usuarios	usuarios
invitado	invitado

Una vez asignados los roles y los distintos usuarios, también se le debe asignar permisos como podemos ver en la pantalla siguiente.



También existe la opción de Roles- Permisos como podemos observar en la pantalla siguiente, esta opción permite al administrador seleccionar los diferentes permisos y roles para cada usuario del sistema.



En la opción de usuarios le permite al administrador poner nombre de usuario y contraseña y el rol que se le asigno con anterioridad y tambien podemos observar los roles secundarios que se le fueron asignados.

The screenshot shows a web interface titled "Gestión de usuarios". At the top, there are tabs for "Roles", "Permisos", "Roles-Permisos", and "Usuarios", with "Usuarios" selected. Below the tabs, there are three input fields: "Usuario:" with the value "user2", "Contraseña:" (empty), and "Rol:" with a dropdown menu showing "usuarios". Below these fields are three buttons: "Nuevo", "Guardar", and "Modificar".

Below the input fields, there is a section titled "Roles existentes:" containing a table with the following data:

Usuario	Contraseña	Rol
user 1	202cb962ac59075b964b...	usuarios
user 2	202cb962ac59075b964b...	usuarios

Below the table is an "Eliminar" button.

Una vez que el Administrador asigna los permisos y roles aparecerá la siguiente pantalla

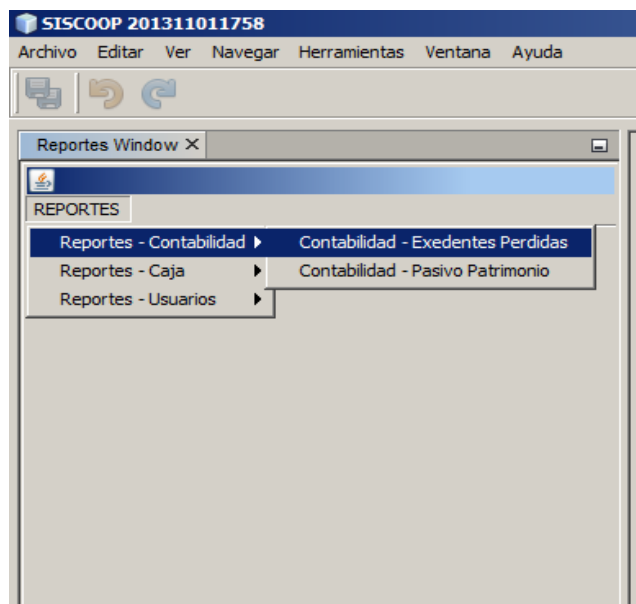
The screenshot shows the same "Gestión de usuarios" interface. The input fields are now empty. The "Rol:" dropdown menu still shows "usuarios". The "Nuevo" button is now disabled, and the "Modificar" and "Guardar" buttons are enabled. The table below now contains two rows of user data:

Usuario	Contraseña	Rol
user 1	202cb962ac59075b964b07152d234b70	usuarios
user 2	202cb962ac59075b964b07152d234b70	usuarios

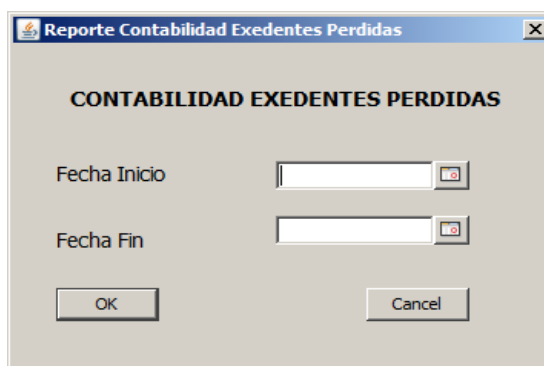
The "Eliminar" button is still present at the bottom.

6. REPORTE

En la siguiente figura se observa la interfaz principal para generar los distintos reportes que serán generados en el sistema.



La aplicación es capaz de emitir reportes para ello se presentara la siguiente pantalla para el administrador el cual debe ingresar la fecha de Inicio y la Fecha fin para poder realizar dichos reporte, además tiene una descripción y para poder obtener los reportes debemos presionar la botón OK (Aceptar). En la siguiente figura se observa la interfaz de ingreso de fecha de Inicio y fecha Final, en el caso que el rango de fechas no sea valido se emitirá un mensaje de error.

The image shows a dialog box titled "Reporte Contabilidad Exedentes Perdidas". The dialog box has a title bar with a close button. The main content area is titled "CONTABILIDAD EXEDENTES PERDIDAS". Below the title, there are two input fields: "Fecha Inicio" and "Fecha Fin". Each input field has a small calendar icon to its right. At the bottom of the dialog box, there are two buttons: "OK" and "Cancel".

La pantalla siguiente nos presenta un reporte denominado "Exedentes - Perdidas", en este reporte se encuentra el logo de la Cooperativa, el nombre de la misma y la fecha de generación del reporte.

REPORTE GENERAL

ESTADO DE EXCEDENTES O PERDIDAS
COOPERATIVA DE AHORRO Y CREDITO CHUNCHI LTDA.

Trabajando (verificar por el sistema...)

DEL 14 marzo 2008 AL 06 marzo 2014

CODIGO	NOMBRE	14 marzo 2008	AL	06 marzo 2014
4.1.	Ingresos Operacionales			6000.0
4.1.3.	Por interes y descuentos ganados			6000.0
4.1.3.01.	Interes en cartera de credito por vencer	3000.0		
4.1.3.02.	Interes en cartera de credito por vencer	3000.0		
5.1.	Gastos operacionales			3000.0
5.1.4.	Gastos de personal			1500.0
5.1.4.01.	Obligaciones con el publico	500.0		
5.1.4.02.	Obligaciones financieras	500.0		
5.1.4.03.	Otros interes causados	500.0		
5.1.5.	Gastos generales			1500.0
5.1.5.01.	Obligaciones financieras	1500.0		
UTILIDAD DE OPERACION				3000.0
4.2.	Ingresos no operacionales			18444.0
4.2.1.	Por utilidades financieras			18000.0
4.2.1.01.	En venta de activos	9000.0		
4.2.1.05.	Otras utilidades financieras	9000.0		
4.2.2.	Por servicios no operacionales			444.0

Pagina 1 de 1

En la siguiente figura se observa el reporte de Estado de Situación Financiera, para generar este reporte se debe especificar la fecha de inicio y la fecha final, si el rango de fechas no es válido se emitirá un mensaje de error.

REPORTE GENERAL

ENTIDAD: COOPERATIVA DE AHORRO Y CREDITO CHUNCHI LTDA.

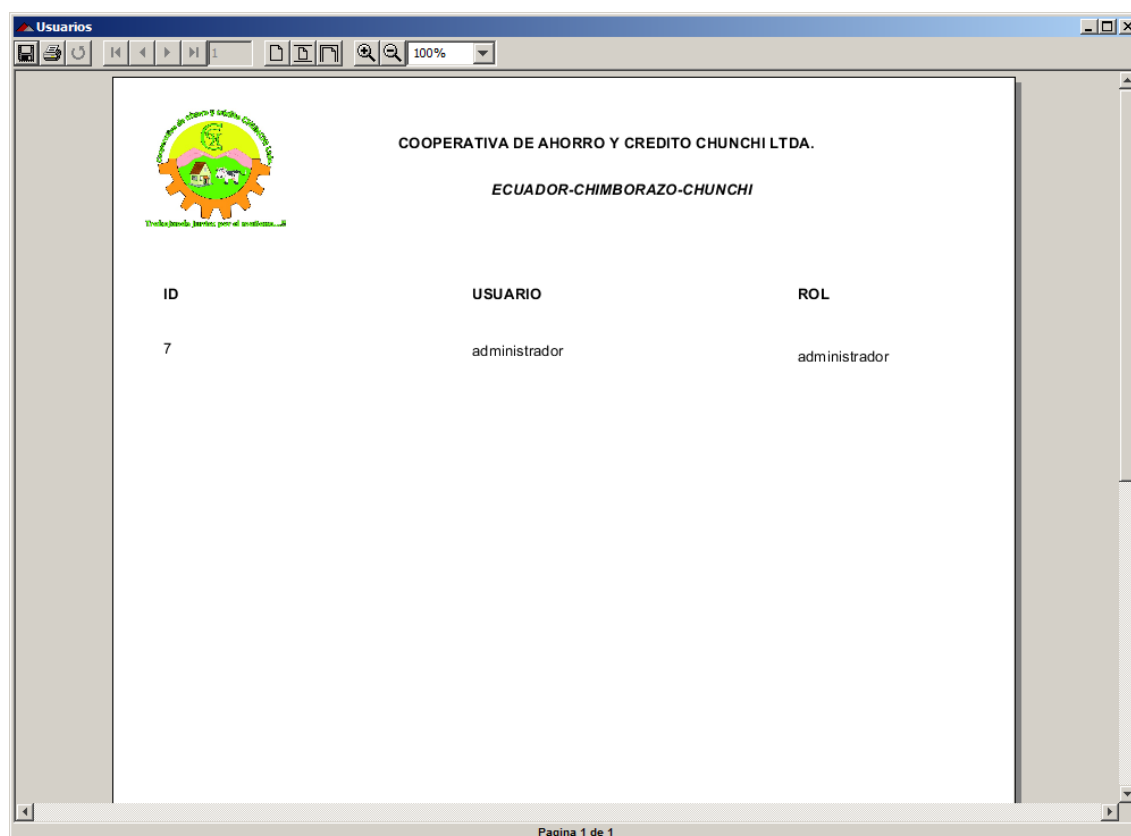
ESTADO DE SITUACION FINANCIERA
EXPRESADO EN DOLARES DE LOS ESTADOS UNIDOS DE NORTEAMERICA

Trabajando (verificar por el sistema...)

CODIGO	NOMBRE		
1.	ACTIVO		
1.1.	CORRIENTES		2200.0
1.1.1.	DISPONIBLE		2100.0
1.1.1.01.	CAJA	1600.0	
1.1.1.02.	BANCOS Y OTRAS INSTITUCIONES	500.0	
1.1.2.	Cartera de credito		100.0
1.1.2.01.	Cartera de credito por vencer	100.0	
1.2.	No corrientes		300.0
1.2.1.	Fijos		300.0
1.2.1.01.	Bienes realizables	300.0	
TOTAL ACTIVO			2500.0
2.	PASIVO		
2.1.	Corrientes		1300.0
TOTAL PASIVO			1300.0
3.	PATRIMONIO		
3.1.	Capital		2000.0
3.1.1.	Aportacion Societaria		2000.0

Pagina 1 de 1

En la siguiente figura se observa el reportes de usuarios existente actualmente en le sistema, en este se especifica el ID que identifica al usuario y que es único, el Usuario que es el nombre con cual ingresara al sistema y el rol que tiene ese usuario. Por motivos de seguridad no se visualiza el campo Contraseña del usuario.



ID	USUARIO	ROL
7	administrador	administrador

7. RECOMENDACIONES DE USO

En caso de que usted no sea un usuario experto se recomienda que lea este manual, con la finalidad de tener una idea clara de las funciones que puede realizar el sistema. Si tiene dificultades en el manejo, no funciona correctamente el software comuníquese directamente con los desarrolladores, quienes podrán dirigirse para solucionar los posibles problemas.

8. SERVICIOS AL CLIENTE

Quienes formamos parte del equipo de desarrollo estamos siempre dispuestos a resolver problemas, dudas de los usuarios y también a recibir consejos por parte de los mismos, para cualquier consulta o información de soporte técnico.

Usted puede contactarnos de la siguiente manera:

Llamando al número telefónico (celular): 0986944180

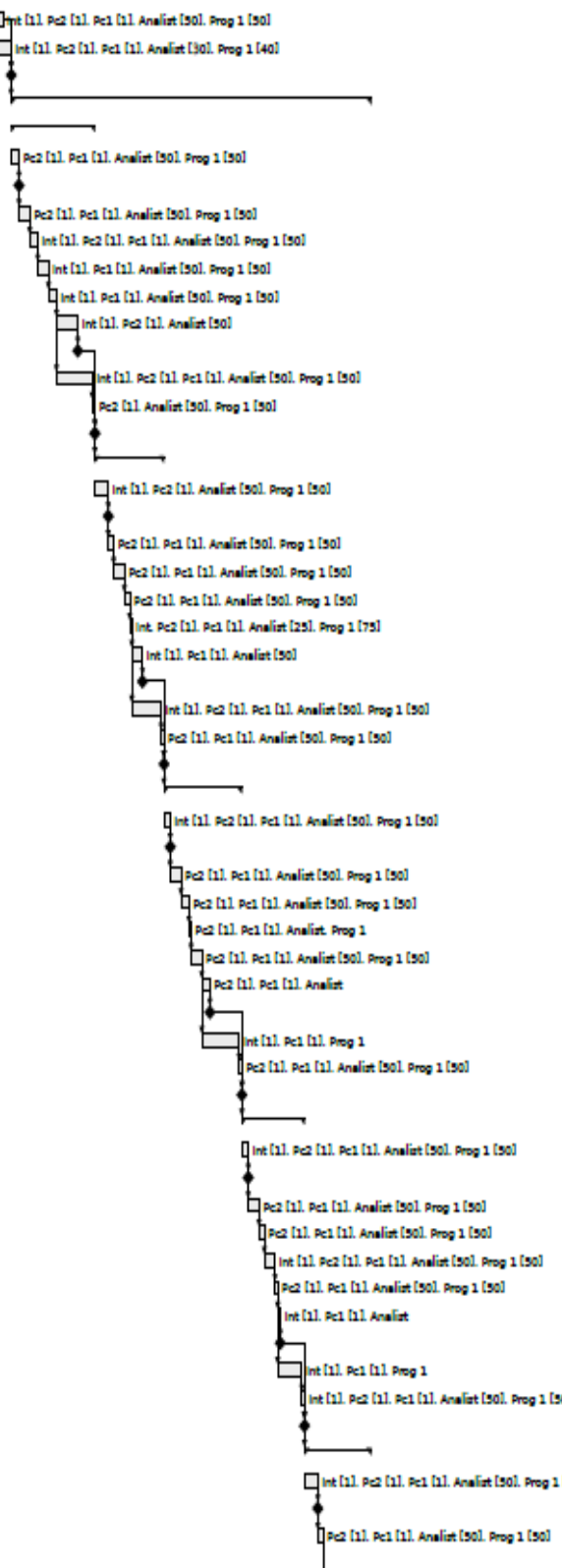
También enviando un e-mail a siscoop@yahoo.com.es

Usted recibirá una solución más detallada a su problema particular o pregunta.

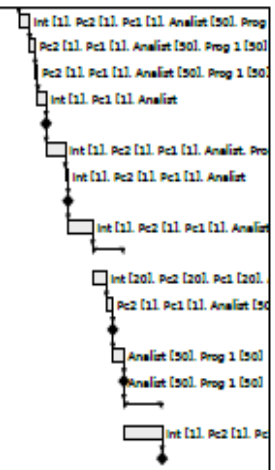
ANEXO IX

Planificación

Name	Work	2013, H1							
		Nov 2012	Dec 2012	Jan 2013	Feb 2013	Mar 2013	Apr 2013	May 2013	...
Inicio									
Definición	35d								
Definición y Especificación de Requerimientos	20d								
Análisis y selección de Herramientas en software Libre	15d								
Documento con la especificación de los requerimientos									
Desarrollo	111d...								
Módulo 1	27d								
Planificación del módulo caja	3d								
Plan del módulo caja									
Definición de los casos de uso	2d								
Definición del modelo conceptual	3d								
Definición del modelo de base de datos	2d								
Definición de laz (UI)	3d								
Definición de los reportes	3d								
Documento casos de uso, clases, diagrama BD, etc									
Implementación del módulo	10d								
Pruebas del módulo	1d								
Entrega módulo									
Módulo 2	19d								
Planificación del módulo microcrédito	3d								
Plan del módulo microcrédito									
Definición de los casos de uso	2d								
Definición del modelo conceptual	2d								
Definición del modelo de base de datos	2d								
Definición de laz (UI)	1d								
Definición de los reportes	1d								
Documento casos de uso, clases, diagrama de BD, etc									
Implementación del módulo	7d								
Pruebas del módulo	1d								
Entrega módulo									
Modulo 3	24d 7h								
Planificación del módulo clientes	3d								
Plan del módulo clientes									
Definición de los casos de uso	2d								
Definición del modelo conceptual	3d								
Definición del modelo de BD	7h 4...								
Definición de laz (UI)	2d								
Definición de los reportes	3d								
Documento casos de uso, clases, diagrama de BD, etc									
Implementación del módulo	10d								
Pruebas del módulo	1d								
Entrega del módulo									
Modulo 4	19d								
Planificación del módulo usuarios	3d								
Plan del módulo usuarios									
Definición de los casos de uso	2d								
Definición del modelo conceptual	2d								
Definición del modelo de BD	2d								
Definición de laz (UI)	1d								
Definición de los reportes	1d								
Documento casos de uso, clases, diagrama de BD, etc									
Implementación del módulo	7d								
Pruebas del módulo	1d								
Entrega del módulo usuarios									
Modulo 5	22d								
Planificación del módulo contabilidad	3d								
Plan del módulo contabilidad									
Definición de los casos de uso contabilidad	2d								



Definición del modelo conceptual contabilidad	2d
Definición del modelo de BD contabilidad	2d
Definición de las UI contabilidad	1d
Definición de los reportes contabilidad	1d
Documentos casos de uso, clases, diagrama de BD, etc	
Implementación del módulo contabilidad	10d
Pruebas del módulo contabilidad	1d
Entrega del módulo contabilidad	
Pruebas generales del sistema completo	7d
Instalación	8d
Adecuación de hardware y software	3d
Verificación y Validación	3d
Documento de V&V	
Instalación del Sistema	2d
Entrega del Sistema	
Mantenimiento	10d
Determinación y corrección de errores	10d
Fin	



WBS	Nombre	Start	Finish	Work	Duration	Slack	Cost	Assigned to
1	Inicio	Oct 15	Oct 15	N/A	N/A	33d	0	
2	Definición	Oct 15	Nov 12	33d	20d 4h	120d	1,230.61	
2.1	Definición y Especificación de Requerimientos	Oct 15	Nov 9	20d	13d 3h	33d 7h	701.77	Analist. Pc1. Pc2. Int. Prog 1
2.2	Análisis y selección de Herramientas en software Libre	Oct 15	Nov 12	15d	20d 4h	30d 6h	328.84	Analist. Pc1. Pc2. Int. Prog 1
2.3	Documento con la especificación de los requerimientos	Nov 12	Nov 12	N/A	N/A	30d 6h	0	
3	Desarrollo	Nov 12	Mar 26	111d 7h	95d 5h	24d 3h	3,963.79	
3.1	Módulo 1	Nov 12	Dec 13	27d	23d 3h	96d 5h	939.01	
3.1.1	Planificación del módulo caja	Nov 12	Nov 15	3d	2d 7h		106.28	Analist. Pc1. Pc2. Prog 1
3.1.2	Plan del módulo caja	Nov 15	Nov 15	N/A	N/A		0	
3.1.3	Definición de los casos de uso	Nov 15	Nov 15	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.1.4	Definición del modelo conceptual	Nov 15	Nov 22	3d	2d 7h		105.27	Analist. Pc1. Pc2. Int. Prog 1
3.1.5	Definición del modelo de base de datos	Nov 22	Nov 26	2d	1d 7h		70.81	Analist. Pc1. Int. Prog 1
3.1.6	Definición de las GUI	Nov 26	Nov 29	3d	2d 7h		106.26	Analist. Pc1. Int. Prog 1
3.1.7	Definición de los reportes	Nov 29	Dec 7	3d	5d 6h	4d 7h	52.84	Analist. Pc2. Int
3.1.8	Documento casos de uso, clases, diagrama BD, etc	Dec 7	Dec 7	N/A	N/A	4d 7h	0	
3.1.9	Implementación del módulo	Nov 29	Dec 12	10d	9d 5h		350.95	Analist. Pc1. Pc2. Int. Prog 1
3.1.10	Pruebas del módulo	Dec 12	Dec 13	1d	7h 36min		35.8	Analist. Pc2. Prog 1
3.1.11	Entrega módulo	Dec 13	Dec 13	N/A	N/A		0	
3.2	Módulo 2	Dec 13	Jan 8	19d	17d	70d 5h	651.69	
3.2.1	Planificación del módulo microcrédito	Dec 13	Dec 18	3d	2d 7h		106.26	Analist. Pc2. Int. Prog 1
3.2.2	Plan del módulo microcrédito	Dec 18	Dec 18	N/A	N/A		0	
3.2.3	Definición de los casos de uso	Dec 18	Dec 20	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.2.4	Definición del modelo conceptual	Dec 20	Dec 24	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.2.5	Definición del modelo de base de datos	Dec 24	Dec 26	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.2.6	Definición de las GUI	Dec 26	Dec 27	1d	3h 38min		20.9	Analist. Pc1. Pc2. Int. Prog 1
3.2.7	Definición de los reportes	Dec 27	Dec 31	1d	1d 7h	5d 6h	30.95	Analist. Pc1. Int
3.2.8	Documento casos de uso, clases, diagrama de BD, etc	Dec 31	Dec 31	N/A	N/A	5d 6h	0	
3.2.9	Implementación del módulo	Dec 27	Jan 7	7d	6d 6h		245.67	Analist. Pc1. Pc2. Int. Prog 1
3.2.10	Pruebas del módulo	Jan 7	Jan 8	1d	7h 31min		35.47	Analist. Pc1. Pc2. Prog 1
3.2.11	Entrega módulo	Jan 8	Jan 8	N/A	N/A		0	
3.3	Modulo 3	Jan 8	Feb 6	24d 7h	20d 7h	58d 5h	907.78	
3.3.1	Planificación del módulo clientes	Jan 8	Jan 10	3d	2d 7h		105.27	Analist. Pc1. Pc2. Int. Prog 1
3.3.2	Plan del módulo clientes	Jan 10	Jan 10	N/A	N/A		0	
3.3.3	Definición de los casos de uso	Jan 10	Jan 14	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.3.4	Definición del modelo conceptual	Jan 14	Jan 17	3d	2d 7h		106.28	Analist. Pc1. Pc2. Prog 1
3.3.5	Definición del modelo de BD	Jan 17	Jan 18	7h 4min	3h 30min		31.54	Analist. Pc1. Pc2. Prog 1
3.3.6	Definición de las GUI	Jan 18	Jan 22	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.3.7	Definición de los reportes	Jan 22	Jan 25	3d	2d 7h	7d 6h	54.5	Analist. Pc1. Pc2
3.3.8	Documento casos de uso, clases, diagrama de BD, etc	Jan 25	Jan 25	N/A	N/A	7d 6h	0	
3.3.9	Implementación del módulo	Jan 22	Feb 5	10d	9d 6h		393.05	Pc1. Int. Prog 1
3.3.10	Pruebas del módulo	Feb 5	Feb 6	1d	7h 31min		35.47	Analist. Pc1. Pc2. Prog 1
3.3.11	Entrega del módulo	Feb 6	Feb 6	N/A	N/A		0	
3.4	Modulo 4	Feb 6	Mar 1	19d	17d 4h	41d	694.48	
3.4.1	Planificación del módulo usuarios	Feb 6	Feb 8	3d	2d 7h		105.27	Analist. Pc1. Pc2. Int. Prog 1
3.4.2	Plan del módulo usuarios	Feb 8	Feb 8	N/A	N/A		0	
3.4.3	Definición de los casos de uso	Feb 8	Feb 12	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.4.4	Definición del modelo conceptual	Feb 12	Feb 14	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.4.5	Definición del modelo de BD	Feb 14	Feb 18	2d	1d 7h		70.26	Analist. Pc1. Pc2. Int. Prog 1
3.4.6	Definición de las GUI	Feb 18	Feb 19	1d	7h 31min		35.47	Analist. Pc1. Pc2. Prog 1
3.4.7	Definición de los reportes	Feb 19	Feb 20	1d	7h 31min	6d 6h	31.53	Analist. Pc1. Int
3.4.8	Documento casos de uso, clases, diagrama de BD, etc	Feb 20	Feb 20	N/A	N/A	6d 6h	0	
3.4.9	Implementación del módulo	Feb 19	Feb 28	7d	6d 6h		275.13	Pc1. Int. Prog 1
3.4.10	Pruebas del módulo	Feb 28	Mar 1	1d	7h 46min		35.13	Analist. Pc1. Pc2. Int. Prog 1
3.4.11	Entrega del módulo usuarios	Mar 1	Mar 1	N/A	N/A		0	
3.5	Modulo 5	Mar 1	Mar 26	22d	18d 4h	24d 3h	770.83	
3.5.1	Planificación del módulo contabilidad	Mar 1	Mar 6	3d	2d 7h		105.27	Analist. Pc1. Pc2. Int. Prog 1
3.5.2	Plan del módulo contabilidad	Mar 6	Mar 6	N/A	N/A		0	
3.5.3	Definición de los casos de uso contabilidad	Mar 6	Mar 8	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.5.4	Definición del modelo conceptual contabilidad	Mar 8	Mar 12	2d	1d 7h		70.26	Analist. Pc1. Pc2. Int. Prog 1

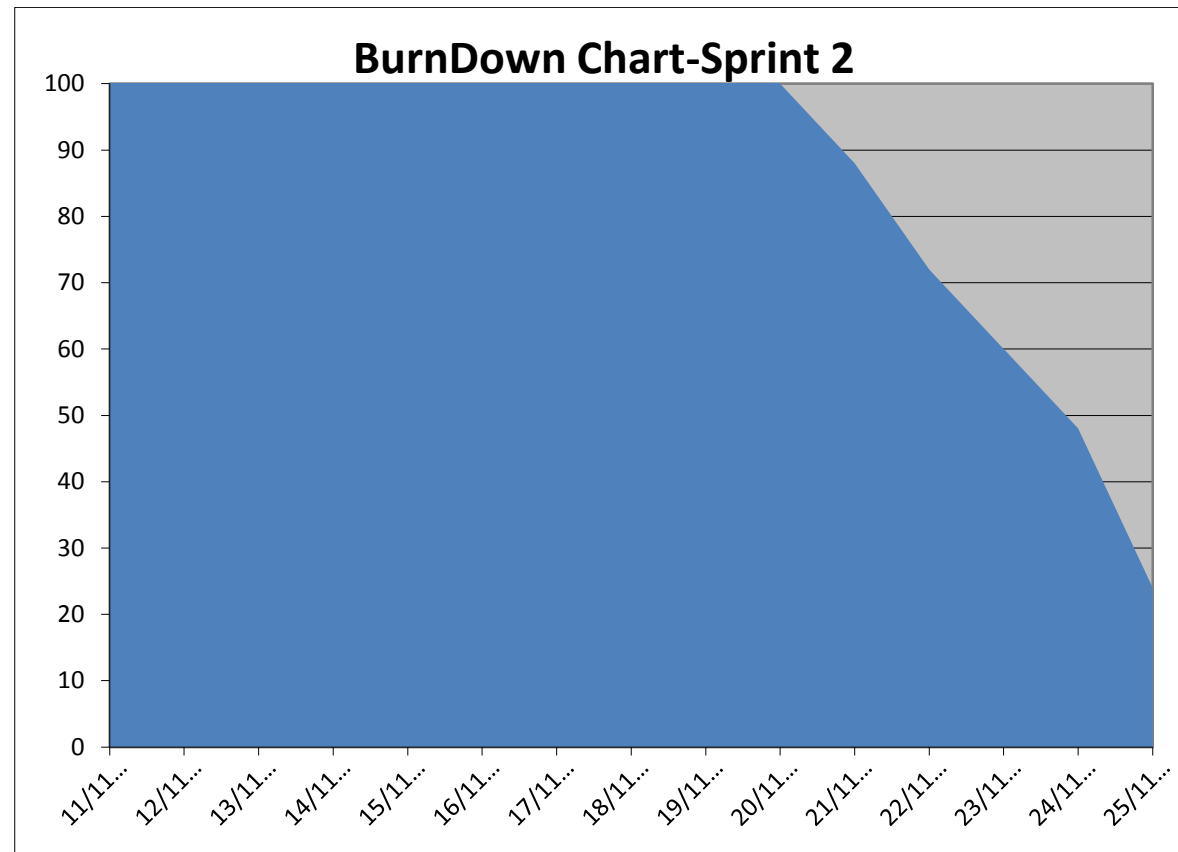
3.5.5	Definición del modelo de BD contabilidad	Mar 12	Mar 14	2d	1d 7h		70.81	Analist. Pc1. Pc2. Prog 1
3.5.6	Definición de las GUI contabilidad	Mar 14	Mar 15	1d	7h 51min		35.47	Analist. Pc1. Pc2. Prog 1
3.5.7	Definición de los reportes contabilidad	Mar 15	Mar 18	1d	7h 51min		31.53	Analist. Pc1. Int
3.5.8	Documentos casos de uso, clases, diagrama de BD, etc	Mar 18	Mar 18	N/A	N/A		0	
3.5.9	Implementación del módulo contabilidad	Mar 18	Mar 25	10d	4d 7h		355.42	Analist. Pc1. Pc2. Int. Prog 1
3.5.10	Pruebas del módulo contabilidad	Mar 25	Mar 26	1d	7h 46min		31.24	Analist. Pc1. Pc2. Int
3.5.11	Entrega del módulo contabilidad	Mar 26	Mar 26	N/A	N/A		0	
4	Pruebas generales del sistema completo	Mar 26	Apr 4	7d	6d 6h		245.67	Analist. Pc1. Pc2. Int. Prog 1
5	Instalación	Apr 4	Apr 15	8d	7d 7h	9d 5h	229.64	
5.1	Adecuación de hardware y software	Apr 4	Apr 3	3d	3d		51.36	Analist. Pc1. Pc2. Int. Prog 1
5.2	Verificación y validación	Apr 3	Apr 11	3d	2d 7h		106.28	Analist. Pc1. Pc2. Prog 1
5.3	Documento de V&V	Apr 11	Apr 11	N/A	N/A		0	
5.4	Instalación del Sistema	Apr 11	Apr 15	2d	2d		72	Analist. Prog 1
5.5	Entrega del Sistema	Apr 15	Apr 15	N/A	N/A		0	Analist. Prog 1
6	Mantenimiento	Apr 15	Apr 29	10d	9d 5h		350.95	
6.1	Determinación y corrección de errores	Apr 15	Apr 25	10d	9d 5h		350.95	Analist. Pc1. Pc2. Int. Prog 1
7	Fin	Apr 25	Apr 25	N/A	N/A		0	

Name	Short name	Type	Group	Email	Cost
Internet	Int	Material	Equipos		0.5
Computador2	Pc2	Material	Equipos		0.6
Computador1	Pc1	Material	Equipos		0.6
Analista	Analist	Work	Programadores		4
Programador1	Prog 1	Work	Programadores		5

Sprint 2

Fecha Entrega	25/11/2012																
Fecha Inicio	11/11/2012																
Días que quedan sprint		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Ítem / Quién	Descripción	11/11/2012	12/11/2012	13/11/2012	14/11/2012	15/11/2012	16/11/2012	17/11/2012	18/11/2012	19/11/2012	20/11/2012	21/11/2012	22/11/2012	23/11/2012	24/11/2012	25/11/2012	
3	Como: Cajero Quiero: Gestionar los reportes de caja De tal forma que: pueda realizar la impresión de libretas, verificar el estado de cuenta y realizar el cierre diario																
Angel	Cierre diario	4	4	0	0	4	0	0	0	4	0	4	0	0	4	4	
Angel	Estado de cuenta	0	4	4	4	4	0	0	0	4	0	0	4	4	4	0	
Angel	Impresión de Libretas	0	0	0	0	0	4	4	4	0	4	0	0	0	0	0	
José	Pruebas	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	
José	Manual de usuario, técnico	4	0	0	0	0	0	0	0	0	0	0	4	4	4	4	
4	Como: Cajero Quiero: Gestionar los procesos de contabilidad De tal forma que: pueda registrar un asiento contable, manejar amortizaciones y depreciaciones																
Angel	Registro de un asiento contable	4	4	4	0	0	0	4	4	4	4	0	0	0	0	0	
Angel	Manejo de Depreciaciones y Amortizaciones	0	0	4	4	4	4	4	4	0	0	0	0	0	0	0	
José	Cierre del año	4	4	0	4	4	4	4	0	4	4	4	0	0	0	0	
José	Pruebas	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	
José	Manual de usuario, técnico	0	0	4	4	4	0	0	0	0	0	4	0	0	0	0	
		16	16	16	16	20	12	16	12	16	12	16	12	12	24	16	

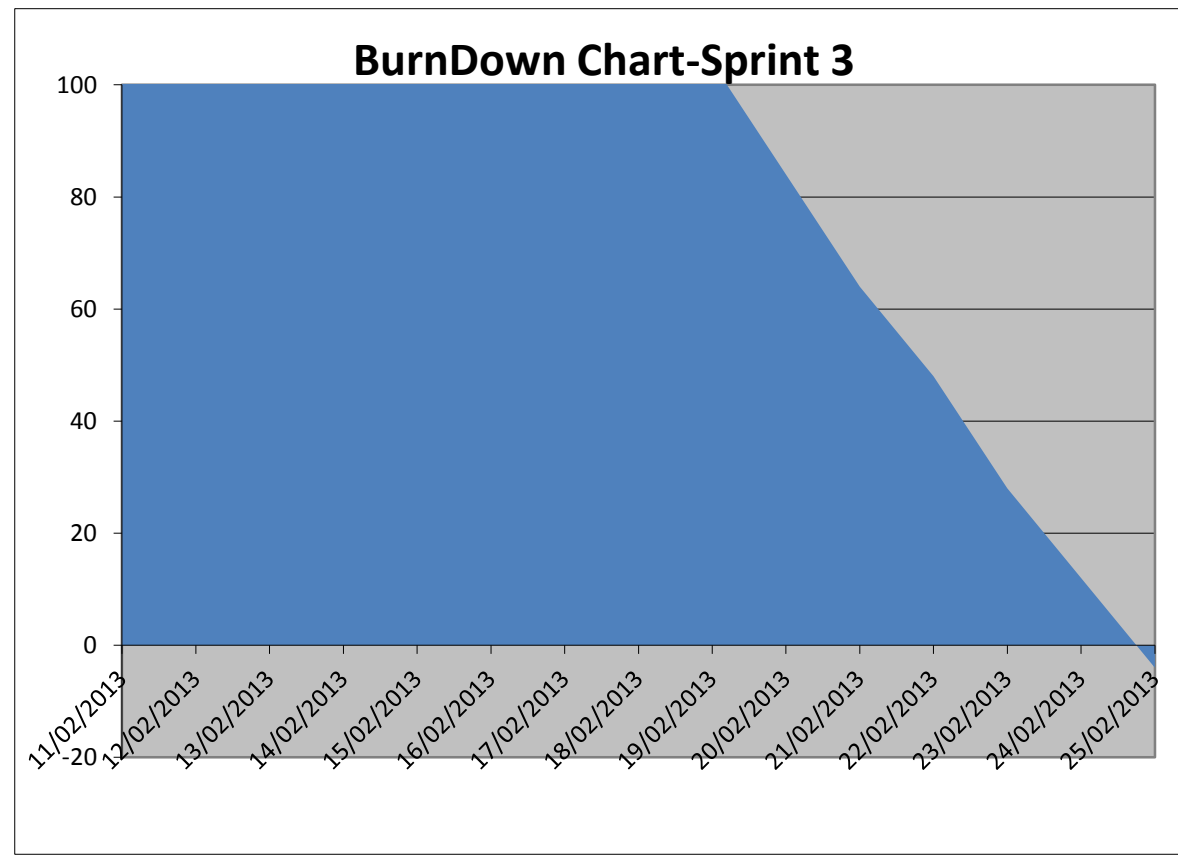
		232	7	100	100	125	75	100	75	100	75	100	75	75	150	100
	HORAS ESTIMADAS DEL SPRINT	240	224	208	192	176	156	144	128	116	100	88	72	60	48	24



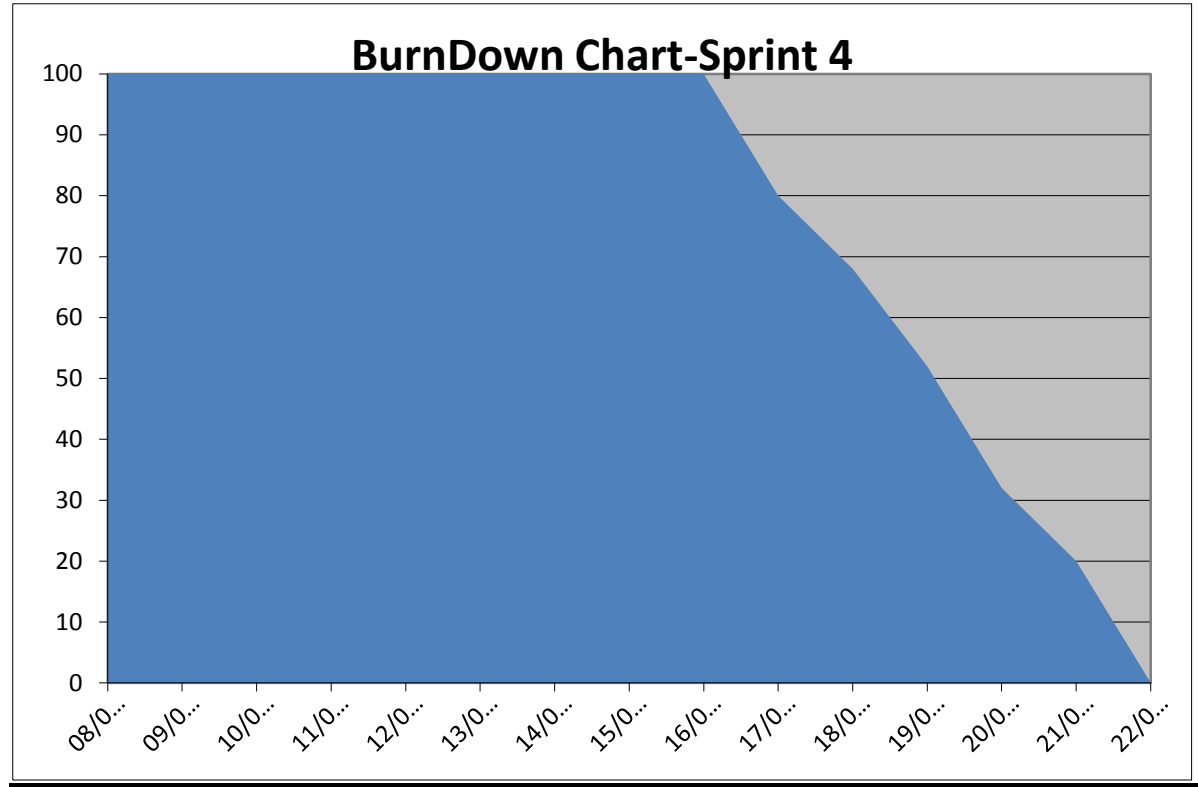
Sprint 3

Fecha Entrega	25/02/2013															
Fecha Inicio	11/02/2012															
Días que quedan sprint		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ítem / Quién	Descripción	11/02/2013	12/02/2013	13/02/2013	14/02/2013	15/02/2013	16/02/2013	17/02/2013	18/02/2013	19/02/2013	20/02/2013	21/02/2013	22/02/2013	23/02/2013	24/02/2013	25/02/2013
2	Como: Asesor de crédito Quiero: Gestionar los Procesos de Crédito De tal forma que: se pueda imprimir una solicitud de crédito y agilizar los procesos															
José	Análisis de la capacidad de pago de los clientes	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
José	Registro de solicitudes de crédito	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0
José	Generación de tablas presuntivas	0	0	0	4	4	4	4	0	0	0	0	0	0	0	0
José	Registro de garantes	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0
Angel	Liquidación de créditos	0	0	0	0	0	0	0	0	4	4	4	0	0	0	0
Angel	Cobros de cuotas	4	0	0	0	0	0	0	4	0	0	4	4	0	0	4
Angel	Cobros de intereses	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0
José	Cobros de multas	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0
José	Cobros de comisiones	4	0	0	0	0	0	0	0	0	4	0	0	4	4	0
José	Generación de notificaciones por atraso	0	0	0	0	0	0	0	0	4	0	4	4	0	0	4
José	Renovación de créditos	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0
Angel	Seguimiento y cobranza de la cartera de crédito	0	4	4	4	4	0	0	0	0	4	0	0	0	0	0
Angel	Reclasificación diaria de la cartera	4	4	4	4	0	0	0	0	4	0	0	0	0	0	0
Angel	Pruebas	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4
Angel	Manual de usuario, técnico	4	0	4	0	4	0	4	4	0	4	0	4	0	4	0

		24	16	20	16	20	12	16	12	20	20	16	20	16	16	12
		256	6	83	67	83	50	67	50	83	83	67	83	67	67	50
	HORAS ESTIMADAS DEL SPRINT	240	216	200	180	164	144	132	116	104	84	64	48	28	12	-4



José	Estado de cambio de patrimonio	0	0	0	0	4	0	0	0	0	0	0	0	0	0	4
José	Diario por cuenta	0	0	0	0	4	4	4	4	4	4	4	0	0	0	0
José	Pruebas	0	0	0	0	0	0	0	0	0	0	4	4	4	4	0
José	Manual de usuario, técnico	0	0	4	4	0	4	0	0	4	0	0	4	0	4	4
		16	12	24	16	20	24	16	12	20	12	16	20	12	20	16
		256	5	150	100	125	150	100	75	125	75	100	125	75	125	100
	HORAS ESTIMADAS DEL SPRINT	240	224	212	188	172	152	128	112	100	80	68	52	32	20	0



Sprint 5

Fecha Entrega	24/06/2013															
Fecha Inicio	10/06/2012															
Días que quedan sprint		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ítem / Quién	Descripción	10/06/2013	11/06/2013	12/06/2013	13/06/2013	14/06/2013	15/06/2013	16/06/2013	17/06/2013	18/06/2013	19/06/2013	20/06/2013	21/06/2013	22/06/2013	23/06/2013	24/06/2013
8	Como: Cajero Quiero: Gestionar usuarios De tal forma que: pueda realizar el registro de nuevos usuarios, asignar roles y permisos, autenticar usuarios y realizar auditoria de usuarios.															
José	Registro de nuevos usuarios	4	4	0	0	0	0	0	0	4	0	0	0	0	0	0
José	Asignación de roles y permisos a los usuarios	4	4	0	0	0	4	4	0	0	0	4	0	4	0	0
José	Autenticación de usuarios	0	0	8	4	4	4	0	4	4	0	0	0	0	0	0
Angel	Auditoria de usuarios	4	4	0	0	0	0	0	4	4	0	4	4	4	4	0
Angel	Pruebas	0	0	4	4	4	0	0	0	0	0	4	0	4	0	4
Angel	Manual de usuario, técnico	4	4	0	4	0	4	4	4	0	4	0	4	0	4	4
9	Como: Cajero Quiero: Generar reporte de usuarios De tal forma que: pueda realizar el reporte de usuarios registrados, accesos fallidos.															
José	Usuarios registrados	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0
José	Accesos de usuarios en determinada fecha	0	0	0	4	4	0	0	4	4	4	4	0	0	0	0
Angel	Accesos fallidos	0	0	0	0	0	0	0	0	0	4	0	4	4	4	0

Angel	Pruebas	0	0	0	0	0	0	0	0	4	0	4	0	0	4	4
Angel	Manual de usuario, técnico	0	0	0	4	4	0	4	0	0	4	0	0	4	0	4
		16	16	12	20	16	12	16	20	24	16	20	12	20	16	16
		252	6	75	125	100	75	100	125	150	100	125	75	125	100	100
	HORAS ESTIMADAS DEL SPRINT	240	224	208	196	176	160	148	132	112	88	72	52	40	20	4

