



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA

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

APLICATIVO

Desarrollo e Implementación de los Módulos de CARTERA Y
CLIENTES para la Cooperativa de Ahorro y Crédito Unión
Cochapamba.

AUTOR: BAIRON PATRICIO GUDIÑO VILLALBA

DIRECTOR: Dra. C.P.A. MARÍA DE LA PORTILLA VERA M.B.A.

Ibarra - Ecuador

2011 - 2012



CERTIFICACIÓN

Certifico que la Tesis “**SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA**” con el aplicativo “**DESARROLLO E IMPLEMENTACIÓN DE LOS MÓDULOS DE CARTERA Y CLIENTES PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA**” ha sido realizada en su totalidad por el señor: Bairon Patricio Gudiño Villalba portador de la cédula de identidad número: 1002299566.

Dra. C.P.A. María De la Portilla Vera M.B.A.

DIRECTORA DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
INVESTIGACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, BAIRON PATRICIO GUDIÑO VILLALBA, con cédula de identidad Nro. 100229956-6, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”** con el aplicativo **“ DESARROLLO E IMPLEMENTACIÓN DE LOS MÓDULOS DE CARTERA, CLIENTES Y AUDITORÍA INFORMÁTICA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

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

BAIRON PATRICIO GUDIÑO VILLALBA

C.I: 100229956-6

Ibarra a los 17 días del mes de Julio del 2012



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

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

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	100229956'6
APELLIDOS Y NOMBRES	GUDIÑO VILLALBA BAIRON PATRICIO
DIRECCIÓN	Av. Real Audiencia Oe3-211 y Ciruelos
EMAIL	bairongv@gmail.com
TELÉFONO FIJO	02 2470718
TELÉFONO MÓVIL	087 518369

DATOS DE LA OBRA	
TÍTULO	“SISTEMA INTEGRADO DE INFORMACIÓN FINANCIERA PARA LA COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN COCHAPAMBA-MÓDULO CARTERA Y CLIENTES”
AUTOR	GUDIÑO VILLALBA BAIRON PATRICIO
FECHA	17 DE JULIO DEL 2012
PROGRAMA	PREGRADO
TÍTULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	DRA. C.P.A. MARÍA DE LA PORTILLA VERA M.B.A.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, BAIRON PATRICIO GUDIÑO VILLALBA, con cedula de identidad Nro. 100229956-6, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.

BAIRON PATRICIO GUDIÑO VILLALBA

C.I. 100229956-6

Ibarra a los 17 días del mes de Julio del 2012

DEDICATORIA

A Dios, por concederme el don de la existencia, por guiar mi vida, y ofrendarme una familia maravillosa.

A mis padres Marcia Villalba y Fabián Gudiño, por todo el apoyo que me han brindado, porque gracias a ellos, a su esfuerzo y apoyo hoy puedo completar una meta muy importante. Gracias por inculcar en mí, el deseo y anhelo de superación, por estar presentes en los momentos más difíciles de mi carrera y de mi vida, por toda su guía, amor y consejo.

AGRADECIMIENTOS

Agradezco de manera sincera y especial, a la doctora María De la Portilla Vera, por haber puesto su confianza en mí, por todo el apoyo, paciencia y dirección en el desarrollo de la tesis.

A todos los miembros de mi familia, por acompañarme y brindarme su apoyo incondicional en todo momento.

A Esther Rincón, por darme ánimos y tenerme mucha paciencia en este largo proceso.

Agradezco también a mis compañeros de tesis, por hacer más fácil y ameno la culminación de la presente tesis.

Y a todas aquellas personas, que directa o indirectamente me han acompañado en mi vida estudiantil.

ÍNDICE DE CONTENIDO

1. Introducción	1
1.1. Base legal de la institución	1
1.1.1. Constitución y domicilio.....	1
1.1.2. Objetivos de la Cooperativa.....	1
1.1.3. Principios de la Cooperativa.....	2
1.1.4. Socios fundadores	2
1.2. Situación actual diagnóstico.	2
1.2.1. Misión	2
1.2.2. Visión	2
1.2.3. Valores corporativos.....	2
1.2.4. FODA.....	3
1.2.5. Situación.....	5
1.2.6. Diagnostico 6M	6
1.2.7. Objetivos operativos y actividades del plan anual operativo	6
1.3. Estructura	8
1.3.1. Organigrama estructural	8
1.4. Información general	9
1.4.1. Clientes segmentados por valores.....	9
1.4.2. Préstamos	10
2. Marco teórico	12
2.1. Normas legales	12
2.1.1. Ley general de instituciones del sistema financiero	12
2.1.2. Ley de Cooperativas	12
2.1.3. Normas de Basilea	13
2.2. Estándares abiertos	14
2.2.1. HTML.....	14
2.2.1.1. Características:.....	14
2.2.1.2. Sintaxis general:	15
2.2.1.3. Estructura de un documento HTML.....	15
2.2.2. CSS (Cascading Style Sheets)	16
2.2.2.1. Formas de usar CSS.....	17
2.2.2.2. Ventajas.....	18
2.3. Sistemas operativos (SO)	18
2.3.1. Introducción.....	18
2.3.2. Definición.....	18
2.3.3. Características	19
2.3.3.1. Proyecto Fedora	22
2.3.3.2. Características.....	23
2.3.4. CentOS	23
2.3.4.1. Características.....	24
2.4. Servidor de aplicaciones Apache Tomcat	25
2.4.1. Introducción a Apache Tomcat.....	25
2.4.2. Funcionamiento del servidor de aplicaciones Apache Tomcat.....	26
2.4.3. Estructura de directorios	27
2.5. Seguridad en sistemas web	27
2.5.1. Introducción.....	27

2.5.2.	Seguridad en el cliente.....	28
2.5.3.	Seguridad en la red.....	28
2.5.4.	Seguridad en el servidor.....	29
2.5.4.1.	Servidor web.....	29
2.5.4.2.	Servidor de base de datos	30
2.5.5.	Seguridad en la aplicación.....	30
2.5.5.1.	Control de acceso.....	30
2.5.5.2.	Validación de datos de entrada	31
2.6.	Bases de datos	32
2.6.1.	PostgreSQL.....	32
2.6.1.1.	Propiedades.....	34
2.6.1.2.	Características.....	34
2.7.	Factibilidad de plataformas y frameworks	36
2.7.1.	Plataforma Java.....	36
2.7.1.1.	Definición.....	36
2.7.1.2.	Tipos de programas en Java.....	37
2.7.2.	JSF (Java Server Faces).....	37
2.7.2.1.	Introducción	37
2.7.2.2.	Definición.....	38
2.7.2.3.	Ventajas.....	38
2.7.2.4.	Ciclo de vida	39
2.7.2.5.	Estructura de una aplicación JSF.....	39
2.7.3.	JSF con Netbeans.....	40
2.7.4.	Netbeans IDE	40
2.7.4.1.	Introducción	40
2.7.4.2.	Definición.....	41
2.7.4.3.	Ventajas.....	41
2.7.5.	Java EE (Java Platform, Enterprise Edition)	41
2.7.5.1.	Introducción a JEE	41
2.7.5.2.	Definición.....	42
2.7.5.3.	Arquitectura de JEE.....	44
2.7.6.	Hibernate.....	45
2.7.6.1.	Definición.....	45
2.7.6.2.	Características.....	46
2.8.	Herramienta de reportes.....	46
2.8.1.	Introducción a JasperReports	46
2.8.2.	Definición.....	47
2.8.3.	Requerimientos de JasperReports	47
2.8.4.	Funcionamiento de JasperReports.....	47
3.	Funcionamiento del sistema	50
3.1.	Vista general.....	50
3.1.1.	Características del sistema.....	51
3.1.2.	Dependencias para la implementación	52
3.1.2.1.	Costos	52
3.1.2.2.	Licenciamiento	52
3.1.2.3.	Instalación.....	53
3.1.3.	Funcionamiento	53
3.2.	Arquitectura del sistema	54
3.3.	Descripción de los módulos.....	55
3.3.1.	Módulo de Préstamos-Cartera	55
3.3.1.1.	Procesos del módulo Préstamos-Cartera.....	57

3.3.1.2.	Acciones del módulo Préstamo-Cartera.....	58
3.3.1.3.	Casos de uso del módulo Préstamos-Cartera.....	60
3.3.2.	Módulo de Clientes.....	61
3.3.2.1.	Procesos del módulo Clientes.....	62
3.3.2.2.	Acciones del módulo Clientes.....	63
3.3.2.3.	Casos de uso del módulo Clientes.....	65
3.3.3.	Módulo de Auditoría Informática.....	66
3.3.3.1.	Procesos del módulo Auditoría Informática.....	66
3.3.3.2.	Acciones del módulo Auditoría Informática.....	67
3.3.3.3.	Casos de uso del módulo Auditoría Informática.....	68
4.	<i>Diseño y desarrollo del aplicativo</i>	70
4.1.	Fase de inicio	71
4.1.1.	Documento de visión.....	71
4.1.1.1.	Introducción.....	71
4.1.1.2.	Posicionamiento.....	71
4.1.1.3.	Descripción de los interesados y usuarios.....	74
4.1.1.4.	Vista general del producto.....	82
4.1.1.5.	Características del producto.....	91
4.1.1.6.	Restricciones.....	91
4.1.1.7.	Rangos de calidad.....	91
4.1.1.8.	Precedencia y prioridad.....	91
4.1.1.9.	Otros requerimientos del producto.....	92
4.1.2.	Plan de desarrollo de software.....	92
4.1.2.1.	Introducción.....	92
4.1.2.2.	Vista general del proyecto.....	94
4.1.2.3.	Organización del proyecto.....	99
4.1.2.4.	Gestión del proceso.....	100
4.2.	Fase de elaboración	108
4.2.1.	Documento de arquitectura.....	108
4.2.1.1.	Introducción.....	108
4.2.1.2.	Representación de la arquitectura.....	108
4.2.1.3.	Objetivos y restricciones de la arquitectura.....	109
4.2.1.4.	Vista de restricciones.....	115
4.2.1.5.	Vistas lógicas.....	117
4.2.1.6.	Vista de despliegue.....	120
4.2.2.	Casos de uso.....	121
4.3.	Fase de construcción	121
4.3.1.	Modelo de datos.....	121
4.3.1.1.	Diseño de las tablas del módulo Préstamos-Cartera.....	122
4.3.1.2.	Diseño de las tablas del módulo Clientes.....	123
4.3.1.3.	Diseño de las tablas del módulo Auditoría Informática.....	124
Fuente: propia.....		124
4.3.2.	Plan de pruebas.....	125
4.3.2.1.	Introducción.....	125
4.3.2.2.	Estrategia de pruebas.....	125
4.3.2.3.	Herramientas.....	132
4.3.2.4.	Recursos.....	132
4.3.2.5.	Entregables.....	133
4.3.3.	Lista de riesgos.....	133
4.4.	Fase de transición	134
4.4.1.	Manual de instalación.....	134
4.4.1.1.	Introducción.....	134

4.4.1.2.	Instalación de la máquina virtual de Java	135
4.4.1.3.	Configuración de JAVA_HOME	135
4.4.1.4.	Instalación de PostgreSQL	136
4.4.1.5.	Instalación y configuración de Apache Tomcat.	144
4.4.1.6.	Configuración del SSL para HTTPS	147
5.	<i>Conclusiones y recomendaciones</i>	152
5.1.	<i>Conclusiones</i>	152
5.2.	<i>Recomendaciones</i>	153
6.	<i>Glosario</i>	154
6.1.	Introducción	154
6.2.	Propósito	154
6.3.	Referencias	154
6.4.	Glosario:	154
7.	<i>Anexos</i>	158
7.1.	Copia del anteproyecto	158
7.2.	Diagrama entidad relación	160
7.3.	Diagrama general de casos de uso	162
8.	<i>Referencias</i>	165

ÍNDICE DE TABLAS

Tabla 1. Análisis de la Cooperativa (FODA)	4
Tabla 2. Tipos de préstamos.	10
Tabla 3. Lista de sistemas operativos.	20
Tabla 4. Comparativa de sistemas operativos Windows y Linux	21
Tabla 5. Límites de PostgreSQL	35
Tabla 6. Características del sistema.....	52
Tabla 7. Acciones del módulo Préstamos-Cartera.....	59
Tabla 8. Acciones del módulo Clientes.	64
Tabla 9. Acciones del módulo Auditoría.....	68
Tabla 10. Definición del problema.	73
Tabla 11. Lista de interesados.	74
Tabla 12. Resumen de los usuarios.	75
Tabla 13. Perfil de los interesados.....	76
Tabla 14. Responsables del proyecto.	77
Tabla 15. Responsable funcional.....	78
Tabla 16. Administrador del sistema.....	79
Tabla 17. Administrador funcional del sistema.....	80
Tabla 18. Usuario del sistema.....	80
Tabla 19. Necesidades de los usuarios.	82
Tabla 20. Módulo de Ahorros	84
Tabla 21. Módulo de Caja.	85
Tabla 22. Módulo de Administración y Seguridad.	86
Tabla 23. Módulo Contabilidad.....	87
Tabla 24. Módulo Préstamos-Cartera.....	88
Tabla 25. Módulo Clientes.	89
Tabla 26. Módulo Auditoría.	90
Tabla 27. Roles y responsabilidades.	100
Tabla 28. Plan de fases	100
Tabla 29. Hitos del proyecto.....	102
Tabla 30. Objetivo de las iteraciones.	103
Tabla 31. Artefactos generados en la fase de inicio.	106
Tabla 32. Artefactos generados en la fase de elaboración.....	107
Tabla 33. Descripción de los interesados.	111
Tabla 34. Prioridad de casos de uso.	113
Tabla 35. Tecnologías empleadas.....	117
Tabla 36. Prueba de integridad de datos.....	126
Tabla 37. Pruebas del sistema.	126

Tabla 38. Pruebas del ciclo del negocio.	127
Tabla 39. Pruebas de interfaz de usuario.	128
Tabla 40. Pruebas de desempeño.	130
Tabla 41. Pruebas de seguridad y control de acceso.	131
Tabla 42. Herramientas.	132
Tabla 43. Recursos.	132
Tabla 44. Entregables.	133
Tabla 45. Lista de riesgos.	134

ÍNDICE DE GRÁFICOS

Figura 1. Diagnóstico de la Cooperativa (6 M).....	6
Figura 2. Organigrama estructural.....	8
Figura 3. Segmentación de clientes.....	9
Figura 4. Documento HTML.....	16
Figura 5. Estilo CSS.....	16
Figura 6. Función de los sistemas operativos.....	19
Figura 7. Funcionamiento de Apache Tomcat.....	26
Figura 8. HTTP sobre SSL.....	28
Figura 9. Sesión SSL.....	29
Figura 10. Componentes de Postgres.....	33
Figura 11. Plataforma JAVA.....	36
Figura 12. Ciclo de vida de JSF.....	39
Figura 13. Estructura de directorio de una aplicación JSF.....	40
Figura 14. Componente en Java EE.....	43
Figura 15. Evolución de la plataforma Java EE.....	44
Figura 16. Arquitectura de aplicaciones Java EE.....	45
Figura 17. Implementaciones con JasperReports.....	48
Figura 18. Prospectiva del sistema.....	50
Figura 19. Funcionamiento general del sistema.....	54
Figura 20. Arquitectura de la aplicación.....	55
Figura 21. Procesos del módulo Préstamos-Cartera.....	57
Figura 22. Casos de uso del módulo Préstamos-Cartera.....	60
Figura 23. Procesos del módulo Clientes.....	62
Figura 24. Casos de uso del módulo Clientes.....	65
Figura 25. Procesos del módulo Auditoría.....	66
Figura 26. Casos de uso del módulo Auditoría.....	68
Figura 27. Perspectiva del sistema.....	83
Figura 28. Calendario del Proyecto.....	104
Figura 29. Actores.....	110
Figura 30. Modelo de casos de negocios.....	112
Figura 31. Patrón arquitectónico MVC.....	118
Figura 32. Arquitectura lógica del sistema.....	119
Figura 33. Interfaz de usuario.....	119
Figura 34. Vista de despliegue.....	121
Figura 35. Tablas del módulo Préstamos-Cartera.....	122
Figura 36. Tablas del módulo Clientes.....	123
Figura 37. Tablas del módulo Auditoría.....	124
Figura 38. Instalación de la máquina virtual de Java.....	135
Figura 39. Configuración del JAVA_HOME.....	136

Figura 40. Configuración del PATH	136
Figura 41. Asistente de instalación de PostgreSQL.	137
Figura 42. Asistente de instalación de PostgreSQL.	137
Figura 43. Asistente de instalación de PostgreSQL.	138
Figura 44. Asistente de instalación de PostgreSQL.	138
Figura 45. Asistente de instalación de PostgreSQL.	139
Figura 46. Asistente de instalación de PostgreSQL.	139
Figura 47. Asistente de instalación de PostgreSQL.	140
Figura 48. Asistente de instalación de PostgreSQL.	140
Figura 49. Creación de nuevo usuario.	141
Figura 50. Creación de nuevo usuario.	141
Figura 51. Creación de nuevo usuario.	142
Figura 52. Creación de una nueva base de datos.	142
Figura 53. Subir backup a la base de datos.	143
Figura 54. Subir backup a la base de datos.	143
Figura 55. Subir backup a la base de datos.	144
Figura 56. Script de inicio de Apache.	145
Figura 57. Página de administración de Apache Tomcat.	146
Figura 58. Página de inicio del sistema.	147
Figura 59. Generación de la clave.	147
Figura 60. Parámetros de la clave.	148
Figura 61. Configuración del server.xml.	148
Figura 62. Página de inicio del sistema en modo HTTPS.	149

RESUMEN

El desarrollo del Sistema Financiero “FINANSYS” se ha realizado con el fin de proporcionar a la Cooperativa de Ahorro y Crédito Unión Cochapamba un aplicativo integrado, estable y sólido, que agilice y realice sus procesos de forma más fácil y ordenada.

El presente sistema tiene como finalidad automatizar los procesos financieros de la Cooperativa, esto con el objetivo de agilizar los trámites y papeleos que en él se realizan, para brindar un mejor servicio a los socios.

Todo el Sistema Financiero está desarrollado en siete módulos: Administración y Seguridad, Clientes, Contabilidad, Cartera (Préstamos), Ahorros, Caja y Auditoría Informática; Bajo mi cargo están los módulos de Cartera, Clientes y Auditoría Informática.

La metodología empleada para el análisis, implementación y documentación en el desarrollo del sistema ha sido mediante RUP (RATIONAL UNIFIED PROCESS), la misma que consta de cuatro fases: Inicio, Elaboración, Construcción y Transición; En la primera fase se definen los objetivos y justificación del proyecto, en la segunda fase establecen los requerimientos y la arquitectura del sistema, en la tercer fase se desarrollan los módulos de acuerdo a las especificaciones detalladas en las fases anteriores y en la última fase se pone el sistema en producción.

Todas las herramientas empleadas en la elaboración e implementación del sistema, son herramientas libres u Open Source y orientadas a la web.

SUMMARY

The development of the Financial System "FINANSYS" has been performed in order to provide the Cooperative Savings and Credit Union an application Cochapamba integrated, stable and sound, speed up and make processes easier and more orderly.

This system aims to automate the financial processes of the Cooperative, this in order to streamline the red tape that he made to provide better service to members.

The entire financial system is developed in seven modules: Administration and Security Customers, Accounting, Portfolio (loans), Savings, Cash and Computer Audit, under my charge are modules Portfolio, Clients, and Computer Audit.

The methodology used for analysis, implementation and documentation in the development of the system has been using RUP (Rational Unified Process), it consists of four phases: Start, Elaboration, Construction and Transition, In the first phase defines the objectives and justification for the project, in the second phase establishes the requirements and system architecture, in the third stage modules are developed according to the specifications detailed in the upstream and in the last phase the system is put into production.

All equipment used in the development and implementation of the system are free or Open Source tools and web-oriented.

TEMA

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

APLICATIVO

Desarrollo e implementación de los módulos de CARTERA, CLIENTES Y AUDITORÍA INFORMÁTICA para la Cooperativa de Ahorro y Crédito Unión Cochapamba.

PROBLEMA

Los procesos de CARTERA y CLIENTES de la Cooperativa de Ahorro y Crédito Unión los manejan con un sistema informático desactualizado, lo cual provoca que se labore con un flujo de información inexacta, poco fiable y nada permanente, haciendo difícil su administración y mantenimiento y tampoco cuenta con un módulo de AUDITORÍA INFORMÁTICA.

JUSTIFICACIÓN

El presente estudio de factibilidad se realiza para determinar la viabilidad del sistema, el cual tiene como finalidad automatizar los procesos financieros de la Cooperativa de Ahorro y Crédito Unión Cochapamba, esto con el objetivo de agilizar los trámites y papeleos que en él se realizan, para brindar un mejor servicio a los socios.

Mediante visitas realizadas a la Cooperativa se observó la falta de un sistema que les permita a ellos como Institución agilizar y automatizar los procesos que hasta ahora se los ha realizado de manera incorrecta, además no cuenta con las respectivas políticas de seguridad, dado que en la Cooperativa no existe separación de funciones, es decir no hay un profesional asignado a cada área, esto hace que la información sea vulnerable a manipulaciones por parte de cualquier; Al implementar este sistema se pretende reducir el tiempo de respuesta de las diferentes instancias y departamentos de la Cooperativa.

OBJETIVOS

OBJETIVO GENERAL:

Analizar, Diseñar e Implementar los módulos de CARTERA, CLIENTES Y AUDITORÍA INFORMÁTICA para el Sistema Integrado de Información Financiera para la Cooperativa de Ahorro y Crédito Unión Cochapamba mediante el uso de herramientas libres y estándares abiertos, para mejorar y agilizar el control, administración e INTEGRACIÓN de la información, CON SEGURIDAD, INTEGRIDAD Y SIN REDUNDANCIA DE INFORMACIÓN.

OBJETIVOS ESPECÍFICOS:

1. Estudiar y realizar una reingeniería de los procesos involucrados con la CARTERA y CLIENTES de la Cooperativa de Ahorro y Crédito Unión Cochapamba.
2. Investigar el uso de herramientas gratuitas apropiadas para utilizar en el desarrollo del proyecto.
3. Analizar las ventajas y desventajas que se obtiene al implementar sistemas web utilizando herramientas libres y estándares abiertos.
4. Desarrollar los módulos de CARTERA, CLIENTES y AUDITORÍA INFORMÁTICA de tal manera que sean fáciles de utilizar para los usuarios finales, creando interfaces amigables y que su funcionamiento sea bastante intuitivo.
5. Realizar la correcta integración e interacción de los módulos de CARTERA, CLIENTES y AUDITORÍA INFORMÁTICA con el resto del sistema.
6. Establecer las debidas seguridades del sistema para mantener protegida la información.

Capítulo 1: Introducción



RESUMEN DEL CAPÍTULO

- *Base legal de la institución.*
- *Situación actual y diagnóstico.*
- *Estructura de la Cooperativa.*
- *Información general.*

1. Introducción

1.1. Base legal de la institución

1.1.1. Constitución y domicilio

La Cooperativa de Ahorro y Crédito “Unión Cochapamba” en adelante “CUC” con domicilio en la parroquia de Ambuquí, comunidad de Chaupi Guaranguí, Cantón Ibarra de la Provincia de Imbabura, se constituye como una Cooperativa sin fines de lucro y se regirá por la ley de Cooperativas, su Reglamento General, por otras leyes que fueren aplicables y por un estatuto único aprobado por los directivos pertinentes.

La responsabilidad de la Cooperativa ante terceros, está limitada a su capital social y al capital que hubieren suscrito personal en la entidad.

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

1.1.2. Objetivos de la Cooperativa

Los objetivos de la Cooperativa son los siguientes:

- a) Promover la cooperación económica entre sus socios, para cuyo cumplimiento recibirá los ahorros, certificados de aportación y depósitos que éstos realicen, efectuará cobros y pagos así como todas aquellas operaciones necesarias para el fortalecimiento de la Cooperativa crediticia dentro del marco legal permitido por las Cooperativas.
- b) Otorgar préstamos a sus miembros de conformidad al reglamento que para el efecto se establezca.
- c) Proporcionar a sus asociados mayor capacidad en lo económico y social, mediante una adecuada educación cooperativista.
- d) Establecer nexos dentro y fuera del país, con entidades similares en beneficio de la Cooperativa.
- e) Obtener fuentes de financiamiento interno y externo para el desarrollo de la institución.
- f) Establecer otros servicios y otras actividades que estén encuadradas en la ley y Reglamento de Cooperativas y otras leyes que sean aplicables, que conlleven al mejoramiento social y económico de sus miembros.
- g) Efectuar actividades inherentes a buscar financiamiento con la finalidad de capitalizar y otorgar más beneficios a los asociados.

1.1.3. Principios de la Cooperativa

La Cooperativa regulara sus actividades, de conformidad y de acuerdo con los siguientes principios:

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

1.1.4. Socios fundadores

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

1.2. Situación actual diagnóstico.

1.2.1. Misión

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

1.2.2. Visión

“La Cooperativa de Ahorro y Crédito “Unión Cochapamba” hasta el año 2012, desarrollará e implementará estrategias administrativas y financieras que nos permitan tener sucursales grandes y competitivas, que generen microempresas sin descuidar el medio ambiente, de tal manera que sus socios interactúen como una familia en armonía con Dios y la naturaleza”.

1.2.3. Valores corporativos

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

1.2.4. FODA

La Cooperativa es una institución que tiene un buen prestigio financiero y es bien recibida por las comunidades rurales de la provincia de Imbabura.

Existen algunos aspectos que alteran el desempeño normal de la institución, como son el tecnológico, hacen que instituciones de éstas características tengan que buscar mejores oportunidades e innovaciones constantes.

A continuación se presenta un pequeño diagnostico a través de un análisis FODA.

FORTALEZAS	OPORTUNIDADES
<ul style="list-style-type: none"> a) Prestigio institucional. b) Directivos y personal comprometidos. c) Alianzas estratégicas permiten brindar mayores servicios. d) Morosidad decreciente. 	<ul style="list-style-type: none"> ✓ Desarrollo de micro y pequeña empresa. ✓ Posibilidad de utilizar nuevas fuentes de ingresos para aumentar los fondos de la CUC. ✓ Utilización de línea de Corporación Financiera Nacional para financiamiento de largo plazo.

<p>e) Solvencia patrimonial.</p> <p>f) Alto nivel de rentabilidad.</p> <p>g) Servicio social de la comisión de educación.</p> <p>h) Ayuda social mediante actividades del mismo orden.</p>	<p>✓ Apoyo de entidades del exterior como el proyecto FIE (Fondo Ítalo Ecuatoriano).</p> <p>✓ Apertura de sucursales de la CUC.</p> <p>✓ Creciente desarrollo cooperativo en el país.</p>
DEBILIDADES	AMENAZAS
<p>✓ Servicio segmentado a comunidades de las zonas rurales.</p> <p>✓ Falta de sucursales.</p> <p>✓ Bajo nivel de control interno.</p> <p>✓ No contar con un sistema de información integrado.</p> <p>✓ Eficiencia financiera inferior al promedio.</p> <p>✓ Demora en el trámite de créditos.</p> <p>✓ Datos erróneos previos a la presentación de informes financieros.</p> <p>✓ Corrección de errores manual.</p> <p>✓ No existe plan estratégico.</p>	<p>✓ Fortalecimiento del Sistema Financiero de la Cooperativa.</p> <p>✓ Contexto socio-político del país.</p> <p>✓ Situación económica financiera del país.</p> <p>✓ Mercado competitivo.</p> <p>✓ Política estatal sobre el sistema cooperativo.</p> <p>✓ Creación de nuevas instancias reguladoras.</p>

Tabla 1. Análisis de la Cooperativa (FODA)

Fuente: propia.

1.2.5. Situación

Conclusión del diagnóstico

La Cooperativa de ahorro y crédito Unión Cochapamba realiza una serie de procesos distribuidos en diferentes áreas específicas, dichos procesos se los realiza mediante el uso de un sistema programado en C++, el cual muestra muchas limitaciones, porque no controla y administra todos los procesos de forma transparente y unificada, por ello emplea mucho tiempo en procesar la información.

Los datos no se guardan en ninguna base de datos, se manejan y almacenan de acuerdo a cada módulo en archivos visibles para el usuario, lo que hace que este Sistema sea por demás vulnerable y susceptible a cualquier tipo de fraude o estafa.

Todos los módulos funcionan de forma separada e independiente uno del otro, repitiendo información importante en cada módulo y proceso, ocasionando pérdida de tiempo e inconsistencias en la información institucional.

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

1.2.6. Diagnóstico 6M



Figura 1. Diagnóstico de la Cooperativa (6 M)

Fuente: propia.

1.2.7. Objetivos operativos y actividades del plan anual operativo

1. Ser una institución financiera rural posicionada en su área de intervención, con socios que están empoderados, con capacidad de dirigir y administrar los componentes socio-organizativos y financieros de la Cooperativa, en alianza con entidades afines. (FODA)
2. Capacitar a los nuevos directivos en finanzas populares
3. Mantener alianzas con CODESARROLLO y FODEMI, para acceder a créditos.
 - a. Realizar una propuesta conjunta con la UOCC (Unión de Organizaciones Campesinas de Cochapamba), para la utilización de los fondos que se mantiene en administración en FODEMI.
 - b. Rotar al menos 200 créditos.

- c. Realizar tres asambleas generales, dos en el mes de enero, la una extraordinaria, para las elecciones de los nuevos directivos.
 - d. Realizar cuatro reuniones de consejo administrativo.
 - e. Realizar cuatro reuniones de consejo de vigilancia.
4. Preservar el medio ambiente como factor de desarrollo para las comunidades a través de la reforestación y forestación anual de por lo menos 20.000 plantas.
- a. Fomentar la capacitación en el manejo de los recursos naturales in situ a cada beneficiario.
 - b. Producción de 8.000 plantas forestales para las parcelas campesinas de los socios de la CUC.
5. Fomentar la cultura del ahorro y fortalecer el patrimonio de la Cooperativa vía incremento certificados de aportación por parte de los socios con un aporte de \$20 USD anuales.
- a. Promover el ahorro y los depósitos a plazo fijo.
 - b. Reformar el reglamento interno en lo que corresponde a los certificados de aportación.
6. Promover la creación de micro emprendimientos productivos como un nuevo modelo de desarrollo comunitario, que permita mejorar el nivel de vida de los socios.
- a. Establecer 2 alianzas estratégicas para el fomento de micro emprendimientos.
 - b. Apoyar en lo administrativo, a dos micro emprendimientos ya existentes (quesera y asociación 20 de diciembre).
 - c. Apoyar 4 micro-emprendimientos en la zona.
7. Fortalecer el talento humano y administrativo de la Cooperativa.
- a. Fortalecer el equipo técnico - administrativo de la CUC con enfoque de género a través de capacitación, acuerdos y convenios.
 - b. Cumplir con las normas de seguridad y control interno de los bienes y recursos de la CUC (Seguros, Garantías)

1.3. Estructura

1.3.1. Organigrama estructural

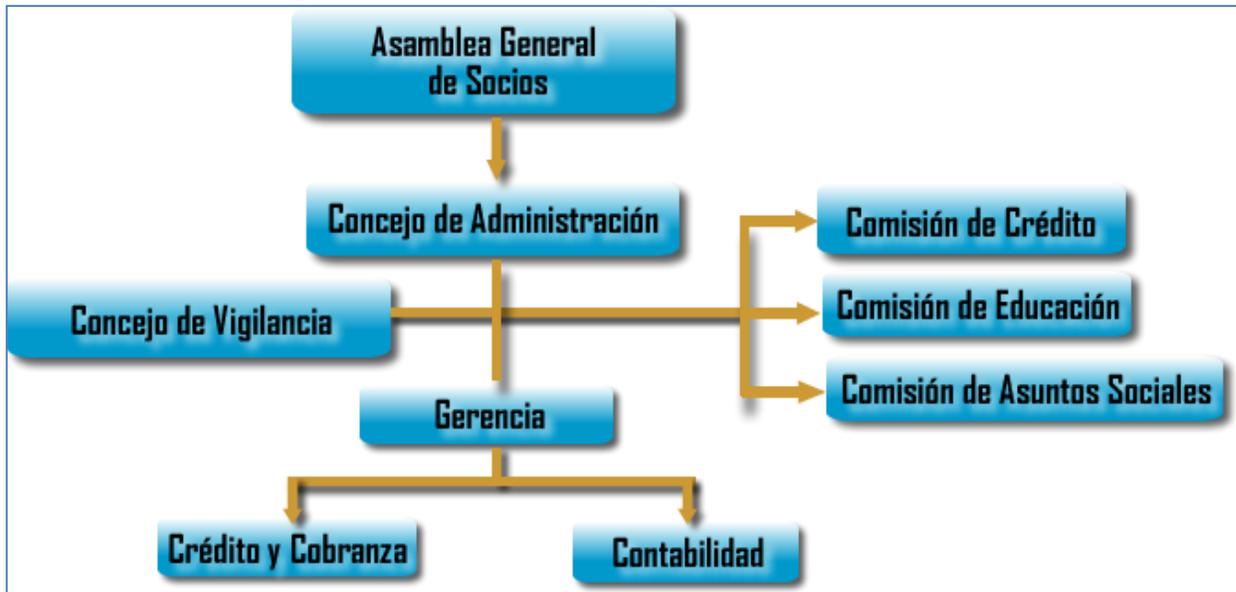


Figura 2. Organigrama estructural.

Fuente: propia.

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

a) Asamblea general de socios

Es la máxima autoridad de la Cooperativa y está constituida por todos los socios que consten en el registro respectivo y que estuvieren en uso y goce de sus derechos.

b) Concejo de administración

Es el organismo directo de la Cooperativa quien se atribuye la elección de presidente, gerente, secretario y vocales pertinentes de acuerdo con la ley.

c) Concejo de vigilancia

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

d) Gerencia

Administración general de la Cooperativa.

e) **Comisiones**

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

1.4. **Información general**

1.4.1. **Clientes segmentados por valores**

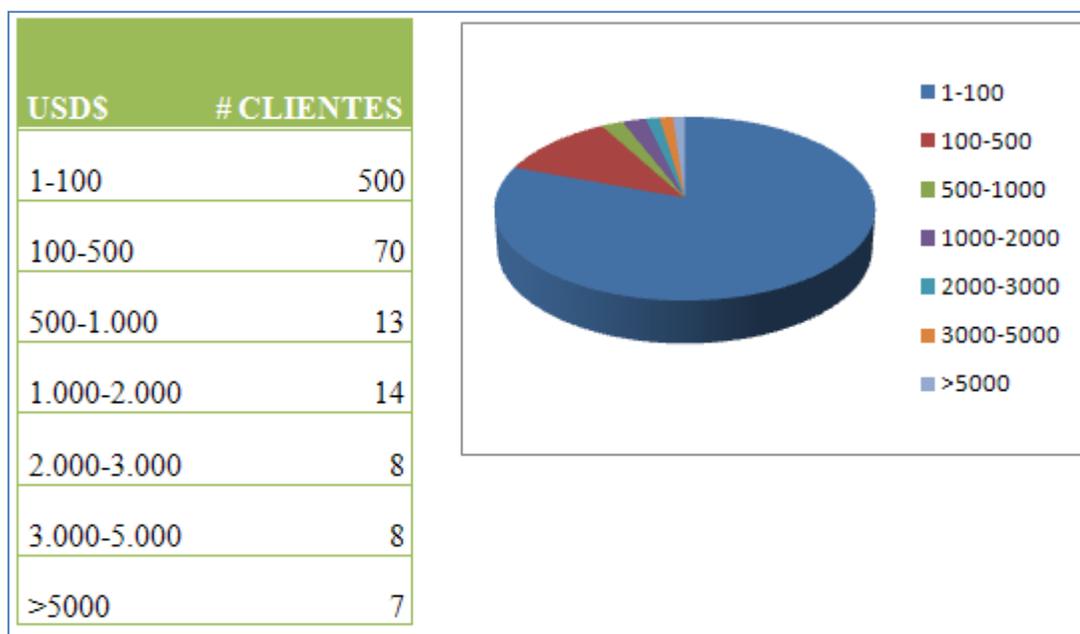


Figura 3. Segmentación de clientes.

Fuente: Cooperativa Unión Cochapamba.

1.4.2. Préstamos

Plazos y montos de préstamos	
Crédito	Descripción
Pecuarios:	Se prestará un máximo de \$1.500 USD, plazo máximo 2 años, pagos mensuales, trimestrales y máximo semestrales.
Agrícolas:	Se prestará un máximo de \$1.500 USD, plazo dependiendo del cultivo máximo 18 meses y pagos serán mensuales, trimestrales y máximo semestrales.
Salud:	Serán solo en casos de enfermedades del socio o de los familiares, plazo máximo de 1 año, pagos trimestrales y se prestará máximo hasta \$1.000 USD.
Viajes de trabajo:	Se entregará a personas que requieran fondos para viajes de trabajo en el exterior, se presta un monto máximo de \$1000 USD.
Viviendas:	Plazo máximo 4 años, pagos mensuales, trimestrales, semestrales, monto máximo \$5.000 USD.
Compra de terrenos:	Plazo máximo será de 4 años pagaderos mensuales y trimestrales y un monto máximo de \$5.000 USD.
Educación:	Plazo hasta 18 meses, monto máximo hasta de \$1.500 USD.
Crédito forestal:	Plazo hasta 3 años, monto hasta \$2.000 USD, pagos mensuales y trimestrales.
Credi-empresas:	Plazos hasta 4 años, monto máximo \$5.000 USD, pagos mensuales y trimestrales.

Tabla 2. Tipos de préstamos.

Fuente: Cooperativa Unión Cochapamba.

Capítulo 2: Marco Teórico



RESUMEN DEL CAPÍTULO

- *Normas legales.*
- *Estándares abiertos.*
- *Seguridad en sistemas web.*
- *Sistemas operativos.*
- *Servidor de aplicaciones.*
- *Bases de datos.*
- *Plataformas y frameworks.*
- *Herramientas de reportes.*

2. Marco teórico

En éste capítulo se presentan los conceptos generales de las normativas para instituciones financieras y las herramientas empleadas para el desarrollo del aplicativo.

2.1. Normas legales

“La regulación y supervisión de las instituciones financieras son elementos muy importantes para la seguridad financiera, puesto que permiten establecer normas prudenciales bien definidas y velan por su aplicación garantizando que el ascenso de riesgos sea limitado o, al menos, que los riesgos se encuentren apropiadamente cubiertos, de manera que se reduzca la probabilidad de crisis en el sistema bancario”¹.

La regulación financiera tiene como principal objetivo proteger a los pequeños depositantes y proteger el sistema de pagos y el sistema financiero, para evitar que se repitan las crisis bancarias y económicas.

2.1.1. Ley general de instituciones del sistema financiero

“ARTÍCULO 1.- La Ley regula la creación, organización, actividades, funcionamiento y extinción de las instituciones del sistema financiero privado, así como la organización y funcionamiento de la Superintendencia de Bancos, entidad encargada de la supervisión y control del sistema financiero en todo lo cual se tiene presente la protección de los intereses del público.

ARTÍCULO 2.- Son instituciones financieras privadas:

- a. Bancos;
- b. Sociedades Financieras;
- c. Asociaciones Mutualistas de Ahorro y Crédito para la Vivienda; y,
- d. Cooperativas de Ahorro y Crédito que realizan intermediación financiera con el público”²

2.1.2. Ley de Cooperativas

“La Ley de Cooperativas y su Reglamento General son preceptos legales de aplicación obligatoria para el sistema cooperativo ecuatoriano. En 1937 bajo la presidencia del Gral. Alberto Enríquez Gallo, se promulga la primera Ley de Cooperativas del Ecuador... La segunda Ley de Cooperativas que se

¹ (<http://www.dspace.espol.edu.ec/>, p. 12)

² (<http://www.sbs.gob.ec/medios>)

encuentra en vigencia hasta la fecha actual y cuya codificación fue publicada en Registro Oficial No. 400 del 29 de agosto del 2001...

Por mandato legal, corresponde a la Dirección Nacional de Cooperativas, ejecutar las políticas estatales de promoción del sector y realizar todos los trámites para la aprobación y registro de las organizaciones Cooperativas; las fiscaliza y asesora; aprueba sus planes de trabajo y aplica inclusive sanciones morales y pecuniarias, respetando obviamente la autonomía de las Cooperativas, en el sentido de que éstas se rigen además de la Ley y Reglamento General de Cooperativas, por sus propios Estatutos y sus Reglamentos Internos.”³

“El reglamento de la Ley de Cooperativas rige la constitución, organización, funcionamiento y liquidación de las Cooperativas de ahorro y crédito que realizan intermediación financiera con el público, así como de las Cooperativas de segundo piso, sujetas al control de la Superintendencia de Bancos y Seguros”⁴

2.1.3. Normas de Basilea

La situación crítica que afectó a las instituciones financieras del Ecuador, impulsó la importancia en la aplicación de las normas sobre regulación bancaria emitidas por el Comité de Basilea.

“La Junta Bancaria con el apoyo de organismos multilaterales de crédito adoptaron la aplicación de una serie de reformas para homologar la legislación bancaria a las denominadas Normas de Basilea, que son una guía para armonizar políticas y procedimientos que faciliten la coordinación en el sistema de control eficaz de la banca, teniendo como objetivo el asegurar que los entes financieros midan, vigilen y controlen adecuadamente los riesgos de liquidez y de mercado, como una garantía de solvencia y de confianza para sus clientes.

Las Normas de Basilea son el resultado de las decisiones del grupo de los 10 países más industrializados del mundo, quienes representados por los bancos centrales y organismos de supervisión de cada país miembro, han elaborado normas de regulación prudencial para el sistema bancario de cada nación; Éste Comité periódicamente revisa y ajusta dicha normativa a los requerimientos del momento.”⁵

³ (García, 2008)

⁴ (<http://www.sbs.gob.ec>)

⁵ (Pérez)

2.2. Estándares abiertos

Estándar abierto es una especificación disponible públicamente para lograr una tarea específica.

Definición.- “Un estándar abierto ofrece el medio para hacer algo que es independiente del fabricante o vendedor, habilitando así a programas de software en competencia para usar los mismos formatos de archivo libremente; HTML, XML y ODF son ejemplos de estándares abiertos para documentos.

Un estándar abierto cumple los siguientes requisitos:

- Está bien documentado, con la especificación completa disponible públicamente, ya sea gratis o por un cargo normal.
- Puede ser copiado, distribuido y usado libremente. La propiedad intelectual del estándar está irrevocablemente disponible sin necesidad de pagar derechos.
- Es estandarizado y mantenido en un foro independiente y abierto (también denominado “organización de estándares”) usando un proceso abierto.”⁶

2.2.1. HTML

HTML es una implementación del estándar SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.

2.2.1.1. Características:

- a) Permite crear lenguajes de codificación descriptivos.
- b) Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- c) Proporciona una especificación formal completa del documento.
- d) No tiene un conjunto implícito de convenciones de señalización.
- e) Soporta un conjunto flexible de juegos de etiquetas.
- f) Los documentos generados por él son legibles.

⁶ (Barnes)

2.2.1.2. Sintaxis general:

“Son válidos todos los caracteres incluidos en ISO 8859-1, El formato es libre. El formato introducido en el fichero fuente (saltos de línea, líneas en blanco, etc.) es irrelevante para el formato final del documento.

- a) Caracteres de significado especial:
 - < Marca el comienzo de una etiqueta.
 - > Marca el final de una etiqueta.
 - & Marca el comienzo de una referencia a entidad

En algunos navegadores las letras acentuadas y algunos caracteres como la “ñ” no se visualizan correctamente, por lo cual deben ser sustituidos por la referencia a su entidad.

Etiquetas: Son los textos que delimitan los distintos elementos que componen un documento. No son sensibles a mayúsculas y minúsculas.

Hay dos tipos de etiquetas:

- a) Etiquetas de comienzo de elemento. Delimitadas por los caracteres “<” y “>”
- b) Etiquetas de final de elemento. Delimitadas por los caracteres “</” y “>”

Estructura de una etiqueta:

- c) De comienzo:
 - <identificador [atributos]>
- d) De final:
 - </identificador>

2.2.1.3. Estructura de un documento HTML

Un documento HTML consta de las siguientes piezas:

- a) Identificación SGML
- b) Una etiqueta <HTML>
- c) Cabecera (iniciada por la etiqueta <HEAD> y cerrada por </HEAD>)
- d) Cuerpo del documento (iniciada por la etiqueta <BODY> y cerrada por </BODY>)
- e) Una etiqueta de fin de documento </HTML>

```
<!DOCTYPE PUBLIC HTML "-//IETF/DDT HTML 2.0/EN">
<HTML>
<HEAD>

...

</HEAD>
<BODY>

...

</BODY>
</HTML>
```

Figura 4. Documento HTML

Fuente: Fuente especificada no válida.

2.2.2. CSS (Cascading Style Sheets)

CSS es un lenguaje estándar usado para definir la presentación de un documento estructurado escrito en HTML, XHTML o XML.

La función principal es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

```
body {
  margin: 4px;
  border: 3px dotted #;
  font-family: sans-serif;
  color: #000000;
  background-color: #FFFFFF;
}

h1 {
  padding: 5px;
  margin: 10px;
  border: 1px solid #C0C0C0;
  color: #FF0000;
  background-color: #0000FF;
}
```

Figura 5. Estilo CSS

Fuente: (<http://upload.wikimedia.org/wikipedia/commons/thumb/8/86/CSS.svg/160px-CSS.svg.png>)

2.2.2.1. Formas de usar CSS

“Para dar formato a un documento HTML, puede emplearse CSS de tres formas distintas:

1. Mediante CSS introducido por el autor del HTML
 - a. **Un estilo en línea** (inline) es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página. Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo. No es todo lo claro o estructurado que debería ser, pero funciona.

Dado que los clientes de correo electrónico no soportan las hojas de estilos externas, y que no existen estándares que los fabricantes de clientes de correo respeten para utilizar CSS en este contexto, la solución más recomendable para maquetar correos electrónicos, es utilizar CSS dentro de los propios elementos (inline).
 - b. **Una hoja de estilo interna**, que es una hoja de estilo que está incrustada dentro de un documento HTML, dentro del elemento <head>, marcada por la etiqueta <style>. De esta manera se obtiene el beneficio de separar la información del estilo del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página Web.
 - c. **Una hoja de estilo externa**, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.
2. Estilos CSS introducidos por el usuario que ve el documento, mediante un archivo CSS especificado mediante las configuraciones del navegador, y que sobre escribe los estilos definidos por el autor en una, o varias páginas web.
3. Los estilos marcados "por defecto" por los user agent, para diferentes elementos de un documento HTML, como por ejemplo, los enlaces

2.2.2.2. Ventajas

Algunas ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- a. Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- b. Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.
- c. Optimización del ancho de banda de la conexión, pues pueden definirse los mismos estilos para muchos elementos con un sólo selector; o porque un mismo archivo CSS puede servir para una multitud de documentos.
- d. Mejora en la accesibilidad del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño (como las tablas), y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales”⁷.

2.3. Sistemas operativos (SO)

Un sistema operativo es un conjunto de programas que gestiona los recursos o hardware y proporciona servicios a los programas de aplicación que emplea el usuario final.

2.3.1. Introducción

“Sin el software, una computadora no es más que una masa metálica sin utilidad. Con el software, una computadora puede almacenar, procesar y recuperar información... El programa fundamental que controla todos los recursos de la computadora y proporciona la base sobre la cual pueden escribirse los programas de aplicación se denomina Sistema Operativo.”⁸

2.3.2. Definición

Un Sistema Operativo es el software que se encarga de controlar y coordinar adecuadamente el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios. Es un administrador de los recursos de hardware del sistema.

⁷ (<http://es.wikipedia.org>)

⁸ (Ruiz, Candido)

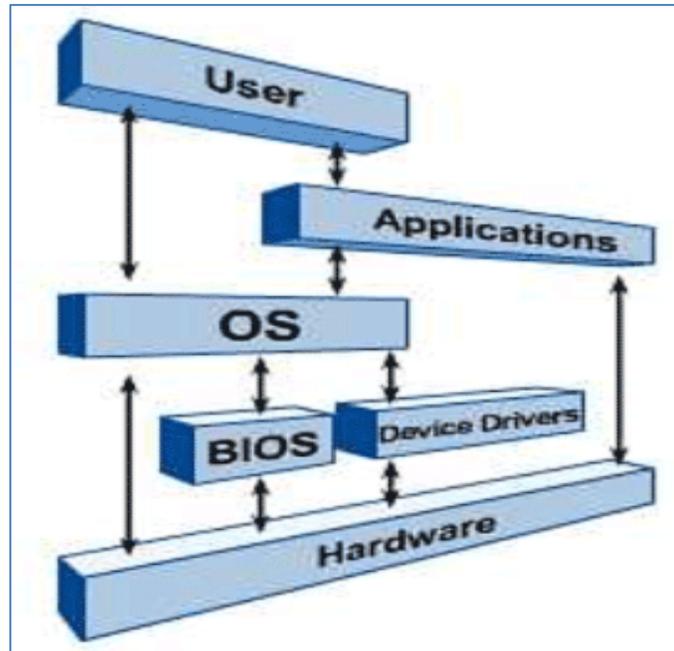


Figura 6. Función de los sistemas operativos.

Fuente: Fuente especificada no válida.

El SO es el encargado de brindar al usuario una forma amigable y sencilla de operar, interpretar, codificar y emitir las órdenes al procesador central para que éste realice las tareas necesarias y específicas para completar una orden.

2.3.3. Características

“En general, se puede decir que un Sistema Operativo tiene las siguientes características:

- a. Conveniencia.- Hace más conveniente el uso de una computadora.
- b. Eficiencia.- Permite que los recursos de la computadora se usen de la manera más eficiente posible.
- c. Habilidad para evolucionar.- Deberá construirse de manera que permita el desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.
- d. Administrar el hardware.- Se encarga de manejar de una mejor manera los recursos de la computadora en cuanto a hardware se refiere, esto es, asignar a cada proceso una parte del procesador para poder compartir los recursos.
- e. Relacionar dispositivos.- Se debe encargar de comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.

- f. Manejar las comunicaciones en red.- Permite al usuario manejar con alta facilidad todo lo referente a la instalación y uso de las redes de computadoras.
- g. Facilitar las entradas y salidas.- Debe hacerle fácil al usuario el acceso y manejo de los dispositivos de Entrada/Salida de la computadora.
- h. Evita que otros usuarios interfieran.- Evita que los usuarios se bloqueen entre ellos, informándoles si esa aplicación está siendo ocupada por otro usuario.
- i. Procesamiento por bytes de flujo a través del bus de datos.
- j. Organizar datos para acceso rápido y seguro.
- k. Técnicas de recuperación de errores.
- l. Generación de estadísticas.
- m. Permite que se puedan compartir el hardware y los datos entre los usuarios.”⁹

Ejemplos de sistema operativo

Familia UNIX	- Familia GNU/Linux	Familia Windows	- Familia Macintosh
- AIX	- Arch Linux	- Windows NT	- Mac OS 7
- GNU / Linux	- Debian	- Windows 2000	- Mac OS 8
- GNU / Hurd	- Fedora	- Windows XP	- Mac OS 9
- HP-UX	- Ubuntu	- Windows Server 2003	- Mac OS X
- IRIX	- OpenSuse	- Windows Mobile	
- Minix	- Centos	- Windows Server 2008	
- System V	- Mandriva	- Windows Vista	
- Solaris	- Red Hat	- Windows 7	

Tabla 3. Lista de sistemas operativos.

Fuente: Fuente especificada no válida.

⁹ (uzcátegui)

Tabla comparativa de los sistemas operativos Windows y Linux

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

Tabla 4. Comparativa de sistemas operativos Windows y Linux

Fuente: Fuente especificada no válida.

Fedora



Para el desarrollo del software FINANSYS, hemos decidido usar como sistema operativo Fedora, por ser una distribución muy estable, orientada al desarrollo y más que todo, es Open Source.

Fedora es un sistema operativo de propósitos generales basado en Linux, es una colección de software que hace funcionar un computador. Éste sistema operativo es libre y gratuito, se mantiene gracias a una comunidad internacional (Proyecto Fedora) de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías¹⁰

2.3.3.1. Proyecto Fedora

Es el nombre de una comunidad de personas en todo el planeta que comparten la filosofía, utilizan y construyen software libre. La meta de esta comunidad, es liderar la creación y la distribución tanto de código como de contenidos libres. Fedora es patrocinado por Red Hat, el proveedor de tecnología de código abierto más confiable en todo el mundo. Red Hat invierte en Fedora para estimular la colaboración y la innovación en tecnologías de software libre.¹¹

Algo que hay que destacar es que los desarrolladores prefieren hacer cambios en las fuentes originales en lugar de aplicar los parches específicos en su distribución, de esta forma se asegura que las actualizaciones estén disponibles para todas las variantes de GNU/Linux.

En la actualidad han lanzado 16 versiones de este sistema operativo, durante sus primeras 6 versiones se llamó Fedora Core, debido a que solo incluía los paquetes más importantes del sistema operativo. La última versión es Fedora 16 Verne, puesta a disposición del público el 8 de Noviembre del 2011. El anuncio de la versión #17 denominada Beefy Miracle cuya versión final está programada para 8 de mayo de 2012.

Cuenta con una buena documentación, traducida en alrededor de 40 idiomas y disponible en múltiples formatos; La documentación de Fedora describe como instalar y usar el sistema operativo y el software empaquetado.

¹⁰ (<http://es.wikipedia.org>)

¹¹ (<http://fedoraproject.org/es>)

Cuenta con soporte o ayuda interactiva a través de foros, por chat o soporte a través de correo electrónico, brindando de manera fácil ayuda en cualquier problema que pueda surgir.

La guía de seguridad detalla la planificación y describe las herramientas involucradas en la creación de un entorno de computación seguro, para centros de datos, estaciones de trabajo, o el hogar. Con un conocimiento administrativo apropiado, vigilancia, y herramientas, los sistemas ejecutando Linux pueden ser funcionales y al mismo tiempo seguro frente a los métodos de intrusión y explotación más comunes.¹²

2.3.3.2. Características

- a. Trata de ser el líder en el ámbito de usar software libre y código abierto.
- b. Prefieren hacer cambios en las fuentes que aplicar un parche específico para su distribución, de esta forma se asegura que las actualizaciones estén disponibles para todas las variantes de Linux.
- c. Yum es el administrador de paquetes del sistema.
- d. Se destaca en seguridad y utiliza SELinux entre otras medidas de seguridad.
- e. Soporta redes instaladas sobre HTTP, FTP y NFS.
- f. El entorno de escritorio por defecto es GNOME, e incluye KDE en algunas versiones.
- g. Está diseñado para ser fácil de instalar y configurar, incluyendo para esto instaladores y herramientas gráficas.
- h. El sistema de archivos por defecto es ext3 sobre LVM.
- i. Su navegador por defecto es Firefox desde su versión Fedora Core 3 y superior.
- j. Incluye el paquete ofimático OpenOffice.org desde su versión 4.

2.3.4. CentOS



Para el alojamiento del software FINANSYS, nos hemos inclinado por el magnífico sistema operativo CentOS, por ser una distribución GNU/Linux muy robusta y orientada a servidores.

CentOS es una distribución Linux de clase empresarial derivada de fuentes libremente ofrecidos al público por un prominente proveedor de América del Norte Enterprise Linux.¹³

¹² (<http://docs.fedoraproject.org>)

¹³ (<http://www.centos.org/>)

Red Hat Enterprise Linux se compone de software libre y código abierto, pero se publica en formato binario usable. Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la licencia pública general de GNU y otras licencias. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es muy similar al Red Hat Enterprise Linux y está libremente disponible para ser bajado y usado por el público, pero no es mantenido ni asistido por Red Hat. Existen otras distribuciones también derivadas de las fuentes de Red Hat.

Se ajusta plenamente a la política de redistribución del proveedor original y aspira a ser 100% compatible a nivel binario.

Es desarrollado por un creciente grupo de desarrolladores del núcleo. A su vez los desarrolladores principales son apoyados por una activa comunidad de usuarios como los administradores de sistemas, administradores de red, los usuarios empresariales, gerentes, colaboradores núcleo Linux y los entusiastas de Linux de todo el mundo.

Es una distribución empresarial plenamente funcional, estable y lista para usar sin el elevado costo económico de los servidores privativos.

Se puede usar como servidor web, servidor de correo, servidor de bases de datos o servidor de aplicaciones. Incorpora todas las sofisticadas funcionalidades de compatibilidad con Windows incluidas en Red Hat Enterprise Linux. Se puede usar CentOS como controlador de dominios primarios o secundarios, o como servidor de clientes Windows basado en Samba y SMB.

Es también un sistema de escritorio Linux plenamente funcional. Cuenta con la mayoría de las herramientas de escritorio más populares.

Ya sea trabajando en un servidor empresarial, en un servidor para la oficina o incluso en un sistema común de escritorio, es una distribución estable, de alto rendimiento, fácil de administrar y de utilizar¹⁴

2.3.4.1. Características

- a. Sumamente estable.
- b. Soporte de muchos programas comerciales o no, que soportan RedHat Enterprise Linux.
- c. Tiene varios repositorios que en suma alcanzan más de los 10 mil paquetes (entre base, extras, rpmforge, atrpms y el de karan).
- d. Yum hace todo el trabajo de instalación o desinstalación.¹⁵

¹⁴ (<http://www.linux-magazine.es>)

2.4. Servidor de aplicaciones Apache Tomcat



Un servidor de aplicaciones gestiona las funciones de lógica de negocio y de acceso a los datos de la aplicación. Un servidor de aplicaciones se diferencia de un servidor web por el uso de contenido dinámico y por su frecuente integración con bases de datos.

2.4.1. Introducción a Apache Tomcat

“Apache Tomcat es una implementación de software de código abierto de la tecnología Java Servlet y Java Server Pages, ambas desarrolladas bajo la Java Community Process.

Apache Tomcat es desarrollado en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2.”¹⁶

Tomcat es el servidor web más utilizado a la hora de trabajar con Java en entornos web.

Es una implementación completamente funcional de los estándares de JSP y Servlets. Puede especificarse como el manejador de las peticiones de JSP y Servlets recibidas por servidores web populares como el servidor Apache.

Apache es el servidor web configurable, robusto y estable.

Características de Apache:

- Corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
 - a) Apache es una tecnología gratuita de código fuente abierto.
 - b) Apache es un servidor altamente configurable de diseño modular.
 - c) Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas JSP. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
 - d) Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
 - e) Tiene una alta configurabilidad en la creación y gestión de logs.¹⁷

¹⁵_(Oliva, 2007)

¹⁶ (<http://tomcat.apache.org/>)

“Tomcat es mantenido y desarrollado por miembros de Apache software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License.

Las primeras distribuciones de Tomcat fueron las versiones 3.0x. Las versiones más recientes son las 7.x.

Incluye el compilador Jasper, que compila JSPs convirtiéndolas en Servlets. El motor de Servlets a menudo se presenta en combinaciones con el servidor web Apache.”¹⁸

2.4.2. Funcionamiento del servidor de aplicaciones Apache Tomcat

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

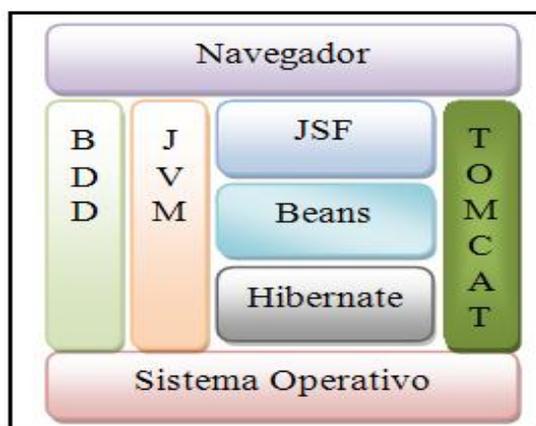


Figura 7. Funcionamiento de Apache Tomcat

Fuente: propia.

¹⁷ (<http://lists.centos.org>)

¹⁸ (Cuenca, 2012)

2.4.3. Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios

2.5. Seguridad en sistemas web

“Hoy en día Internet juega un papel fundamental en las comunicaciones, en el mundo de los negocios, la información, el entretenimiento, las finanzas, etc. Cada vez más, se desarrollan aplicaciones web que involucran información de carácter confidencial y que requieren mecanismos de seguridad que garanticen que dicha información no será modificada, sustraída o falsificada por personas ajenas. Esto es, un sistema web debe garantizar la autenticidad, integridad y confidencialidad de toda la información involucrada en el sistema.

2.5.1. Introducción

La principal vulnerabilidad de la información en una aplicación web, se presenta cuando la información viaja a través de la red, puesto que ésta puede ser sustraída por cualquier persona, no obstante, existen una serie de vulnerabilidades como la falsificación de identidad, el robo de sesiones autenticadas, la falsificación o modificación de los mensajes, entre otras. Una aplicación web que se diga segura, debe contemplar mecanismos que garanticen que efectivamente ninguna persona ajena al sistema, pueda modificar, obtener o falsificar datos de dicha aplicación web.

En la medida en que las aplicaciones web críticas (que involucren información confidencial como números de tarjetas de crédito, números de identificación personal, etc.) se consideren seguras, los usuarios harán un mayor uso de ellas, sin el temor de que la información que están proporcionando va a ser utilizada de forma ilícita.”¹⁹

¹⁹ (Aguilar, 2006)

Los problemas de seguridad relacionados con el protocolo HTTP se dividen en grandes grupos en función de los datos a los que puede afectar.

2.5.2. Seguridad en el cliente

Es necesario garantizar al usuario que descarga páginas de un servidor que no va a perjudicar a la seguridad de su equipo. Se deben evitar Applets maliciosos, programas con virus o simples cuelgues al acceder a las páginas.

2.5.3. Seguridad en la red

Cuando un usuario se conecta a un servidor web se produce un intercambio de información cliente-servidor; Es vital garantizar que los datos que recibe el cliente desde el servidor sean los mismos que se están enviando (esto es que no sufran modificaciones de terceros en el camino) y también garantizar que la información que el usuario envía al servidor no sea capturada, destruida o modificada por un atacante.

SSL (Secure Socket Layer) es un protocolo para asegurar el transporte de datos entre el cliente y el servidor web.

Diseñado inicialmente por Netscape, hoy día es soportado por la mayoría de los servidores web, se puede reconocer una conexión HTTP sobre SSL porque aparece el prefijo 'https' en lugar de 'http' en la URL



Figura 8. HTTP sobre SSL.

Fuente: propia.

La versión actual de SSL es la 3 y a partir de ella se está desarrollando un protocolo público que se conoce como TLS (Transport Layer Security) y es compatible con SSL.

SSL proporciona una comunicación segura entre cliente y servidor permitiendo la autenticación mutua, el uso de firmas digitales y garantizando la privacidad mediante encriptación. Una sesión SSL se establece según una secuencia de operaciones.

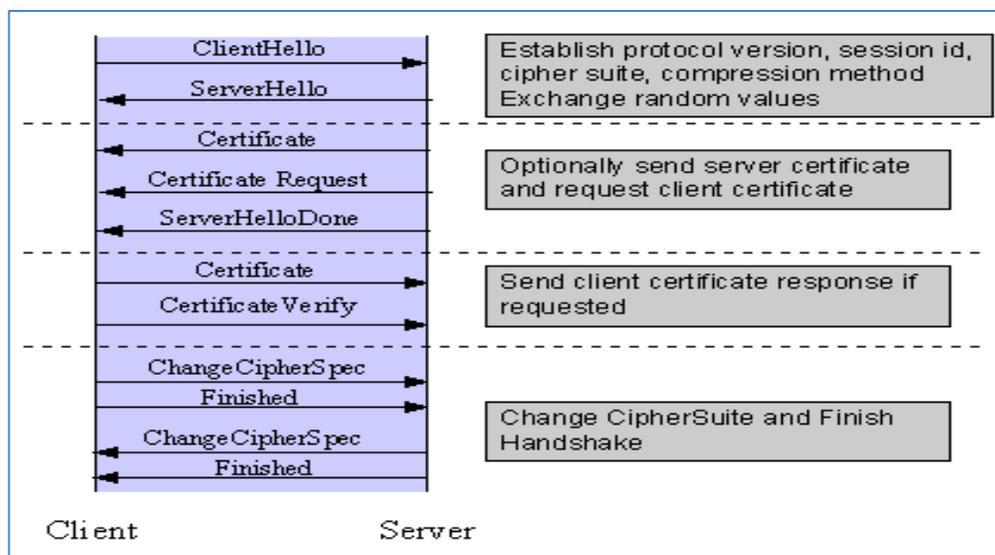


Figura 9. Sesión SSL

Fuente: seguridad en las aplicaciones web.ppt página 138

2.5.4. Seguridad en el servidor

Es necesario garantizar que la información almacenada en el servidor no pueda ser modificada sin autorización, que permanezca disponible y que sólo pueda ser accedida por los usuarios con los respectivos privilegios.

El desarrollo de una aplicación web requiere de una serie de herramientas: servidor web, servidor de aplicaciones, servidor de base de datos, lenguajes de servidor, etc., éstas herramientas pueden plantear problemas que comprometan a la aplicación:

- Vulnerabilidades debidas al uso de versiones no actualizadas.
- Configuraciones por defecto inadecuadas
- Activación de cuentas por defecto.

Las herramientas deben estar actualizadas y bien configuradas para impedir ataques a la aplicación.

2.5.4.1. Servidor web

Un servidor web proporciona muchos servicios y es muy probable que algunos de ellos no sean necesarios para el funcionamiento de la aplicación, en tal caso es conveniente deshabilitarlos.

Sugerencias:

- a) Establecer permisos adecuados para los ficheros del servidor y los documentos; Es conveniente definir un usuario y grupo especiales.
- b) Evitar seguimiento de enlaces simbólicos si se enlazan ficheros externos al árbol de documentos.
- c) Permitir el acceso sólo a usuarios de confianza y restringirlo a un directorio especial.
- d) Revisar periódicamente los ficheros de log para detectar posibles ataques al servidor.

2.5.4.2. Servidor de base de datos

Riesgos:

- a) Descubrimiento de información sobre datos de conexión al servidor o información sobre la estructura de la base de datos.
- b) Modificación de las instrucciones SQL enviadas al servidor, construidas de forma dinámica a partir de los datos recibidos del usuario.
- c) Acceso no autorizado a información restringida.

Sugerencias:

- a) Vigilar la configuración por defecto del servidor.
- b) No ejecutar el servidor como usuario administrador por defecto.
- c) Restringir el acceso remoto al servidor.
- d) No dar a un usuario más permisos que los que estrictamente necesite.
- e) Almacenar los datos sensibles de forma encriptada.
- f) Validar las instrucciones SQL antes de enviarlas al servidor.
- g) No revelar información sobre la base de datos en los mensajes de error.
- h) Proteger el código donde aparezca información sensible para el acceso al servidor.

2.5.5. Seguridad en la aplicación.

2.5.5.1. Control de acceso

Un aspecto muy importante de una aplicación web es el control de acceso de los usuarios a zonas restringidas de la aplicación; Intervienen dos conceptos: Autenticación, Autorización.

Autenticación.-Es el proceso de determinar si un usuario es quien dice ser.

Contraseñas.- Siempre que se utilizan contraseñas para autenticar usuarios hay que seguir unas sugerencias:

- a) Almacenas las contraseñas de forma segura.
- b) Actualizar las contraseñas periódicamente.
- c) Bloquear una cuenta cuando se detecta un número determinado de intentos de acceso incorrecto.
- d) Contar con políticas de recuperación de contraseñas.

Sesiones.- La gestión de las sesiones es responsabilidad del programador. Normalmente los lenguajes de servidor disponen de funciones diseñadas específicamente para ello.

Es recomendable establecer un tiempo límite de vida para la sesión.

Autorización.- Es el acto de comprobar si un usuario tiene el permiso adecuado para acceder a un cierto fichero o realizar una determinada acción, una vez que ha sido autenticado.

Sugerencias:

- a) Determinar la información que será accesible para cada usuario.
- b) Determinar el nivel de acceso de cada usuario.
- c) Especificar un mecanismo para otorgar y revocar permisos a los usuarios.

2.5.5.2. Validación de datos de entrada

El problema más frecuente que se presenta en las aplicaciones web es no validar correctamente los datos de entrada; Esto da lugar a algunas vulnerabilidades.

Sugerencias:

- a) Aceptar únicamente datos válidos conocidos.
- b) Rechazar datos no válidos.
- c) Mostrar los errores al digitar datos no válidos.
- d) Toda la validación de datos debe realizarse en el servidor.

2.6. Bases de datos

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

Las bases de Datos tienen muchos usos: nos facilitan el almacenamiento de grandes cantidades de información; permiten la recuperación rápida y flexible de información, con ellas se puede organizar y reorganizar la información, así como imprimirla o distribuirla en formas diversas²⁰

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



2.6.1. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual. Es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. PostgreSQL está considerado como la base de datos de código abierto más avanzada. Proporciona un

²⁰ (<http://html.rincondelvago.com>)

gran número de características que normalmente sólo se encontraban en las bases de datos comerciales.²¹

Se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que puede funcionar en la mayoría de los sistemas operativos, puede ser integrada al ambiente Windows permitiendo a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes. Permite desarrollar o migrar aplicaciones desde Access, Visual Basic, FoxPro, Visual FoxPro, C/C++, Visual C/C++, Delphi, etc., para que utilicen a PostgreSQL como servidor de BD.

A continuación se muestra un gráfico que ilustra de manera general los componentes más importantes:

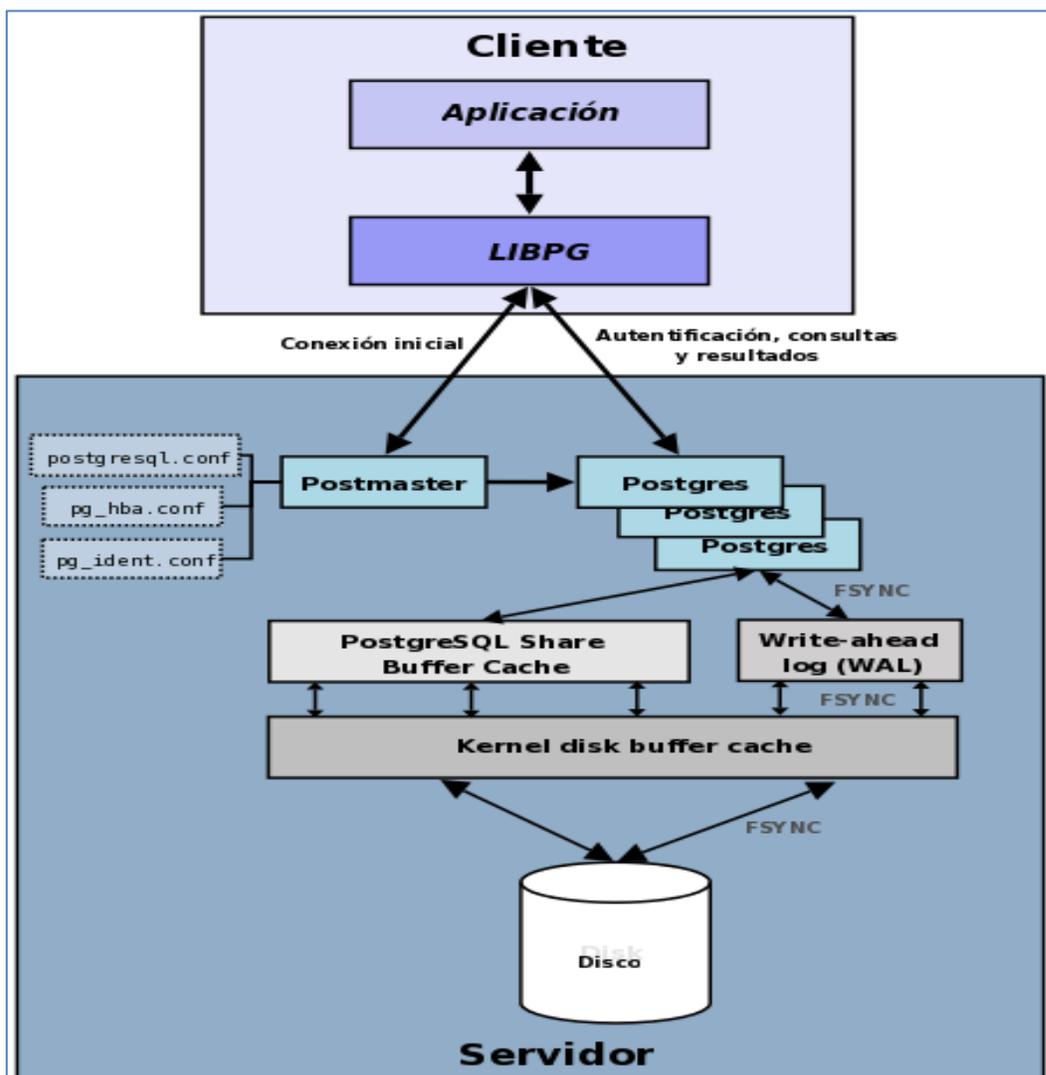


Figura 10. Componentes de Postgres.

Fuente: Fuente especificada no válida.

²¹ (<http://www.manualesdeayuda.com>)

- a) **Aplicación cliente:** Es la aplicación cliente que se utiliza como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- b) **Demonio postmaster:** Es el proceso principal. Es el encargado de escuchar por un puerto/socket las conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargarán de autenticar dichas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- c) **Ficheros de configuración:** Los tres ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf.
- d) **Procesos hijos postgres:** Procesos hijos que se encargan de autenticar a los clientes, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- e) **PostgreSQL share buffer cache:** Memoria compartida usada para almacenar datos en caché.
- f) **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- g) **Kernel disk buffer cache:** Caché de disco del sistema operativo.
- h) **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que funcione

2.6.1.1. Propiedades

- a) **Atomicidad.-** (Indivisible) es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- b) **Consistencia.-** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- c) **Aislamiento.-** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generará ningún tipo de error.
- d) **Durabilidad.-** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

2.6.1.2. Características

- a) Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, entre otros.
- b) Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
- c) Comunidades muy activas, varias comunidades en castellano.
- d) Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, etc.

- e) Drivers: Odbc, Jdbc, .Net, etc.
- f) Soporte de todas las características de una base de datos profesional (triggers, storeprocedures – funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.).
- g) Implementación del estándar SQL92 y SQL99.
- h) Soporte de protocolo de comunicación encriptado por SSL.
- i) Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.
- j) Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- k) Permite la declaración de funciones propias, así como la definición de disparadores y la creación de tipos de datos propios.
- l) Soporta el uso de índices, reglas y vistas.
- m) Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- n) Diseñada para entornos con altos volúmenes de tráfico/transacciones.²²

Algunos de los límites de PostgreSQL son:

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende del sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 – 1.600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Tabla 5. Límites de PostgreSQL

Fuente: Fuente especificada no válida.

²² (http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf)

2.7. Factibilidad de plataformas y frameworks

Para el desarrollo del aplicativo se optó por la utilización de plataformas y frameworks que viabilizan el desarrollo del sistema, dado la facilidad de uso, la documentación existente y el acoplamiento a las necesidades requeridas en las interfaces de usuarios de la Cooperativa.

Dichas plataformas y frameworks se detallan a continuación, en ellos se resalta los aspectos más relevantes y motivos primordiales por los cuales han sido elegidas:

2.7.1. Plataforma Java

Una plataforma es el ambiente de hardware o software en el cual se ejecutan los programas. En general, la mayoría de las plataformas pueden ser descritas como una combinación de hardware y sistema operativo. Algunas de las plataformas más populares son Windows, Solaris, Linux y MacOS.

2.7.1.1. Definición

La plataforma Java difiere de las anteriores en que ésta es una plataforma basada únicamente en software que corre por encima de las plataformas basadas en hardware.

La plataforma Java consta de dos componentes:

- La Máquina Virtual de Java (JVM)
- La Interfaz de Programación de Aplicaciones de Java (API Java)

A continuación se muestra una representación de los elementos que forman parte de la plataforma Java.

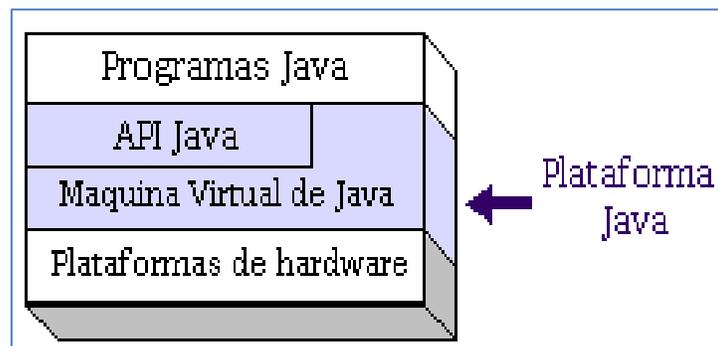


Figura 11. Plataforma JAVA.

Fuente: (uzcátegui)

“Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.”²³

Existe un gran número de aplicaciones y sitios web que no funcionan a menos que Java esté instalado, y muchas más que se crean a diario. Java es rápido, seguro y fiable. Se encuentra presente en portátiles, centros de datos, consolas de juegos, súper equipos científicos, teléfonos móviles, Internet, Java está en todas partes.

2.7.1.2. Tipos de programas en Java

Los programas en Java suelen estar en una de las siguientes categorías:

Applets.- Los applets son pequeños programas que se incorporan en una página Web y que por lo tanto, necesitan de un Navegador Web compatible con Java para poder ejecutarse. A menudo los applets se descargan junto con una página HTML desde un Servidor Web y se ejecutan en la máquina cliente.

Aplicaciones.- Las aplicaciones son programas independientes de propósito general que normalmente se ejecutan desde la línea de comandos del sistema operativo. Con Java se puede realizar cualquier programa que normalmente se crearía con algún otro lenguaje de programación.

Servlets.- Los servlets al contrario de los applets son programas que están pensados para trabajar en el lado del servidor y desarrollar aplicaciones Web que interactúen con los clientes.

2.7.2. JSF (Java Server Faces)

JSF es un framework de desarrollo web basado en Java.

2.7.2.1. Introducción

“El desarrollo de aplicaciones o sistemas web sin el uso de una metodología o un patrón de diseño que permita mantener un orden y una separación de la codificación y demás elementos hace que en muchas

²³ (<http://www.java.com>)

ocasiones se mezcle en un mismo archivo JSP la interfaz de usuario, las reglas de validación, el acceso a la base de datos, etc. Lo cual impide el mantenimiento de la aplicación y a la vez acorta su tiempo de vida útil, es por ello que se hace necesaria la utilización de herramientas que faciliten el desarrollo y nos permitan trabajar de manera ordenada. JSF ofrece un marco de trabajo que permite desarrollar aplicaciones, separando las diferentes capas de una arquitectura: presentación, reglas y entidades de negocio.²⁴

2.7.2.2. Definición

“JSF es una tecnología y framework para aplicaciones Java, orientadas a la web que simplifican el desarrollo de interfaces de usuario en aplicaciones JEE. JSF usa Java Server Pages (JSP) como tecnologías que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

Es un framework orientado a la interfaz gráfica de usuario, facilitando el desarrollo de éstas, y que sin embargo, realiza una separación entre comportamiento y presentación, además proporciona su propio servlet como controlador, implementando así los principios del patrón de diseño MVC (Modelo Vista Controlador), lo que da como resultado un desarrollo más simple y una aplicación mejor estructurada.

2.7.2.3. Ventajas

- a. Es una especificación estándar en que pueden encontrarse implementaciones de distintos fabricantes, esto permite no vincularse exclusivamente con un proveedor concreto y poder seleccionar el que más interese a cada caso según el número de componentes que suministra, el rendimiento de éstos, soporte proporcionado, precio, políticas de evolución, etc.
- b. Es muy flexible, permite crear propios componentes y crear propios renderizadores para pintar los componentes de la forma más conveniente.
- c. Ofrece una clara separación entre el comportamiento y la presentación.
- d. Separar la lógica de negocios de la presentación permite a cada miembro del equipo de desarrollo de la aplicación web se centre en su parte asignada del proceso de diseño, y proporciona un modelo sencillo de programación para enlazar todas las piezas.
- e. Mejora los conceptos de componente-UI y capa web sin limitar a una tecnología de script particular o lenguaje de marcas.
- f. Proporciona una amplia arquitectura para manejar el estado de los componentes, procesar los datos,

²⁴ (<http://www.proactiva-calidad.com/>)

validar la entrada del usuario y manejar eventos”.²⁵

2.7.2.4. Ciclo de vida

Un aspecto importante dentro de JSF es su ciclo de vida, el cual es similar al de una página JSP; el cliente hace una petición HTTP y el servidor responde con la página HTML.

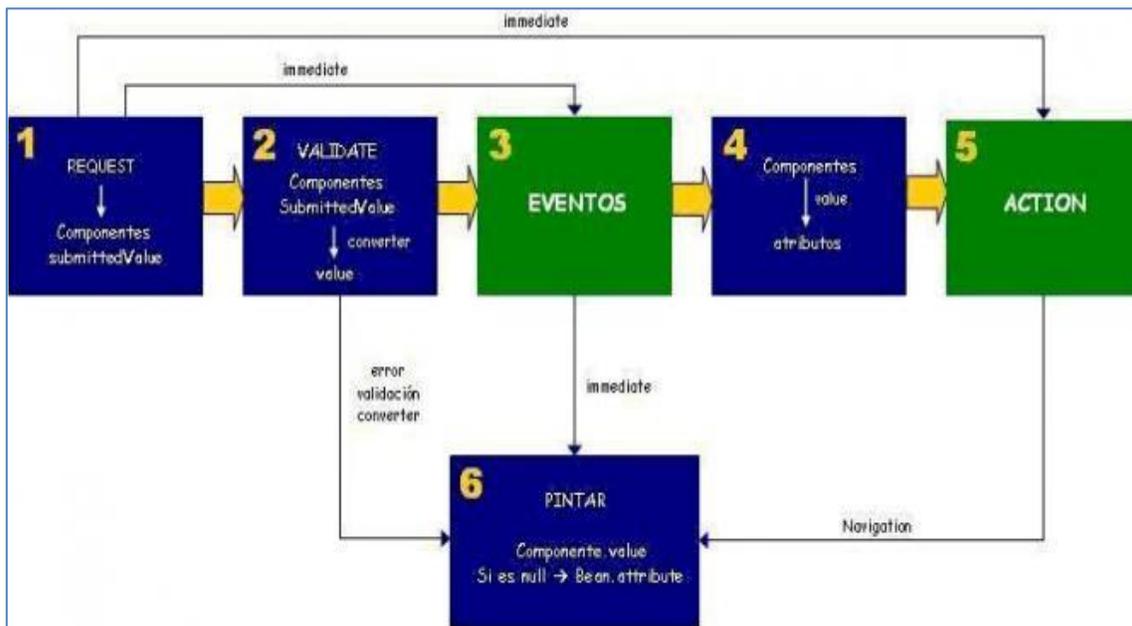


Figura 12. Ciclo de vida de JSF

Fuente: (<http://www.postgresql.org.es/>)

2.7.2.5. Estructura de una aplicación JSF

Se necesitan dos cosas para correr aplicaciones con JSF: un contenedor web para Java y una implementación de la especificación de JSF.

Los elementos que conforman una aplicación son típicamente los siguientes:

- Archivos JSP, que constituyen la interfaz gráfica de la aplicación y que contienen las diversas funcionalidades de JSF.
- Archivos XML, específicamente el archivo faces-config.xml que almacena las diferentes configuraciones y elementos a utilizar en la aplicación.
- Archivos Java, desempeñando el rol de beans.

²⁵ (<http://www.sicuma.uma.es>)

d) Archivos de paquetes de mensajes (message bundles).

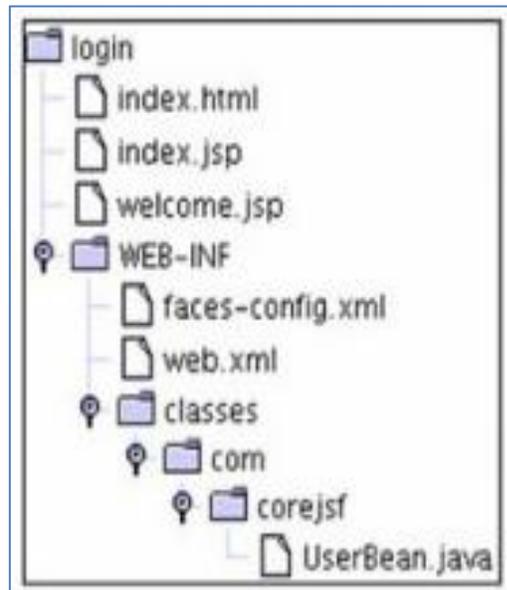


Figura 13. Estructura de directorio de una aplicación JSF.

Fuente: Fuente especificada no válida.

2.7.3. JSF con Netbeans

Netbeans es un entorno de desarrollo integrado el cual se describe más adelante, JSF se integra en Netbeans desde la versión 5.0, mediante bibliotecas que contienen los siguientes archivos jar:

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

Las cuales contienen diferentes clases necesarias para usar los tags en las páginas jsp. El uso de jsf en Netbeans solo es necesario crear un proyecto nuevo y especificar el uso de JSF.

2.7.4. Netbeans IDE

2.7.4.1. Introducción

NetBeans es un entorno de desarrollo integrado - una herramienta para programadores creada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para otros lenguajes de programación.

Es de código abierto escrito completamente en Java, soporta el desarrollo de todos los tipos de aplicación Java.

2.7.4.2. Definición

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto Netbeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.²⁶

2.7.4.3. Ventajas

- a) Multilenguaje completo y modular
 - Soporte para Java SE, Java EE, Java ME.
 - Gran cantidad de módulos de terceros (plugins).
 - Desarrollo intuitivo.
 - Soporte para Java, C/C++, Ruby on Rails, PHP, Groovy, Python, JavaScript entre otros.
- b) Gratis y Open Source.
 - Open source desde el 2000.
 - Gran comunidad de usuarios y desarrolladores.
- c) Plataforma para construir aplicaciones
 - Aplicaciones completas para el cliente.
 - Características visuales para el desarrollo web.
 - Creador gráfico de juegos para celular.
 - Crea fácilmente ventanas, menús, barras de herramientas, etc.
 - Se pueden construir aplicaciones de escritorio, web, móviles, Enterprise.
- d) Multi plataforma.
 - Funciona en OpenSolaris, Linux, Windows y Mac OS.

2.7.5. Java EE (Java Platform, Enterprise Edition)

2.7.5.1. Introducción a JEE

Java Platform, Enterprise Edition (antes J2EE, ahora Java EE) es un estándar para el desarrollo de aplicaciones empresariales (portables, robustas, escalables y seguras) usando tecnología JAVA.

²⁶ (<http://es.wikipedia.org/wiki/NetBeans>)

2.7.5.2. Definición

“Java EE es una plataforma de programación -parte de la Plataforma Java- para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java Community Process / JCP.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc., y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), Java Server Pages y varias tecnologías de servicios web. Ello permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.”²⁷

²⁷ (http://es.wikipedia.org/wiki/Java_EE)

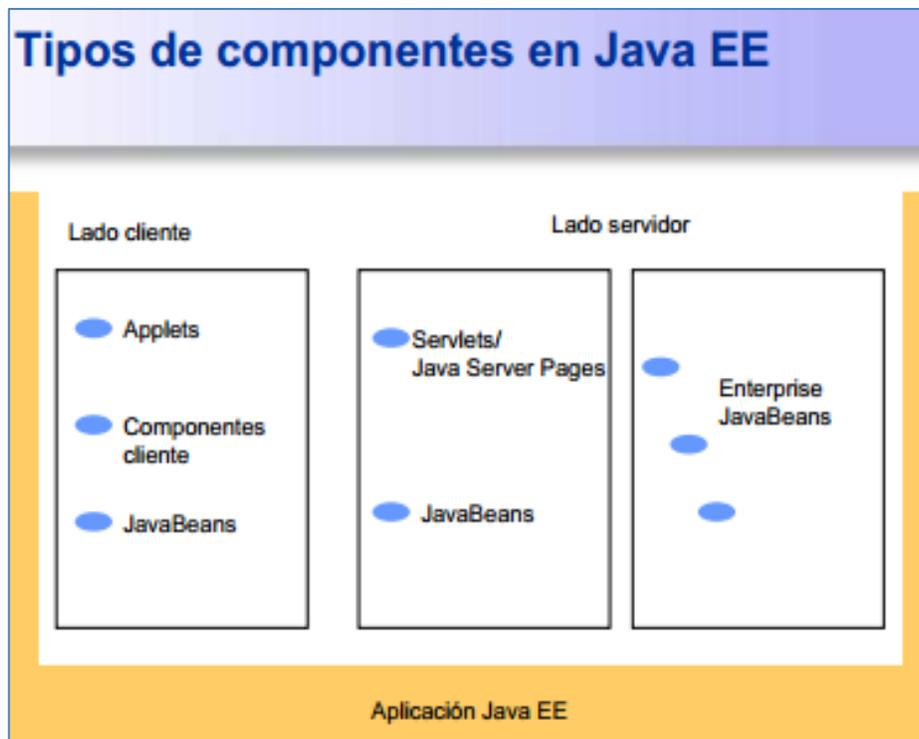


Figura 14. Componente en Java EE

Fuente: (http://ocw.uc3m.es/ingenieria-telematica/software-de-comunicaciones/transparencias/3_cmpnts-JavaEE.pdf, p. 19)

Applet.- Applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. A diferencia de un programa, un applet no puede ejecutarse de manera independiente. Un applet normalmente lleva a cabo una función muy específica que carece de uso independiente.²⁸

Servlets y Java Server Pages.- Son componentes a nivel de presentación, permiten generar páginas web dinámicas; En los servlets es más fácil controlar el flujo de acciones.

Las páginas JSP son un lenguaje de marcado basado en etiquetas, la representación de la información es más fácil; Internamente una JSP se traduce a servlet.

Enterprise JavaBean.- (EJB) es una completa especificación de arquitectura para componentes de servicio.

²⁸ (<http://es.wikipedia.org/wiki/Applet>)

Java EE se ha convertido en un estándar de la industria de desarrollo de software que permite trabajar aplicaciones empresariales basadas en servidores. Provee APIs que permiten administrar, entre otros, web services, modelo de componentes, comunicación entre objetos y que hacen posible la implementación de la arquitectura empresarial orientada a servicios (SOA) y aplicaciones web 2.0.

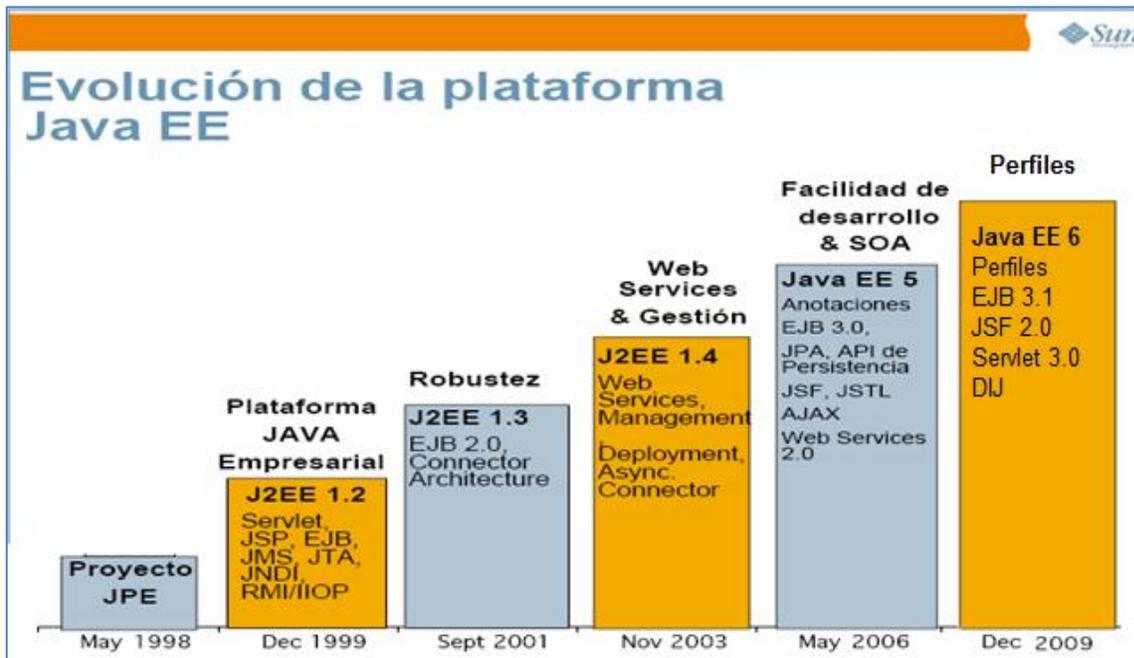


Figura 15. Evolución de la plataforma Java EE

Fuente: Fuente especificada no válida.

2.7.5.3. Arquitectura de JEE

En la arquitectura JEE se contemplan cuatro capas, en función del tipo de servicio y contenedores:

- Capa de cliente**, también conocida como capa de presentación o de aplicación.
- Capa web**. Intermediario entre el cliente y otras capas. Sus componentes principales son los servlets y las JSP. Aunque componentes de capa cliente (applets o aplicaciones) pueden acceder directamente a la capa EJB, lo normal es que Los servlets/JSPs pueden llamar a los EJB.

- c) **Capa Enterprise JavaBeans.** Permite a múltiples aplicaciones tener acceso de forma concurrente a datos y lógica de negocio. Los EJB se encuentran en un servidor EJB, que no es más que un servidor de objetos distribuidos. Un EJB puede conectarse a cualquier capa, aunque su misión esencial es conectarse con los sistemas de información empresarial
- d) **Capa de sistemas de información empresarial.**

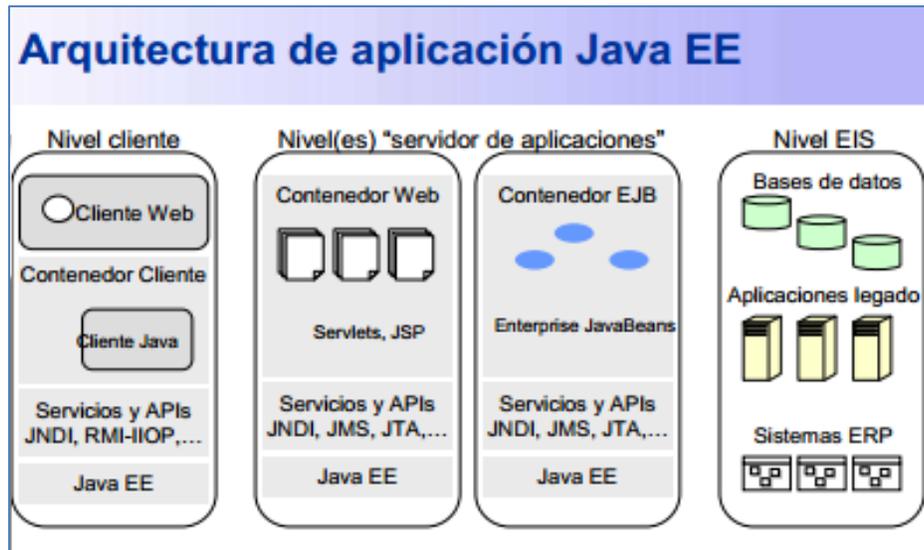


Figura 16. Arquitectura de aplicaciones Java EE

Fuente: (<http://ocw.uc3m.es/ingenieria-telematica/>, p. 18)

La visión de la arquitectura es un esquema lógico, no físico.²⁹

2.7.6. Hibernate



Es una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

2.7.6.1. Definición

Hibernate es una herramienta ORM (Mapeo Objeto-Relacional) completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente

²⁹ (<http://www.proactiva-calidad.com/java/>)

como el producto Open Source líder en este campo gracias a sus prestaciones, buena documentación y estabilidad.

2.7.6.2. Características

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

2.8. Herramienta de reportes

Un punto muy importante de los sistemas se presenta a la hora de mostrar la información resultante de los procesos.

2.8.1. Introducción a JasperReports

JasperReports es una herramienta para generar reportes en JAVA, con la habilidad de producir contenido completo en diferentes formatos.

2.8.2. Definición

“JasperReports es una herramienta de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible.

JasperReports se usa comúnmente con iReport, un front-end gráfico de código abierto para la edición de informes.

Se encuentra bajo licencia libre GNU, por lo que es Software libre. Forma parte de la iniciativa apilada open source Lisog.”³⁰

2.8.3. Requerimientos de JasperReports

Se requiere tener instalado en el equipo el JDK 1.4 (SDK) o posterior. No basta con tener instalado el J2RE (Run Time Environment).

2.8.4. Funcionamiento de JasperReports

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

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

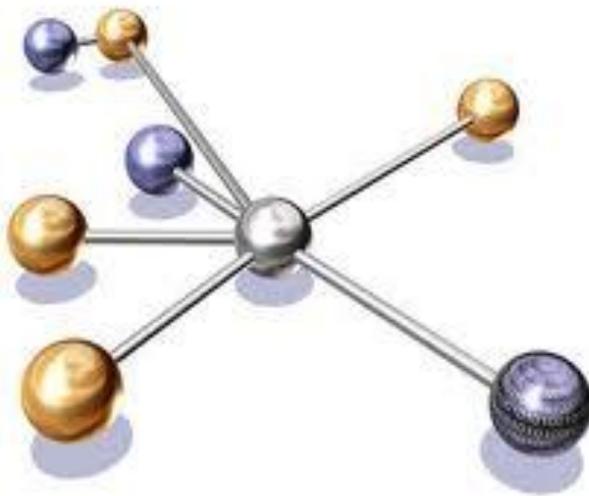
³⁰ (<http://es.wikipedia.org/wiki/JasperReports>)



Figura 17. Implementaciones con JasperReports

Fuente: (http://es.wikipedia.org/wiki/Java_EE)

Capítulo 3: Funcionamiento Del Sistema



RESUMEN DEL CAPÍTULO

- *Vista general.*
- *Arquitectura del sistema.*
- *Descripción de los módulos.*

3. Funcionamiento del sistema

3.1. Vista general

En el presente capítulo se detallan los Modelos Funcionales elaborados como base para el desarrollo informático del sistema integrado de información financiera.

En este documento la administración financiera se define como un sistema que integra los subsistemas de Administración y Seguridad, Ahorros, Préstamos-Cartera, Clientes, Caja, Contabilidad, y Auditoría Informática, los cuales se encuentran interrelacionados en cuanto a su normatividad, operatividad e información que generan, manteniendo así las características y necesidades propias de cada uno de los subsistemas.

A partir de la descripción de las funciones propias de cada uno de los departamentos existentes en la Cooperativa y de la información que generan, hemos diseñado al sistema integrado de información financiera teniendo como módulos principales los siguientes:

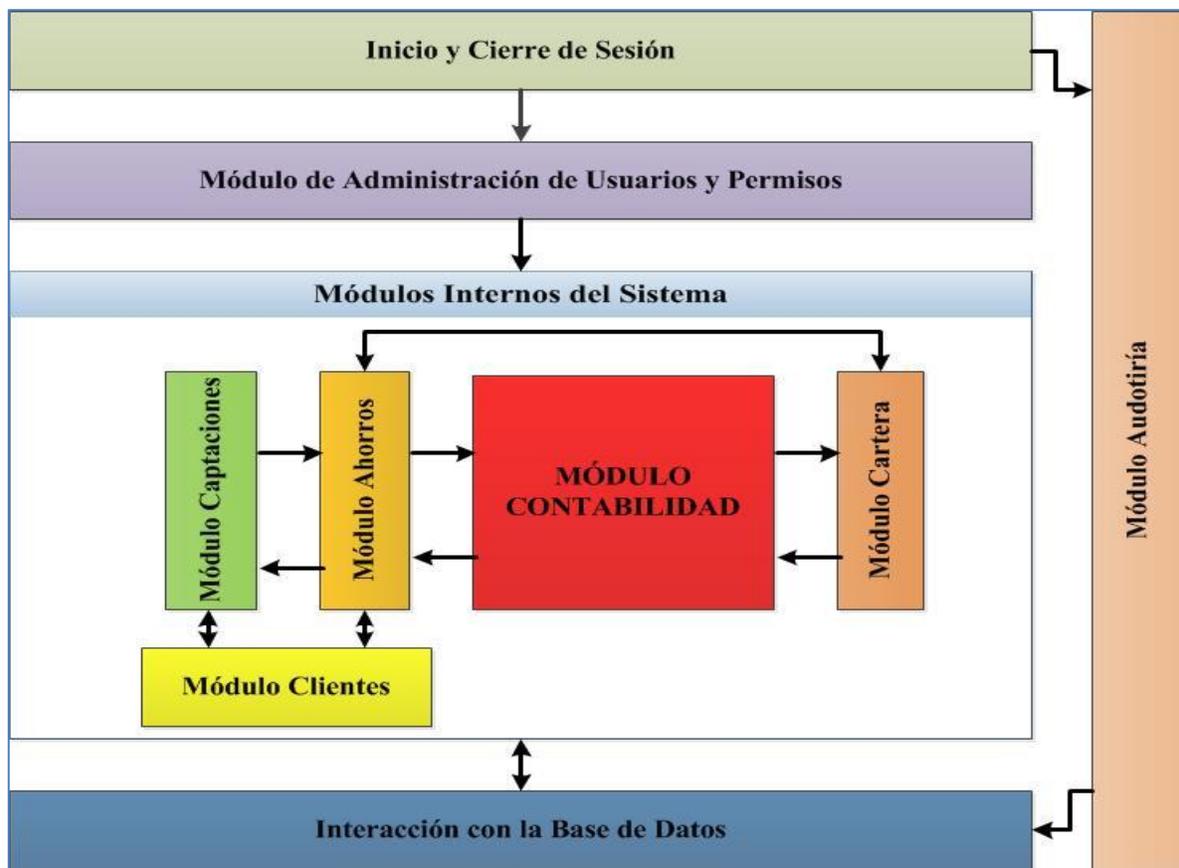


Figura 18. Prospectiva del sistema

Fuente: propia

3.1.1. Características del sistema

El sistema integrado de información financiera al ser diseñado como un sistema web tiene algunos beneficios para los usuarios y consta de las siguientes características:

Característica	Descripción
Administración agrupada:	Se instala y actualiza únicamente del lado del servidor ya que es una aplicación web y no requiere que se apliquen dichos cambios a los clientes.
Visualización en navegadores web:	Podrá funcionar en cualquier navegador que tenga soporte para java y java script puesto que la aplicación fue desarrollada con tecnologías de código abierto como JEE (java Enterprise Edition) pero se recomienda el uso del navegador Firefox.
Facilidades en el registro de información:	La interfaz es amigable y de fácil manejo lo que agiliza el proceso de capacitación.
Alta disponibilidad:	Los usuarios podrán acceder inmediatamente al sistema desde cualquier punto de la intranet local.
Eficiencia en la presentación de información:	Al ser un sistema integrado se puede acceder rápidamente a la información que se encuentra en los diferentes módulos del sistema y tener toda la información necesaria para una tarea en una misma pantalla ya que se pueden definir menús y roles personalizados.
Fácil de usar:	El diseño y la funcionalidad son muy simples ya que consta de un menú intuitivo, fácil navegación entre opciones y su información es consistente.
Facilidad para analizar la información:	A través de los diferentes reportes y opciones de búsquedas y consultas con los que consta el sistema.

Sistema multiplataforma:	El acceso al sistema se podrá realizar desde cualquier navegador web de cualquier tipo de sistema operativo.
--------------------------	--

Tabla 6. Características del sistema

Fuente: propia

3.1.2. Dependencias para la implementación

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

3.1.2.1. Costos

Para la implementación del sistema se realizará la adquisición de un servidor con características adecuadas para el funcionamiento del sistema, dicha adquisición queda bajo la responsabilidad de la Cooperativa. El sistema no aplica en costos de licenciamiento ya que se desarrolló a través de una plataforma de código abierto.

3.1.2.2. Licenciamiento

Las tecnologías utilizadas para el desarrollo e implementación del sistema no necesitan de algún pago para la adquisición de licencias ya que las herramientas están regidas por las cláusulas de GPL establecido en el software libre. Para el desarrollo del sistema integrado de información financiera se utilizaron las siguientes herramientas:

- a) Como metodología de desarrollo RUP (Rational Unified Process).
- b) Postgres como servidor de base de datos.
- c) Framework Hibernate para mapeo de datos.
- d) JEE (Java Enterprise Edition) como plataforma de desarrollo.
- e) JSF (Java Server Faces) para la interfaz.
- f) RichFaces para la integración de funcionalidades Ajax en JSF.
- g) JasperReports para presentación de reportes.
- h) Netbeans como IDE de desarrollo.

3.1.2.3. Instalación

No existe mayor problema en la instalación ya que realizará de forma centralizada, la información estará ubicada en un solo punto llamado servidor, al cual los clientes accederán a través de un navegador web desde cualquier punto de la intranet local de la institución.

3.1.3. Funcionamiento

El flujo de la aplicación inicia en la página de login o autenticación, si un usuario intenta acceder directamente a otra página, la aplicación verifica que no se ha autenticado y le re-direcciona a la página de autenticación. Luego de autenticarse el usuario podrá navegar por las opciones que presenta cada módulo de la aplicación dependiendo del rol y el tipo de usuario. A continuación se muestra el funcionamiento en general de sistema.

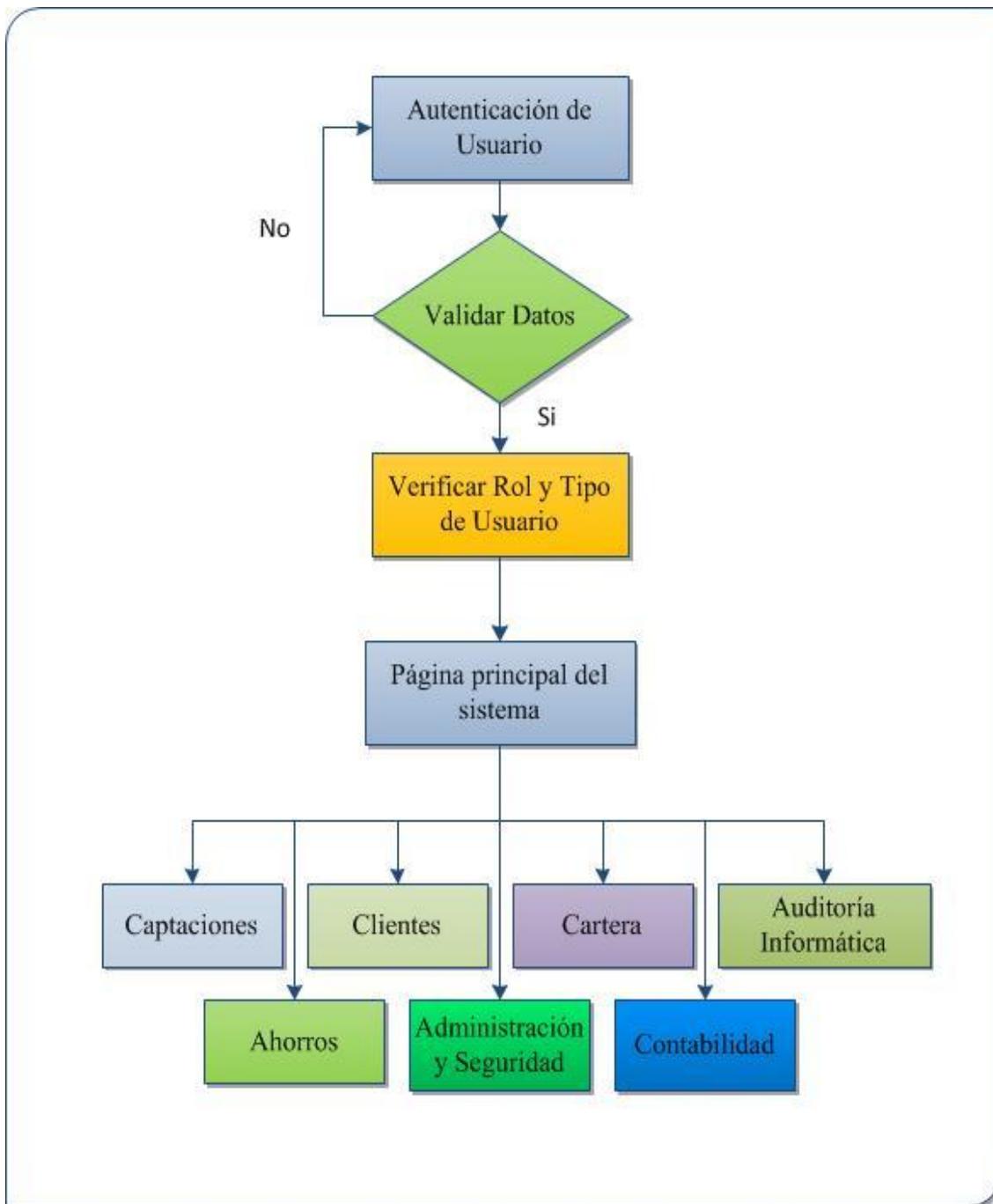


Figura 19. Funcionamiento general del sistema

Fuente: Propia

3.2. Arquitectura del sistema

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

- a) **EL MODELO.-** representa la estructura de los datos en la aplicación, es decir que en ésta capa se realizan las consultas y manipulación de datos (CRUD), además de los métodos, funciones u operaciones con estos datos.
- b) **LA VISTA.-** es la capa que presenta los datos al usuario, es la interfaz gráfica de la aplicación.
- c) **EL CONTROLADOR.-** traduce las acciones de los usuarios realizadas en la VISTA y llama a los métodos de negocio creados en la capa de MODELO y selecciona la vista apropiada según haya sido el requerimiento del usuario.

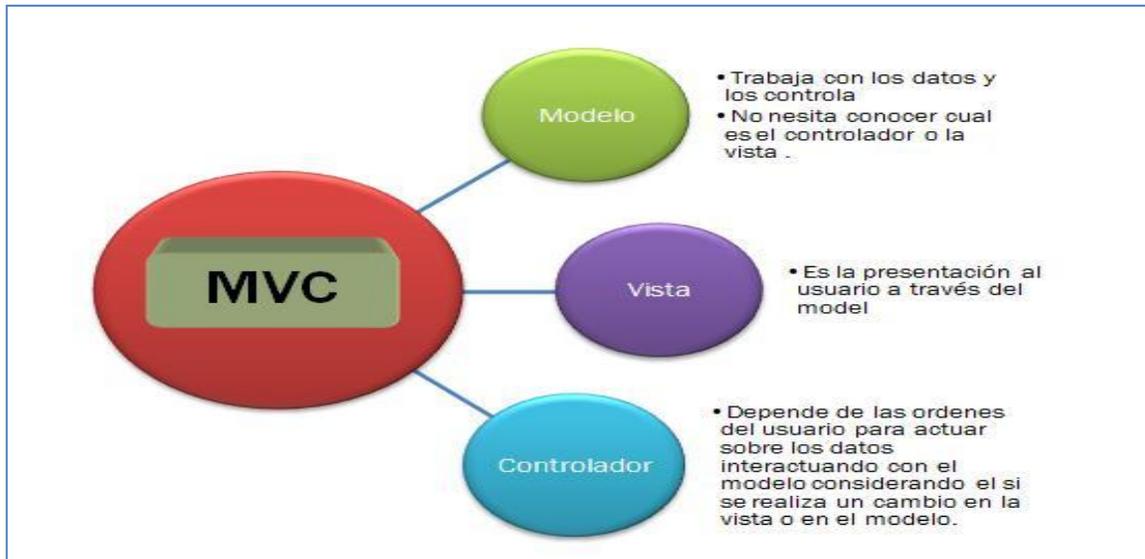


Figura 20. Arquitectura de la aplicación

Fuente: (<http://es.wikibooks.org/wiki>)

3.3. Descripción de los módulos

A continuación se describen los módulos de Préstamos-Cartera, Clientes y Auditoría Informática.

3.3.1. Módulo de Préstamos-Cartera

El módulo de Cartera o Créditos está diseñada para administrar todos los procesos relacionados con los préstamos. Éste módulo tramita las operaciones desde la recopilación de la información, la solicitud del préstamo, la aprobación o negación del préstamo hasta el manejo de los respectivos pagos.

Los productos de créditos que ofrece la Cooperativa a sus clientes, depende de las necesidades de la empresa, puesto que es totalmente parametrizable.

Al momento de ingresar los datos el módulo verifica las tasas de interés y los montos máximos permitidos, de acuerdo a las características asignadas al tipo de préstamo que desee el cliente.

El módulo de Préstamos-Cartera también documenta la situación económica del cliente y las garantías para contar con todos los datos adicionales necesarios.

3.3.1.1. Procesos del módulo Préstamos-Cartera

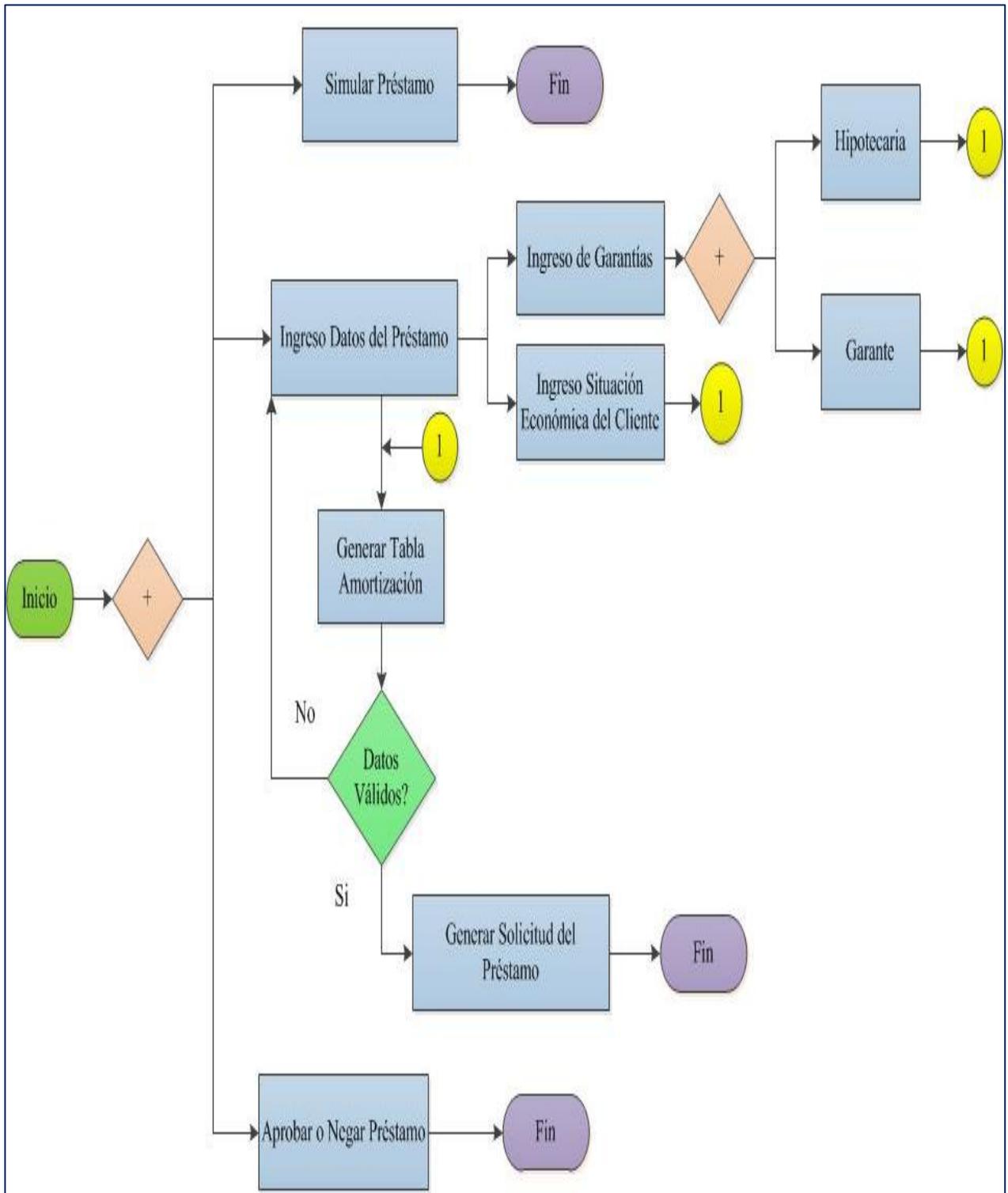


Figura 21. Procesos del módulo Préstamos-Cartera

Fuente: propia.

3.3.1.2. Acciones del módulo Préstamo-Cartera

Paso	Acción
1.	Autenticarse como usuario con acceso al módulo de préstamos-cartera.
2.	Ingresar al menú cartera
3.	Seleccionar una de las siguientes opciones: <ol style="list-style-type: none">1. Simulación de préstamos.2. Solicitud de préstamo.3. Aprobación o negación de préstamos.4. Pago cuotas.5. Consultas.6. Repostes.
3.1	Simulación de un préstamo <ul style="list-style-type: none">• Selección del grupo de préstamo.• Selección de línea de crédito.• Selección de tipo de amortización.• Ingreso de la cantidad a solicitar.• Ingreso del número de cuotas.• Selección de la frecuencia de pagos.• Simular las cuotas.
3.2	Solicitud de préstamo <ul style="list-style-type: none">• Selección del grupo de préstamo.• Selección de línea de crédito.• Selección de tipo de amortización.• Selección del tipo de garantía.• Selección de la cuenta del socio.• Ingreso de la cantidad a solicitar.• Ingreso del número de cuotas.

	<ul style="list-style-type: none"> • Selección de la frecuencia de pagos. • Generar tabla de amortización. • Ingreso de la situación económica del cliente. • Ingreso del garante o bienes según el tipo de garantía seleccionada. • Generar solicitud.
3.3	Aprobación de préstamo <ul style="list-style-type: none"> • Selección de la solicitud del préstamo. • Aprobar o denegar solicitud.
3.4	Pago de cuotas <ul style="list-style-type: none"> • Selección del cliente. • Seleccionar el ícono pagar del préstamo del cliente. • Ingresar el valor a pagar. • Pagar cuota(s).
3.5	Consultas
3.6	Reportes

Tabla 7. Acciones del módulo Préstamos-Cartera.

Fuente: propia.

3.3.1.3. Casos de uso del módulo Préstamos-Cartera

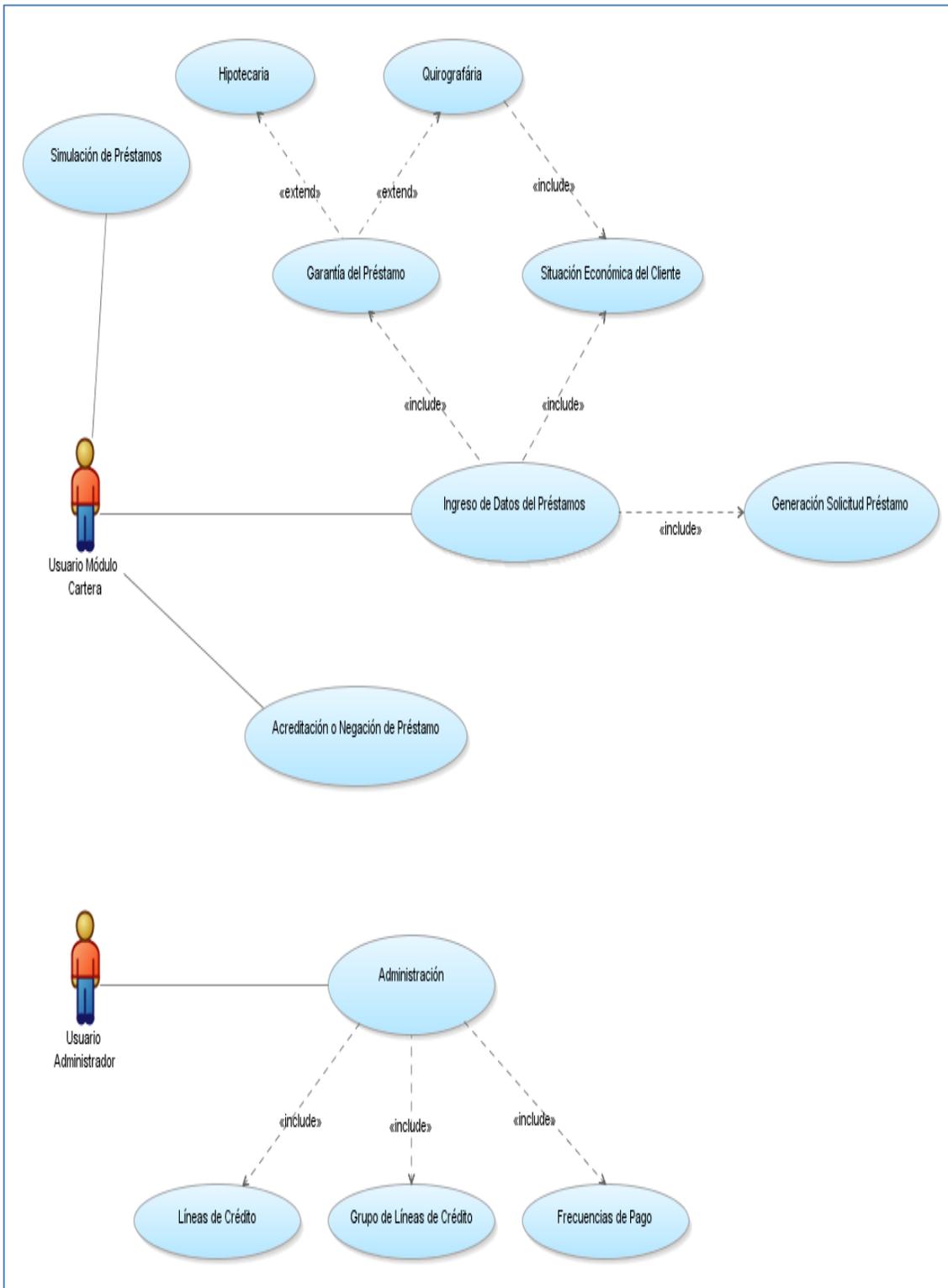


Figura 22. Casos de uso del módulo Préstamos-Cartera

Fuente: Propia.

3.3.2. Módulo de Clientes

El sistema FINANSYS dispone del módulo de Clientes para la gestión de la información de los clientes de la Cooperativa, la actualización y corrección de los datos, consultas de diferentes tipos y el cierre de cuentas.

Este módulo del sistema es de mucha importancia para la Cooperativa, puesto que aquí se guardan y registran todos los clientes pertenecientes a la empresa.

El módulo de clientes permite administrar toda la información personal y financiera de cada cliente, existe también la función para agregar una fotografía del cliente a sus datos personales, facilitando la verificación de la identidad del cliente cuando se presente a realizar alguna operación en la Cooperativa.

El módulo también guarda los datos adicionales que están relacionados al cliente, como ser las referencias personales, los datos del cónyuge en caso de tenerlo, la actividad laboral, y para el caso de los clientes menores de edad, los datos del representante.

Por medio de éste módulo también se gestionan los cierres de cuentas, cuando el cliente definitivamente desee dejar de pertenecer a la Cooperativa.

También cuenta con la funcionalidad de generar reportes personalizados de acuerdo a las necesidades específicas de los usuarios encargados de manejar el módulo.

3.3.2.1. Procesos del módulo Clientes

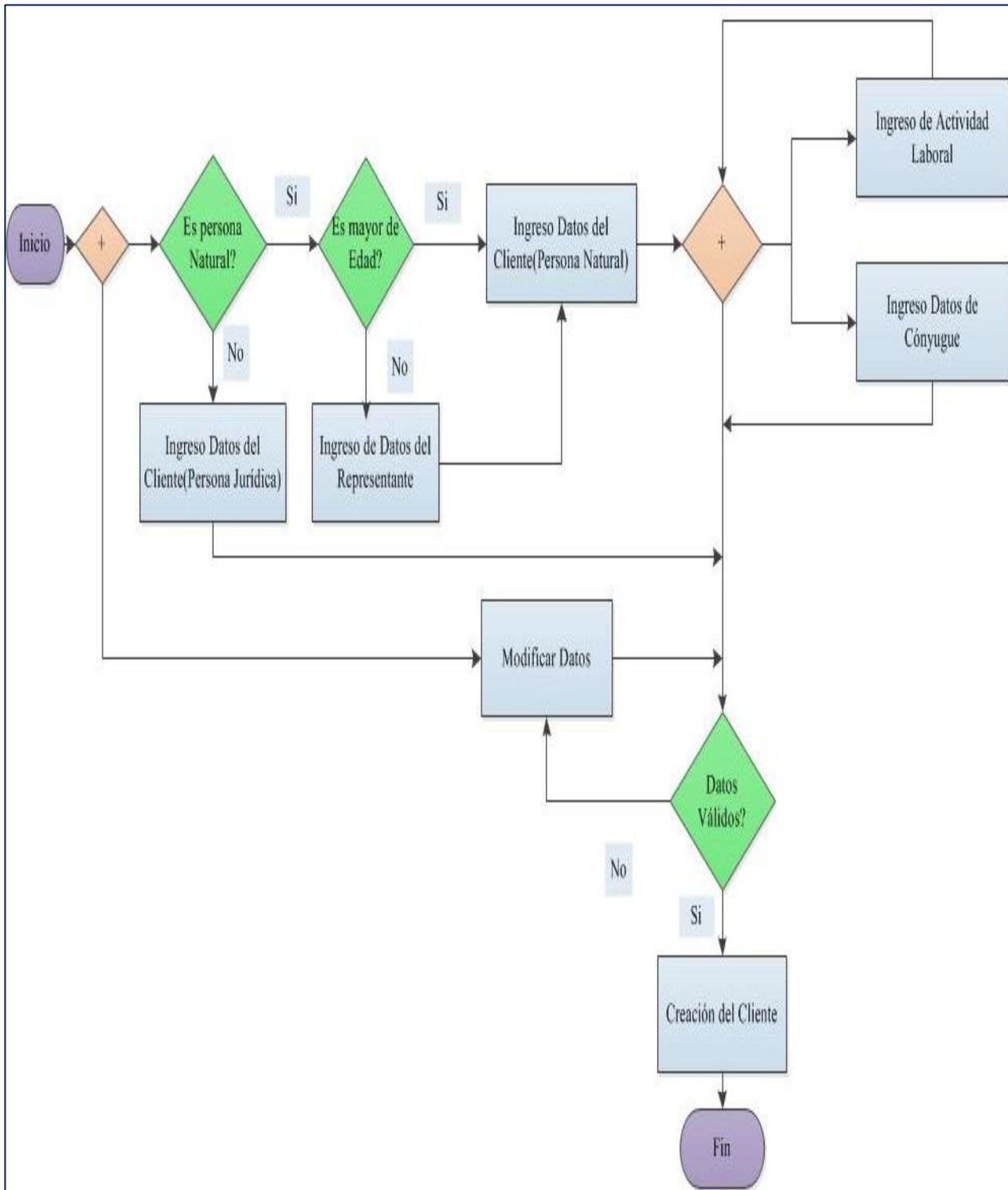


Figura 23. Procesos del módulo Clientes.

Fuente: propia.

3.3.2.2. Acciones del módulo Clientes

Pasos	Acción
1.	Autenticarse como usuario con acceso al módulo de clientes.
2.	Ingresar al menú clientes.
3.	Seleccionar una de las siguientes opciones: 1 Personas. 2 Personas jurídicas. 3 Nuevo cliente. a. Persona natural b. Persona jurídica 4 Cerrar cuenta 5 Búsquedas: a. Personas naturales b. Personas jurídicas c. Clientes. 6 Reportes.
3.1	Personas <ul style="list-style-type: none">• Ingresar datos personales.• Ingresar fotografía en caso de tenerla.• Guardar datos.
3.2	Personas jurídicas <ul style="list-style-type: none">• Ingreso de datos generales.• Ingreso de datos de los representantes legales y direcciones de los mismos.• Guardar datos.
3.3.a	Nuevo cliente-persona natural <ul style="list-style-type: none">• Selección de la oficina y grupo.

	<ul style="list-style-type: none"> • Selección o ingreso de la cédula de la persona. • Ingreso de los datos de la dirección del cliente. • Ingreso de la actividad económica del cliente (opcional). • Ingreso del cónyuge del cliente (en caso de tenerlo). • Ingreso de las referencias personales. • Ingreso de los datos del representante. (en caso de ser cliente menor de edad). • Guardar datos.
3.3.b	Nuevo cliente-persona jurídica <ul style="list-style-type: none"> • Selección de la oficina y grupo. • Selección o ingreso de la persona jurídica. • Ingreso de los datos de la dirección del cliente. • Guardar datos.
3.4	Cerrar cuenta <ul style="list-style-type: none"> • Selección de la cuenta a cerrar. • Cerrar cuenta.
3.5	Consultas
3.6	Reportes

Tabla 8. Acciones del módulo Clientes.

Fuente: propia.

3.3.2.3. Casos de uso del módulo Clientes

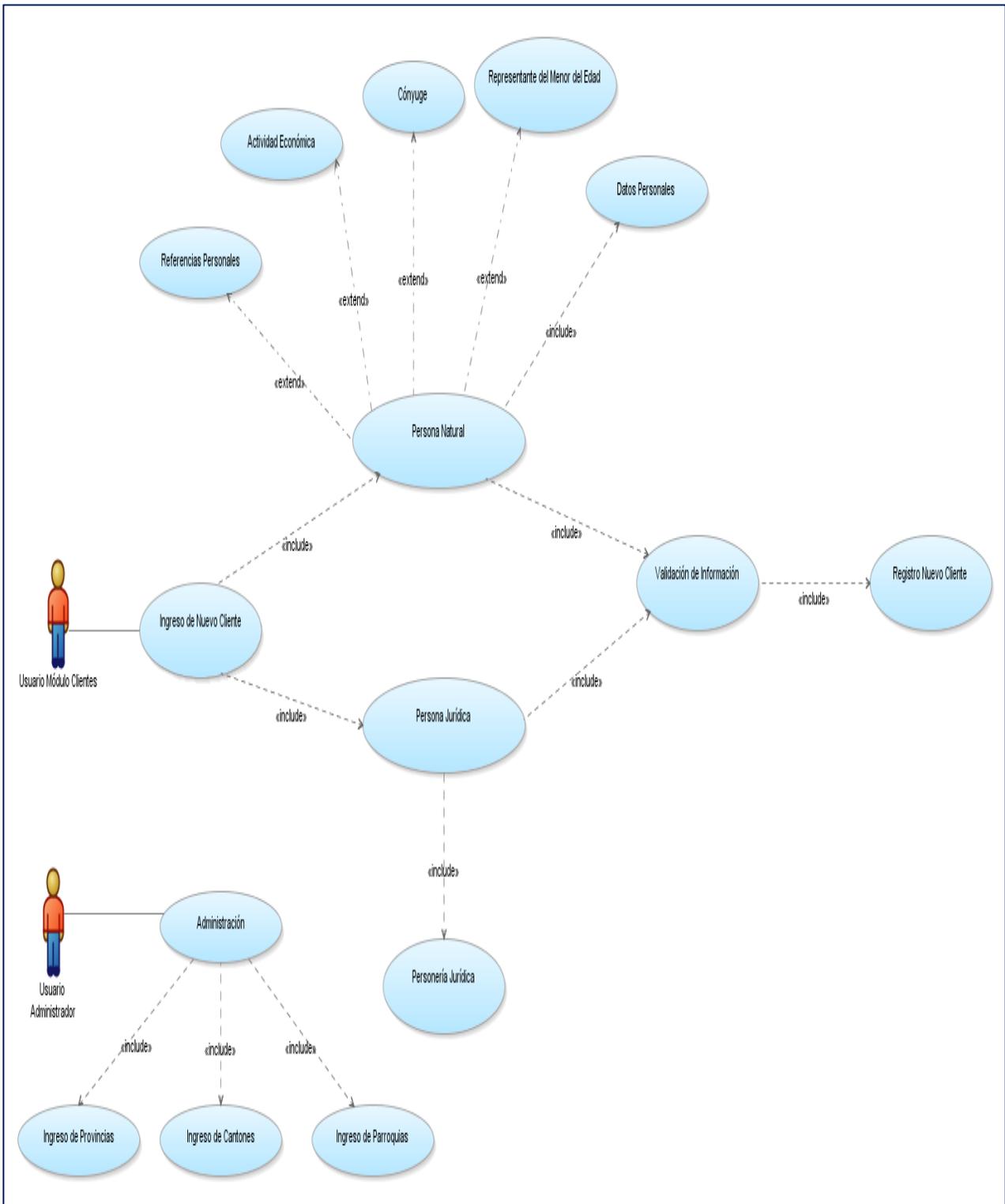


Figura 24. Casos de uso del módulo Clientes.

Fuente: propia.

3.3.3. Módulo de Auditoría Informática

El módulo de auditoría informática permite tener un constante control interno y monitoreo de la información manejada en el sistema.

Cada vez que un usuario registrado ingrese o cierre la sesión del sistema, automáticamente se guarda el registro con la fecha y hora de la misma.

De igual forma, cada vez que se realiza alguna acción de ingreso, modificación o eliminación de registros en la base de datos, automáticamente se guarda ese detalle.

3.3.3.1. Procesos del módulo Auditoría Informática

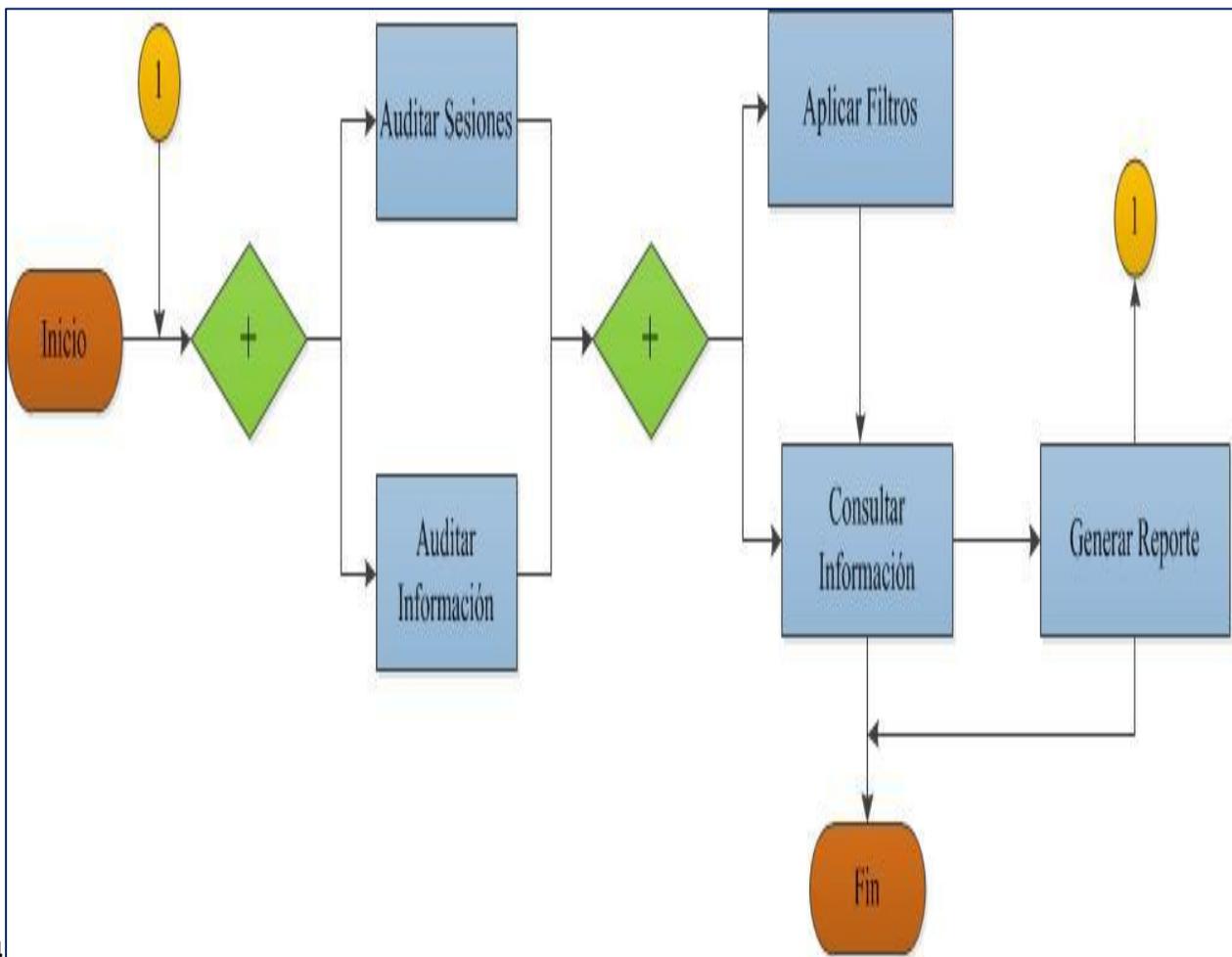


Figura 25. Procesos del módulo Auditoría.

Fuente: propia.

3.3.3.2. Acciones del módulo Auditoría Informática

Pasos	Acción
1.	Autenticarse como usuario con acceso al módulo de Auditoría
2.	Ingresar al menú Auditoría Informática
3.	Seleccionar una de las siguientes opciones: 1. Interacción con la BDD 2. Sesiones 3. Reportes
3.1	Interacción con la BDD <ul style="list-style-type: none">• Selección de la tabla a auditar.• Selección de la acción realizada en la tabla.• Selección del usuario.• Selección de la sucursal.• Ingreso de la fecha de inicio.• Ingreso de la fecha de fin.• Ingreso de la hora de inicio.• Ingreso de la hora de fin.• Seleccionar el botón consultar.
3.2	Sesiones <ul style="list-style-type: none">• Selección del usuario.• Selección de la sucursal.• Ingreso de la fecha de ingreso a la sesión.• Ingreso de la fecha de cierre de la sesión.• Ingreso de la hora de ingreso.• Ingreso de la hora de cierre.• Seleccionar el botón consultar.
3.3	Reportes

- En el formulario de la auditoria de sesiones o de base de datos se encuentra en la parte inferior un botón para imprimir el contenido actual del formulario.

Tabla 9. Acciones del módulo Auditoría.

Fuente: propia.

3.3.3.3. Casos de uso del módulo Auditoría Informática

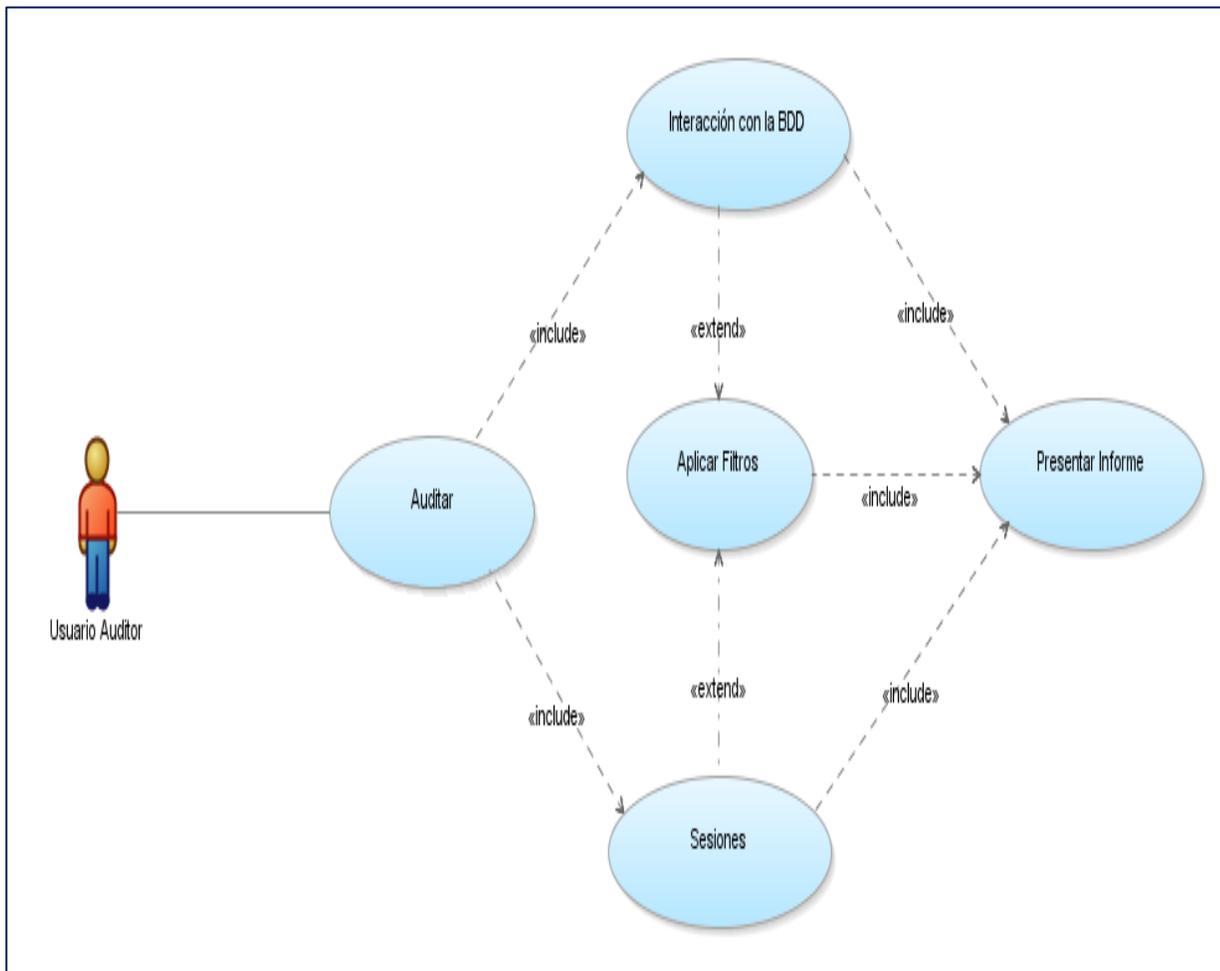


Figura 26. Casos de uso del módulo Auditoría.

Fuente: propia.

Capítulo 4:

Diseño y Desarrollo del Aplicativo



RESUMEN DEL CAPÍTULO

- *Fase de inicio.*
- *Fase de elaboración.*
- *Fase de construcción.*
- *Fase de transición.*

4. Diseño y desarrollo del aplicativo

Uno de los aspectos importantes a la hora de realizar un sistema es la correcta elección de las herramientas adecuadas, para asegurar la calidad del producto terminado y cumplir con los plazos estipulados.

Es por ello que la metodología usada para todo el proceso de diseño desarrollo y documentación del aplicativo hemos elegido emplear RUP (Rational Unified Process).

RUP clasifica el proceso en cuatro fases los cuales se describen a continuación.

Fase de inicio

El objetivo principal de la fase de inicio es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Se inicia el análisis del negocio hasta el punto necesario para justificar la puesta en marcha del proyecto.

Esta fase es muy importante porque se identifica los riesgos relacionados con el proyecto y los requerimientos.

Fase de elaboración

El objetivo de esta fase es establecer la arquitectura base del sistema para proveer bases estables para el diseño e implementación de la siguiente fase.

Fase de construcción

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

En las fases anteriores se da una arquitectura básica que en éste punto es refinado.

Gran parte del trabajo es programación y pruebas, también se documenta el sistema construido como el manejo del mismo.

Fase de transición

Esta fase se enfoca en asegurar que el software esté disponible para su entrega a los usuarios. Los manuales de usuario se completan y pulen de acuerdo a la información anterior.

4.1. Fase de inicio

4.1.1. Documento de visión

4.1.1.1. Introducción

4.1.1.1.1. Propósito

El propósito de este documento es definir los requerimientos del sistema integrado de información financiera - Cooperativa Unión Cochapamba (FINANSYS).

El sistema FINANSYS se encargará de realizar todo el proceso financiero de la Cooperativa Unión Cochapamba de manera detallada y automática.

El detalle de cómo el sistema FINANSYS cubrirá las necesidades de los usuarios se especifica en los Casos de Uso, que es información adicional no especificada en este documento.

4.1.1.1.2. Alcance

Este documento de visión se aplica al **sistema integrado de información financiera - Cooperativa Unión Cochapamba** que será implementado por estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte.

4.1.1.2. Posicionamiento

4.1.1.2.1. Oportunidad de negocio

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

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

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

a) **No existe un gestor de base de datos**

El sistema actual para el manejo de su información usa archivos de texto visibles al usuario, lo que hace que sea un sistema vulnerable y susceptible a cualquier tipo de error.

b) **Parametrización no adecuada**

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

c) **Tecnología desactualizada**

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

d) **Falta de control y auditoría**

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

e) **Seguridad deficiente**

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

Con referencia a lo expuesto anteriormente, la directiva general de la Cooperativa ha planteado rediseñar la aplicación informática existente para permitir trabajar de una manera más eficiente e integrada en un único sistema financiero.

El nuevo sistema FINANSYS brindará las facilidades necesarias para el manejo de la información financiera generada por la Cooperativa.

Rediseñar e implementar un nuevo sistema, obliga a que se cree en la nueva aplicación la mayoría de las opciones del sistema antiguo, como también los nuevos módulos que se desarrollen con el fin de obtener una aplicación eficiente, estos módulos son:

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

4.1.1.2.2. Definición del problema

<p>El problema de:</p>	<p>Redundancia de procesos en el desarrollo de actividades diarias. No poseer un sistema integrado que administre todos los procesos de gestión de la información financiera La aplicación actual carece de seguridad. Aplicación construida con tecnología que ya no tiene soporte. Dificultad para el mantenimiento de la aplicación actual.</p>
<p>Que afecta a:</p>	<p>Todos los usuarios de la Cooperativa involucrados con la gestión de la información financiera. Personal que tiene dificultad para mantener el sistema en producción.</p>
<p>El impacto de ello es:</p>	<p>Falta de información consistente de la gestión financiera. Existen muchas actividades y procesos manuales que no permiten una gestión eficiente. Inconsistencia en la información generada.</p>
<p>Una solución exitosa debería:</p>	<p>Solucionar los requerimientos internos de los involucrados en el proceso de la gestión financiera. Cubrir las necesidades de integración. Automatizar algunos de los procesos que hasta ahora son realizados de forma manual. Brindar información confiable sobre el manejo financiero de la Cooperativa de manera fácil y oportuna. Implementar una solución informática de calidad, soportada por una metodología eficiente de desarrollo de software.</p>

Tabla 10. Definición del problema.

Fuente: propia.

4.1.1.3. Descripción de los interesados y usuarios

4.1.1.3.1. Resumen de los interesados

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

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

Tabla 11. Lista de interesados.

Fuente: propia.

4.1.1.3.2. Resumen de los usuarios

Los usuarios son todas aquellas personas involucradas directamente en el uso del sistema FINANSYS.

A continuación se presenta una lista de los usuarios:

Nombre	Descripción	Responsabilidad
Administrador funcional del sistema (contador)	Persona de la Cooperativa de Ahorros y Crédito Unión Cochapamba que administra el sistema FINANSYS.	Administrar funcionalmente el sistema: creación de nuevos productos, definir los tipos de transacciones, nuevos grupos de clientes, tipos de préstamos, etc.
Usuario del sistema (gerente, contador, cajero, asesor de créditos)	Personal que labora en los diferentes departamentos de la Cooperativa de Ahorros y Crédito Unión Cochapamba que usan el sistema FINANSYS.	Ingresan la información referente a cada uno de los departamentos de la Cooperativa como son registro de información personal para el ingreso de nuevos clientes, créditos, retiros , etc.
Usuario de gestión del sistema (gerente)	Personal de la Cooperativa de ahorro y crédito Unión Cochapamba	Validar la información proveniente de los diferentes departamentos de la Cooperativa generada a través del FINANSYS. Consolidar la información. Aprobación de socios y créditos

Tabla 12. Resumen de los usuarios.

Fuente: propia.

4.1.1.3.3. Entorno actual de los usuarios

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

Debido a que es muy costoso adquirir licencias de software privativo el sistema será implementado en su totalidad con software libre y el lineamiento general es tener las aplicaciones en plataforma web.

Actualmente existen sistemas para instituciones financieras como Bancos y Cooperativas de Ahorro y Crédito pero hay que tomar en cuenta que cada institución tiene lineamientos diferentes por lo que hace difícil la adquisición de estos sistemas ya elaborados, lo que conlleva a las instituciones a diseñar su propio sistema que se ajuste a sus necesidades.

Los Módulos de Ahorros, Captaciones, Clientes, Administración y Seguridad, Préstamos-Cartera, Auditoría y Contabilidad formarán parte del Sistema Integrado de Información Financiero, los mismos que deberán interactuar entre sí.

Perfiles de los interesados

4.1.1.3.3.1. Coordinador de proyecto

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

Tabla 13. Perfil de los interesados.

Fuente: propia.

4.1.1.3.3.2. Responsable del proyecto

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

Tabla 14. Responsables del proyecto.

Fuente: propia.

4.1.1.3.3.3. Responsable funcional

Representante	Irene de la Cruz.
Descripción	Responsables del proyecto por parte de la Cooperativa de ahorro y crédito Unión Cochapamba.
Tipo	Experto en el tema.
Responsabilidades	Responsable de coordinar con los diferentes usuarios la correcta determinación de los requerimientos y la correcta concepción del sistema. Coordinar las pruebas de validación del nuevo sistema. Coordinar y asegurar la capacitación de los usuarios.
Criterios de éxito	Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos.
Implicación	Aprueba las especificaciones funcionales y las pruebas realizadas.
Entregables	Documento de revisión de las especificaciones funcionales. Documento de revisión de las pruebas funcionales
Comentarios	N/A

Tabla 15. Responsable funcional.

Fuente: propia.

4.1.1.3.4. Perfiles de usuario

4.1.1.3.4.1. Administrador del sistema

Representante	N/A
Descripción	Persona del centro de cómputo que administra el sistema FINANSYS.
Tipo	Operador, analista de sistemas
Responsabilidades	Administrar funcionalmente el sistema: gestionar acceso a usuarios, dar mantenimiento al sistema frente a nuevos requerimientos.
Criterios de éxito	N/A
Implicación	N/A
Entregables	Bitácora de control de nuevos requerimientos. Bitácora de control de incidencias del nuevo sistema.

Tabla 16. Administrador del sistema.

Fuente: propia.

4.1.1.3.4.2. Administrador funcional del sistema

Representante	Irene de la Cruz.
Descripción	Persona del centro de cómputo de la Cooperativa que administra el FINANSYS
Tipo	Experto en el tema
Responsabilidades	Creación de nuevas cuentas de usuarios, definición de períodos contables, etc.
Criterios de éxito	N/A
Implicación	N/A
Entregables	N/A
Comentarios	N/A

Tabla 17. Administrador funcional del sistema

Fuente: propia.

4.1.1.3.4.3. Usuario del sistema

Representante	N/A
Descripción	Personal de los diferentes departamentos CUC que harán uso de cada uno de los módulos del FINANSYS.
Tipo	Personal de los diferentes departamentos de la CUC.
Responsabilidades	Ingresa la información concerniente en cada uno de los módulos del FINANSYS. Realizar consultas de información de clientes. Realizar transacciones, etc.

Tabla 18. Usuario del sistema.

Fuente: propia.

4.1.1.3.4.4. Necesidades de los interesados y usuarios

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

Necesidades	Prioridad	Inquietudes	Solución actual	Solución propuesta
La interfaz del sistema debe ser fácil de manejar, cumpliendo con todos los requerimientos establecidos.	Alta	Cumplir con todos los requerimientos de los usuarios.	N/A	Desarrollo con la ayuda de los expertos en cada uno de los departamentos de la CUC.
Lograr la implementación del sistema en el menor tiempo posible para su utilización desde el próximo año.	Alta	Cumplir con las especificaciones de los usuarios.	Actualmente se trabaja con el sistema por módulos independientes .	Trabajar con el FINANSYS desde el año 2012 ya con el sistema implementado y en funcionamiento.

Tabla 19. Necesidades de los usuarios.

Fuente: propia.

4.1.1.3.4.5. Alternativas y competencia

Adquirir un sistema desarrollado externamente por casas Desarrolladoras de Software.

Se ha mostrado interés en buscar alternativas externas para solucionar los diversos requerimientos, pero en la actualidad no existen herramientas en el mercado que se adapten a las necesidades específicas de la Cooperativa.

4.1.1.4. Vista general del producto

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

4.1.1.4.1. Perspectiva del producto

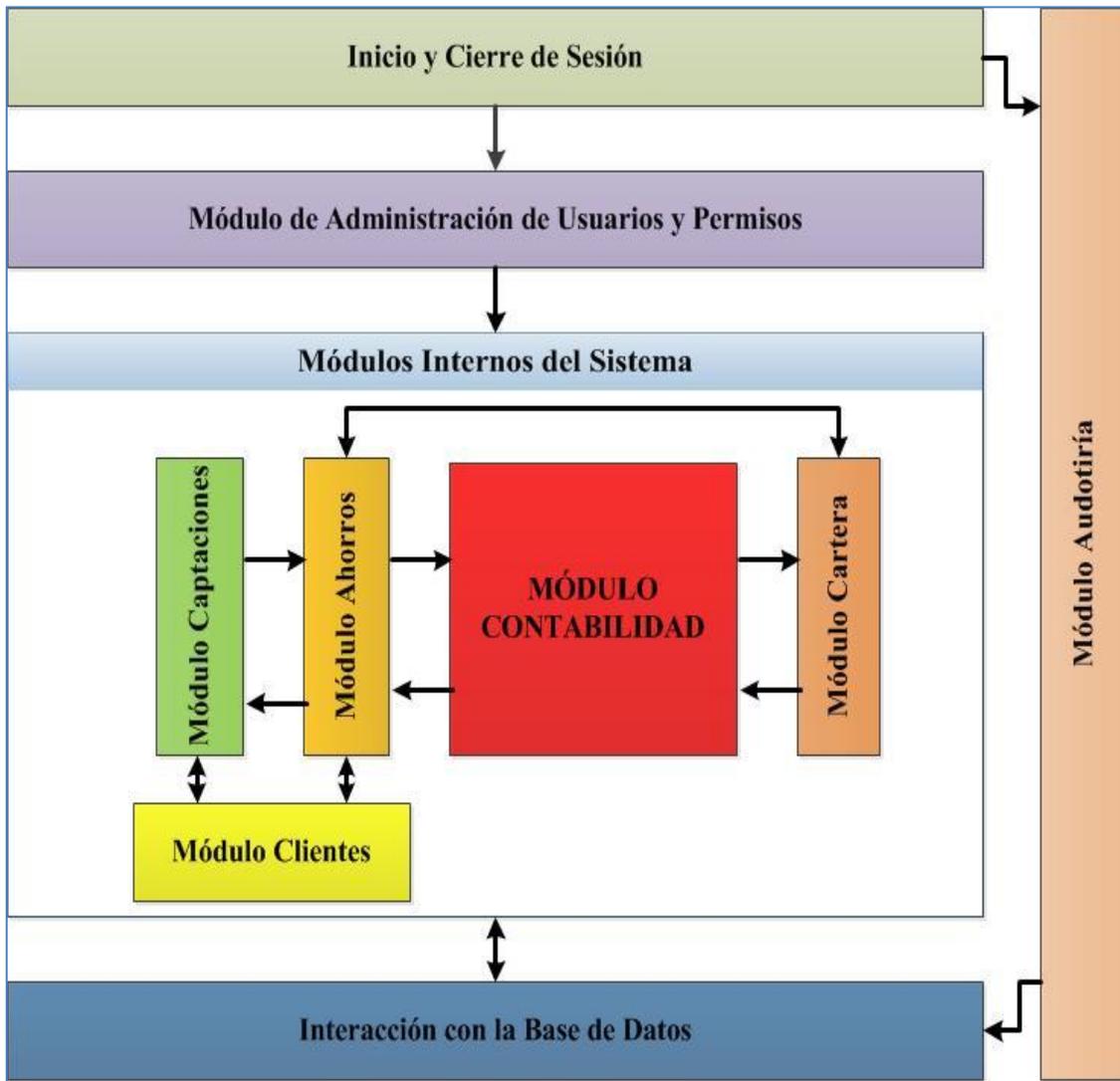


Figura 27. Perspectiva del sistema.

Fuente: propia.

4.1.1.4.2. Resumen de capacidades

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

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

4.1.1.4.2.1. Módulo de Ahorros

Beneficios para el usuario	Características que lo soportan
El registro de la información se realiza con mayor rapidez.	Debido a su facilidad de manejo y entendimiento se podrá ingresar la información con mayor rapidez. El responsable del Manejo del módulo Ahorros tendrá una herramienta de registro y manejo de información congruente y sincronizada con el sistema integrado.
El acceso a la información y acceso a consultas se realizan con mayor facilidad.	Al contar con un sistema computacional, el tiempo de respuesta a una consulta se mejora Brindará diferentes reportes y funciones de consulta.
Se tendrá alta disponibilidad.	El acceso a la información a través del sistema Web permitirá a los usuarios un acceso inmediato desde cualquier punto de la intranet de la Cooperativa.
El módulo de Ahorros y su información interactúa con el resto de módulos del sistema.	Oportunidad de mantener un control sobre la información generada por los módulos que interactúan con el módulo de Ahorros.

Tabla 20. Módulo de Ahorros

Fuente: propia.

4.1.1.4.2.2. Módulo de Caja

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

Tabla 21. Módulo de Caja.

Fuente: propia.

4.1.1.4.2.3. Módulo de Administración y Seguridad

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

Tabla 22. Módulo de Administración y Seguridad.

Fuente: propia.

4.1.1.4.2.4. Módulo Contabilidad.

Beneficios para el usuario	Características que lo soportan
El registro de la información contable será confiable, automático y completo.	El ingreso de información realizada por el encargado del manejo del módulo contable será sistematizado y ordenado. El responsable del manejo del módulo contable tendrá una herramienta de registro y manejo de información congruente y sincronizada con el sistema integrado.
Los usuarios o usuario del módulo contarán con una herramienta unificada.	Al contar con un sistema computacional, los usuarios podrán mejorar sus procesos. Además evitará el uso manual de formatos y documentos no unificados.
Se tendrá alta disponibilidad.	El acceso a la información a través del sistema web permitirá a los usuarios un acceso inmediato desde cualquier punto de la intranet de la Cooperativa.
Facilidades para el análisis de la información.	Brindará diferentes reportes y funciones de consulta.
Contará con información detallada.	Oportunidad de mantener un control sobre la información que se desea mostrar o realizar un reporte.

Tabla 23. Módulo Contabilidad.

Fuente: propia.

4.1.1.4.2.5. Módulo Préstamos-Cartera

Beneficios para el usuario	Características que lo soportan
El registro de la información de los préstamos será de manera fácil, confiable y unificada con el resto del sistema.	<p>El ingreso de información realizada por el usuario encargado del manejo del módulo de PRÉSTAMOS-CARTERA será sencillo, sistematizado y ordenado.</p> <p>El usuario responsable del manejo del módulo PRÉSTAMOS-CARTERA tendrá una herramienta de registro y manejo de información congruente y sincronizada con el sistema integrado.</p> <p>El sistema verificará si el cliente de la Cooperativa cumple con los requisitos básicos necesarios para ser acreedor a un préstamo.</p>
Se tendrá alta disponibilidad.	La información se almacenará de forma unificada permitiendo el acceso a dicha información a través del sistema Web de forma inmediata y desde cualquier punto de red de la intranet de la Cooperativa.
Facilidades para el análisis de la información.	<p>Brindará diferentes funciones de consulta.</p> <p>Mostrará avisos de las personas que están por entrar en morosidad y de los clientes morosos para que el agente de cobranzas se ponga en contacto con esos clientes.</p>
Herramientas de simulación de préstamos.	Los clientes que deseen obtener un préstamo pueden hacer primero una simulación para que las cuotas se vayan acomodando de acuerdo a sus necesidades o posibilidades.
Generación de reportes parametrizables.	El mecanismo de búsqueda de información y generación de reportes es totalmente parametrizable, obteniendo la información precisa que los usuarios requieran.

Tabla 24. Módulo Préstamos-Cartera.

Fuente: propia.

4.1.1.4.2.6. Módulo de Clientes

Beneficios para el usuario	Características que lo soportan
El registro de la información de los clientes será de manera fácil, confiable y unificada con el resto del sistema.	<p>El ingreso de información realizada por el usuario encargado del manejo del módulo de CLIENTES será sencillo, sistematizado y ordenado.</p> <p>El usuario responsable del manejo del módulo CLIENTES tendrá una herramienta de registro y manejo de información congruente y sincronizada con el sistema integrado.</p>
Se tendrá alta disponibilidad.	La información se almacenará de forma unificada permitiendo el acceso a dicha información a través del sistema Web de forma inmediata y desde cualquier punto de red de la intranet de la Cooperativa.
Facilidades para el análisis de la información.	Brindará diferentes funciones de consulta.
Generación de reportes parametrizables.	El mecanismo de búsqueda de información y generación de reportes es totalmente parametrizable, obteniendo la información precisa que los usuarios requieran.

Tabla 25. Módulo Clientes.

Fuente: propia.

4.1.1.4.2.7. Módulo Auditoría

Beneficios para el usuario	Características que lo soportan
Registro de los movimientos del sistema.	El ingreso de la información de los movimientos de la información en el sistema se realiza de forma transparente para los usuarios.
Facilidades para el análisis de la información.	Brindará diferentes funciones de consulta. Seguimiento detallado de las acciones realizadas por un usuario específico. Seguimiento detallado la información almacena por tabla, tipo, fecha, hora, usuario, etc.
Generación de reportes parametrizables.	El mecanismo de búsqueda de información y generación de reportes es totalmente parametrizable, obteniendo la información precisa que los usuarios requieran.

Tabla 26. Módulo Auditoría.

Fuente: propia.

4.1.1.4.2.8. Suposiciones y dependencias

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

4.1.1.4.3. Licenciamiento e instalación

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

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

4.1.1.5. Características del producto

4.1.1.5.1. Facilidad de acceso y uso

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

4.1.1.5.2. Unificación de la información

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

4.1.1.5.3. Mejor control y validación de la información

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

4.1.1.6. Restricciones

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

4.1.1.7. Rangos de calidad

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

4.1.1.8. Precedencia y prioridad

Las prioridades del sistema son:

- a) Mantener la información unificada y congruente.
- b) Realizar de forma correcta los procesos financieros de la Cooperativa.
- c) Interactuar entre los diferentes módulos del sistema.

- d) Proporcionar una herramienta que tenga una interfaz amigable con el usuario.
- e) Desarrollar un sistema multiusuario con el respectivo control de acceso y privilegios a cada usuario de acuerdo al rol que éste tenga.
- f) Desarrollar métodos fáciles de búsqueda de información

4.1.1.9. Otros requerimientos del producto

Requisitos del Sistema.

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

- a) Un computador servidor con las siguientes aplicaciones y características:
 - Sistema operativo CentOS.
 - Base de datos PostgreSQL versión mínima 8.4.
 - Servidor web Apache Tomcat versión mínima 6.0.26.
 - Puerto de red.
- b) Red de área local.
- b) Navegador web Firefox desde la versión 4 o superior.
- c) No requiere una conexión a internet.
- d) Facilidad de uso.
- e) Que sea administrable.
- f) Que realice cálculos exactos.

4.1.2. Plan de desarrollo de software

4.1.2.1. Introducción

Este Plan de desarrollo del software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto FINANSYS para la Cooperativa de ahorro y crédito “Unión Cochapamba” (en adelante CUC).

Este documento presenta una visión general del enfoque de desarrollo propuesto.

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

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

4.1.2.1.1. Propósito

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

Participantes:

Los usuarios del Plan de Desarrollo del Software son:

Analistas del proyecto, los mismos que definen los tiempos, esquemas, recursos y herramientas a utilizarse en el desarrollo del cronograma de actividades.

Los miembros del equipo de desarrollo que ejecutan y desarrollan el contenido de la planificación del cronograma.

4.1.2.1.2. Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo de FINANSYS.

Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base principal de cada etapa.

Para la versión 1.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos por medio del stakeholder representante de la “CUC” para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

4.1.2.2. Vista general del proyecto

4.1.2.2.1. Propósito, alcance y objetivos

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

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

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

Módulo de Administración y Seguridad:

- Administración de todos los usuarios del sistema.
- Administración de sucursales de la Cooperativa.
- Administración de los roles que se asignan a los usuarios.
- Creación de tipos de usuario.
- Administrar parámetros generales.
- Autenticación de usuarios.
- Administración de sesiones.

Módulo de Ahorros:

- Cuentas de ahorros.
- Otros ingresos.
- Realizar bloqueos y desbloqueos.
- Depósitos a plazo fijo.

- Realizar transferencias, débitos y créditos.
- Consultas y reportes.

Módulo de Cajero:

- Apertura de cuenta.
- Depósito de ahorros.
- Retiros en efectivo.
- Retiros con cheque.
- Cobro de préstamos.
- Registro otros ingresos.
- Apertura de caja.
- Arqueos de caja.
- Consultas y reportes.

Módulo de Contabilidad:

- Administración del plan de cuentas.
- Administrar todos los comprobantes.
- Administrar los tipos de comprobante.
- Administración de períodos contables.
- Administración de períodos mensuales.
- Administración información de bancos.
- Cerrar y mayorizar saldos del mes.
- Conciliaciones bancarias.
- Realizar el control de presupuesto.
- Asientos de fin de año.
- Realizar balances y estados financieros.
- Consultas.
- Reportes.

Módulo de Préstamos-Cartera.

- Crear grupos y tipos de préstamos.
- Simular monto y tiempo del préstamo.
- Generar solicitudes de préstamos.

- Generar tabla amortización.
- Estudio de factibilidad del préstamo.
- Aprobación o negación de préstamos.
- Entrega del préstamo.
- Cobro de cuotas de préstamos.
- Consultas.
- Reportes.

Módulo de Clientes.

- Administración información de provincias, cantones, parroquias y comunidades.
- Administración de datos personales del cliente.
- Administración de datos relacionados al cliente (cónyuge, referencias, actividad laboral).
- Creación de clientes naturales y jurídicos.
- Cierre de cuentas.
- Consultas.
- Reportes.

Módulo de Auditoría.

- Ingresar y administrar pistas de auditoría.
- Auditar sesiones.
- Auditar acciones en la base de datos.
- Consultas.
- Reportes.

4.1.2.2.2. Suposiciones y restricciones

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

Debe contemplarse las implicaciones de los siguientes puntos críticos:

Sistemas seguros: protección de la información.

Seguridad de transacciones e intercambio de información.

Adaptación a la normativa de Protección de Datos.

Adaptabilidad y la facilidad de uso.

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

4.1.2.2.3. Entregables del proyecto

En la metodología RUP, a lo largo de todo el desarrollo del software se van generando documentos entregables, que son documentos explicativos de cada actividad.

A continuación se describen cada uno de los artefactos que serán generados y utilizados por el proyecto.

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

Plan de desarrollo del software

Es el presente documento.

Glosario

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

Modelo de casos de uso

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

Visión

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

Especificaciones de casos de uso

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

Prototipos de interfaces de usuario

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

Modelo de datos

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

Casos de prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

Lista de riesgos

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

Manual de instalación

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

Producto

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

Actas de trabajo

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

4.1.2.2.4. Evolución del plan de desarrollo del software

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

4.1.2.3. Organización del proyecto

4.1.2.3.1. Participantes en el proyecto

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

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

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

4.1.2.3.2. Interfaces externas

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

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

4.1.2.3.3. Roles y responsabilidades

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

Puesto	Responsabilidad
Analista	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.

Programador	Son los responsables del desarrollo del proyecto, desarrollando los módulos del proyecto, creando versiones de los módulos hasta llegar a la versión final.
-------------	---

Tabla 27. Roles y responsabilidades.

Fuente: propia.

4.1.2.4. Gestión del proceso

4.1.2.4.1. Plan del proyecto

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

4.1.2.4.1.1. Plan de las fases

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

Fase	Nro. Iteraciones	Duración
Fase de inicio	1	5 semanas
Fase de elaboración	1	8 semanas
Fase de construcción	2	16 semanas
Fase de transición	1	4 semanas

Tabla 28. Plan de fases

Fuente: propia.

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

Descripción	Hito
Fase de inicio	<p>En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto visión, previamente establecidos los requerimientos en las actas de trabajo. Los principales casos de uso serán identificados y se hará un refinamiento del plan de desarrollo del proyecto. La aceptación del cliente / usuario del artefacto visión y el plan de desarrollo marcan el final de esta fase.</p>
Fase de elaboración	<p>En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera versión de la fase de construcción deben estar analizados y diseñados (en el modelo de análisis / diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el modelo de análisis / diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.</p>
Fase de construcción	<p>Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el modelo de análisis / diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el usuario. Se comienza la elaboración de material de apoyo al usuario.</p>

	El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla 29. Hitos del proyecto.

Fuente: propia.

4.1.2.4.1.2. Objetivos de las iteraciones

Fase	Iteración	Descripción	Hitos asociados	Objetivos
Incepción	Iteración inicial	Definición del modelo de negocio y plan de proyecto	Actas de trabajo. Documento de visión.	Clasificar de forma general los requerimientos del usuario. Determinar el alcance e impacto del proyecto. Determinar la factibilidad del proyecto desde el punto de vista del negocio.
Elaboración	Desarrollar prototipo	Desarrollar el prototipo de la arquitectura.	Documento de arquitectura.	Definir arquitectura y herramientas. Realizar la lista de riesgos

		Definir y diseñar todos los casos de uso.		técnicos. Prototipo temprano para revisión de usuarios.
Construcción	Primera iteración: Desarrollar versión beta del producto.	Implementar y probar los casos de uso para la versión beta.	Versión beta del producto	Implementación de todas las características claves obtenidas de la arquitectura realizada previamente y desde la perspectiva del usuario.
	Segunda iteración: Desarrollar el reléase inicial.	Implementación y desarrollo de los casos de uso restantes, corregir errores y observaciones de la versión beta.	Producto (software)	Software revisado por los usuarios involucrados en cada uno de los módulos del sistema. Alta calidad del software.
	Tercera iteración: Desarrollo del sistema completo.	Corregir y mejorar los defectos del reléase inicial. Desarrollo del sistema completo.	Producto (software)	Toda la funcionalidad del sistema completa para su liberación.
Transición	Liberar el software	Empaquetar e instalar el reléase.	Software liberado	Sistema en producción.

Tabla 30. Objetivo de las iteraciones.

Fuente: propia.

4.1.2.4.1.3. Calendario del proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

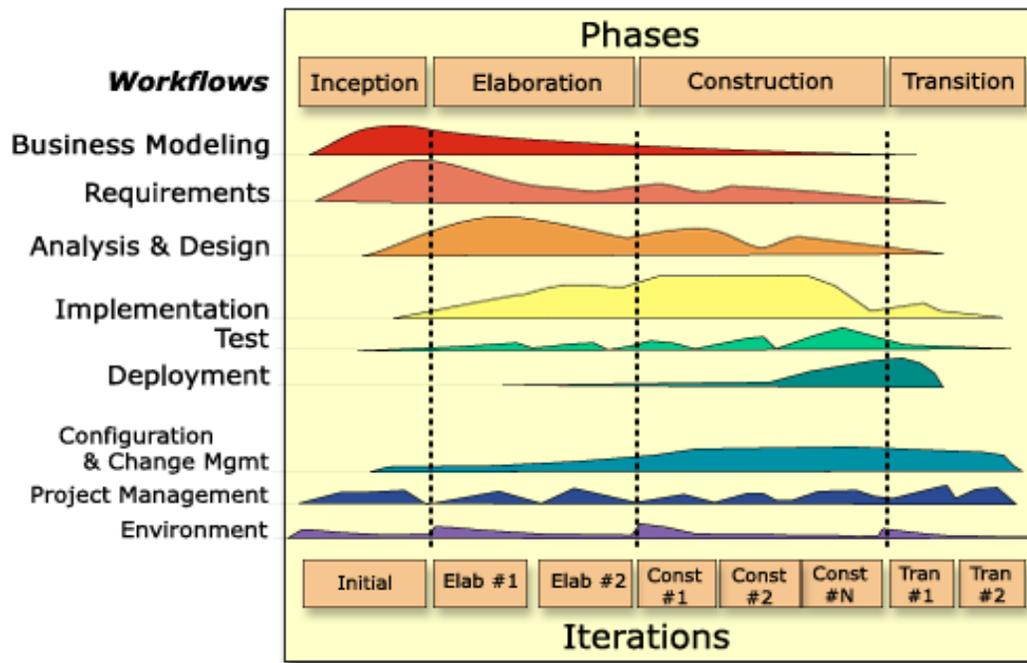


Figura 28. Calendario del Proyecto.

Fuente: (<http://www.dspace.espol.edu.ec/>)

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

Disciplinas / artefactos generados o modificados Durante la fase de inicio	Comienzo	Aprobación
Modelado del negocio		
Modelo de casos de uso del negocio y modelo de objetos del negocio	Semana 1 14/10 – 20/10	Semana 3 28/10 – 3/11
Requisitos		
Glosario	Semana 1 14/10 – 20/10	Semana 3 28/10 – 3/11
Visión	Semana 2 21/10 – 27/10	Semana 3 28/10 – 3/11
Modelo de casos de uso	Semana 3 28/10 – 3/11	Siguiente fase
Especificación de casos de uso	Semana 3 28/10 – 3/11	Siguiente fase
Análisis / diseño		
Modelo de análisis / diseño	Semana 2 21/10 – 27/10	Siguiente fase
Modelo de datos	Semana 2 21/10 – 27/10	Siguiente fase
Implementación		
Modelo de implementación	Semana 3 28/10 – 3/11	Siguiente fase
Pruebas		
Casos de pruebas funcionales	Semana 3 28/10 – 3/11	Siguiente fase
Gestión de cambios y configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de desarrollo del software en su versión 1.0 y planes de las iteraciones	Semana 1 14/10 – 20/10	Semana 3 28/10 – 3/11

Ambiente	Durante todo el proyecto
----------	--------------------------

Tabla 31. Artefactos generados en la fase de inicio.

Fuente: propia.

Disciplinas / artefactos generados o modificados Durante la fase de elaboración	Comienzo	Aprobación
Modelado del negocio		
Modelo de casos de uso del negocio y modelo de objetos del negocio	Semana 1 14/10 – 20/10	Aprobado
Requisitos		
Glosario	Semana 1 14/10 – 20/10	Aprobado
Visión	Semana 2 21/10 – 27/10	Aprobado
Modelo de casos de uso	Semana 3 28/10 – 3/11	Semana 5 11/12 – 17/12
Especificación de casos de uso	Semana 3 28/10 – 3/11	Semana 5 11/12 – 17/12
Análisis / diseño		
Modelo de datos	Semana 2 21/10 – 27/10	Revisar en cada iteración
Implementación		
Modelo de implementación	Semana 3 28/10 – 3/11	Revisar en cada iteración
Pruebas		
Casos de pruebas funcionales	Semana 3 28/10 – 3/11	Revisar en cada iteración
Gestión de cambios y configuración	Durante todo el proyecto	

Gestión del proyecto		
Plan de desarrollo del software en su versión 2.0 y planes de las iteraciones	Semana 4 4/11 – 10/11	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Tabla 32. Artefactos generados en la fase de elaboración.

Fuente: propia.

4.1.2.4.2. Seguimiento y control del proyecto

4.1.2.4.2.1. Gestión de requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

4.1.2.4.2.2. Control de plazos

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

4.1.2.4.2.3. Control de calidad

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

4.1.2.4.2.4. Gestión de riesgos

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

4.1.2.4.2.5. Gestión de configuración

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

4.2. Fase de elaboración

4.2.1. Documento de arquitectura

4.2.1.1. Introducción

4.2.1.1.1. Propósito

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

4.2.1.1.2. Alcance

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

4.2.1.2. Representación de la arquitectura

El presente documento presenta la arquitectura como una serie de vistas los modelos han sido desarrollados usando Visio, Jdeveloper y el lenguaje UML.

El modelo propuesto por RUP para representar la arquitectura utiliza el siguiente conjunto de vistas:

- a) **Vista de casos de uso:** lista los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura.
- b) **Vista lógica:** describe las partes arquitectónicamente significativas del modelo de diseño, como ser la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel que se considere adecuado.
- c) **Vista de procesos:** describe la descomposición del sistema en sub-procesos. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que éstos se comunican.

- d) **Vista de despliegue:** describe uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos.
- e) **Vista de implementación:** describe la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación.
- f) **Vista de datos:** describe los elementos principales del Modelo de Datos, brindando un panorama general de dicho modelo en términos de tablas, vistas, índices, etc.

4.2.1.3. Objetivos y restricciones de la arquitectura

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

- a) Diseño basado en capas de propósito claro y concreto y con alto grado de cohesión.
- b) Desacoplamiento entre capas que permita el fácil reemplazo de los mismos.
- c) Capas altamente reutilizables.
- d) Todos los requerimientos descritos en el documento de Visión deben ser tomados en consideración para el desarrollo de la arquitectura definida.

4.2.1.3.1. Proceso (ver capítulo 3)

Actores

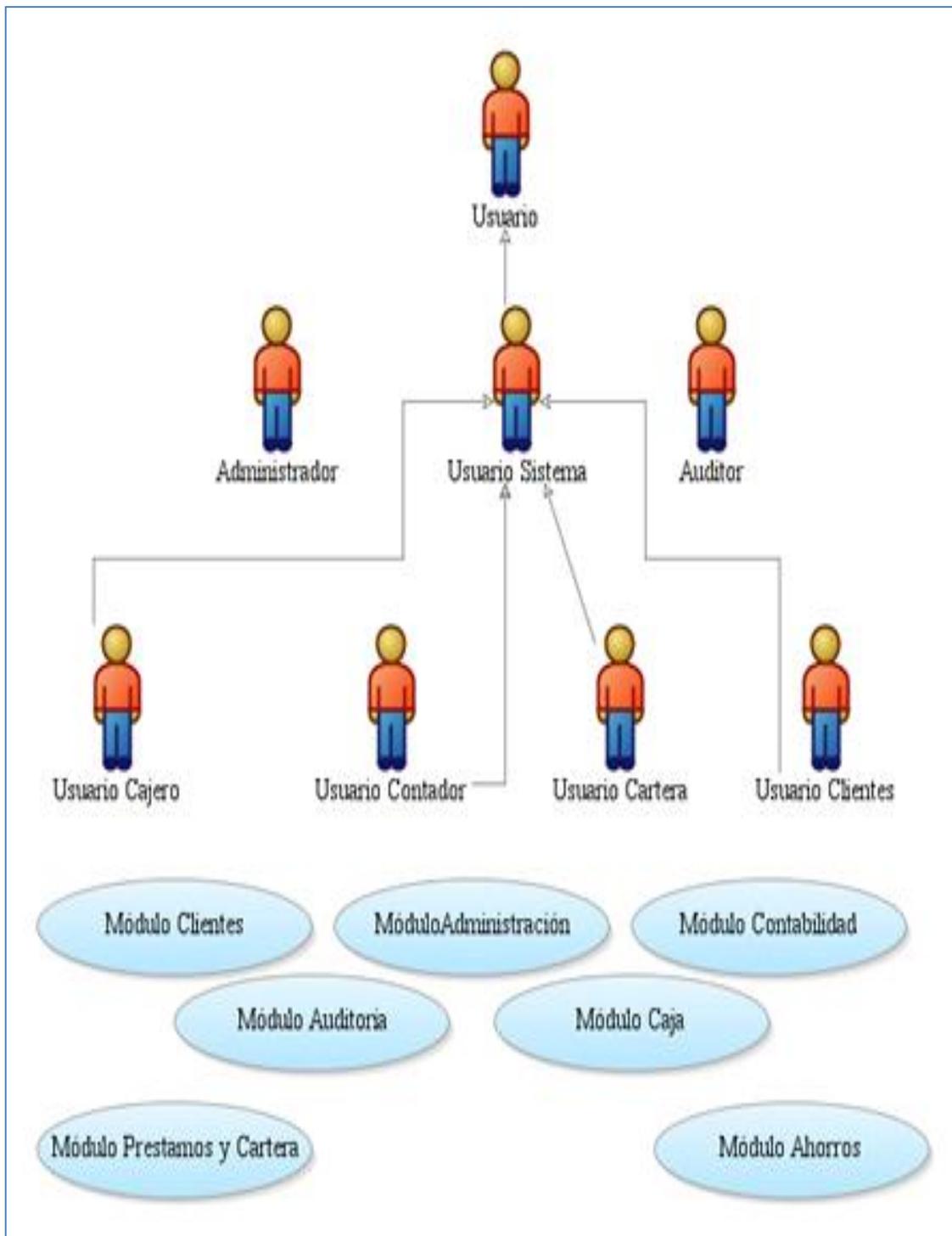


Figura 29. Actores.

Fuente: propia.

Descripción de los interesados.

Nro.	Stakeholder	Descripción
1	Usuario	Usuario general del sistema que puede ingresar requerimientos y consultar la base de datos
2	Administrador	Administrador del sistema quién gestiona los objetos y recursos del sistema
4	Usuario Auditor	Realiza consultas de ingresos y salidas de usuarios, movimientos (ingreso, modificación, eliminación) de la información del sistema.
5	Usuario Contador	Procesa todas las operaciones contables que implica el sistema financiero.
6	Usuario Clientes	Se encarga de actividades relacionadas con clientes nuevos y existentes.
7	Usuario Cartera	Analiza la simulación, viabilidad, registro y asignación de préstamos a los clientes.
8	Usuario Cajero	Se encarga de realizar todas las transacciones de los clientes en las actividades de depósitos retiros, cobros de préstamos, etc.

Tabla 33. Descripción de los interesados.

Fuente: propia.

4.2.1.3.2. Modelos de casos de negocio

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

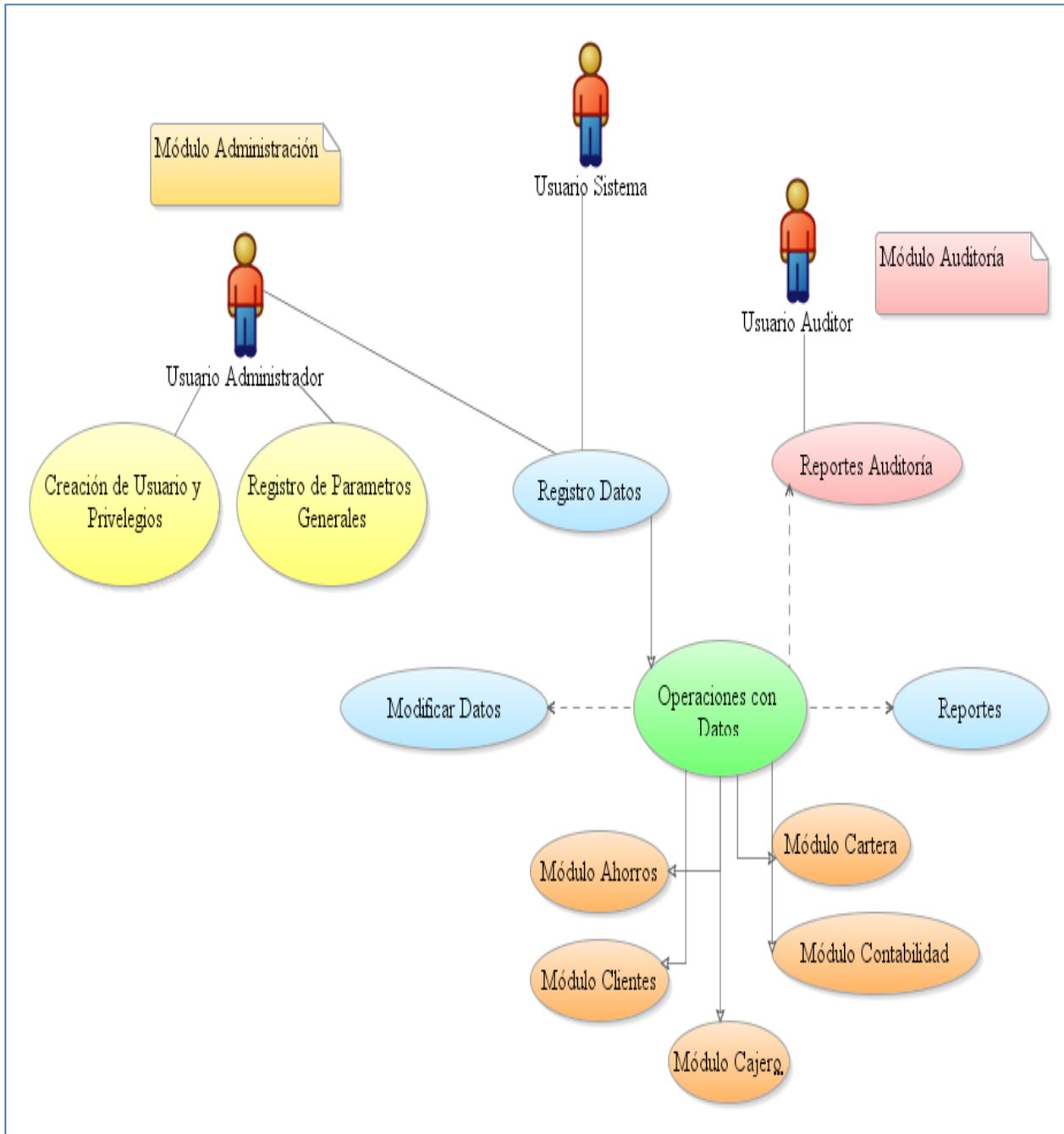


Figura 30. Modelo de casos de negocios.

Fuente: propia.

4.2.1.3.3. Prioridad de casos de uso

Caso de uso	Prioridad para el negocio	Prioridad técnica
1. Módulo préstamos-cartera		
1. Simular préstamo	Media	Baja
2. Registrar datos del préstamo	Alta	Alta
3. Situación económica.	Alta	Alta
4. Garantías del préstamo	Alta	Alta
5. Estudio de factibilidad del préstamo	Alta	Alta
6. Generar solicitud de préstamo	Alta	Alta
7. Aprobar o negar préstamos	Alta	Alta
8. Administrar líneas de crédito	Alta	Alta
9. Administrar grupos de líneas de crédito	Alta	Alta
2. Módulo clientes		
1. Ingreso nuevo cliente.	Alta	Alta
2. Persona natural	Alta	Alta
3. Datos personales	Alta	Alta
4. Referencias personales	Media	Media
5. Actividad económica	Media	Media
6. Cónyuge	Media	Media
7. Representante del menor	Alta	Alta
8. Persona jurídica	Alta	Alta
9. Personería jurídica	Alta	Alta
10. Creación del cliente	Alta	Alta
11. Administrar información de provincias, cantones y parroquias	Alta	Alta
3. Módulo auditoría		
1. Auditar	Alta	Alta
2. Interacción con la bdd.	Alta	Alta
3. Sesiones	Alta	Alta
4. Aplicar filtros	Media	Media
5. Presentar informe	Alta	Alta

Tabla 34. Prioridad de casos de uso.

Fuente: propia.

4.2.1.3.4. Modelos de casos de uso (Ver capítulo 3)

4.2.1.3.5. Descripción de los casos de uso más relevantes

4.2.1.3.5.1. Descripción módulo Préstamos-Cartera

Simular préstamo

Permite realizar una estimación de las cuotas a pagar dependiendo de la cantidad de las variables usadas en la aprobación de un préstamo.

Ingreso de datos del préstamo

Se ingresa toda la información relacionada al préstamo, como ser, el cliente que solicita el préstamo, el tipo de préstamo, la cantidad, el número de cuotas, la frecuencia de pagos, las garantías, la situación económica del cliente.

Generar solicitud del préstamo

Los datos ingresados generan un informe y pasa a la comisión de créditos, para que realicen el estudio de factibilidad del préstamo

Aprobar o negar préstamos

De acuerdo al estudio realizado por la comisión, se le aprueba o niega el préstamo al cliente.

4.2.1.3.5.2. Descripción módulo Clientes

Persona natural

Se ingresan todos los datos relacionados a la persona natural como ser los datos personales, referencias personales, actividad económica, cónyuge y representante de menor de edad si fuera el caso.

Persona jurídica

Se ingresan los datos de la empresa o persona jurídica como ser razón social, ruc y los datos personales y dirección de los representantes legales.

Creación del cliente

De acuerdo al tipo de persona seleccionado (persona natural o jurídica) se crea un nuevo cliente de la Cooperativa.

4.2.1.3.5.3. Descripción módulo Auditoría Informática

Auditar Sesiones

Puede ver los ingresos y cierres de sesión por parte de los usuarios del sistema.

Auditar interacción con la BDD

Puede ver el movimiento del flujo de información realizada en la base de datos.

Aplicar filtros

En ambos tipos de auditorías, ya sea por sesiones o por interacción con la BDD, se pueden aplicar filtros, para ver sólo la información que desee el usuario.

4.2.1.4. Vista de restricciones

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

4.2.1.4.1. Normativas

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

4.2.1.4.2. Licenciamiento

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

4.2.1.4.3. Estándares

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

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

4.2.1.4.4. Tecnología

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

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

		páginas web dinámicas.
Control de versiones	CVS	Mantiene el registro de todo el trabajo y los cambios en los ficheros que forman parte de un proyecto, permitiendo que distintos desarrolladores colaboren.
Sistema operativo	De desarrollo: Fedora	Distribución Linux muy estable, de propósitos generales basada en RPM respaldada por Red Hat.
	De producción: Centos	Distribución Linux popular en el segmento de servidores web, derivada y con total compatibilidad de Red Hat.
Herramienta de reportes	JasperReports	Herramienta escrita en Java para la creación de informes con contenido enriquecido.
Metodología	RUP	Metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos

Tabla 35. Tecnologías empleadas.

Fuente: propia.

4.2.1.4.5. Sistemas existentes

4.2.1.4.5.1. Sistema financiero CUC

La CUC (Cooperativa Unión Cochabamba) en la actualidad trabaja con un sistema programado en C++, El cual se encarga en la totalidad de procesar toda la información financiera generada, sin embargo dicho sistema presenta limitaciones, porque no controla y administra todos los procesos de forma unificada, además los datos no se guardan en ningún gestor de base de datos sino en archivos planos.

4.2.1.5. Vistas lógicas

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

4.2.1.5.1. Arquitectura del sistema

FINANSYS será desarrollado utilizando tecnología Web, desarrollando la parte visual con RichFaces optimizando de esta forma el rendimiento y facilidad de uso por parte de los usuarios, creando pantallas con una presentación clara y atractiva gracias a títulos barras de estado, íconos intuitivos, tipografía legible, etc. La figura 33 indica la arquitectura básica de desarrollo y separa sus funcionalidades en tres capas diferenciando claramente el funcionamiento y el proceso del sistema. Esta lógica se basa en el patrón arquitectónico MVC (Modelo Vista Controlador).

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

Sus capas principales son el Modelo, las Vistas y los Controladores las cuales se manipulan como entidades separadas, lo que significa que cualquier cambio realizado en el Modelo se manifieste automáticamente en las Vistas.

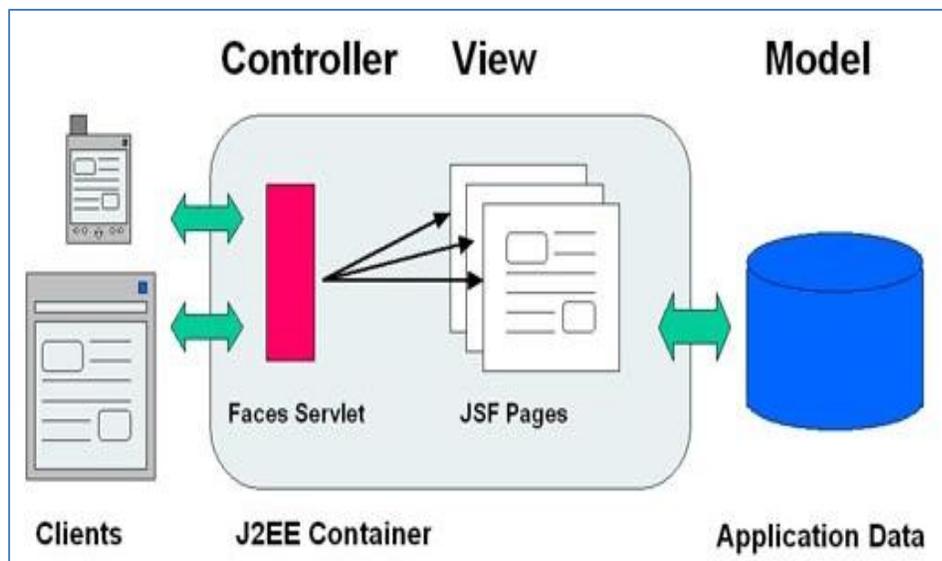


Figura 31. Patrón arquitectónico MVC.

Fuente: (<http://fedoraproject.org/es>)

4.2.1.5.2. Arquitectura lógica

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

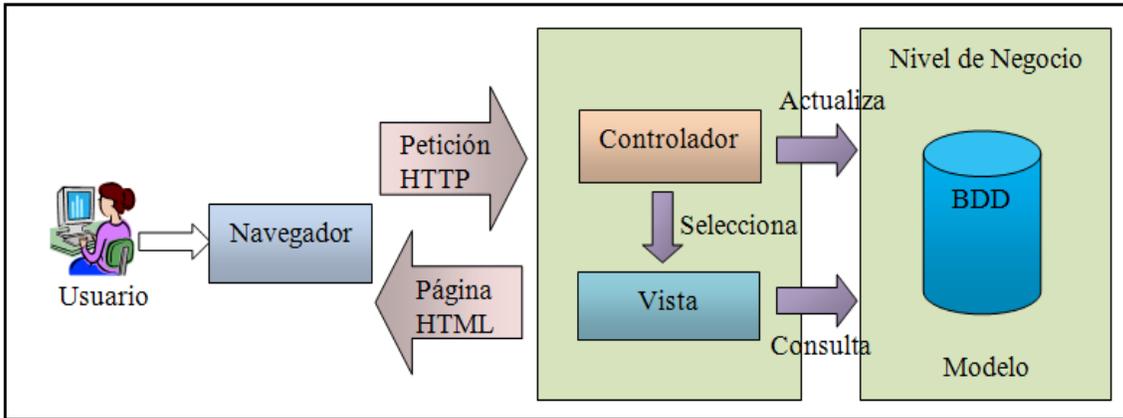


Figura 32. Arquitectura lógica del sistema.

Fuente: propia.

4.2.1.5.2.1. Interfaz de usuario

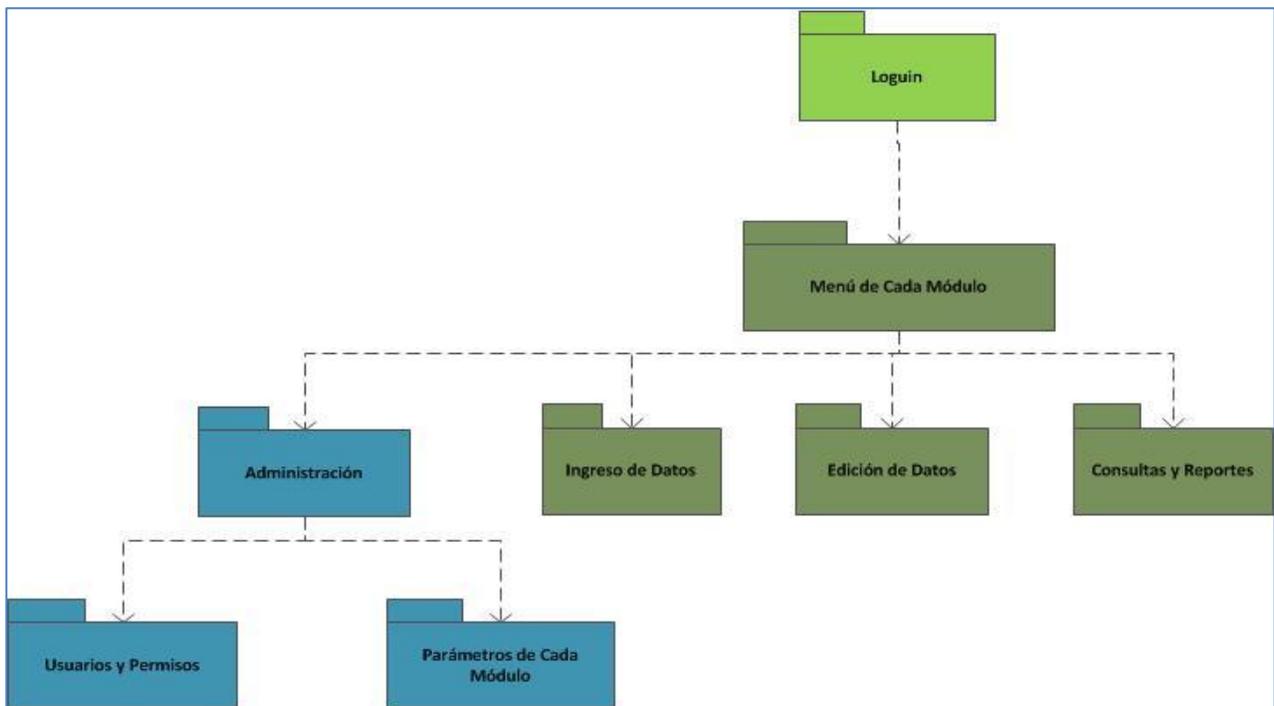


Figura 33. Interfaz de usuario.

Fuente: propia.

4.2.1.5.3. Servicios del sistema

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

4.2.1.5.4. Infraestructura

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

4.2.1.5.4.1. Aplicación

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

4.2.1.5.4.2. Datos

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

4.2.1.5.4.3. Sistema operativo

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

4.2.1.5.5. Vista de datos

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

4.2.1.6. Vista de despliegue

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

Como se puede observar en la figura, un usuario podrá acceder a la aplicación mediante un browser siendo especificado Firefox como browser necesario el cual ejecutará la página JSF-RichFaces con

extensión .jsp, el mismo que muestra la portada de login que permite el acceso al sistema. La conexión a la base de datos se realiza a través de Hibernate (Herramienta ORM).

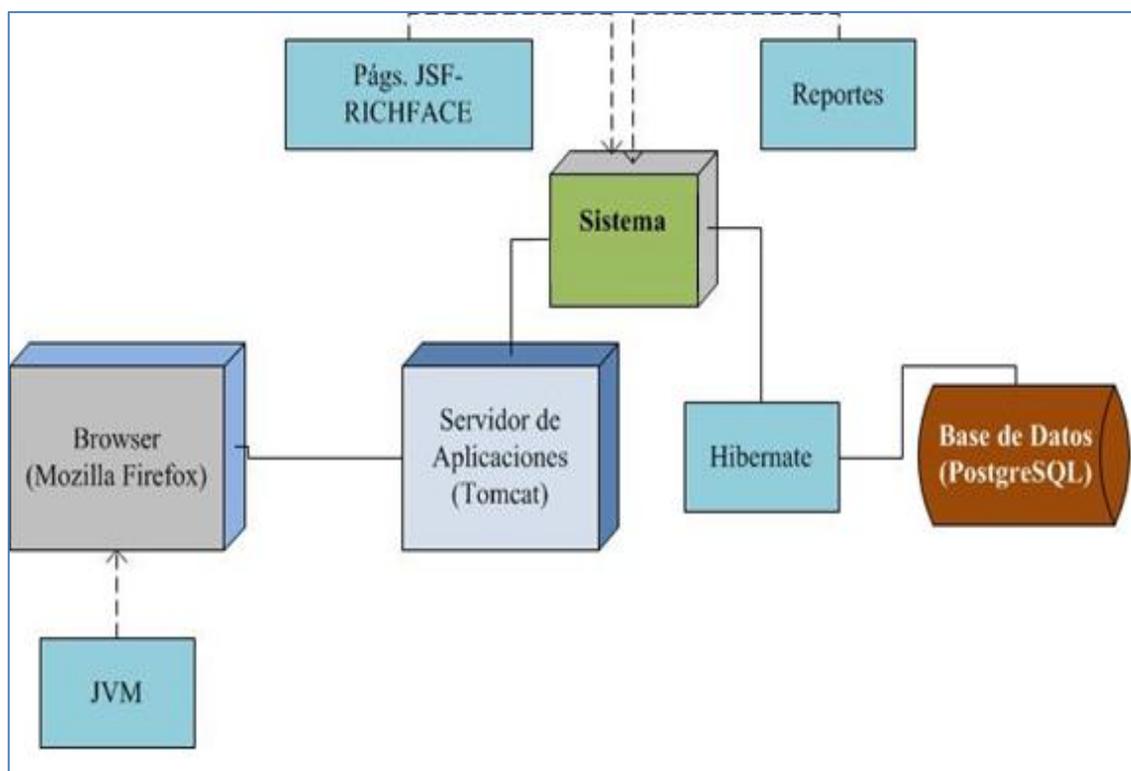


Figura 34. Vista de despliegue.

Fuente: propia.

4.2.2. Casos de uso

Las especificaciones de los casos de uso de los módulos de Clientes, Préstamos-Cartera y Auditoría Informática se encuentran en los anexos del CD.

4.3. Fase de construcción

4.3.1. Modelo de datos

A continuación se describe gráficamente la estructura de las tablas empleada por el sistema para almacenar la información relacionada con los módulos de Clientes, Cartera y Auditoría.

4.3.1.1. Diseño de las tablas del módulo Préstamos-Cartera

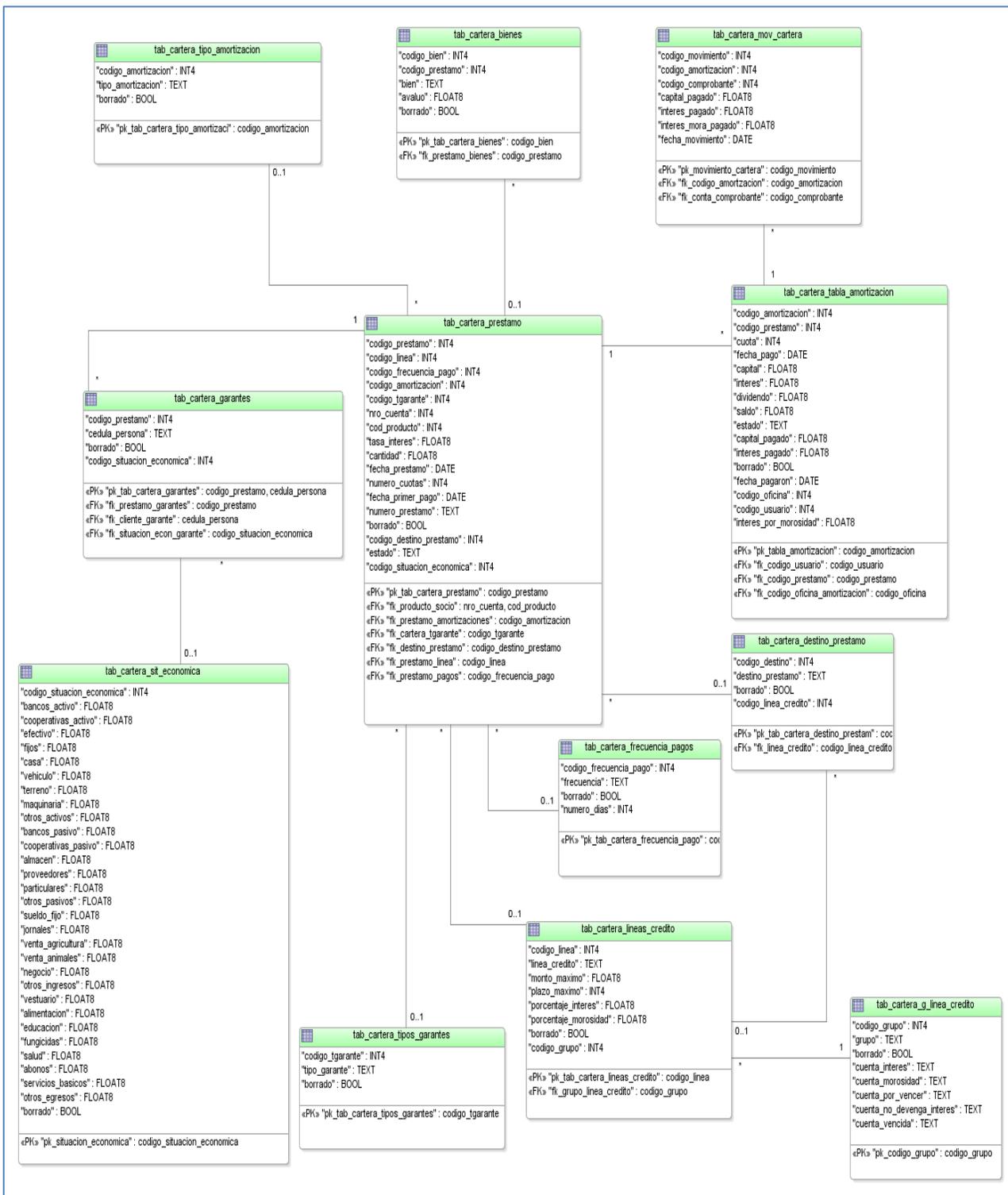


Figura 35. Tablas del módulo Préstamos-Cartera.

Fuente: propia.

4.3.1.2. Diseño de las tablas del módulo Clientes

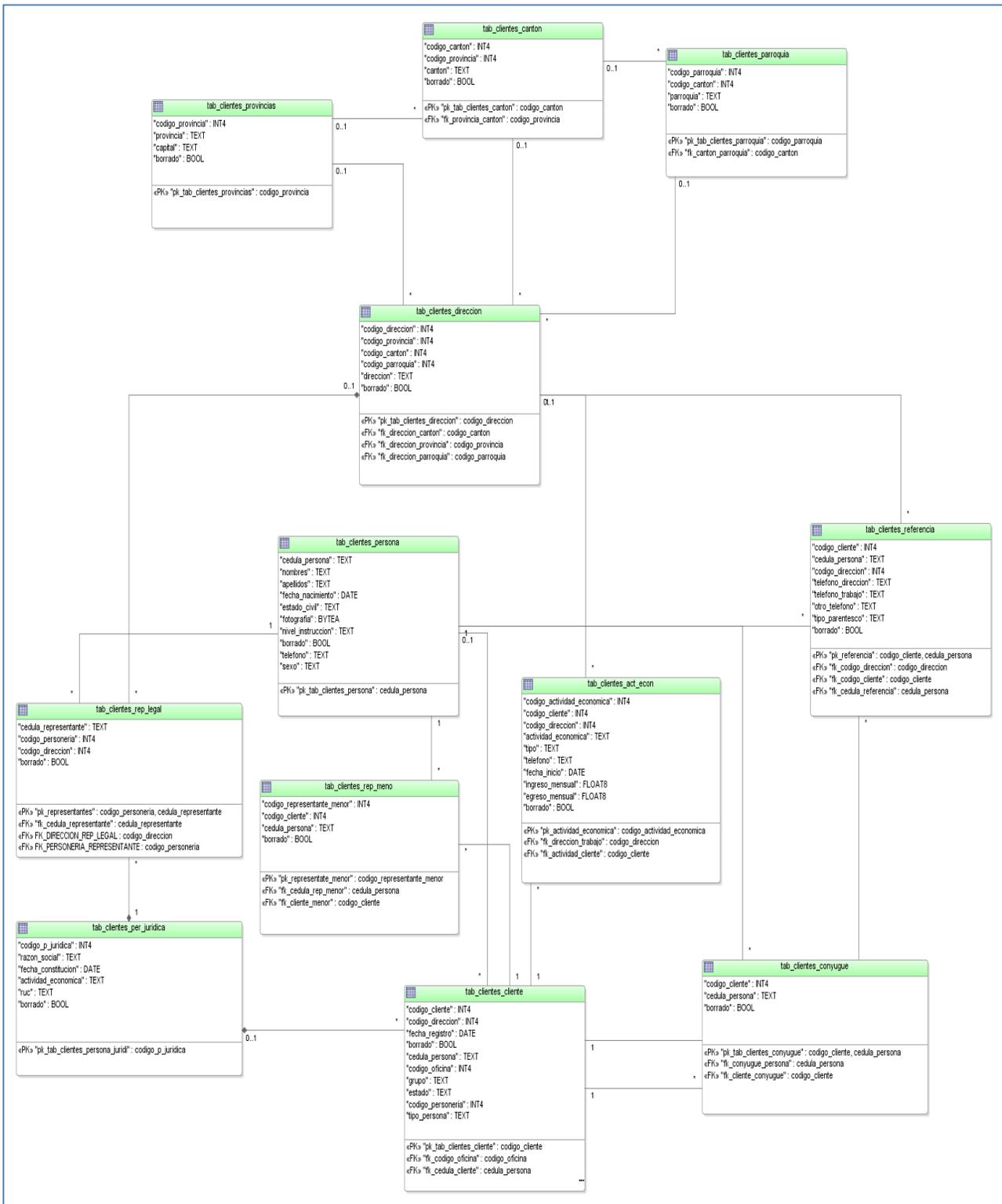


Figura 36. Tablas del módulo Clientes.

Fuente: propia.

4.3.1.3. Diseño de las tablas del módulo Auditoría Informática

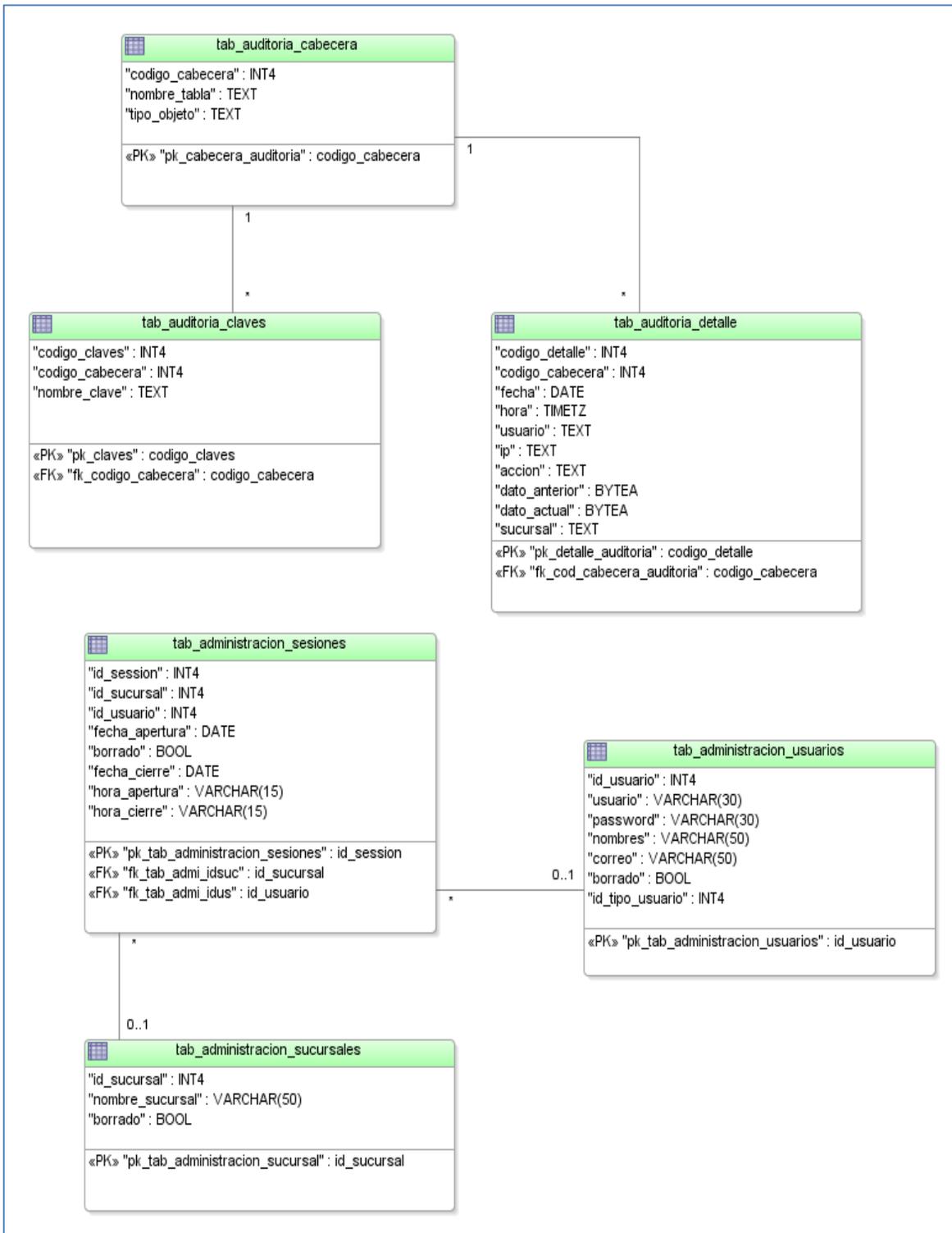


Figura 37. Tablas del módulo Auditoría.

Fuente: propia.

4.3.2. Plan de pruebas

4.3.2.1. Introducción

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

4.3.2.1.1. Propósito

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

4.3.2.1.2. Alcance

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

4.3.2.2. Estrategia de pruebas

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

4.3.2.2.1. Tipos de pruebas

3.3.2.2.1.1. Pruebas de integridad de datos

Objetivo:	Asegurar la integridad de datos
Técnica:	Registrar datos con tipos válidos. Registrar datos en entidad que tengan relación con otras. Revisar el esquema de base de datos para asegurarse que los datos se han guardado satisfactoriamente y de acuerdo a los estándares definidos.
Criterio de completitud:	Todos los métodos de acceso y procesos de la Base de datos funcionan como fueron diseñados.

Consideraciones especiales:	<p>Se debe utilizar un conjunto pequeño de datos para incrementar la visibilidad de cualquier evento anormal o inesperado.</p> <p>Los datos de pruebas deberían ser reales y de uso común.</p>
------------------------------------	--

Tabla 36. Prueba de integridad de datos.

Fuente: propia.

3.3.2.2.1.2. Pruebas del sistema

Objetivo:	Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.
Técnica:	<p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <p>Los resultados esperados ocurren cuando se utiliza un dato válido.</p> <p>Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.</p> <p>Cada regla de negocios es aplicada adecuadamente.</p>
Criterio de completitud:	<p>Todas las pruebas planeadas han sido ejecutadas.</p> <p>Todos los defectos que se identificaron han sido tenidos en cuenta.</p>
Consideraciones especiales:	Considerar aspectos que impactan la implementación y ejecución de las pruebas del Sistema

Tabla 37. Pruebas del sistema.

Fuente: propia.

3.3.2.2.1.3. Pruebas del ciclo del negocio

Objetivo:	Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo.
Descripción de la prueba:	Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en él a través del tiempo. Debería identificarse un período, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un período de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales.
Técnicas:	<p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <p>Incrementa el número de veces en que una función es ejecutada para simular diferentes usuarios sobre un período especificado</p> <p>Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o períodos de tiempo.</p> <p>Todas las funciones que ocurren en un período de tiempo serán ejecutadas en el tiempo apropiado.</p> <p>Los resultados esperados ocurren cuando los datos válidos son usados.</p> <p>Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.</p> <p>Cada regla de negocios es aplicada adecuadamente.</p>
Criterio de completitud:	<p>Todas las pruebas planeadas han sido ejecutadas.</p> <p>Todos los defectos que se identificaron han sido tenidos en cuenta.</p>
Consideraciones especiales:	<p>Las fechas y eventos del sistema pueden requerir actividades especiales de soporte.</p> <p>Se requiere un modelo de negocios para identificar requisitos y procedimientos de prueba apropiados.</p>

Tabla 38. Pruebas del ciclo del negocio.

Fuente: propia.

4.3.2.2.2. Pruebas de interfaz de usuario

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

Tabla 39. Pruebas de interfaz de usuario.

Fuente: propia.

4.3.2.2.3. Pruebas de desempeño

Objetivo:	<p>Validar y verificar los requisitos de desempeño especificados para el sistema.</p> <p>Validar el tiempo de respuesta para las transacciones o procesos bajo las siguientes condiciones:</p> <p>Volumen normal anticipado.</p> <p>Volumen máximo anticipado.</p>
Descripción de la prueba:	<p>Las pruebas de desempeño miden los tiempos de respuesta que tiene el sistema y otros aspectos sensibles al tiempo.</p> <p>Las pruebas se realizan varias veces, cambiando entre una y otra cargas diferentes. La prueba inicial debe ejecutarse con una carga similar a la esperada, mientras que la prueba final se ejecuta utilizando una carga máxima esperada.</p> <p>Las pruebas se pueden utilizar para calibrar el desempeño del sistema en función de condiciones como el hardware, usuarios, etc.</p> <p>Algunas características que pueden afectar el desempeño son:</p> <p>Cuellos de botella en el disco.</p> <p>Cuellos de botella en el CPU.</p> <p>Capacidad de almacenamiento.</p> <p>Capacidades físicas del hardware donde está alojado el sistema.</p> <p>Virus informáticos.</p> <p>Congestión en la red de datos.</p>

Técnicas:	<p>Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema.</p> <p>Los usuarios del sistema son reales y trabajan en su área de trabajo normal.</p> <p>Se pide a los usuarios que todos accedan a una determinada acción o proceso al mismo tiempo.</p> <p>Los desarrolladores no están presentes.</p> <p>Los usuarios son advertidos que el sistema puede fallar.</p>
Criterio de completitud:	<p>Se establece un período de pruebas, en el que los errores detectados no sean clasificados como críticos para el sistema.</p>
Consideraciones especiales:	<p>Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados.</p>

Tabla 40. Pruebas de desempeño.

Fuente: propia.

4.3.2.2.4. Pruebas de seguridad y control de acceso

Objetivo:	<p>Seguridad en el Funcionamiento y Datos, verificar que los usuarios puedan acceder solo aquellas funciones y datos para los cuales se le ha otorgado permisos al momento de crear su perfil de usuario.</p> <p>Seguridad en Administración del Sistema, comprobar que solo aquellos usuarios con permisos privilegiados puedan acceder a las funciones parametrizables del sistema y opciones del sistema.</p>
Descripción de la prueba:	<p>Las pruebas se realizará enfocándonos en dos aspectos principales:</p> <p>Seguridad en la aplicación controlando el acceso a determinada información y funciones del negocio, y</p> <p>Seguridad del sistema realizando registro de accesos de usuarios al sistema.</p>
Técnicas:	<p>Seguridad de Datos y Funciones, identificar los tipos de usuarios y asignar funciones a las que tiene acceso el tipo de usuarios para asignarle al usuario.</p> <p>Realizar pruebas para cada tipo de usuario y verificar los permisos creado transacciones para cada tipo de usuario.</p> <p>Re direccionamiento a la página de autenticación si el usuario aún no se ha logeado o registrado para el acceso al sistema.</p> <p>Modificar los tipos de usuario y verificar si los permisos han cambiado para el usuario.</p>
Criterio de completitud:	<p>Para cada tipo de dato se puede asignar las funciones y datos apropiados para su desempeño.</p>
Consideraciones especiales:	<p>El acceso al sistema debe ser revisado con el administrador de la red y de la base de datos. Con esto podremos visualizar cualquier anomalía.</p>

Tabla 41. Pruebas de seguridad y control de acceso.

Fuente: propia.

4.3.2.3. Herramientas

Tarea	Herramientas
Registro de defectos	Microsoft Word
Otras herramientas de prueba	SQL PLUS
Gestión de proyecto	JDEVELOPER, Microsoft Project Microsoft Word
Herramientas DBMS	PGADMIN III

Tabla 42. Herramientas.

Fuente: propia.

4.3.2.4. Recursos

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

Tabla 43. Recursos.

Fuente: propia.

4.3.2.5. Entregables

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

Tabla 44. Entregables.

Fuente: propia.

4.3.3. Lista de riesgos

Ranking	Descripción del Riesgo e Impacto	Estrategia de reducción del riesgo
7	La liberación del sistema FINANSYS podría no estar lista para el mes de enero del 2012, mes en que se inicia un nuevo período contable.	Aumentar el esfuerzo.
7	Que la Cooperativa no haga la adquisición del servidor a tiempo.	Recordar a la administración de la Cooperativa la adquisición del servidor.
5	Que las características del servidor donde será alojado el sistema, no cumpla con los requerimientos necesarios para un correcto funcionamiento del sistema.	Antes de su uso especificar los debidos requerimientos de hardware y software.
3	Abandono del proyecto por parte de los desarrolladores.	Aumentar el compromiso de terminar todo el sistema.
3	Incompatibilidad del navegador de internet de los usuarios.	Instalar el navegador compatible con el sistema.

3	Modificación de los requerimientos puestos en marcha.	Realizar actas de reunión de trabajo, definir bien los requerimientos y que el interesado apruebe dicha acta con la respectiva firma.
3	Saturación de la red informática por parte de los usuarios en actividades ajenas al sistema.	Sugerir políticas de uso de internet.
2	Presencia de virus en las computadores de los usuarios finales.	Instalar un antivirus y actualizar constantemente la base de datos del anti virus.

Tabla 45. Lista de riesgos.

Fuente: propia.

4.4. Fase de transición

4.4.1. Manual de instalación

4.4.1.1. Introducción

A continuación se describen los pasos necesarios para configurar en el servidor todas las aplicaciones y configuraciones necesarias para el correcto funcionamiento del sistema.

Copiar el software al escritorio

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

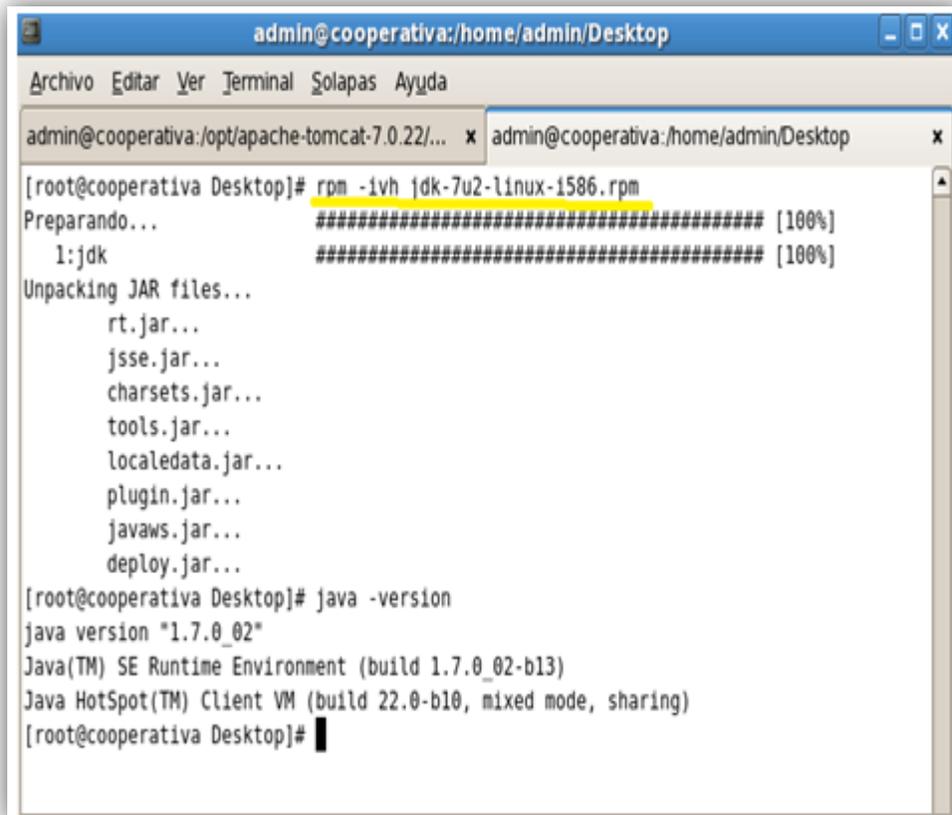
Se abre una consola, debe de autenticarse como súper usuario (root), acceder desde consola al directorio del escritorio (*cd /home/admin/Desktop*) y dar todos los permisos a todos los archivos (*chmod 777 **).

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

4.4.1.2. Instalación de la máquina virtual de Java

Desde la consola se debe digitar el siguiente comando:

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



```
admin@cooperativa:/home/admin/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
admin@cooperativa:/opt/apache-tomcat-7.0.22/... x admin@cooperativa:/home/admin/Desktop x
[root@cooperativa Desktop]# rpm -ivh jdk-7u2-linux-i586.rpm
Preparando... ##### [100%]
1:jdk ##### [100%]
Unpacking JAR files...
  rt.jar...
  jsse.jar...
  charsets.jar...
  tools.jar...
  localedata.jar...
  plugin.jar...
  javaws.jar...
  deploy.jar...
[root@cooperativa Desktop]# java -version
java version "1.7.0_02"
Java(TM) SE Runtime Environment (build 1.7.0_02-b13)
Java HotSpot(TM) Client VM (build 22.0-b10, mixed mode, sharing)
[root@cooperativa Desktop]#
```

Figura 38. Instalación de la máquina virtual de Java.

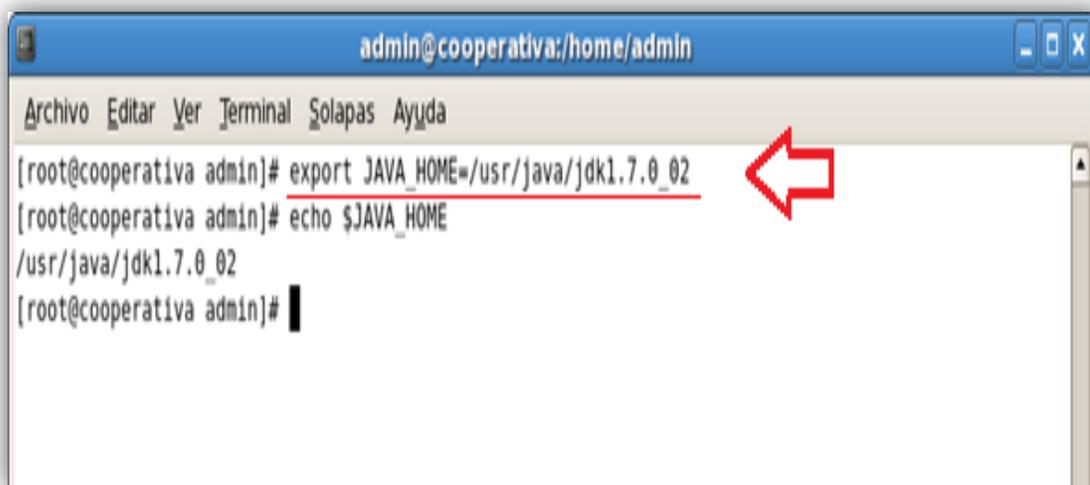
Fuente: propia.

El resultado debe ser como se muestra en el gráfico.

Digitando el comando `java -version` podemos comprobar la versión del java que acabamos de instalar.

4.4.1.3. Configuración de JAVA_HOME

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

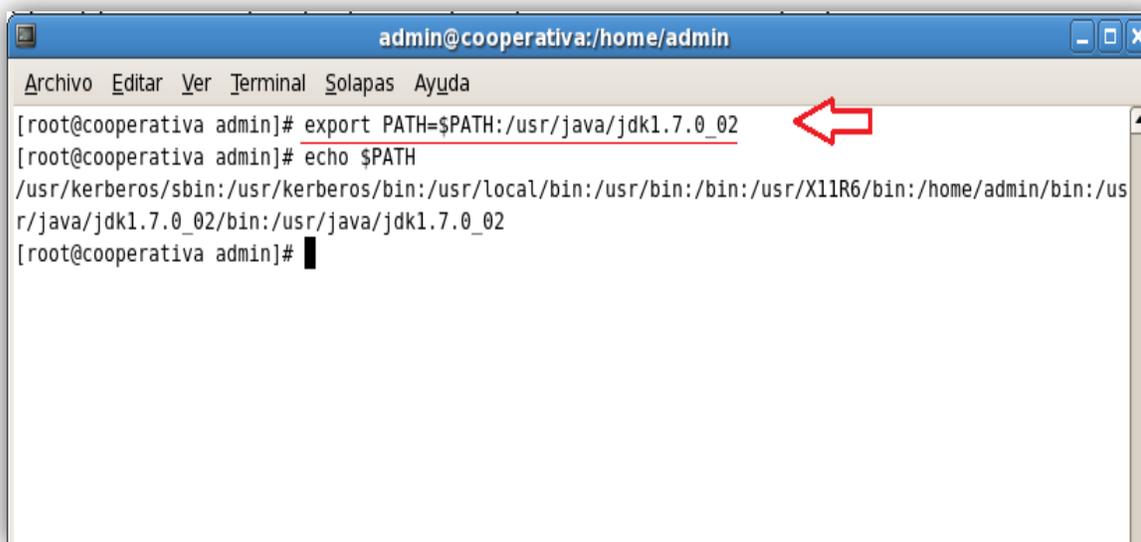


```
admin@cooperativa:/home/admin
Archivo Editar Ver Terminal Solapas Ayuda
[root@cooperativa admin]# export JAVA_HOME=/usr/java/jdk1.7.0_02
[root@cooperativa admin]# echo $JAVA_HOME
/usr/java/jdk1.7.0_02
[root@cooperativa admin]#
```

Figura 39. Configuración del JAVA_HOME

Fuente: propia.

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



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

Figura 40. Configuración del PATH

Fuente: propia.

4.4.1.4. Instalación de PostgreSQL

Desde la consola se debe digitar el siguiente comando:

```
# ./postgresql-9.1.2-1-linux.run
```

A continuación se carga un asistente gráfico de instalación, los pasos son los siguientes:



Figura 41. Asistente de instalación de PostgreSQL.

Fuente: propia.



Figura 42. Asistente de instalación de PostgreSQL.

Fuente: propia.

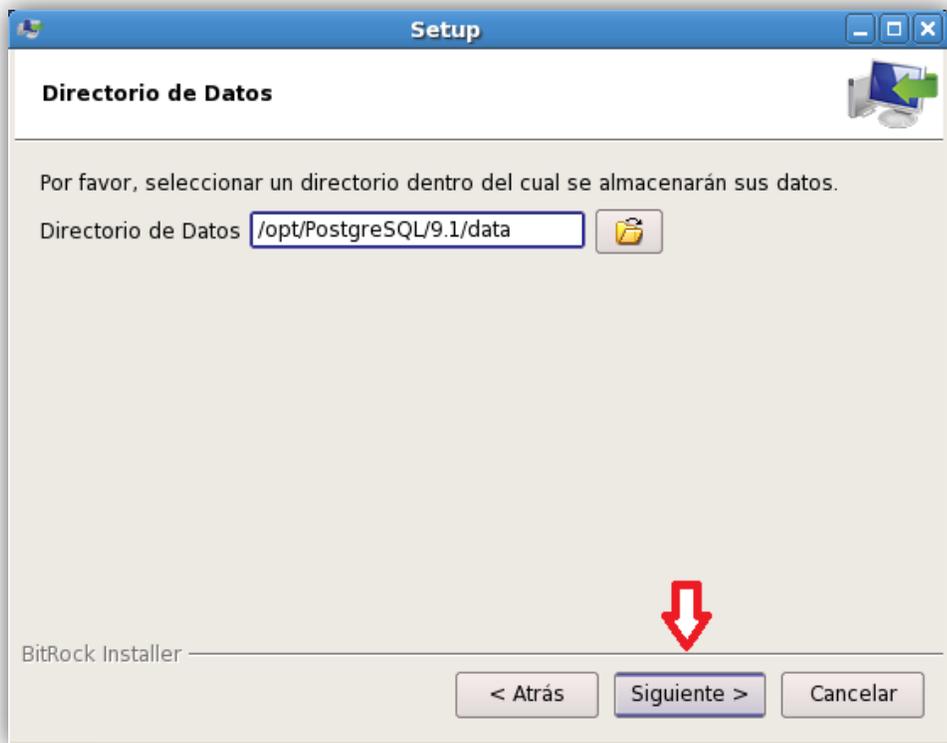


Figura 43. Asistente de instalación de PostgreSQL.

Fuente: propia.

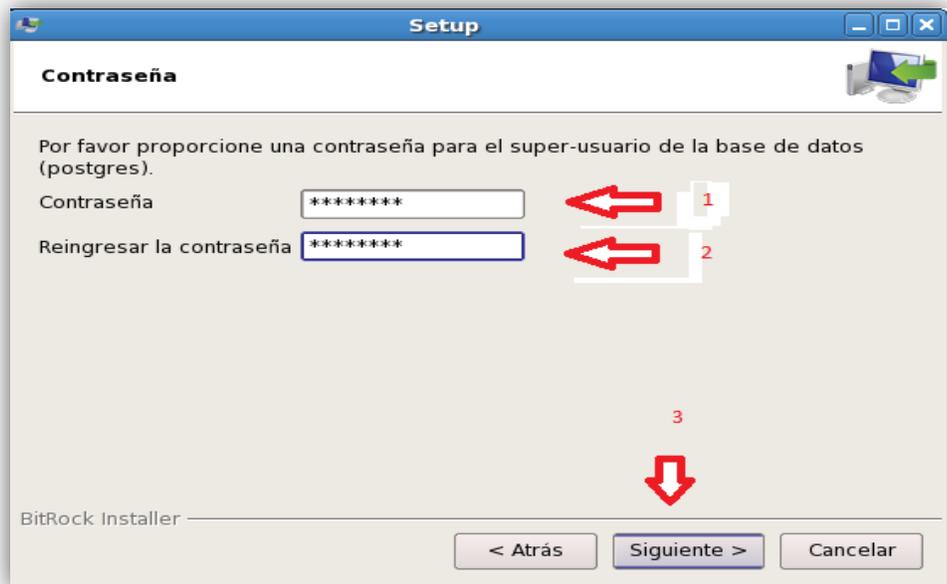


Figura 44. Asistente de instalación de PostgreSQL.

Fuente: propia.

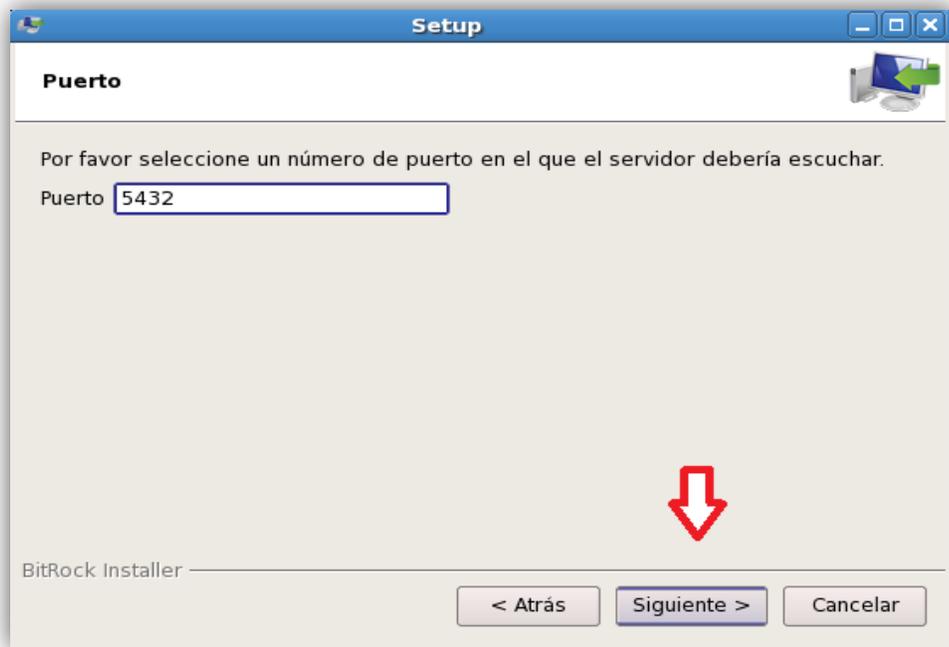


Figura 45. Asistente de instalación de PostgreSQL.

Fuente: propia.



Figura 46. Asistente de instalación de PostgreSQL.

Fuente: propia.

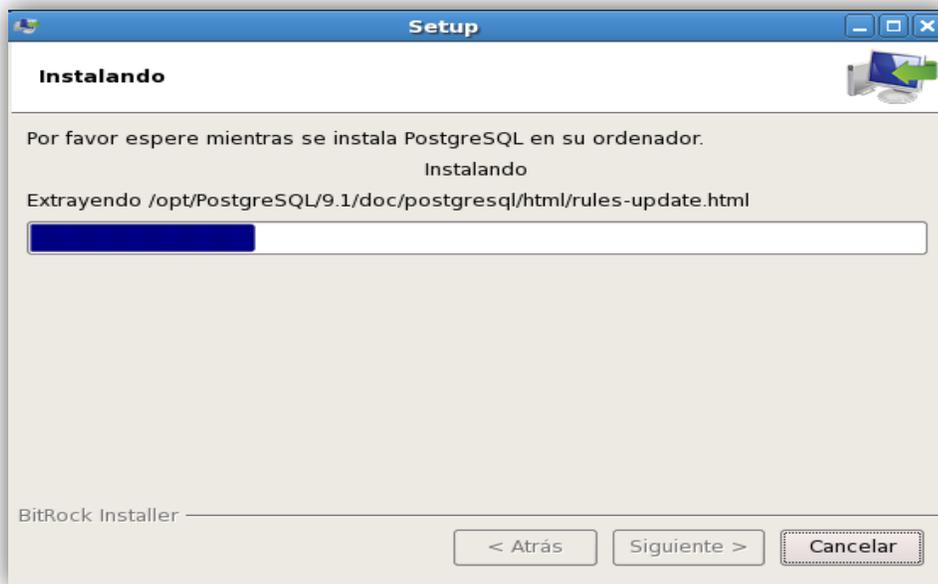


Figura 47. Asistente de instalación de PostgreSQL.

Fuente: propia.



Figura 48. Asistente de instalación de PostgreSQL.

Fuente: propia.

Una vez instalado la base de datos, procedemos a abrir PgAdmin III, para realizar acciones en el motor de base de datos.

Se debe autenticar con el password proporcionado en el proceso de instalación. Dentro de PgAdmin III, dar clic derecho en Login Roles y seleccionar New Login Role como muestra la imagen.

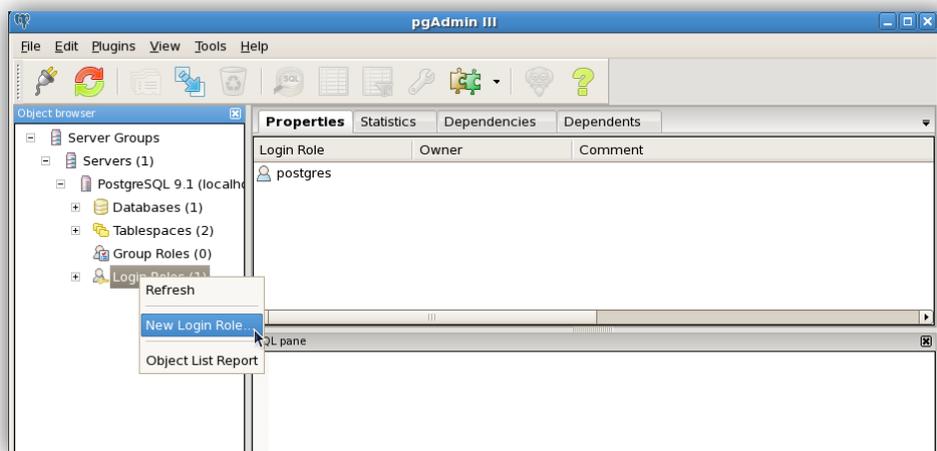


Figura 49. Creación de nuevo usuario.

Fuente: propia.

En Role name escribir FINANSYS.

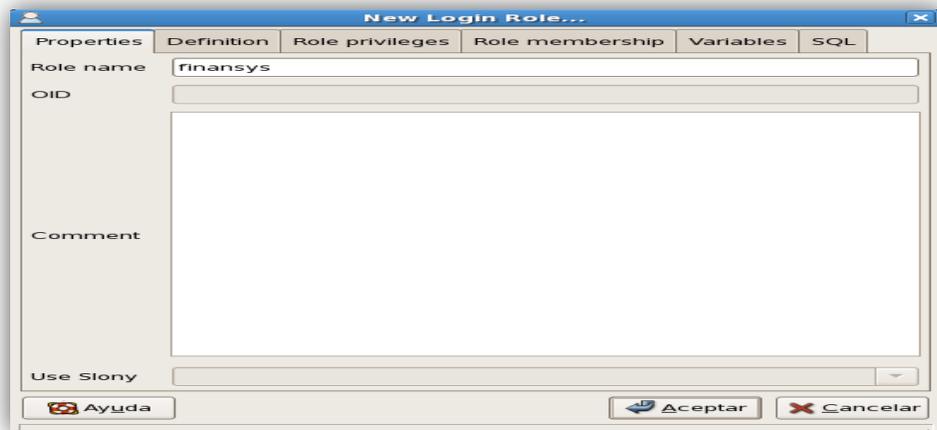


Figura 50. Creación de nuevo usuario.

Fuente: propia.

Luego en la pestaña Definition asignar el password que tendrá dicho usuario.

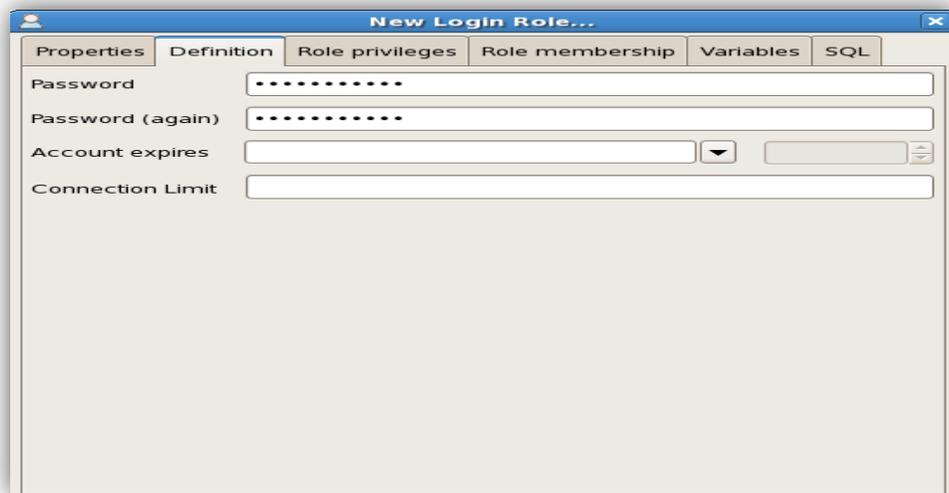


Figura 51. Creación de nuevo usuario.

Fuente: propia.

De igual forma se procede a crear una nueva base de datos dando clic derecho en la opción Databases y seleccionando la opción new database.

Crear la base de datos con el nombre FINANSYS y elegir al usuario creado como se muestra en el gráfico.

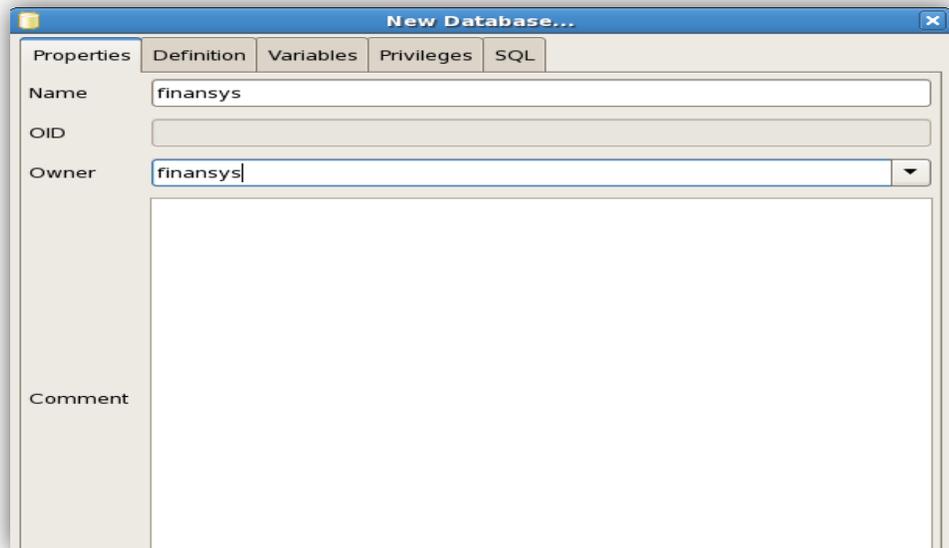


Figura 52. Creación de una nueva base de datos.

Fuente: propia.

Por último se procede a cargar el backup, el mismo que contiene el script de las tablas y algunos datos básicos que necesita el sistema.

En la base de datos creada, dar clic derecho y seleccionar la opción Restore.

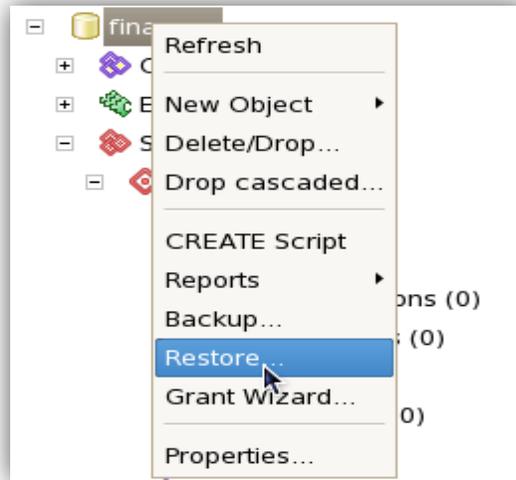


Figura 53. Subir backup a la base de datos.

Fuente: propia.

A continuación en la ventana que se carga, primero se debe seleccionar un archivo con extensión .backup, asignarle el usuario creado y presionar el botón Restore tal como se muestra en la imagen.

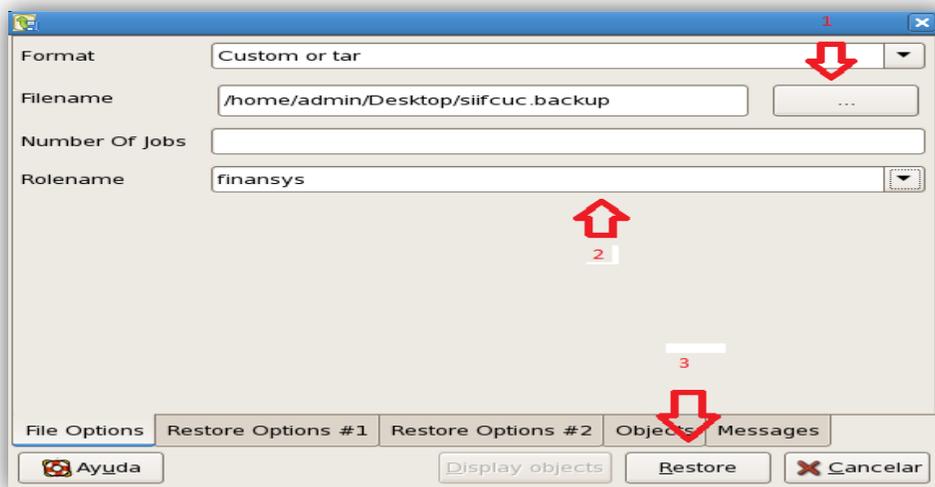


Figura 54. Subir backup a la base de datos.

Fuente: propia.

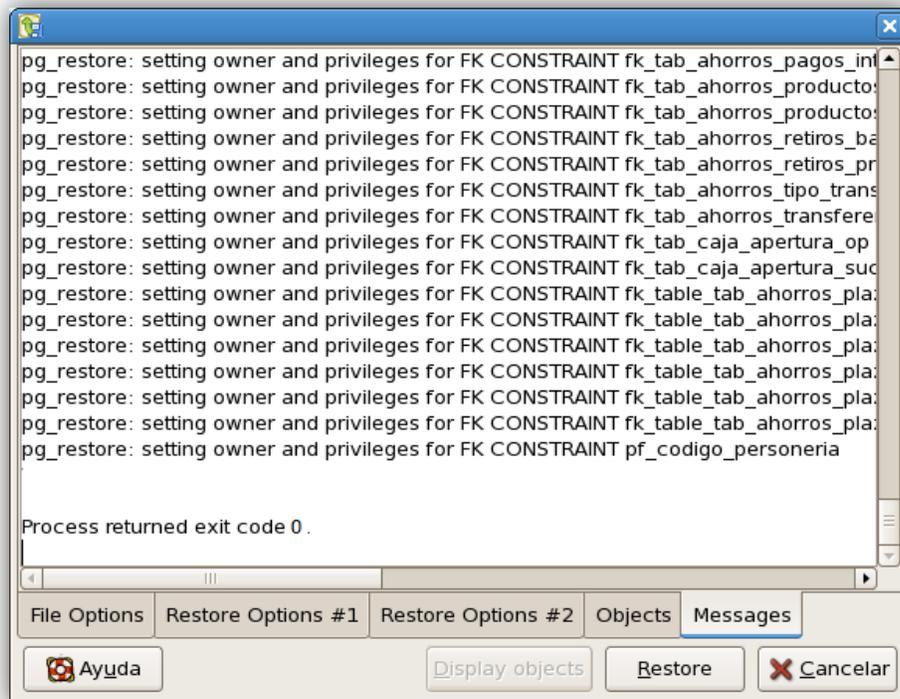


Figura 55. Subir backup a la base de datos.

Fuente: propia.

4.4.1.5. Instalación y configuración de Apache Tomcat.

En el escritorio, donde se copiaron todos los archivos, existe un archivo comprimido llamado *apache-tomcat-7.0.22-x86.zip*, se accede nuevamente a la consola que estaba abierta y ubicada en el directorio del escritorio, para descomprimir el archivo de digita el siguiente comando

```
# unzip apache-tomcat-7.0.22-windows-x86.zip
```

Al concluir la ejecución del comando anterior, se crea en el escritorio una carpeta llamada *apache-tomcat-7.0.22*,

```
# mv apache-tomcat-7.0.22 /opt/
```

```
# cd /opt/ apache-tomcat-7.0.22/bin
```

```
# chmod 777 *
```

Hasta ahí ya está el apache tomcat listo; para iniciar el servicio hay que ejecutar el fichero *startup.sh*.

Para hacer que el servicio se inicie automáticamente al inicial el sistema operativo se debe seguir los siguientes pasos:

```
# cd /etc/init.d/
```

```
# vi apache
```

El script apache debe quedar de la siguiente forma

```
#!/bin/sh
# Script ejemplo para arranque de servicios en /etc/init.d/
#

case "$1" in
start)
    echo "Iniciando servicio..."
    /opt/apache-tomcat-7.0.22/bin/startup.sh
    ;;
stop)
    echo "Deteniendo servicio..."
    /opt/apache-tomcat-7.0.22/bin/shutdown.sh
    ;;
*)
    echo "Modo de empleo: /etc/init.d/mi_script {start|stop}"
    exit 1
    ;;
esac
exit 0
```

Figura 56. Script de inicio de Apache.

Fuente: propia.

Posteriormente, hemos de crear un enlace simbólico en el runlevel correspondiente para que se ejecute cada vez que arranquemos la máquina

```
#ln -s /etc/init.d/apache /etc/rc3.d/S98apache
```

Y listo.

Reiniciar el servidor para verificar que el servicio si se inicia automáticamente al inicial el sistema.

Una vez iniciado el sistema, para verificar que el servidor Apache Tomcat este funcionando correctamente, en el path en un web browser digitar localhost y debe cargar la siguiente página:

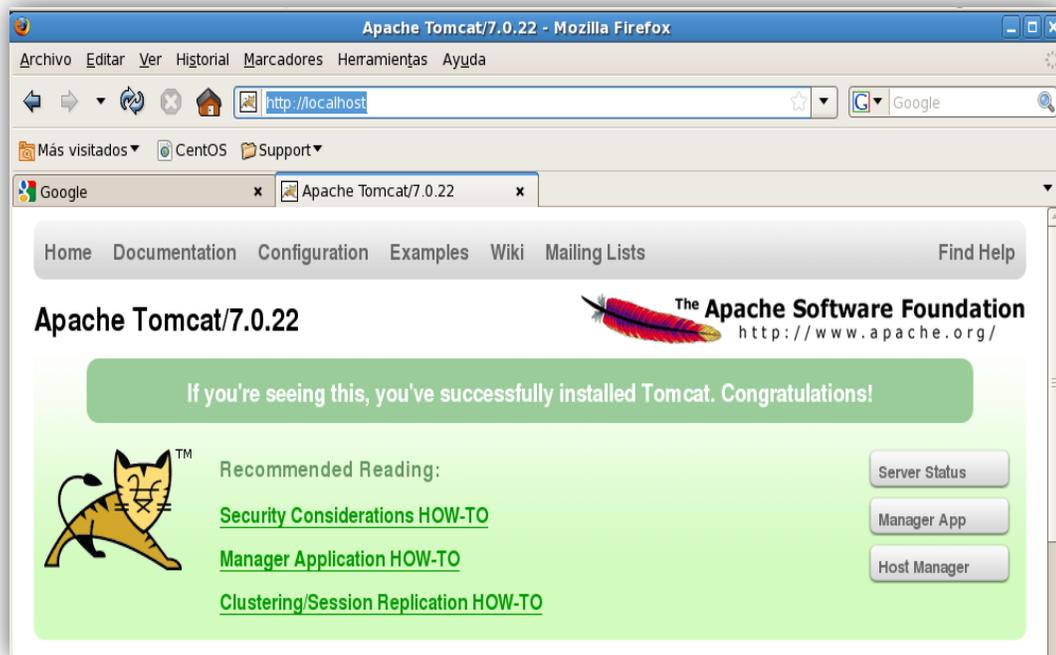


Figura 57. Página de administración de Apache Tomcat.

Fuente: propia.

Por último el archivo con extensión .war ubicado en el escritorio, pegarlo en

`/opt/apache-tomcat-7.0.22/webapp/` y listo.

Para verificar que la aplicación funcione correctamente en el navegador Firefox digitar

<http://localhost/FINANSYS>



Figura 58. Página de inicio del sistema.

Fuente: propia.

4.4.1.6. Configuración del SSL para HTTPS

Después de configurar correctamente el JAVA_HOME, se procede a generar una clave con el algoritmo de encriptación RSA, a dicha clave debemos asignarle un alias; Se debe configurar de la siguiente manera:

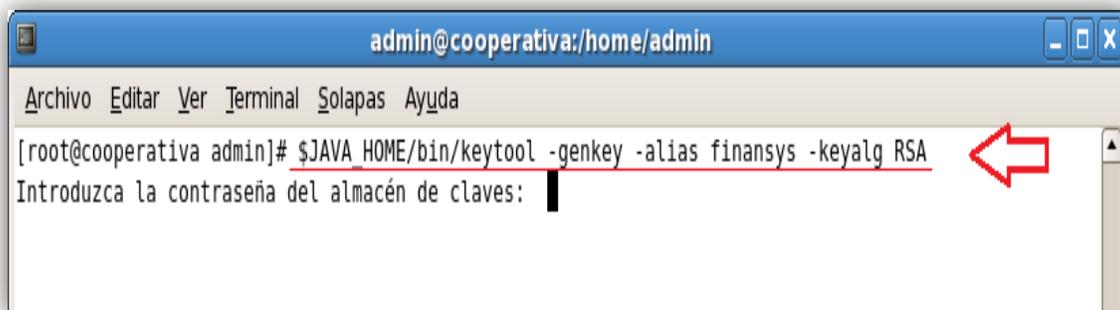


Figura 59. Generación de la clave.

Fuente: propia.

Luego se debe ingresar una contraseña, se debe completar unos datos que le va solicitando:

```
[root@cooperativa admin]# $JAVA_HOME/bin/keytool -genkey -alias finansys -keyalg RSA
Introduzca la contraseña del almacén de claves:
¿Cuáles son su nombre y su apellido?
 [Unknown]: Bairon Gudino
¿Cuál es el nombre de su unidad de organización?
 [Unknown]: FINANSYS
¿Cuál es el nombre de su organización?
 [Unknown]: FINANSYS
¿Cuál es el nombre de su ciudad o localidad?
 [Unknown]: IBARRA
¿Cuál es el nombre de su estado o provincia?
 [Unknown]: IMBABURA
¿Cuál es el código de país de dos letras de la unidad?
 [Unknown]: EC
¿Es correcto CN=Bairon Gudino, OU=FINANSYS, O=FINANSYS, L=IBARRA, ST=IMBABURA, C=EC?
 [no]: si

Introduzca la contraseña de clave para <finansys>
      (INTRO si es la misma contraseña que la del almacén de claves): _____
Volver a escribir la contraseña nueva: _____
[root@cooperativa admin]# █
```

Figura 60. Parámetros de la clave.

Fuente: propia.

Para terminar la configuración del SSL, se procede a configurar el servidor apache tomcat.

El Archivo server.xml ubicado en la carpeta `/opt/apache-tomcat-7.0.22/conf`, debe quedar de la siguiente forma:

```
-->
<!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the JSSE configuration, when using APR, the
      connector should be using the OpenSSL style configuration
      described in the APR documentation -->

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
      maxThreads="150" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS"
      keystoreFile="${user.home}/.keystore"
      keystorePass="changeit"/>

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Descomentar y configurar ésta sección de la siguiente forma

Figura 61. Configuración del server.xml

Fuente: propia.

Para verificar la configuración, reiniciar el servidor apache y en un navegador web acceder al siguiente link: <https://localhost:8443/FINANSYS/>

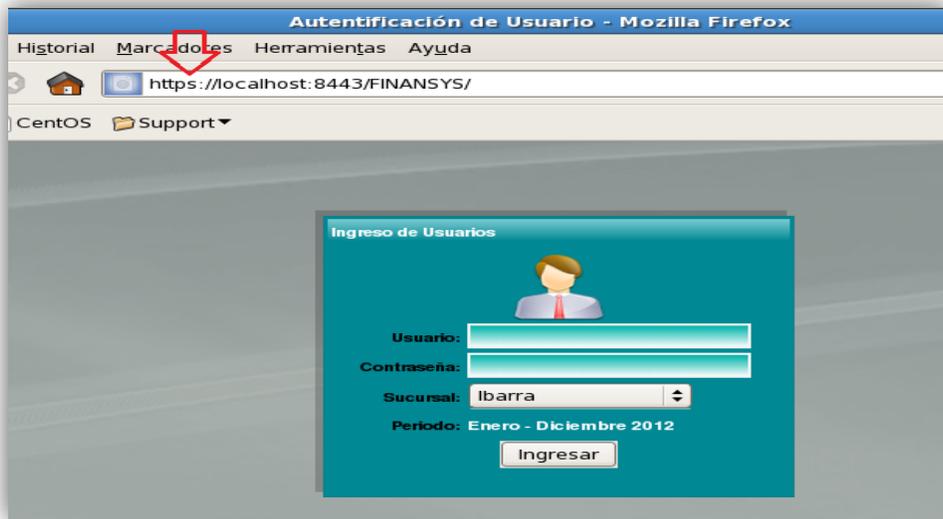


Figura 62. Página de inicio del sistema en modo HTTPS.

Fuente: propia.

Capítulo 5: Conclusiones y Recomendaciones



5. Conclusiones y recomendaciones

5.1. Conclusiones

- a) La metodología de desarrollo RUP, permite identificar fácilmente los requerimientos, procesos, secuencias y demás componentes necesarios en todas las fases; También hace que todo el proceso de desarrollo se lo vaya realizando de forma correcta y ordenada.
- b) El aspecto de las interfaces de usuario al desarrollar las páginas con algún Framework (RichFaces), mejora significativamente su presentación y se logra mantener un estándar en las diferentes formas
- c) El modelo de desarrollo en capas de MVC y JEE, hace mucho más fácil la comunicación interna de los módulos del sistema.
- d) Las herramientas de código abierto, cuentan con un conjunto de características que proporcionan gran estabilidad y seguridad para desarrollar sistemas empresariales.
- e) El desarrollo de aplicaciones en Java, proporciona al desarrollador un control completo de los componentes y de todo el código fuente.
- f) Todas las herramientas que hemos empleado, cuentan con una excelente documentación y foros de apoyo, logrando solucionar las inquietudes y problemas de forma rápida y en muchos sitios de internet
- g) Mediante el uso de Hibernate para el mapeo de las tablas, se me hizo más fácil al manejar la información de las tablas a través de objetos.
- h) El desarrollo del sistema de información financiera, ayudó de mucho a los usuarios de la Cooperativa al manejar toda la información de una forma centralizada, más ordenada y segura.

5.2. Recomendaciones

- a) En todo proyecto que se realice se debe emplear siempre una metodología de desarrollo para llevar todo el proyecto ordenadamente.
- b) Se debe definir una plantilla con un formato estándar para ahorrar tiempo al momento de crear nuevas interfaces de usuario.
- c) En todo proyecto que se desarrolle se debe emplear programación de n-capas para separar adecuadamente las funcionalidades del proyecto.
- d) Existe una gran cantidad de herramientas de código abierto que cubren satisfactoriamente las necesidades y requerimientos de nuestros proyectos.
- e) A la hora de buscar nuevas herramientas, primero se debe estudiar la documentación con la que cuenta para tener soluciones rápidas y precisas a nuestros problemas.
- f) Al desarrollar proyectos con base de datos siempre se debe utilizar alguna herramienta de persistencia hacia la base de datos.
- g) La correcta capacitación del uso del sistema ayuda a reducir el tiempo de manejo por parte de los usuarios logrando una mejor atención a los clientes.

6. Glosario

6.1. Introducción

El presente glosario forma parte del sistema “FINANSYS” y es una recopilación de términos empleados en todo el sistema.

Este glosario no es un diccionario informático; representa una guía general de las definiciones y conceptos que se usan con frecuencia y tienen una relevante importancia para la adecuada comprensión dentro del sistema.

6.2. Propósito

La función principal del presente glosario es proporcionar un estándar de términos y conceptos asociados a los términos técnicos del sistema.

6.3. Referencias

Éste documento hace referencia a términos empleados en toda la documentación del sistema.

6.4. Glosario:

A

Actions. -Métodos de capa de vista, para el uso de componentes en páginas jsp.

API.- Interfaz de Programación de Aplicaciones (**A**pplication **P**rogramming **I**nterface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Artefacto.- Un artefacto es un producto tangible resultante del proceso de desarrollo de software. En ocasiones un artefacto puede referirse a un producto terminado, pero más habitualmente se refiere a la documentación generada a lo largo del desarrollo del producto.

B

Buteo (Boot).- Es la secuencia de arranque, es el proceso que inicia el sistema operativo cuando el usuario enciende una computadora. Se encarga de la inicialización del sistema y de los dispositivos.

C

Casos de uso.- Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

CRUD.- (Create, Read, Update and Delete). Es usado para referirse a las funciones básicas realizadas en las bases de datos o la capa de persistencia en un sistema de software.

CSS.- El nombre hojas de estilo en cascada viene del inglés **CascadingStyle Sheets**.

CUC.- Cooperativa Unión Cochapamba.

F

FINANSYS.- Nombre del sistema financiero desarrollado.

FODA.- Fortalezas, Oportunidades, Debilidades, Amenazas.

Framework.- es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

G

GNU/Linux.- GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU.

GPL.- (General Public License) es una licencia creada por la Free Software Foundation orientada principalmente a proteger la libre distribución, modificación y uso de software.

H

HTML.- (Siglas de **Hyper Text Markup Language**) Lenguaje de marcado de hipertexto.

HQL.- Hibernate Query Language

J

Jdeveloper.- Es un entorno de desarrollo integrado, desarrollado por Oracle.

JSF.- es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

L

Live CD.- es un sistema operativo almacenado en un medio extraíble, que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora.

O

On-the-fly.-Sobre marcha, en proceso

Open source.- Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente.

ORM.- Es el Mapeo Objeto-Relacional (**Object-Relational Mapping**) es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

P

POO.-Programación Orientada a Objetos.

R

RichFaces.- Framework de código abierto que añade la capacidad de Ajax en aplicaciones JSF sin recurrir a JavaScript.

Rup.- El Proceso Unificado de Rational (**Rational Unified Process**) es un proceso de desarrollo de software y junto con el UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

S

SELinux.- es una característica de seguridad de Linux que provee una variedad de políticas de seguridad.

Servlets.- Los servlets, son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ejemplo: Tomcat) y extienden su funcionalidad.

Skin.- Mascara, interfaz visual de distintas características.

Skinnability.- Cambiar de skin.

SMB.-Server Message Block es un Protocolo de red que permite compartir archivos e impresoras entre nodos de una red.

T

Tag.- Etiqueta. Las etiquetas le dicen al browser las instrucciones y características necesarias para presentar la página en la pantalla.

U

UML.- Lenguaje Unificado de Modelado (*Unified Modeling Language*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

V

Visio.- Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más.

Vista de deployment.- es un tipo de diagrama UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Y

YUM.- Es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software.

7. Anexos

7.1. Copia del anteproyecto

A continuación se anexa una copia del anteproyecto de tesis aprobado por el Consejo Directivo de la Facultad.

7.2. Diagrama entidad relación

A continuación se presenta el diagrama Entidad-Relación de la BDD de FINANSYS, para el cual se ha utilizado el modelador de datos PowerDesigner.

7.3. Diagrama general de casos de uso.

A continuación se presenta el Diagrama General de Casos de Uso de FINANSYS JDeveloper.

Los siguientes archivos se encuentran en el documento anexos, ubicado en el CD del proyecto.

- A. Manual de usuario
- B. Manual técnico
- C. Entregables de RUP
 - C.1 Actas de trabajo
 - C.2 Casos de uso

8. Referencias

BIBLIOGRAFÍA

- Aguilar, D. A. (2006, Mayo). <http://zarza.fis.usal.es>. Retrieved from <http://zarza.fis.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/SeguridadAppWeb.pdf>
- Barnes, R. (n.d.). <http://www.odfauthors.org/openoffice.org/es>. Retrieved from http://www.odfauthors.org/openoffice.org/es/guia-de-comienzo-ooo-3.2/publicado/0116GS3-CodigoAbiertoEstandaresDoc.pdf/at_download/file
- Cid, J. (n.d.). <http://www.jtech.ua.es/jornadas/07/charlas/JaimeCid1pp.pdf>. Retrieved from <http://www.jtech.ua.es/jornadas/07/charlas/JaimeCid1pp.pdf>
- Cuenca, E. R. (2012, Junio). <http://mistock.lcompras.biz/sysoper/1362-sysop2011-servidor-de-aplicaciones-apache-tomcat>. Retrieved from <http://mistock.lcompras.biz/sysoper/1362-sysop2011-servidor-de-aplicaciones-apache-tomcat>
- García, D. J. (2008, Abril 2). <http://abogadosmanabi.blogspot.com>. Retrieved from <http://abogadosmanabi.blogspot.com>: <http://abogadosmanabi.blogspot.com/2008/04/la-ley-de-Cooperativas-y-su-reglamento.html>
- <http://certamenlitoral.proydesa.org>. (n.d.). Retrieved from <http://certamenlitoral.proydesa.org/it1/cont6.asp>
- <http://docs.fedoraproject.org>. (n.d.). Retrieved from http://docs.fedoraproject.org/es-ES/Fedora/16/html/Security_Guide/index.html
- <http://es.wikibooks.org/wiki>. (n.d.). Retrieved from http://es.wikibooks.org/wiki/Aplicaciones_Distribuidas:_Un_enfoque_pr%C3%A1ctico
- <http://es.wikipedia.org>. (n.d.). Retrieved from http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada
- <http://es.wikipedia.org>. (n.d.). Retrieved from [http://es.wikipedia.org/wiki/Fedora_\(distribución_Linux\)](http://es.wikipedia.org/wiki/Fedora_(distribuci%C3%B3n_Linux))
- http://es.wikipedia.org/wiki/Anexo:Implementaciones_de_sistemas_operativos. (n.d.). Retrieved from http://es.wikipedia.org/wiki/Anexo:Implementaciones_de_sistemas_operativos
- <http://es.wikipedia.org/wiki/Applet>. (n.d.). Retrieved from <http://es.wikipedia.org/wiki/Applet>

<http://es.wikipedia.org/wiki/JasperReports>. (n.d.). Retrieved from <http://es.wikipedia.org/wiki/JasperReports>

http://es.wikipedia.org/wiki/Java_EE. (n.d.). Retrieved from http://es.wikipedia.org/wiki/Java_EE

<http://es.wikipedia.org/wiki/NetBeans>. (n.d.). Retrieved from <http://es.wikipedia.org/wiki/NetBeans>

<http://fedoraproject.org/es>. (n.d.). Retrieved from <http://fedoraproject.org/es/about-fedora>

<http://html.rincondelvago.com>. (n.d.). Retrieved from <http://html.rincondelvago.com/concepto-de-base-de-datos.html>

<http://jansoblog.wordpress.com/2007/10/23/ciclo-de-vida-en-jsf/>. (n.d.). Retrieved from <http://jansoblog.wordpress.com/2007/10/23/ciclo-de-vida-en-jsf/>

<http://lists.centos.org>. (n.d.). Retrieved from <http://lists.centos.org/pipermail/centos-es/2007-March/000317.html>

<http://ocw.uc3m.es/ingenieria-telematica/>. (n.d.). Retrieved from http://ocw.uc3m.es/ingenieria-telematica/software-de-comunicaciones/transparencias/3_cmpnts-JavaEE.pdf

http://ocw.uc3m.es/ingenieria-telematica/software-de-comunicaciones/transparencias/3_cmpnts-JavaEE.pdf. (n.d.). Retrieved from http://ocw.uc3m.es/ingenieria-telematica/software-de-comunicaciones/transparencias/3_cmpnts-JavaEE.pdf

http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html. (n.d.). Retrieved from http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html

<http://tomcat.apache.org/>. (n.d.). Retrieved from <http://tomcat.apache.org/>

<http://upload.wikimedia.org/wikipedia/commons/thumb/8/86/CSS.svg/160px-CSS.svg.png>. (n.d.). Retrieved from <http://upload.wikimedia.org/wikipedia/commons/thumb/8/86/CSS.svg/160px-CSS.svg.png>

<http://www.asptutor.com>. (n.d.). Retrieved from <http://www.asptutor.com>

<http://www.bi-spain.com/articulo/71281>. (n.d.). Retrieved from <http://www.bi-spain.com/articulo/71281/open-source-software-libre/todos/integracion-e-implementacion-de-jasperreports-server-con-aplicaciones-java-webinar-de-1-hora>

<http://www.centos.org/>. (n.d.). Retrieved from <http://www.centos.org/>

<http://www.dspace.espol.edu.ec/>. (n.d.).

<http://www.java.com>. (n.d.). Retrieved from
http://www.java.com/es/download/faq/whatis_java.xml

<http://www.librosweb.es/>. (n.d.). Retrieved from
http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html

<http://www.linux-magazine.es>. (n.d.). Retrieved from <http://www.linux-magazine.es/issue/16/DVD.pdf>

<http://www.manualesdeayuda.com>. (n.d.). Retrieved from
<http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>

<http://www.postgresql.org.es/>. (n.d.). Retrieved from
http://www.postgresql.org.es/sobre_postgresql

http://www.postgresql.org.es/sobre_postgresql. (n.d.). Retrieved from
http://www.postgresql.org.es/sobre_postgresql

http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf. (n.d.). Retrieved from
http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf

<http://www.proactiva-calidad.com/>. (n.d.). Retrieved from <http://www.proactiva-calidad.com/java/jsf/introduccion.html>

<http://www.proactiva-calidad.com/java/>. (n.d.). Retrieved from <http://www.proactiva-calidad.com/java/arquitectura/index.html>

<http://www.sbs.gob.ec>. (n.d.). Retrieved from
http://www.sbs.gob.ec/medios/PORTALDOCS/downloads/normativa/decreto_194_Cooperativas_29_dic_09.pdf

<http://www.sbs.gob.ec/medios>. (n.d.). Retrieved from
<http://www.sbs.gob.ec/medios/PORTALDOCS/downloads/normativa/Reglamento%20Ley%20de%20Inst.%20Financieras.pdf>

<http://www.sicuma.uma.es>. (n.d.). Retrieved from
<http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>

<http://www.taringa.net>. (n.d.). Retrieved from
http://www.taringa.net/posts/linux/6239365/comparativa-gnu_linux-vs-windows___muy-bueno.html

<http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>. (n.d.). Retrieved from
<http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm>

Oliva, J. (2007). *<http://lists.centos.org/pipermail/centos-es/2007-March/000317.html>*.
Retrieved from <http://lists.centos.org/pipermail/centos-es/2007-March/000317.html>

Pérez, A. S. (n.d.). *<http://www.dspace.espol.edu.ec/bitstream>*. Retrieved from
<http://www.dspace.espol.edu.ec/bitstream/123456789/1921/1/3824.pdf>

Ruiz, Candido. (s.f.). *<http://www.slideshare.net/CANDIDOALBERTO>*. Obtenido de
<http://www.slideshare.net/CANDIDOALBERTO/1-caractersticas-del-sistema-operativo-1714927>

uzcátegui, A. (s.f.). *<http://www.monografias.com>*. Obtenido de
<http://www.monografias.com/trabajos5/sisop/sisop.shtml>