

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UN SOFTWARE QUE PERMITA LA
AUTOMATIZACIÓN DEL REGISTRO Y PROCESAMIENTO DE LAS
OPERACIONES CONTABLES HISTORICAS QUE SE GENERAN EN UNA
EMPRESA O ACTIVIDAD PRODUCTIVA.**

REALIZADO POR:

Br. Jesús Miguel Malavé Caraballo

C.I: V-16.841.728

Trabajo de Grado presentado ante la Universidad de Oriente como requisito parcial
para optar al título de Ingeniero en Computación

Barcelona, Octubre de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UN SOFTWARE QUE PERMITA LA
AUTOMATIZACIÓN DEL REGISTRO Y PROCESAMIENTO DE LAS
OPERACIONES CONTABLES HISTORICAS QUE SE GENERAN EN UNA
EMPRESA O ACTIVIDAD PRODUCTIVA.**

ASESORADO POR:

Prof. Pedro Dorta

Barcelona, Octubre de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UN SOFTWARE QUE PERMITA LA
AUTOMATIZACIÓN DEL REGISTRO Y PROCESAMIENTO DE LAS
OPERACIONES CONTABLES HISTORICAS QUE SE GENERAN EN UNA
EMPRESA O ACTIVIDAD PRODUCTIVA.**

JURADO CALIFICADOR:

Prof. Mónica Saettone
Jurado Principal

Prof. Manuel Carrasquero
Jurado Principal

Prof. Pedro Dorta
Asesor Académico

Barcelona, Octubre de 2009



REGLAMENTO

De acuerdo con el artículo N° 44 del reglamento sobre trabajo de grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario”.



RESUMEN

El presente proyecto consistió en el desarrollo de un software que permitirá la automatización del registro y procesamiento de las operaciones contables históricas que se generan en una empresa o actividad productiva. La sistematización y automatización de la contabilidad tiene como finalidad liberar al contador de esta fase del proceso, permitiéndole dedicar más tiempo a labores de mayor importancia, como el análisis e interpretación de la información. El software se elaboró utilizando tecnologías de software libre, rigiéndose por el decreto 3390, el cuál dispone que “La Administración Pública Nacional deberá emplear prioritariamente Software Libre desarrollado con Estándares Abiertos, en todos sus Sistemas, Proyectos y Servicios Informáticos”. El desarrollo del proyecto estuvo dirigido por el Proceso Unificado de Desarrollo de Software que permite dividir el proyecto en fases y éstas en flujos de trabajos, y el lenguaje de modelado UML. Para construir el software se utilizó Java como lenguaje de programación y como manejador de bases de datos MySQL.



DEDICATORIA Y AGRADECIMIENTOS

Dedicatoria

A Dios todopoderoso por estar a mi lado ayudándome en todo momento, a mis Padres, Jesús Malavé y Nairobi Caraballo, por darme siempre su apoyo para alcanzar mis metas e inculcarme que el estudio es una de las cosas más importantes en la vida. A mis hermanas Naigré y Desirée y al pequeño de la casa Miguel, de quienes me he perdido tanto debido a la distancia que nos separa, pero que siempre los he tenido presentes en mi corazón. Los quiero mucho a todos.

Agradecimientos

Dios, te doy gracias por darme la dicha de culminar mi carrera universitaria.

A mis padres por su apoyo incondicional a lo largo de mi vida, pero en especial en esta etapa de formación universitaria.

A mi tía Petra Malavé, la segunda madre que me dio la vida y para quien no existen palabras que me permitan demostrarle lo mucho que le debo y le estoy agradecido.

Al profesor Pedro Dorta por haberme asesorado y ayudado en este proyecto.

A mis amigos y a todas aquellas personas que han estado conmigo en el desarrollo de este proyecto.



Índice	Pag.
REGLAMENTO	IV
RESUMEN.....	V
DEDICATORIA Y AGRADECIMIENTOS	VI
CAPITULO I: PLANTEAMIENTO DEL PROBLEMA	17
1.1 INTRODUCCIÓN.....	17
1.2 EL PROBLEMA	18
1.3 PROPÓSITO Y METODOLOGÍA.....	18
1.4 ALCANCE E IMPORTANCIA	19
1.5 ORIGINALIDAD.....	20
1.6 OBJETIVOS.....	21
1.6.1 Objetivo General.....	21
1.6.2 Objetivos Específicos	21
CAPÍTULO II: MARCO TEORICO	22
2.1 ANTECEDENTES	22
2.2 PROGRAMACIÓN ORIENTADA A OBJETOS	24
2.3 Características de los Lenguajes de Programación Orientados a Objetos....	25
2.3.1 Definición de Clases y Objetos	27
2.3.2 Ventajas de la Programación Orientada a Objetos en el ciclo de vida de desarrollo.....	27
2.4 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE	28
2.5 LENGUAJE DE MODELADO UNIFICADO (UML).....	28
2.6 BASES DE DATOS.....	29
2.6.1 Tipos de Bases de Datos	30
2.6.2 Sistema de Gestión de Base de Datos (SGBD)	32
2.6.3 Descripción de la arquitectura Cliente-Servidor.....	33
2.7 CONTABILIDAD.....	33
2.7.1 Partida Doble	33
2.7.2 Asientos Contables	34



2.7.3	Ecuación Patrimonial.....	34
2.7.4	Concepto de Cuenta.....	35
2.7.5	Balance General.....	35
2.7.6	Depreciación.....	36
2.7.7	Libros de Contabilidad.....	36
2.7.8	Empresa.....	36
2.8	SOFTWARE LIBRE.....	37
2.8.1	Historia del Software Libre.....	37
2.8.2	Libertades del Software Libre.....	38
2.8.3	Tipos de Licencias mas usadas en Software Libre.....	40
CAPITULO III. FASE DE INICIO.....		41
3.1	INTRODUCCIÓN.....	41
3.2	PLANIFICACIÓN DE LA FASE DE INICIO.....	42
3.3	ESTUDIO DEL CONTEXTO DEL SISTEMA.....	43
3.3.1	Modelo de Dominio del Sistema Propuesto.....	43
3.4	RIESGOS DEL SISTEMA.....	44
3.4.1	Riesgos Críticos del Sistema.....	45
3.5	FLUJO DE TRABAJO DE REQUISITOS.....	47
3.5.1	Requisitos Funcionales.....	48
3.5.2	Requisitos No Funcionales.....	49
3.5.3	Modelado de Casos de Uso.....	49
3.6	FLUJO DE TRABAJO DE ANÁLISIS.....	58
3.6.1	Identificación de las Clases de Análisis.....	59
3.6.2	Diagramas de Clases de Análisis.....	64
3.6.3	Paquetes del Análisis.....	77
3.7	EVALUACION DE LA FASE DE INICIO.....	79
CAPÍTULO IV. FASE DE ELABORACIÓN.....		80
4.1	INTRODUCCIÓN.....	80
4.2	PLANIFICACIÓN DE LA FASE DE ELABORACIÓN.....	80



4.3	FLUJO DE TRABAJO DE REQUISITOS	81
4.3.1	Modelo de Caso de Uso	81
4.4	FLUJO DE TRABAJO DE ANÁLISIS	82
4.4.1	Identificación de Clases de Análisis	82
4.4.2	Diagramas de Colaboración.....	82
4.5	FLUJO DE TRABAJO DE DISEÑO.....	98
4.5.1	Clases de Diseño del software J-Contab.....	98
4.5.2	Diagramas de secuencia.....	101
4.5.3	Identificación de paquetes de diseño	124
4.5.4	Diseño de la Arquitectura	125
4.5.5	Diseño de la Base de Datos del Sistema.....	126
4.6	EVALUACIÓN DE LA FASE DE ELABORACIÓN	129
CAPÍTULO V. FASE DE CONSTRUCCIÓN.....		131
5.1	INTRODUCCIÓN.....	131
5.2	PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN	131
5.3	FLUJO DE TRABAJO DE IMPLEMENTACIÓN	131
5.3.1	Implementación de los Subsistemas.....	132
5.3.2	Integración del sistema	137
5.4	PRUEBAS.....	140
5.4.1	Partición Equivalente.....	140
5.4.2	Aplicación de Casos de Prueba.....	143
5.4.3	Identificación de los Casos de Prueba	146
5.5	RESUMEN DE LA FASE DE CONSTRUCCIÓN.....	148
CONCLUSIONES		149
RECOMENDACIONES		150
BIBLIOGRAFÍA		151
ANEXO A: MANUAL DE USUARIO		153
A.1	DESCRIPCION DE J-CONTAB	153
A.2	PANTALLA PRINCIPAL	153



A.3	MENÚ	155
A.3.1	Menú Archivo	155
A.3.2	Menú Transacciones	159
A.3.3	Menú Reportes	164
A.3.4	Menú Configuración	167
A.3.5	Menú Ayuda	176
A.4	REQUISITOS OPERACIONALES DE HARDWARE Y SOFTWARE .	178



Índice de Figuras	Pag.
Figura 3.1 Modelo de Dominio del sistema	44
Figura 3.2 Caso de Uso General del sistema.	52
Figura 3.3 Caso de Uso Administrar Usuarios	53
Figura 3.4 Caso de Uso Manejar Respaldos	53
Figura 3.5 Caso de Uso Administrar Empresas	54
Figura 3.6 Caso de Uso Administrar Cuentas.....	55
Figura 3.7 Caso de Uso Manejar Asientos.....	56
Figura 3.8 Caso de Uso Emitir Reportes.....	57
Figura 3.9 Diagrama de Análisis para el Caso de Uso Crear Usuario	64
Figura 3.10 Diagrama de Análisis para el Caso de Uso Eliminar Usuario	65
Figura 3.11 Diagrama de Análisis para el Caso de Uso Exportar Archivo de Respaldo	66
Figura 3.12 Diagrama de Análisis para el Caso de Uso Importar Archivo de Respaldo	66
Figura 3.13 Diagrama de Análisis para el Caso de Uso Crear Empresa.....	67
Figura 3.14 Diagrama de Análisis para el Caso de Uso Modificar Empresa.....	68
Figura 3.15 Diagrama de Análisis para el Caso de Uso Eliminar Empresa.....	68
Figura 3.16 Diagrama de Análisis para el Caso de Uso Crear Cuenta	69
Figura 3.17 Diagrama de Análisis para el Caso de Uso Modificar Cuenta	70
Figura 3.18 Diagrama de Análisis para el Caso de Uso Eliminar Cuenta	70
Figura 3.19 Diagrama de Análisis para el Caso de Uso Crear Asiento	71
Figura 3.20 Diagrama de Análisis para el Caso de Uso Modificar Asiento	72
Figura 3.21 Diagrama de Análisis para el Caso de Uso Eliminar Asiento	72
Figura 3.22 Diagrama de Análisis para el Caso de Uso Emitir Balance General.....	73
Figura 3.23 Diagrama de Análisis para el Caso de Uso Emitir Balance de Comprobación.....	74
Figura 3.24 Diagrama de Análisis para el Caso de Uso Emitir Estados de Resultados	75



Figura 3.25 Diagrama de Análisis para el Caso de Uso Emitir Libro Diario General	76
Figura 3.26 Diagrama de Análisis para el Caso de Uso Emitir Libro Mayor	77
Figura 3.27 Diagrama de Análisis para el Caso de Uso Emitir Mayor Analítico.....	77
Figura 3.28 Paquetes de Análisis	78
Figura 4.1 Diagrama de Colaboración para el Caso de Uso Crear Usuario.....	83
Figura 4.2 Diagrama de Colaboración para el Caso de Uso Eliminar Usuario.....	83
Figura 4.3 Diagrama de Colaboración para el Caso de Uso Exportar Archivo de Respaldo.....	84
Figura 4.4 Diagrama de Colaboración para el Caso de Uso Importar Archivo de Respaldo.....	84
Figura 4.5 Diagrama de Colaboración para el Caso de Uso Crear Empresa	85
Figura 4.6 Diagrama de Colaboración para el Caso de Uso Modificar Empresa	86
Figura 4.7 Diagrama de Colaboración para el Caso de Uso Eliminar Empresa	87
Figura 4.8 Diagrama de Colaboración para el Caso de Uso Crear Cuenta.....	87
Figura 4.9 Diagrama de Colaboración para el Caso de Uso Modificar Cuenta.....	88
Figura 4.10 Diagrama de Colaboración para el Caso de Uso Eliminar Cuenta.....	89
Figura 4.11 Diagrama de Colaboración para el Caso de Uso Crear Asiento.....	90
Figura 4.12 Diagrama de Colaboración para el Caso de Uso Modificar Asiento.....	90
Figura 4.13 Diagrama de Colaboración para el Caso de Uso Eliminar Asiento.....	91
Figura 4.14 Diagrama de Colaboración para el Caso de Uso Emitir Balance General	92
Figura 4.15 Diagrama de Colaboración para el Caso de Uso Emitir Balance de Comprobación.....	93
Figura 4.16 Diagrama de Colaboración para el Caso de Uso Emitir Estados de Resultados	94
Figura 4.17 Diagrama de Colaboración para el Caso de Uso Emitir Libro Diario General	95
Figura 4.18 Diagrama de Colaboración para el Caso de Uso Emitir Libro Mayor ...	96



Figura 4.19 Diagrama de Colaboración para el Caso de Uso Emitir Mayor Analítico	97
Figura 4.20: Diagrama de clases del software J-Contab.....	100
Figura 4.21 Diagrama de Secuencia para el Caso de Uso Crear Usuario	102
Figura 4.22 Diagrama de Secuencia para el Caso de Uso Eliminar Usuario	103
Figura 4.23 Diagrama de Secuencia para el Caso de Uso Exportar Archivo de Respaldo.....	104
Figura 4.24 Diagrama de Secuencia para el Caso de Uso Importar Archivo de Respaldo.....	105
Figura 4.25 Diagrama de Secuencia para el Caso de Uso Crear Empresa.....	106
Figura 4.26 Diagrama de Secuencia para el Caso de Uso Modificar Empresa.....	107
Figura 4.27 Diagrama de Secuencia para el Caso de Uso Eliminar Empresa.....	108
Figura 4.28 Diagrama de Secuencia para el Caso de Uso Crear Cuenta	109
Figura 4.29 Diagrama de Secuencia para el Caso de Uso Modificar Cuenta	111
Figura 4.30 Diagrama de Secuencia para el Caso de Uso Eliminar Cuenta	112
Figura 4.31 Diagrama de Secuencia para el Caso de Uso Crear Asiento	113
Figura 4.32 Diagrama de Secuencia para el Caso de Uso Modificar Asiento	114
Figura 4.33 Diagrama de Secuencia para el Caso de Uso Eliminar Asiento	115
Figura 4.34 Diagrama de Secuencia para el Caso de Uso Emitir Balance General.	117
Figura 4.35 Diagrama de Secuencia para el Caso de Uso Emitir Balance de Comprobación.....	118
Figura 4.36 Diagrama de Secuencia para el Caso de Uso Emitir Estado de Resultados	119
Figura 4.37 Diagrama de Secuencia para el Caso de Uso Emitir Libro Diario General	121
Figura 4.38 Diagrama de Secuencia para el Caso de Uso Emitir Libro Mayor.....	122
Figura 4.39 Diagrama de Secuencia para el Caso de Uso Emitir Mayor Analítico.	123
Figura 4.40 Paquete de Diseño Usuarios	124
Figura 4.41 Vista Lógica de las Capas de la Arquitectura del Sistema	126



Figura 4.42 Modelo Físico de la Base de Datos Local	129
Figura 5.1 Subsistemas de Implementación.....	132
Figura 5.2 Subsistema Administrar Usuarios para la implementación.....	133
Figura 5.3 Subsistema Manejar Respaldos para la implementación.....	134
Figura 5.4 Subsistema Administrar Empresas para la implementación.....	134
Figura 5.5 Subsistema Administrar Cuentas para la implementación.....	135
Figura 5.6 Subsistema Manejar Asientos para la implementación.....	136
Figura 5.7 Subsistema Emitir Reportes para la implementación.....	137
Figura 5.8 Componentes implementados en la primera construcción.....	138
Figura 5.9 Componentes implementados en la segunda construcción.....	139
Figura 5.10 Componentes implementados en la tercera construcción.....	140
Figura A.1: Validar Usuario.....	154
Figura A.2: Pantalla Principal.....	154
Figura A.3: Menú Archivo.....	156
Figura A.4: Seleccionar Empresa.....	157
Figura A.5: Guardar	158
Figura A.6: Abrir.....	159
Figura A.7: Menú Transacciones	160
Figura A.8: Procesar Comprobantes	161
Figura A.9: Nuevo Asiento	162
Figura A.10: Modificar Asiento.....	163
Figura A.11: Insertar Asiento.....	164
Figura A.12: Menú Reportes.....	165
Figura A.13: Menú Configuración.....	167
Figura A.14: Submenú Usuarios.....	168
Figura A.15: Cambiar Contraseña.....	169
Figura A.16: Administrar Usuarios.....	170
Figura A.17: Administrar Empresas	171
Figura A.18: Nueva Empresa.....	172



Índice de Figuras

Figura A.19: Modificar Empresa	173
Figura A.20: Administrar Cuentas	174
Figura A.21: Nueva Cuenta.....	175
Figura A.22: Modificar Cuenta.....	175
Figura A.23: Menú Ayuda	177



Índice de Tablas	Pag.
Tabla 3.1 Actores identificados en el sistema.....	50
Tabla 3.2 Casos de Uso del sistema.....	51
Tabla 3.3 Descripción de las clases de control	60
Tabla 3.4 Descripción de las clases de entidad.....	61
Tabla 3.5 Descripción de las clases de interfaz. 1/2	62
Tabla 3.5 Descripción de las clases de interfaz. 2/2	63
Tabla 4.1 Clases de Diseño. 1/2.....	99
Tabla 4.1 Clases de Diseño. 2/2.....	100
Tabla 5.1 Aplicación de los casos de prueba (1/6).....	143
Tabla 5.2 Aplicación de los casos de prueba (2/6).....	143
Tabla 5.3 Aplicación de los casos de prueba (3/6).....	144
Tabla 5.4 Aplicación de los casos de prueba (4/6).....	144
Tabla 5.5 Aplicación de los casos de prueba (5/6).....	145
Tabla 5.6 Aplicación de los casos de prueba (6/6).....	145
Tabla 5.7 Caso de Prueba para el caso de uso Administrar Usuario.....	146
Tabla 5.8 Caso de Prueba para el caso de uso Administrar Empresas.....	147
Tabla 5.9 Caso de Prueba para el caso de uso Administrar Cuenta.....	147
Tabla 5.10 Caso de Prueba para el caso de uso Crear Asientos.....	148



CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 INTRODUCCIÓN

La contabilidad puede definirse como una ciencia instrumental que se ocupa de sistematizar y generalizar los métodos contables, al tiempo que investiga la consecución de otros nuevos. Es una ciencia instrumental porque el objeto de su estudio consiste en instrumentos o métodos. La ciencia de la contabilidad se dirige a la construcción de un cuerpo metodológico, que proporcione una herramienta de análisis a otras ciencias, particularmente la economía de la empresa.

Tradicionalmente la contabilidad se ha definido como el "lenguaje de los negocios" por el hecho de ser, dentro de la organización, una actividad de servicio encargada de identificar, medir y comunicar la información económica que permite a los diferentes usuarios formular juicios y tomar decisiones.

El objetivo primordial de la contabilidad es el de proporcionar información financiera de la organización a personas naturales y entidades jurídicas interesadas en sus resultados operacionales y en su situación económica. Los administradores de la empresa, los accionistas, el gobierno, las entidades crediticias, los proveedores y los empleados son personas e instituciones que constantemente requieren información financiera de la organización para sus respectivos análisis.



1.2 EL PROBLEMA

La complejidad y gran número de transacciones comerciales se han convertido en un problema desde el punto de vista contable-administrativo, ya que los tiempos de procesamiento y disponibilidad de esta información se elevan considerablemente hecho que afecta directamente la toma de decisiones dentro de una organización. Esto genera la necesidad de automatizar las actividades de registro y procesamiento de todas las operaciones comerciales de una organización, entre otros. Destinado a satisfacer estos requerimientos surge como proyecto el desarrollo de un “Software Contable”, que permita a la empresa llevar un registro histórico de sus operaciones comerciales de manera de poder disponer de dicha información de manera rápida y segura.

Se llama “Software Contable” a los programas de contabilidad o paquetes contables, destinados a sistematizar y simplificar las tareas de contabilidad. El Software contable registra y procesa las transacciones históricas que se generan en una empresa o actividad productiva: las funciones de compras, ventas, cuentas por cobrar, cuentas por pagar, balances, nóminas, etc. Para ello solo hay que ingresar la información requerida, como las pólizas contables, ingresos y egresos, y hacer que el programa realice los cálculos necesarios.

1.3 PROPÓSITO Y METODOLOGÍA

Este proyecto tiene como finalidad desarrollar un “Software Contable”, para sistematizar y simplificar las tareas de contabilidad, mejorando la capacidad de procesamiento, disponibilidad y control de los datos.



El proyecto se llevó a cabo siguiendo las normas del Proceso Unificado de Desarrollo de Software R.U.P. (Proceso Unificado Racional), haciendo uso de la notación del Lenguaje Unificado de Modelado (UML), y cumpliendo con el Decreto Presidencial No. 3390, publicado en gaceta No. 38095, de fecha 28 de diciembre de 2004, mediante el cual se dispone que “la Administración Pública Nacional deberá emplear prioritariamente Software Libre desarrollado con Estándares Abiertos, en todos sus Sistemas, Proyectos y Servicios Informáticos”. La aplicación se desarrollará utilizando el lenguaje de programación JAVA y el manejador de base de datos MySQL.

1.4 ALCANCE E IMPORTANCIA

El alcance de este proyecto contempló la automatización de las siguientes operaciones contables:

- Teneduría de libros: es la parte de la contabilidad que se encarga de recopilar, anotar, verificar y clasificar las operaciones mercantiles por orden cronológico en los libros dispuesto al efecto, según leyes normas y usos establecidos.
- Manejo y configuración de cuentas: las cuentas son un medio contable o instrumento de operación, mediante el cual podemos sub-dividir el activo, pasivo y el capital, y agruparlos de acuerdo a ciertas características de afinidad.
- Balance general: es un estado financiero o de situación que agrupa los bienes, derechos y obligaciones, mostrando con su información la información económico contable de una empresa en un momento dado.



Capítulo I - Planteamiento del Problema

- Balance de comprobación.
- Balance de suma.
- Estado de resultado: demostración en forma resumida de las variaciones que ha sufrido el capital en un determinado ejercicio.
- Cierre de cuentas.

La sistematización y automatización de la contabilidad tiene como finalidad liberar al contador de esta fase del proceso, permitiéndole dedicar más tiempo a labores de mayor importancia, como el análisis e interpretación de la información. De manera que se pueda conocer la capacidad financiera de la empresa.

1.5 ORIGINALIDAD

Una de las características que le otorga mayor originalidad a este proyecto radicó en su generalidad, es decir, el hecho de poder ser implementado en cualquier empresa u organización que así lo amerite, e inclusive utilizado por contadores independientes, puesto que cumple con todas las normas del sistema nacional de contabilidad venezolano. A demás, el uso de herramientas de software libre disminuye notablemente sus costos de desarrollo, lo que conlleva a bajos costo de adquisición para los usuarios finales, convirtiéndose así en una alternativa viable para pequeñas y medianas empresas.



1.6 OBJETIVOS

1.6.1 Objetivo General

Desarrollar un software que permita la automatización del registro y procesamiento de las operaciones contables históricas que se generan en una empresa o actividad productiva.

1.6.2 Objetivos Específicos

- Describir las normas y procedimientos contables asociados a una empresa o actividad productiva.
- Realizar la descripción y especificación de los requerimientos del sistema.
- Diseñar los distintos módulos que estructuran la aplicación y las bases de datos relacionales.
- Efectuar la codificación de los módulos diseñados.
- Realizar la integración de todos los módulos que conforman la aplicación.
- Realizar las pruebas de unidad, integración, validaciones y documentación del sistema.



CAPÍTULO II: MARCO TEORICO

2.1 ANTECEDENTES

Los antecedentes que se presentan se basan en proyectos relacionados con la automatización de procesos administrativos y/o software libre.

“Desarrollo de un Software que permita la automatización de las actividades asociadas al Departamento de Admisión y Control de Estudios de la extensión región centro/sur del Núcleo de Anzoátegui de la Universidad de Oriente” (Junio de 2007). Trabajo de Grado realizado por Luís Carlos Garelli Boada y Luís Eduardo Millán González, presentado como requisito parcial para obtener el título de Ingeniero en Computación en la Universidad de Oriente – Núcleo de Anzoátegui. Fue desarrollado usando la metodología del proceso unificado de desarrollo de software y programación orientada a objetos. [1]

“Desarrollo de una Aplicación para el Control Administrativo de una Organización Farmacéutica” (Febrero de 2004). Trabajo realizado por Juan Carlos Otero Groba, presentado como requisito parcial para obtener el título de Ingeniero en Computación en la Universidad de Oriente – Núcleo de Anzoátegui. [2]

“Diseño de un Sistema de Información para la atención de usuarios de la Biblioteca Pública Central- Julián Temístocles Maza”. Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui por Cherry Rodríguez y Martín Fariás, para optar al título de Ingenieros de Sistemas. El propósito de este proyecto es realizar el diseño del Sistema de Información para los Servicios de



Atención al Usuario de la Biblioteca Pública Central “Julián Temístocles Maza”, para incrementar la eficiencia de las operaciones referentes al manejo de los datos, utilizando el Lenguaje Unificado de Modelado (UML). [3]

“Diseño De Un Sistema De Información De Inventario En El Departamento IT (Información Tecnología) de una Empresa Petrolera. Puerto La Cruz, Sector Venecia”, trabajo de grado presentado en la Universidad de Oriente Núcleo de Anzoátegui por: Hiram Caguaripano, para optar al título de Ingeniero de Sistemas, en el cual se rediseñó un sistema de inventario por razones logísticas y de estructura interna en la empresa. [4]



2.2 PROGRAMACIÓN ORIENTADA A OBJETOS [5]

La programación orientada a objetos (POO) tiene la ventaja de ser un paradigma natural donde se pueden programar sistemas. Los seres humanos perciben el mundo como si estuviera formado por objetos: mesas, sillas, computadoras, coches, cuentas bancarias, partidos de fútbol, etc. También es un instinto humano intentar organizar estos objetos disponiéndolos de una forma concreta, optando por destacar determinadas características de algunos objetos que los destacan de otros.

Los niveles de dichas categorías y los métodos de clasificación de objetos en el mundo son infinitos. La forma en que las personas clasifican las cosas depende, en gran medida, de lo que deseen hacer con ellas y las características que más les llamen la atención. A la vez que se agrupan los objetos atendiendo a esquemas de clasificación, también se tiende a resaltar determinados atributos de objetos mostrando su preferencia sobre otros.

Esta idea de crear jerarquías de objetos relacionados se utiliza en la programación orientada a objetos. En 1960, los investigadores ya observaron que muchas de las entidades del modelo de programas de computadoras se podían nombrar y que se podían describir sus propiedades y comportamiento. Se dieron cuenta que los programas estaban relacionados con cuentas bancarias, matrices, archivos y usuarios; en definitiva, algo parecido a los objetos en el mundo real.

La programación orientada a objetos se puede describir rápidamente como la identificación de objetos importantes, su organización en jerarquías, la adición de atributos a los objetos que describen las características relevantes en el contexto del problema y de la adición de las funciones (métodos) a los objetos para realizar las



tareas necesarias en el objeto. Los detalles son algo más complejo, pero lo fundamental es que se trata de un proceso simple y natural.

No obstante, que sea un proceso simple y natural no significa que sea sencillo, ya que un conjunto de objetos se podría clasificar de muchas formas distintas. La clave es la posibilidad de identificar los atributos importantes de objetos y formar abstracciones y jerarquías idóneas. Incluso en el contexto de un dominio problemático, a veces resulta bastante difícil determinar los niveles correctos de abstracción y jerarquías de clasificación correctas. Simplemente, la decisión de la clase o grupo al que un objeto pertenece puede ser una tarea bastante difícil.

2.3 Características de los Lenguajes de Programación Orientados a Objetos

Los lenguajes de programación orientados a objetos (como C++, C# y Java) se caracterizan por tres conceptos claves: encapsulación, herencia y polimorfismo, que son compatibles con este aspecto natural de identificación y clasificación de objetos.

Encapsulación:

La encapsulación facilita la comprensión de los grandes programas; la ocultación de datos les hace más eficaces.

Los objetos pueden interactuar con otros objetos sólo a través de los atributos y métodos del objeto que se muestran públicamente. Cuantos más atributos y métodos se muestren públicamente, más difícil será modificar la clase sin que ello afecte al código que utiliza la clase. Una variable oculta se podría cambiar de un long a un



double, sin que ello afecte al código que utilicen los objetos creados (instanciados) de esa clase. El programador solo se debería preocupar por los métodos en la clase que han tenido acceso a esa clase, en lugar de por todos los sitios en el programa que un objeto ha instanciado desde los que se puede llamar a la clase.

Herencia:

La herencia proporciona dos ventajas evidentes a los programadores. La primera, y mas importante, es que permite crear jerarquías que expresen las relaciones entre los diferentes tipos, la segunda ventaja es que las clases pueden heredar características más generales de las clases superiores dentro de la jerarquía. En lugar de desarrollar nuevas clases a partir de cero, las clases nuevas podrán heredar las funciones de las clases existentes y, a continuación, modificar o ampliar esta función. La clase principal desde la que la nueva clase hereda las propiedades se conoce como la clase básica y la nueva clase se denomina la clase derivada.

Polimorfismo:

Polimorfismo significa, fundamentalmente, que las clases pueden tener el mismo comportamiento, pero implementarse de distintas maneras. Esto resulta muy útil en términos de programación, ya que permite trabajar con tipos de objetos genéricos cuando lo que interesa no es cómo implementa cada clase las funciones.



2.3.1 Definición de Clases y Objetos

Como su propio nombre lo indica, la programación orientada a objetos está relacionada con objetos. Un objeto se compone de datos que describen al objeto y las operaciones que se pueden realizar en el objeto. No obstante, cuando se crea un programa, en realidad se declaran y definen clases, no objetos.

Una clase es un tipo definido por el usuario, encapsula tanto los datos como los métodos que funcionen en esos datos. Una clase es algo muy parecido a una plantilla, que se utiliza para crear (instanciar) objetos.

2.3.2 Ventajas de la Programación Orientada a Objetos en el ciclo de vida de desarrollo

Son tres las ventajas clave de la programación orientada a objetos: compresión, reutilización del código y posibilidad de ampliación. La división del código en clases contribuye a imponer una estructura a medida que crecen los programas. La idea es unir los programas orientados a objetos de clases escritas previamente, así como realizar las modificaciones necesarias para admitir los nuevos requisitos mediante la herencia para derivar nuevas clases a partir de las ya existentes. La creación de sistemas a partir de componentes que se pueden volver a utilizar conduce, naturalmente, a una mayor productividad, que suele ser la ventaja más citada de todos los métodos orientados a objetos.

Las características de la programación orientada a objetos también proporcionan varias ventajas. La encapsulación facilita la escala de pequeños



sistemas a grandes sistemas. En buena medida, independientemente del tamaño del sistema, el programador simplemente está creando objetos.

Por último, la posibilidad de ocultar datos también genera sistemas más seguros. El estado de objetos sólo se puede modificar mediante métodos expuestos públicamente; esto aumenta la posibilidad de predecir el comportamiento del objeto.

2.4 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE [6]

El proceso unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El proceso unificado utiliza el lenguaje unificado de modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software, de hecho UML es una parte esencial del proceso unificado.

2.5 LENGUAJE DE MODELADO UNIFICADO (UML) [7]

El Lenguaje de Modelado Unificado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la



semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML es una especificación de notación orientada a objetos. Se basa en las anteriores especificaciones Booch, Rumbaugh y Coad-Yourdon. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representan la arquitectura del proyecto. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

2.6 BASES DE DATOS [8]

Una base de datos es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito.

Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado mini mundo o universo de discurso. Las modificaciones del mini mundo se reflejan en la base de datos.
- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.



- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

En otras palabras, una base de datos tiene una fuente de la cual se derivan datos, cierto grado de interacción con los acontecimientos del mundo real y un público que está activamente interesado en el contenido de la base de datos.

La generación y mantenimiento de las bases de datos pueden ser manuales o mecánicos. El catálogo en tarjetas de una biblioteca es un ejemplo de base de datos que se puede crear y mantener manualmente. Las bases de datos computarizadas se pueden crear y mantener con un grupo de programas de aplicación escritos específicamente para esa tarea o bien mediante un sistema de gestión de bases de datos.

2.6.1 Tipos de Bases de Datos

Base de Datos jerárquicos:

Los datos se organizan en grupos unidos entre ellos por relaciones de "posesión", en las que un conjunto de datos puede tener otros conjuntos de datos, pero un conjunto puede pertenecer sólo a otro conjunto. La estructura resultante es un árbol de conjuntos de datos.

**Base de Datos Reticulares:**

El modelo reticular es muy parecido al jerárquico, y de hecho nace como una extensión de este último. También en este modelo conjunto de datos están unidos por relaciones de posesión, pero cada conjunto de datos puede pertenecer a uno o más conjuntos.

Base de Datos Relacionales:

Las bases de datos que pertenecen a esta categoría se basan en el modelo relacional, cuya estructura principal es la relación, es decir una tabla bidimensional compuesta por líneas y columnas. Cada línea, que en terminología relacional se llama tupla, representa una entidad que nosotros queremos memorizar en la base de datos. Las características de cada entidad están definidas por las columnas de las relaciones, que se llaman atributos. Entidades con características comunes, es decir descritas por el mismo conjunto de atributos, formarán parte de la misma relación.

Base de Datos Orientada a Objeto:

El esquema de una base de datos por objetos está representado por un conjunto de clases que definen las características y el comportamiento de los objetos que poblarán la base de datos. La diferencia principal respecto a los modelos examinados hasta ahora es la no positividad de los datos. En efecto, con una base de datos tradicional (entendiendo con este término cualquier base de datos no por objetos), las operaciones que se tienen que efectuar en los datos se les piden a las aplicaciones que los usan. Con una base de datos orientada a objeto, al contrario, los objetos memorizados en la base de datos contienen tanto los datos como las operaciones



posibles con tales datos. En cierto sentido, se podrá pensar en los objetos como en datos a los que se les ha puesto una inyección de inteligencia que les permita saber cómo comportarse, sin tener que apoyarse en aplicaciones externas.

2.6.2 Sistema de Gestión de Base de Datos (SGBD)

En inglés, *Database Management System DBMS*, es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Por tanto, el SGBD es un sistema de software de propósito general que facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones. Para definir una base de datos hay que especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenarán en ella. Construir una base de datos es el proceso de guardar los datos en algún medio de almacenamiento controlado por el SGBD. En la manipulación de una base de datos intervienen funciones como consultar la base de datos para obtener datos muy específicos, actualizar la base de datos para reflejar cambios en el mini mundo y generar informes a partir de datos.

No hace falta un software de SGBD de propósito general para implementar una base de datos computarizada. Se puede escribir un conjunto propio de programas para crear y mantener la base de datos, con lo cual estaríamos creando de hecho nuestro propio software SGBD de propósito específico. En todo caso, ya sea que se utilice un SGBD de propósito general o no, casi siempre se requiere un software de gran capacidad para manipular la base de datos, además de la base de datos misma. Al conjunto formado por la base de datos y el software se le denomina sistema de base de datos.



2.6.3 Descripción de la arquitectura Cliente-Servidor

Se usa para caracterizar un SGBD cuando la aplicación se ejecuta físicamente en una máquina, llamada cliente, y otra, el servidor, se encarga del almacenamiento y el acceso a los datos. Los proveedores ofrecen diversas combinaciones de clientes y servidores; por ejemplo, un servidor para varios clientes.

2.7 CONTABILIDAD [9]

Es la ciencia y técnica que enseña a recopilar, clasificar y registrar de una forma sistemática y estructural, las operaciones mercantiles realizadas por una empresa, con el fin de producir informes que, analizados e interpretados, permitan planear, controlar y tomar decisiones sobre la actividad de la empresa.

2.7.1 Partida Doble

Es un principio universal introducido por Fray Lucas Paciolo, según el cual si se debita una cuenta por un valor, se debe acreditar otra u otras por igual valor. Es un sistema universalmente aceptado para el registro contable de transacciones comerciales. Se fundamenta en el siguiente hecho:

- Cuando una persona vende, hay otra que compra.
- Cuando una persona entrega, hay otra que recibe.



- No puede existir deudor sin acreedor.
- No puede existir acreedor sin deudor.

Tal situación da lugar a los asientos de contabilidad, que no son otra cosa que el registro de las transacciones u operaciones comerciales en cada una de las cuentas afectadas.

Esto quiere decir que toda transacción comercial implica: entrega de mercancías o servicios y entrega de efectivo o aceptación de la deuda.

El principio universal de la Partida Doble significa que en cualquier asiento contable, la suma de los débitos debe ser igual a la suma de los créditos.

2.7.2 Asientos Contables

Son los registros de las operaciones mercantiles en los comprobantes y libros de contabilidad, pueden ser asientos simples y asientos compuestos; simple cuando se utilizan dos cuentas, un débito y un crédito; compuesto cuando se emplean más de dos cuentas.

2.7.3 Ecuación Patrimonial

Es la resultante de comparar el activo y el pasivo de una empresa, con el objeto de obtener el valor del Patrimonio y Capital. Desde su inicio, toda empresa o ente



económico puede expresar su ecuación patrimonial, la que se conservará a través de toda la vida de la empresa, aun cuando los valores que involucra aumenten o disminuyan el Activo o el Pasivo.

2.7.4 Concepto de Cuenta

Es el nombre que se utiliza para registrar, en forma ordenada, las operaciones que diariamente realiza una empresa.

Cuenta es el registro de valores homogéneos relativos a una persona o cosa, bajo un título apropiado, que facilita la interpretación de las operaciones en los libros.

Cuenta es un término usado en contabilidad para designar derechos, bienes, obligaciones y resultados; en ella se hacen registros sistemáticos y análogos que permiten la interpretación de las operaciones de un ente económico.

2.7.5 Balance General

Informe básico que muestra la situación financiera de un ente económico, en una fecha determinada. Contiene información sobre los activos, pasivos y patrimonio, los cuales deben relacionarse unos con otros para reflejar dicha situación financiera.



2.7.6 Depreciación

Pérdida del valor de los elementos del activo fijo o inmovilizado de cualquier institución, o empresa, al prestar la función que le es propia. Resulta de dividir el costo original del elemento o bien por su vida útil. Aunque no es una erogación real de efectivo, la legislación tributaria permite tratar esta partida como un gasto, disminuyendo las utilidades y convirtiendo así en un crédito tributario.

2.7.7 Libros de Contabilidad

Las características básicas de un libro de contabilidad deben remitirse, en principio, a una fecha de registro, una descripción del registro y unas columnas donde se registra el valor de acuerdo al concepto, ya sea entrada o salida (debe o haber). La tecnología actual agiliza el manejo de los libros contables, pues elimina el uso del registro manual, pero esto no quiere decir que pierda sus características enunciadas anteriormente.

2.7.8 Empresa

Se entenderá por empresa toda actividad económica organizada para la producción, transformación, circulación, administración o custodia de bienes o para la prestación de servicios. Dicha actividad se realizará a través de uno o más establecimientos de comercio.



2.8 SOFTWARE LIBRE [10]

Software libre (en inglés free software) es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. Análogamente, el software gratis o gratuito (denominado usualmente freeware) incluye en algunas ocasiones el código fuente; sin embargo, este tipo de software no es libre en el mismo sentido que el software libre, al menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

2.8.1 Historia del Software Libre

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época (los mainframes) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Con este antecedente, en 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation (FSF).



Stallman introdujo una definición para free software y el concepto de "copyleft", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software.

El término free, traducido al castellano, significa tanto libre como gratis, por eso muchas veces suelen confundirse el freeware con el software libre aunque entre ambos existen notables diferencias.

2.8.2 Libertades del Software Libre [11]

Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.

Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades – el acceso al código fuente es condición indispensable para esto.

Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.

Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad – el acceso al código fuente es condición indispensable para esto.

Software libre es cualquier programa en donde los usuarios gocen de estas libertades. De modo que la persona debería ser libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, entre otras cosas, no tener que pedir permiso ni pagar para ello.



Asimismo, se debería ser libre para introducir modificaciones y utilizarlas de forma privada, ya sea en el trabajo o en tiempo libre, sin siquiera tener que mencionar su existencia. Si decidieras publicar estos cambios, no se debería estar obligado a notificarlo de ninguna forma ni a nadie en particular.

La libertad para utilizar un programa significa que cualquier individuo u organización podrá ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto.

La libertad para redistribuir copias supone incluir las formas binarias o ejecutables del programa y el código fuente tanto de las versiones modificadas como de las originales. La distribución de programas en formato ejecutable es necesaria para su adecuada instalación en sistemas operativos libres. No pasa nada si no se puede producir una forma ejecutable o binaria, dado que no todos los lenguajes pueden soportarlo, pero todos debemos tener la libertad para redistribuir tales formas si se encuentra el modo de hacerlo.

Para que las libertades 1 y 3 (la libertad para hacer cambios y para publicar las versiones mejoradas) adquieran significado, se debe disponer del código fuente del programa. Por consiguiente, la accesibilidad del código fuente es una condición necesaria para el software libre.

Para materializar estas libertades, éstas deberán ser irrevocables siempre que no cometamos ningún error; si el desarrollador del software pudiera revocar la licencia sin motivo, ese software dejaría de ser libre.

Sin embargo, ciertas normas sobre la distribución de software libre nos parecen aceptables siempre que no planteen un conflicto con las libertades centrales. Por



ejemplo, el copyleft, grosso modo, es la norma que establece que, al redistribuir el programa, no pueden añadirse restricciones que nieguen a los demás sus libertades centrales.

Esta norma no viola dichas libertades, sino que las protege. De modo que puedes pagar o no por obtener copias de software libre, pero independientemente de la manera en que las obtengas, siempre tendrás libertad para copiar, modificar e incluso vender estas copias.

2.8.3 Tipos de Licencias mas usadas en Software Libre

- BSD (Berkeley Software Distribution): una aplicación licenciada con BSD permite que otras versiones puedan tener otros tipos de licencias, tanto propietarias, como BSD o GPL.
- GPL (General Public License): esta licencia aplica la necesidad del copyleft, haciendo que las nuevas versiones de la aplicación sean siempre libres y licenciadas bajo GPL.
- MPL (Mozilla Public License): aplica licencias dobles al código fuente y a los ejecutables, obligando a devolver al autor los códigos fuentes modificados y permitiendo licenciar los binarios como propietarios.



CAPITULO III. FASE DE INICIO

3.1 INTRODUCCIÓN

En este capítulo se llevó a cabo la primera fase del Proceso Unificado de Desarrollo de Software, llamada Fase de Inicio, en donde se establece el concepto inicial del sistema y se crea un esquema general para su comprensión, mediante la identificación de sus funcionalidades y requisitos; todo esto con el fin de elaborar un conjunto de modelos iniciales que capturen la semántica y comportamiento del sistema.

El objetivo de la fase de inicio es conocer a fondo las actividades y procesos más relevantes dentro del campo de estudio, obtener los requerimientos por parte de los usuarios y consolidar una visión única del sistema para modelar su estructura y dinámica; asegurando que exista un entendimiento común entre los desarrolladores y los usuarios.

Para definir la visión y obtener una estimación del orden de magnitud es necesario llevar a cabo alguna exploración de los requisitos. Sin embargo, el objetivo de la etapa de inicio no es definir todos los requisitos, o generar una estimación creíble o plan de proyecto. Aún a riesgo de simplificar demasiado, la idea es hacer la investigación justa para formar una opinión racional y justificable del propósito global y la viabilidad del nuevo sistema potencial, y decidir si merece la pena invertir en un estudio más profundo. Por lo tanto, la fase de inicio debe ser relativamente corta en la mayoría de los proyectos, una duración de unas pocas semanas.



3.2 PLANIFICACIÓN DE LA FASE DE INICIO

Para llevar a cabo esta fase se realizó un análisis del sistema tomando en cuenta varios aspectos esenciales; en primer lugar, se estudiará el contexto en el cual se desarrollan todas las actividades humanas relacionadas con el tema de interés, analizando todos los procesos y procedimientos involucrados; que serán representados a través del modelo de dominio. En segundo lugar, se identificarán los riesgos que serán mitigados con la realización de las fases en el desarrollo del proyecto. En tercer lugar, se capturarán los requisitos y se describirá la interacción de los actores implicados en el sistema a través de los diagramas de casos de uso. Una vez elaborados estos diagramas, se procederá a modelarlos a través de los diagramas de clases de análisis, para comprender cómo se deberán realizar dichos casos de uso.

El propósito del desarrollo de esta fase es establecer una visión común inicial de los objetivos del proyecto, determinar y decidir cuales elementos e investigaciones valen la pena llevar a cabo en las próximas etapas.

Para este proyecto, en esta fase inicial, se realizó una única iteración, implementando los flujos de trabajo de requisitos y análisis, que comprende el estudio de los requisitos y análisis de los mismos.

Se definen y estudian los riesgos, arquitectura del sistema, modelo de dominio y casos de uso más críticos del sistema.



3.3 ESTUDIO DEL CONTEXTO DEL SISTEMA

En todo proyecto de desarrollo de software se debe tener un acercamiento con el entorno del sistema. En el presente proyecto, se llevó a cabo un estudio de las actividades más importantes relacionadas con la contabilidad de una empresa, con el fin de comprender los métodos y procedimientos utilizados en esta área, y ofrecer una visión representativa del entorno del sistema.

3.3.1 Modelo de Dominio del Sistema Propuesto

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.

Con el modelo del dominio se capturan los objetos más importantes en el contexto del Software J-Contab. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno del sistema. El objetivo de este modelado es comprender el contexto y los requerimientos del sistema propuesto; en otras palabras el modelado del dominio contribuye a entender el problema, que se supone que el sistema resuelve en relación a su contexto.

Por ejemplo, la figura 3.1 muestra el diagrama de dominio parcial del sistema a implementar, en el que podemos observar lo siguiente:

Las empresas u organizaciones deben llevar un registro de todos sus movimientos financieros, tarea que recae sobre uno o varios contadores, quienes se encargan de la teneduría de libros contables. En estos libros se llevan los registros de



forma cronológica y ordenada por cuentas. La cuenta, es un medio o instrumento contable a través del cual son registradas las operaciones de acuerdo a su naturaleza, estos registros son denominados asientos contables. La información contenida en los libros es sometida a criterios y calculo que permiten obtener reportes del estado financiero de la empresa u organización, vitales a efectos legales y de productividad de la empresa.

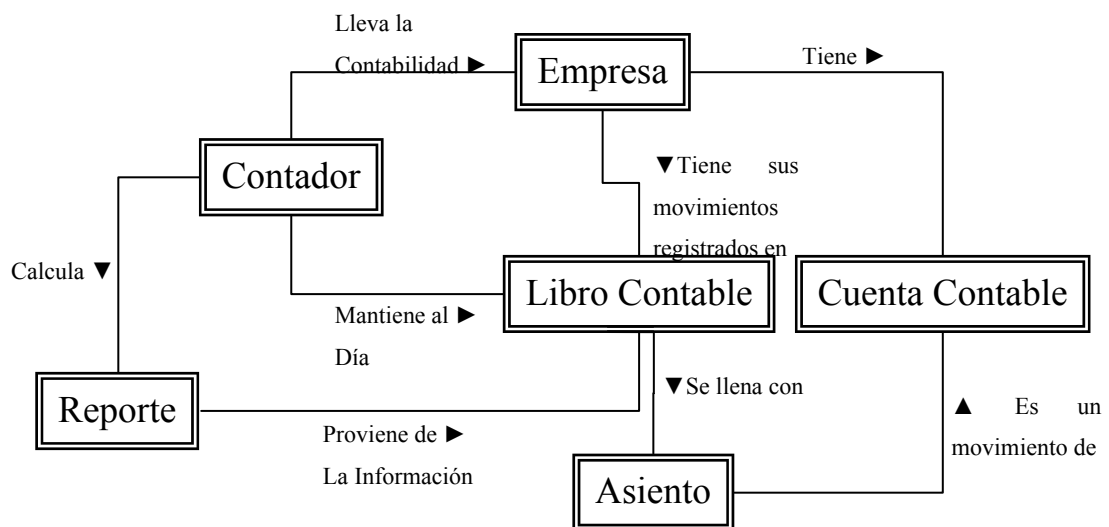


Figura 3.1 Modelo de Dominio del sistema

3.4 RIESGOS DEL SISTEMA

Un riesgo es una variable del proyecto que pone en peligro o impide el éxito del proyecto. Es la probabilidad de que un proyecto experimente sucesos no deseables.



El modo en que se planifica el desarrollo de un nuevo sistema está influenciado en gran medida por los riesgos que se perciben. Por tanto uno de los primeros pasos, al principio de la fase de inicio, es crear una lista de riesgos.

Los riesgos dirigen la viabilidad del sistema en el Proceso Unificado de Desarrollo de Software; ya que existe la probabilidad de que el proyecto se vea afectado en su desarrollo o comportamiento futuro. En tal sentido, los riesgos deben ser ordenados por el nivel de seriedad o por su influencia en el desarrollo.

El principio general es tener un plan de acción a seguir sobre los riesgos. Las fases y las iteraciones dentro de las fases proporcionan el medio de planificar las acciones sobre los riesgos.

Una vez que se han identificado los riesgos, pueden ser tratados de diferentes maneras. Algunos pueden ser evitados llevando a cabo acciones como replanear el proyecto, pueden ser también evitados haciendo cambios en los requisitos, otros podrían ser detectados y aislados de otras partes del proyecto, a manera de afectar solo una parte.

3.4.1 Riesgos Críticos del Sistema

Desconocimiento del ámbito del sistema:

Descripción: El no conocer el ámbito del sistema en base al cual se está trabajando, significa un riesgo crítico que debe ser mitigado en la fase de inicio. El diseñador debe tener un conocimiento del contexto y entender sus aspectos fundamentales.



Acción Preventiva: Realizar una investigación bibliográfica suficientemente extensa del tema a fin de obtener los conocimientos requeridos.

Requisitos mal definidos:

Descripción: No disponer de unos requisitos claros y que abarquen todas las necesidades de los usuarios, pueden significar un riesgo para el sistema. Se deben recolectar todas las necesidades de los usuarios.

Acción Preventiva: Tomar con precisión todas las necesidades del problema, haciendo una revisión de las mismas en la búsqueda de cubrir todos los detalles posibles.

Falta de robustez de la arquitectura:

Descripción: Un riesgo crítico que debe ser mitigado en la fase de inicio es el riesgo de falta de robustez de la arquitectura. El sistema debe poseer una arquitectura robusta que le permita adaptarse a los cambios y al mantenimiento de manera elegante y poco traumática.

Acción Preventiva: Realizar un análisis adecuado de los diferentes casos de uso mediante los diagramas de análisis, proporcionándole al sistema una arquitectura sólida y estable.

Diseño no adecuado de la base de datos:

Descripción: El mal diseño de la base de datos es un riesgo crítico, la base de datos del sistema debe estar diseñada de tal manera que mantenga la integridad



referencial, que no sea redundante y que garantice el acceso a los datos necesarios, datos que le permiten al sistema llevar a cabo sus funciones.

Acción Preventiva: Realizar un diagrama entidad-relación cubriendo todas las necesidades de datos del sistema, posteriormente convertirlo a modelo físico y realizar normalizaciones hasta la Forma Normal de Boyce-Codd, respetando todas las restricciones de integridad, eliminando así las redundancias de datos.

3.5 FLUJO DE TRABAJO DE REQUISITOS

La ingeniería de requisitos es un término utilizado para describir las actividades implicadas en la obtención, documentación y mantenimiento de un conjunto de requisitos para un sistema de software. Trata acerca de descubrir que necesitan los grupos de decisión que el sistema haga por ellos.

Puesto que el sistema de software final se basa sobre un conjunto de requisitos, la ingeniería de requisitos es un factor crítico de éxito en proyectos de desarrollo de software.

Se puede definir un requisito como “una especificación de lo que se debería implementar”. Existen básicamente dos tipos de requisitos, los requisitos funcionales que son los que determinan el comportamiento que debería ofrecer el sistema, y los no funcionales que son los que determinan una propiedad específica o restricción del sistema.

Los requisitos son (o al menos deberían ser) la base de todos los sistemas. En principio, los requisitos deberían ser solamente una declaración de lo que debería



hacer el sistema, y no cómo lo debería hacer. Ésta es una importante distinción. Se puede especificar lo que un sistema debería hacer y qué comportamiento debería mostrar un sistema sin decir necesariamente nada sobre cómo esta funcionalidad debería estar realizada.

3.5.1 Requisitos Funcionales

- El sistema debe permitir el registro de la información contable de varias empresas.
- El sistema debe manejar un plan de cuentas único para cada empresa cargada en él.
- El sistema debe llevar el registro de los asientos en un libro diario único para cada empresa.
- El sistema debe mostrar los siguientes reportes: Ganancias y Pérdidas, Balance General, Balance de Comprobación, Balance de suma, Diario General, Libro Mayor, Mayor Analítico.
- El sistema debe permitir la exportación e importación de la información contable de una determinada empresa.



3.5.2 Requisitos No Funcionales

- El sistema debe estar realizado con herramientas de Software Libre cumpliendo con el Decreto Presidencial N° 3.390, a manera de reducir los costos de desarrollo y su código pueda ser modificado sin ninguna restricción.
- EL sistema debe poseer una interfaz gráfica, amigable e intuitiva que permita la fácil interacción con el usuario, reduciendo así la curva de aprendizaje del mismo.
- El sistema debe estar diseñado con una arquitectura que pueda adaptarse fácilmente a cualquier cambio y mejora estructural.

3.5.3 Modelado de Casos de Uso

El modelado de casos de uso es una forma complementaria de crear y documentar requisitos, permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los mismos.

El caso de uso es el artefacto fundamental utilizado en la captura de requisitos. Cada forma en que los actores usan el sistema se representa con un caso de uso. Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.



3.5.3.1 Identificación de Actores

Un actor especifica un rol que cierta entidad externa adopta cuando interactúa con su sistema directamente. Puede representar un rol de usuario, o un rol desempeñado por otro sistema o hardware que toca el límite del sistema. Para identificar los actores, se necesita considerar quién y que utiliza el sistema y qué roles desempeñan en su interacción con el sistema.

Tabla 3.1 Actores identificados en el sistema.

ACTORES	DESCRIPCION
Administrador	Este usuario cuenta con la permisología absoluta y completo acceso a todos los módulos.
Usuario	Representa cualquier usuario común con permisología a las funciones básicas del sistema, mas no a funciones avanzadas y de configuración.
SMBD	Representa el sistema manejador de bases de datos que se encargara de la seguridad e integridad de la data en el sistema.

3.5.3.2 Identificación de Casos de Uso

Luego de que han sido identificados los actores principales del sistema y se ha establecido una lista de requisitos, se procede a identificar los casos de uso y las relaciones que existen entre ellos. A continuación se presentan los casos de usos identificados para el Software J-Contab.



Tabla 3.2 Casos de Uso del sistema.

Caso de uso	Descripción	Actores
Administrar Usuarios	Permite agregar o eliminar usuarios del sistema y establecer sus privilegios.	Administrador. SMBD.
Manejo de Respaldos	Permite crear o recuperar respaldos de la Base de Datos del sistema.	Administrador. SMBD.
Administrar Empresas	Permite agregar, modificar o eliminar una empresa en el sistema.	Administrador. Usuario. SMBD.
Administrar Cuentas	Permite agregar, modificar o eliminar una cuenta en el sistema.	Administrador. Usuario. SMBD.
Manejar Asientos	Permite ingresar, modificar o eliminar los asientos contables al sistema.	Administrador. Usuario. SMBD.
Emitir Reportes	Permite generar reportes impresos y electrónicos referentes a toda la información consultada por los usuarios.	Administrador. Usuario. SMBD.

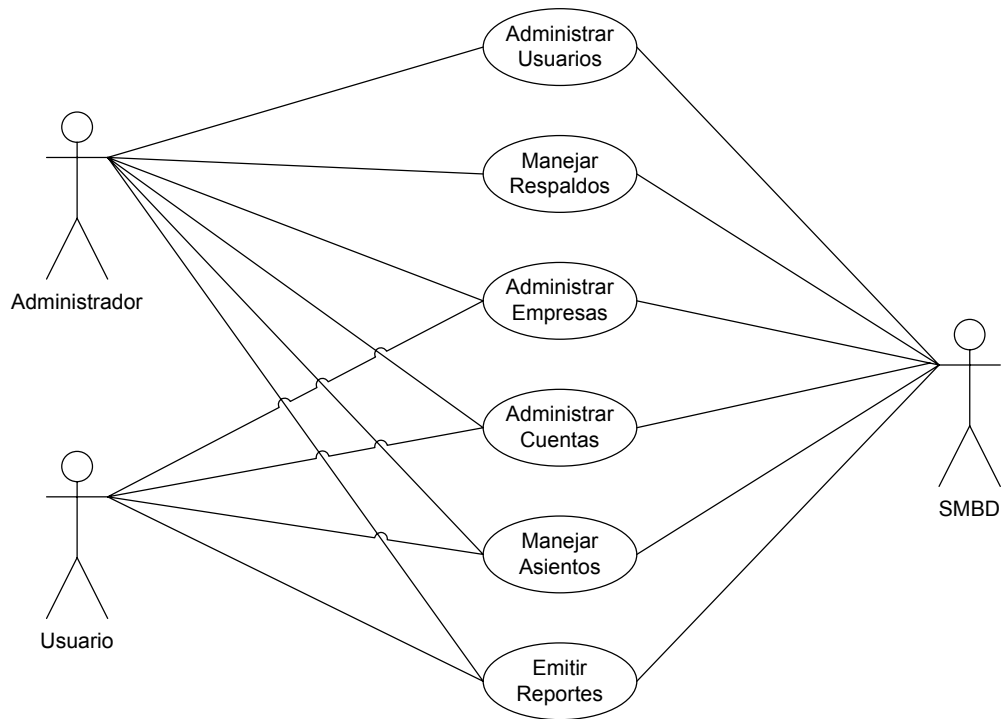


Figura 3.2 Caso de Uso General del sistema.

3.5.3.3 Descripción de Casos de Uso

El caso de uso principal permite dar una visión general del sistema y sólo representa sus funciones principales. A continuación para una mejor comprensión del sistema se presentan a detalle los casos de uso que lo conforman.



Caso de Uso No 1 “Administrar Usuarios”

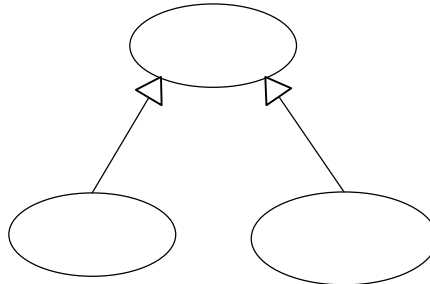


Figura 3.3 Caso de Uso Administrar Usuarios

Nombre del Caso de Uso 1.1: Crear Usuario.

Actores: Administrador, SMBD

Descripción: Ingresa al sistema un nuevo usuario

Nombre del Caso de Uso 1.2: Eliminar Usuario.

Actores: Administrador, SMBD.

Descripción: Elimina un usuario del sistema.

Caso de Uso No 2 “Manejar Respaldos”

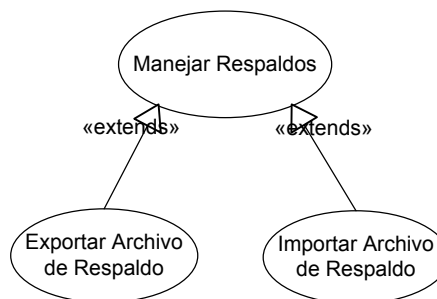


Figura 3.4 Caso de Uso Manejar Respaldos

Crear Usuario

«extends»

Ad
U



Nombre del Caso de Uso 2.1: Exportar Archivo de Respaldo.

Actores: Administrador, SMBD

Descripción: Genera un archivo de respaldo con la información contenida en la base de datos del sistema.

Nombre del Caso de Uso 2.2: Importar Archivo de Respaldo.

Actores: Administrador, SMBD

Descripción: Carga a la base de datos del sistema, la información contenida en un archivo de respaldo.

Caso de Uso No 3 “Administrar Empresas”

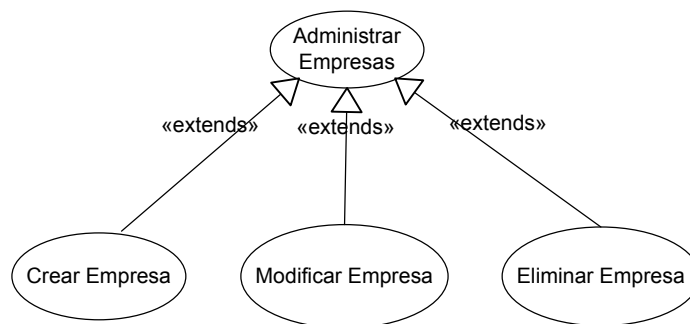


Figura 3.5 Caso de Uso Administrar Empresas

Nombre del Caso de Uso 3.1: Crear Empresa.

Actores: Administrador, Usuario, SMBD

Descripción: Ingresa al sistema una nueva empresa

Nombre del Caso de Uso 3.2: Modificar Empresa.

Actores: Administrador, Usuario, SMBD

Descripción: Modifica los datos de una empresa existente en el sistema.



Nombre del Caso de Uso 3.3: Eliminar Empresa.

Actores: Administrador, SMBD.

Descripción: Elimina una empresa del sistema junto con todos sus datos asociados, existentes en el sistema.

Caso de Uso No 4 “Administrar Cuentas”

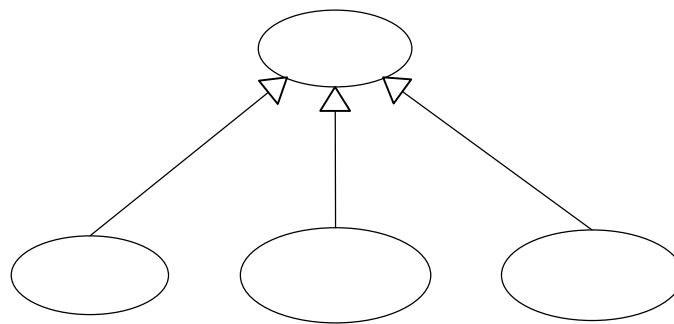


Figura 3.6 Caso de Uso Administrar Cuentas

Nombre del Caso de Uso 4.1: Crear Cuenta.

Actores: Administrador, Usuario, SMBD

Descripción: Ingresa una nueva cuenta al plan de cuentas de una determinada empresa existente en el sistema.

Nombre del Caso de Uso 4.2: Modificar Cuenta.

Actores: Administrador, Usuario, SMBD

Descripción: Modifica los datos de una cuenta perteneciente al plan de cuenta de una determinada empresa existente en el sistema.

«extends»

Nombre del Caso de Uso 4.3: Eliminar Cuenta.

Actores: Administrador, SMBD.

Descripción: Elimina una cuenta perteneciente al plan de cuenta de una determinada empresa existente en el sistema.

Crear Cuenta

Modif

Ad
C

«e



Caso de Uso No 5 “Manejar Asientos”

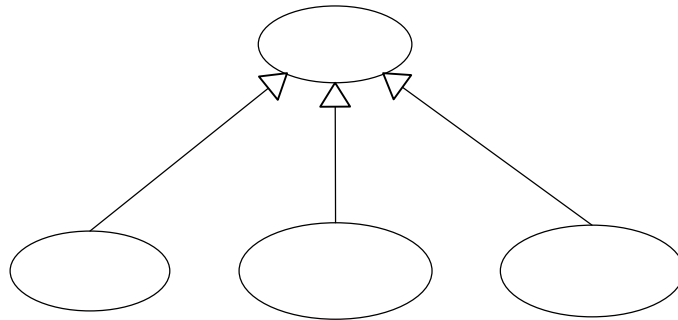


Figura 3.7 Caso de Uso Manejar Asientos

Nombre del Caso de Uso 5.1: Crear Asiento.

Actores: Administrador, Usuario, SMBD

Descripción: Ingresa un nuevo asiento al libro diario de una empresa determinada.

«extends»

Nombre del Caso de Uso 5.2: Modificar Asiento.

Actores: Administrador, Usuario, SMBD

Descripción: Modifica los datos de un asiento existente en el libro diario de una empresa determinada.

M
A

«e

Nombre del Caso de Uso 5.3: Eliminar Asiento.

Crear Asiento

Modifi

Actores: Administrador, SMBD.

Descripción: Elimina un asiento del libro diario de una determinada empresa.



Caso de Uso No 6 “Emitir Reportes”

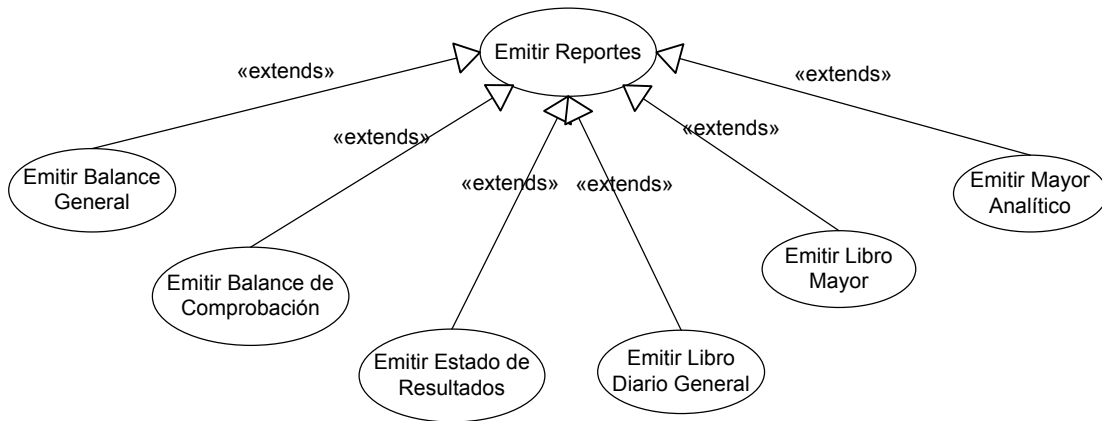


Figura 3.8 Caso de Uso Emitir Reportes

Nombre del Caso de Uso 6.1: Balance General.

Actores: Administrador, Usuario, SMBD.

Descripción: Documento contable que presenta la situación financiera de un negocio a una fecha determinada.

Nombre del Caso de Uso 6.2: Balance de Comprobación.

Actores: Administrador, Usuario, SMBD.

Descripción: Instrumento financiero que se utiliza para visualizar la lista del total de los debitos y de los créditos de las cuentas, junto al saldo de cada una de ellas.

Nombre del Caso de Uso 6.3: Estado de Resultados (Ganancias y Pérdidas).

Actores: Administrador, Usuario, SMBD.

Descripción: Es un documento contable que muestra detallada y ordenadamente la utilidad o pérdida del ejercicio.



Nombre del Caso de Uso 6.4: Libro Diario General.

Actores: Administrador, Usuario, SMBD.

Descripción: En él se registran todos los hechos y operaciones contables.

Nombre del Caso de Uso 6.5: Libro Mayor.

Actores: Administrador, Usuario, SMBD.

Descripción: Ordena los registro del Diario General por cuentas contables.

Nombre del Caso de Uso 6.6: Mayor Analítico.

Actores: Administrador, Usuario, SMBD.

Descripción: Muestra una Cuenta Contable y sus relación de movimientos.

3.6 FLUJO DE TRABAJO DE ANÁLISIS

El objetivo de éste flujo de trabajo, es analizar los requisitos, refinarlos y estructurarlos en un conjunto de objetos conceptuales que se denomina modelo de análisis que permite identificar los elementos significativos del sistema desde un punto de vista de la arquitectura con vista a la construcción de un modelo de diseño.

El análisis de casos de uso se realiza para identificar las clases de análisis cuyos objetos son necesarios para llevar a cabo el flujo de sucesos del caso de uso, y distribuir su comportamiento entre los objetos del análisis.



3.6.1 Identificación de las Clases de Análisis

En este punto se identifican las clases de control, de entidad, y de interfaz para realizar los casos de uso y esbozar sus nombres, responsabilidades, atributos y relaciones.

Existen tres estereotipos básicos para identificar las clases de análisis: clases de control, clases de entidad y clases de interfaz.

La Clase de control modela funcionalidad que no corresponde a ningún objeto en particular y que se presenta en algunos casos de uso. Estos objetos generalmente operan sobre varios objetos entidad, realizan algún algoritmo y retornan algún resultado a un objeto de interfaz. Esta clase representa:

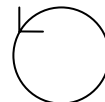
Coordinación

Secuencia

Transacciones

Control de otros objetos

Su estereotipo es:





En la tabla siguiente se muestra una descripción de las clases de control identificadas en el sistema.

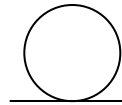
Tabla 3.3 Descripción de las clases de control

Clase de Control	Descripción
Gestor Usuarios	Recibe y valida la información de los usuarios.
Gestor Respaldos	Hace la solicitud de los datos de las empresas y sus asientos correspondientes.
Gestor Empresas	Recibe y valida la información de las empresas. Hace la solicitud de los datos de las empresas.
Gestor Cuentas	Recibe y valida la información de las cuentas. Hace la solicitud de los datos de las cuentas.
Gestor Asientos	Recibe y valida la información de los asientos. Hace la solicitud de los datos de los asientos.
Gestor Reportes	Hace la solicitud de los asientos y realiza los cálculos correspondientes para entregar el reporte solicitado.



La Clase de entidad se utiliza para modelar información que posee una vida larga o que debe sobrevivir cierto período de tiempo.

Su estereotipo es:



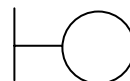
A continuación se muestra una breve descripción de las clases de entidad del sistema.

Tabla 3.4 Descripción de las clases de entidad.

Clase de Entidad	Descripción
Usuario	Almacena temporalmente los datos del usuario.
Empresa	Almacena temporalmente los datos de la empresa.
Cuenta	Almacena temporalmente los datos de la Cuenta.
Asiento	Almacena temporalmente los datos del Asiento.

La Clase de Interfaz de Usuario (IU) se utiliza para modelar información y comportamiento que es dependiente de la interfaz actual del sistema, también modela la interacción entre el sistema y sus actores, esto implica recibir información y peticiones de los usuarios y los sistemas externos. Estas clases representan abstracción de ventanas, Formularios, paneles, interfaces de comunicación interfaces de impresoras, sensores, etc.

Su estereotipo es:



A continuación se describen brevemente las clases de interfaz de usuario del sistema.



Tabla 3.5 Descripción de las clases de interfaz. 1/2

Clase de Interfaz de Usuario	Descripción
Crear Usuario	Permite ingresar al sistema los datos de un nuevo usuario.
Eliminar Usuario	Permite eliminar un usuario existente del sistema.
Exportar Archivo de Respaldo	Permite al usuario generar un archivo de respaldo de los datos de las empresas.
Importar Archivo de Respaldo	Permite ingresar al sistema los datos de las empresas contenidos en un archivo de respaldo.
Crear Empresa	Permite al usuario ingresar al sistema los datos de una nueva empresa.
Modificar Empresa	Permite al usuario modificar los datos de una empresa existente en el sistema.
Eliminar Empresa	Permite al usuario seleccionar una empresa existente en el sistema para luego ser eliminada.
Crear Cuenta	Permite ingresar los datos de una nueva cuenta, al plan de cuentas de una empresa previamente cargada en el sistema.
Modificar Cuenta	Permite al usuario modificar los datos de una cuenta.
Eliminar Cuenta	Permite al usuario eliminar una cuenta, del plan de cuentas de una determinada empresa.
Crear Asiento	Permite ingresar los datos de un nuevo asiento al libro diario de una empresa.
Modificar Asiento	Permite la modificación de los datos de un asiento.



Tabla 3.5 Descripción de las clases de interfaz. 2/2

Clase de Interfaz de Usuario	Descripción
Eliminar Asiento	Permite eliminar un asiento del libro diario de una empresa.
Balance General	Muestra un reporte de la situación financiera de una empresa a una fecha determinada.
Balance de Comprobación	Muestra un reporte con la lista del total de los debitos y de los créditos de las cuentas, junto al saldo de cada una de ellas.
Estado de Resultado	Muestra un reporte detallado y ordenado de la utilidad o perdida del ejercicio.
Libro Diario General	Muestra un reporte con todos los hechos y operaciones contables.
Libro Mayor	Muestra un reporte con los registros del Diario General ordenados por cuentas contables.
Mayor Analítico	Muestra el reporte de una Cuenta Contable y su relación de movimientos.
BD	Permite la comunicación entre el software y el sistema manejador de base de datos.



3.6.2 Diagramas de Clases de Análisis

Los diagramas de clases de análisis plasman las posibles clases del diseño, encajándolas en los tres estereotipos básicos: de interfaz, de control y de entidad. Cada uno de ellos implica una semántica específica, lo cual constituye un método consistente de identificar y describir cada clase, contribuyendo a la creación de un modelo de objetos y una arquitectura robusta.

3.6.2.1 Diagrama de Clases de Análisis para el Caso de Uso Administrar Usuarios

Diagrama de Análisis para el Caso de Uso Crear Usuario

El usuario inicializa el proceso haciendo la petición e ingresando los datos a la IU Crear Usuario, el Gestor de Usuario recibe y valida la información del nuevo usuario y pasa los datos a la entidad Usuario que los almacena temporalmente para luego enviarlos al SMBD.

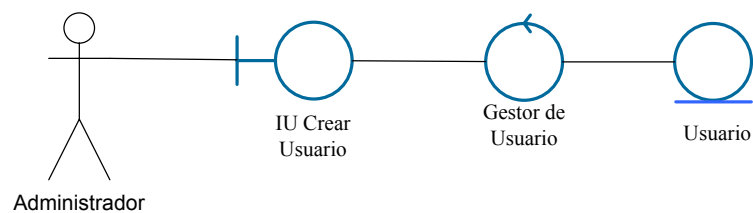


Figura 3.9 Diagrama de Análisis para el Caso de Uso Crear Usuario



Diagrama de Análisis para el Caso de Uso Eliminar Usuario

El usuario inicializa el proceso haciendo la petición a la IU Eliminar Usuario, el Gestor de Usuarios recibe y valida la información del usuario y pasa los datos a la entidad Usuario que los almacena temporalmente para luego solicitar la eliminación al SMBD.

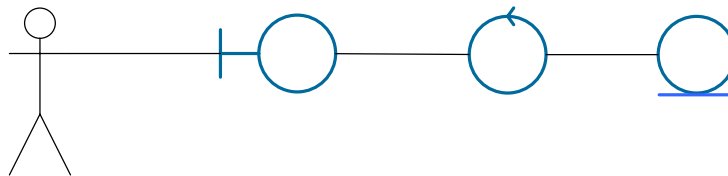


Figura 3.10 Diagrama de Análisis para el Caso de Uso Eliminar Usuario

3.6.2.2 Diagrama de Clases de Análisis para el Caso de Uso Manejar Respaldos

Diagrama de Análisis para el Caso de Uso Exportar Archivo de Respaldo

El usuario inicializa el proceso haciendo la petición a la IU Exportar Respaldo, indicando la ruta y nombre de archivo en el cual se deberá almacenar la información exportada, posteriormente el Gestor de Respaldo hace la solicitud de los datos al SMBD, una vez realizada esta acción el respaldo se guarda en el archivo antes especificado.

IU Elimina
Usuario

Administrador

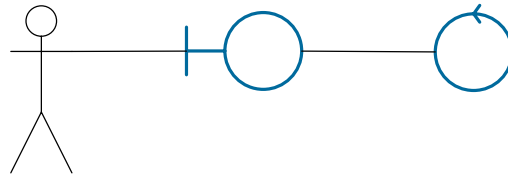


Figura 3.11 Diagrama de Análisis para el Caso de Uso Exportar Archivo de Respaldo

Diagrama de Análisis para el Caso de Uso Importar Archivo de Respaldo

El usuario inicializa el proceso haciendo la petición a la IU Importar Respaldo, indicando la ruta y nombre de archivo desde el cual se deberá importar la información, posteriormente el Gestor de Respaldo hace la solicitud de actualización al SMDB, una vez completada dicha tarea, la información del archivo estará disponible en la Base de Datos para acceso del sistema.

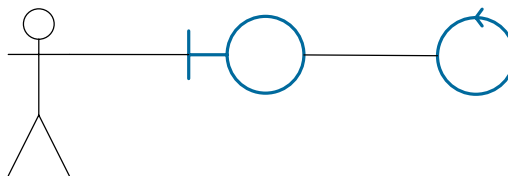


Figura 3.12 Diagrama de Análisis para el Caso de Uso Importar Archivo de Respaldo

3.6.2.3 Diagrama de Clases de Análisis para el Caso de Uso Administrar Empresas



Diagrama de Análisis para el Caso de Uso Crear Empresa

El usuario inicializa el proceso haciendo la petición e ingresando los datos a la IU Crear Empresa, el Gestor de Empresas recibe y valida la información de la nueva empresa y pasa los datos a la entidad Empresa que los almacena temporalmente para luego enviarlos al SMBD.

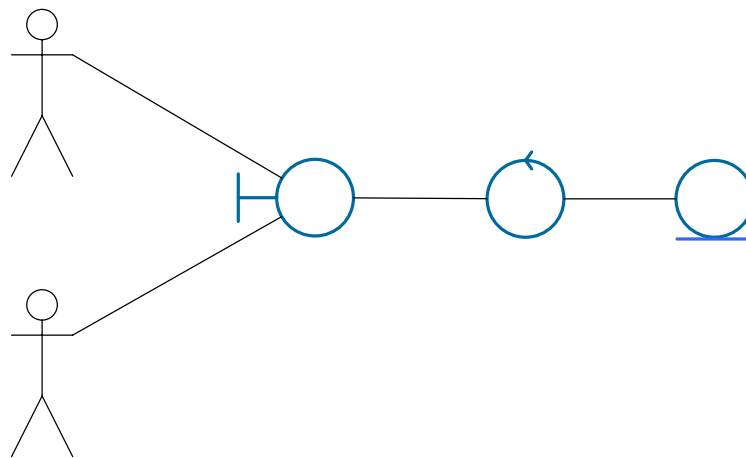


Figura 3.13 Diagrama de Análisis para el Caso de Uso Crear Empresa

Diagrama de Análisis para el Caso de Uso Modificar Empresa

El usuario inicializa el proceso haciendo la petición a la IU Modificar Empresa, el Gestor de Empresas hace la solicitud de los datos de la empresa al SMBD, estos son almacenados temporalmente por la entidad Empresa y se muestran al usuario en la IU Modificar Empresa. El usuario realiza las modificaciones a los datos mostrados y luego el Gestor de Empresa recibe y valida la información modificada de la empresa y pasa los datos a la entidad Empresa que los almacena temporalmente para luego enviarlos al SMBD.

Administrador

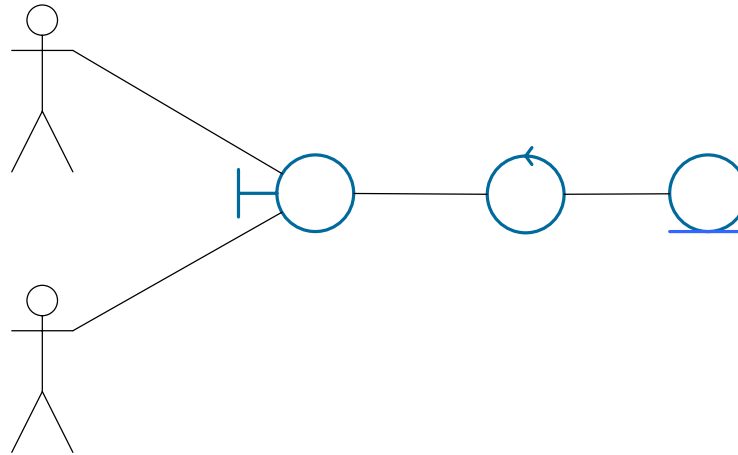


Figura 3.14 Diagrama de Análisis para el Caso de Uso Modificar Empresa

Diagrama de Análisis para el Caso de Uso Eliminar Empresa

El usuario inicializa el proceso haciendo la petición a la IU Eliminar Empresa, el Gestor de Empresas recibe y valida la información de la empresa y pasa los datos a la entidad Empresa que los almacena temporalmente para luego solicitar la eliminación al SMBD.

IU Modif
Empres

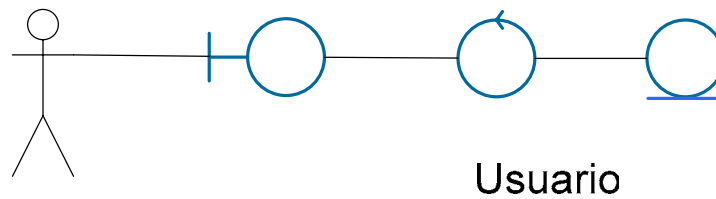


Figura 3.15 Diagrama de Análisis para el Caso de Uso Eliminar Empresa



3.6.2.4 Diagrama de Clases de Análisis para el Caso de Uso Administrar Cuentas

Diagrama de Análisis para el Caso de Uso Crear Cuenta

El usuario inicializa el proceso haciendo la petición e ingresando los datos a la IU Crear Cuenta, el Gestor de Cuentas recibe y valida la información de la nueva cuenta y pasa los datos a la entidad Cuenta que los almacena temporalmente para luego enviarlos al SMBD.

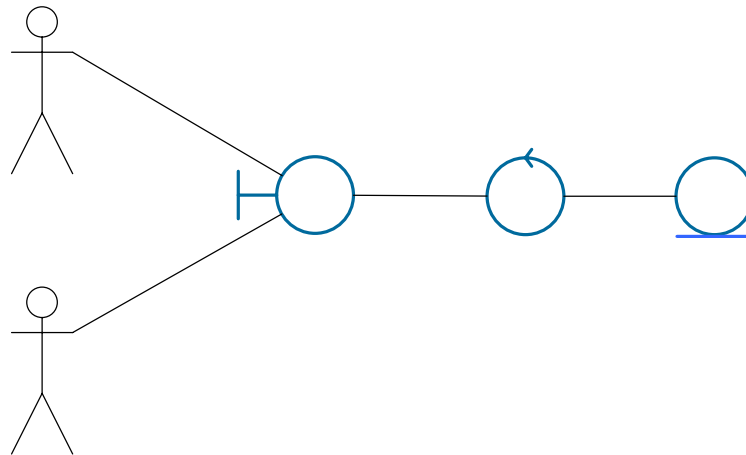


Figura 3.16 Diagrama de Análisis para el Caso de Uso Crear Cuenta

Diagrama de Análisis para el Caso de Uso Modificar Cuenta

El usuario inicializa el proceso haciendo la petición a la IU Modificar Cuenta, el Gestor de Cuentas hace la solicitud de los datos de la cuenta al SMBD, estos son almacenados temporalmente por la entidad Cuenta y se muestran al usuario en la IU Modificar Cuenta. El usuario realiza las modificaciones a los datos mostrados y luego el Gestor de Cuentas recibe y valida la información modificada de la cuenta y pasa



los datos a la entidad Cuenta que los almacena temporalmente para luego enviarlos al SMBD.

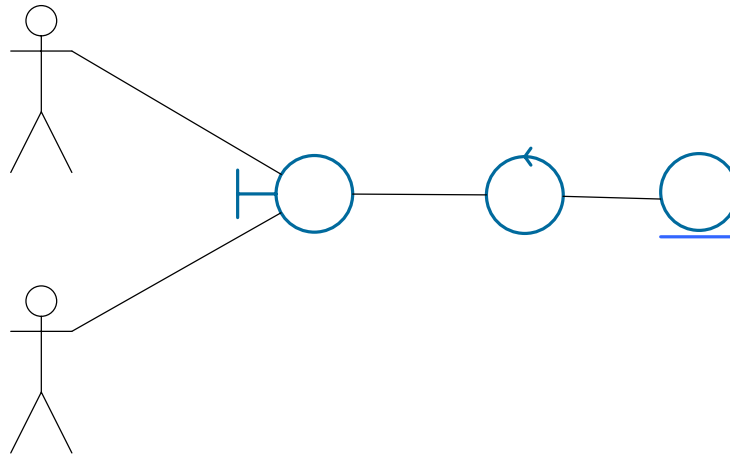


Figura 3.17 Diagrama de Análisis para el Caso de Uso Modificar Cuenta

Diagrama de Análisis para el Caso de Uso Eliminar Cuenta

El usuario inicializa el proceso haciendo la petición a la IU Eliminar Cuenta, el Gestor de Cuentas recibe y valida la información del usuario y pasa los datos a la entidad Cuenta que los almacena temporalmente para luego solicitar la eliminación al SMBD.

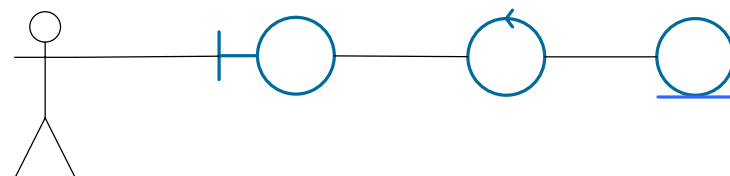


Figura 3.18 Diagrama de Análisis para el Caso de Uso Eliminar Cuenta

Usuario

IU Modif
Cuenta



3.6.2.5 Diagrama de Clases de Análisis para el Caso de Uso Manejar Asientos

Diagrama de Análisis para el Caso de Uso Crear Asiento

El usuario inicializa el proceso haciendo la petición e ingresando los datos a la IU Crear Asiento, el Gestor de Asientos recibe y valida la información del nuevo asiento y pasa los datos a la entidad Asiento que los almacena temporalmente para luego enviarlos al SMBD.

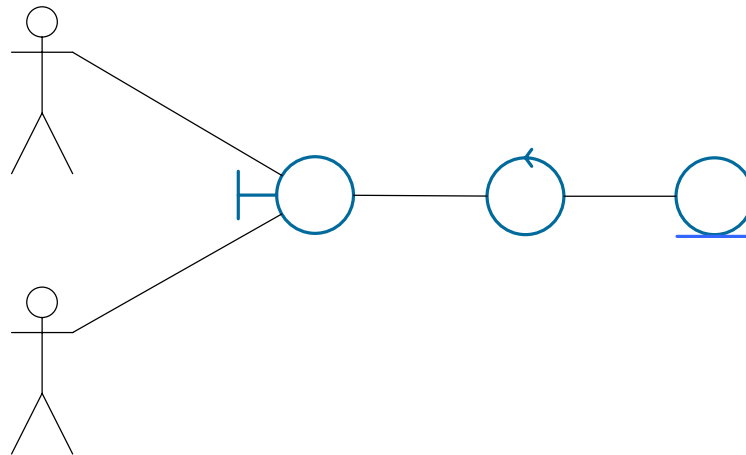


Figura 3.19 Diagrama de Análisis para el Caso de Uso Crear Asiento

Diagrama de Análisis para el Caso de Uso Modificar Asiento

El usuario inicializa el proceso haciendo la petición a la IU Modificar Asiento, el Gestor de Asientos hace la solicitud de los datos del asiento al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Modificar Asiento. El usuario realiza las modificaciones a los datos mostrados y luego el Gestor de Asientos recibe y valida la información modificada del asiento y



pasa los datos a la entidad Asiento que los almacena temporalmente para luego enviarlos al SMBD.

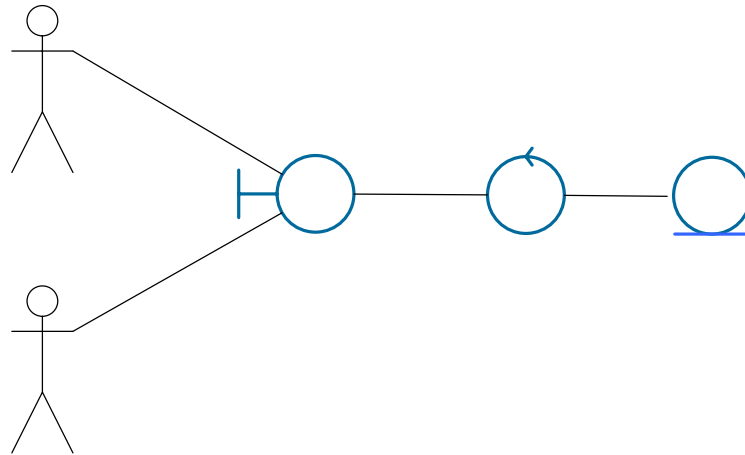


Figura 3.20 Diagrama de Análisis para el Caso de Uso Modificar Asiento

Diagrama de Análisis para el Caso de Uso Eliminar Asiento

El usuario inicializa el proceso haciendo la petición a la IU Eliminar Asiento, el Gestor de Asientos recibe y valida la información del asiento y pasa los datos a la entidad Asiento que los almacena temporalmente para luego solicitar la eliminación al SMBD.

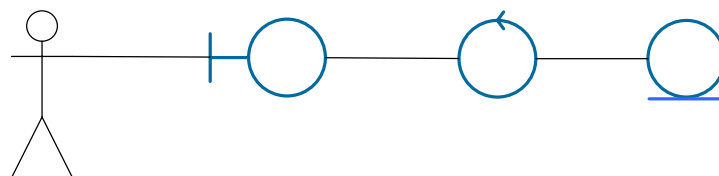


Figura 3.21 Diagrama de Análisis para el Caso de Uso Eliminar Asiento

Usuario

IU Modifi
Asiento



3.6.2.6 Diagrama de Clases de Análisis para el Caso de Uso Emitir Reportes

Diagrama de Análisis para el Caso de Uso Emitir Balance General

El usuario inicializa el proceso haciendo la petición a la IU Balance General, el Gestor de Reportes hace la solicitud de los datos de los asientos al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Balance General una vez realizados los cálculos correspondientes.

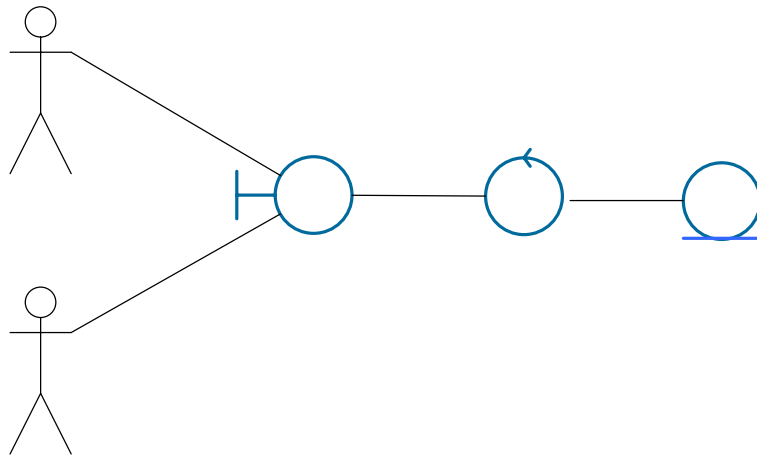


Figura 3.22 Diagrama de Análisis para el Caso de Uso Emitir Balance General

Diagrama de Análisis para el Caso de Uso Emitir Balance de Comprobación

El usuario inicializa el proceso haciendo la petición a la IU Balance de Comprobación, el Gestor de Reportes hace la solicitud de los datos de los asientos al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Balance de Comprobación una vez realizados los cálculos correspondientes.

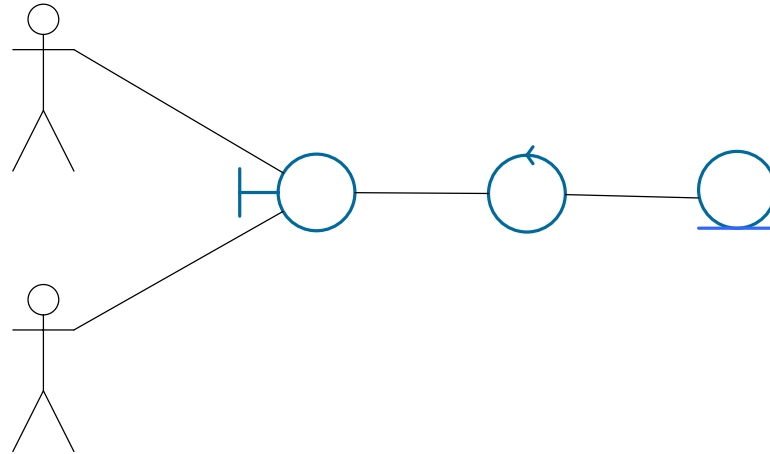


Figura 3.23 Diagrama de Análisis para el Caso de Uso Emitir Balance de Comprobación

Diagrama de Análisis para el Caso de Uso Emitir Estados de Resultado Administrador

El usuario inicializa el proceso haciendo la petición a la IU Estados de Resultado, el Gestor de Reportes hace la solicitud de los datos de los asientos al IU Balance de Comprobación de SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Estados de Resultado una vez realizados los cálculos correspondientes.

Usuario

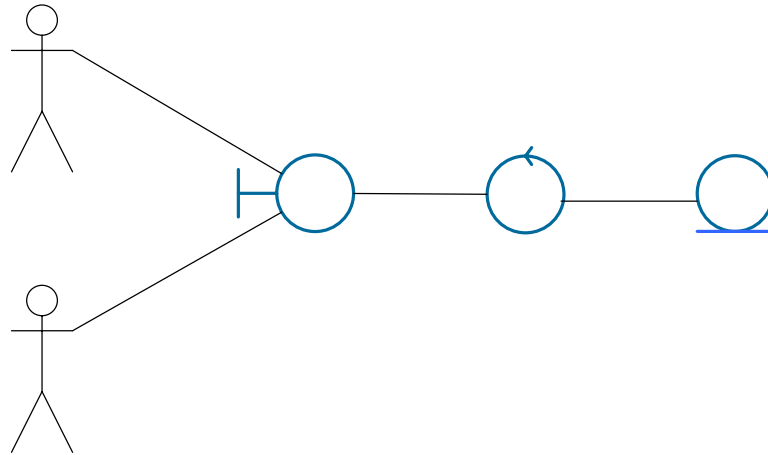


Figura 3.24 Diagrama de Análisis para el Caso de Uso Emitir Estados de Resultados

Diagrama de Análisis para el Caso de Uso Administrador General

El usuario inicializa el proceso haciendo la petición a la IU Libro Diario General, el Gestor de Reportes hace la solicitud de los datos de los asientos al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Libro Diario General una vez realizados los cálculos correspondientes.

IU Estado de Resultados

Usuario

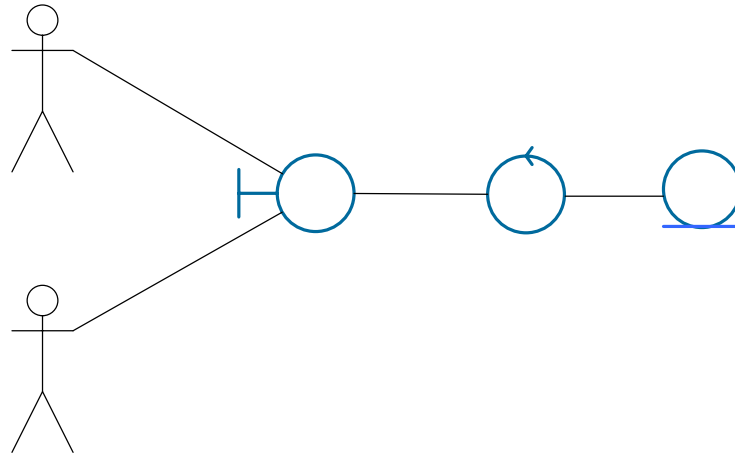


Figura 3.25 Diagrama de Análisis para el Caso de Uso Emitir Libro Diario General

Diagrama de Análisis para el Caso de Uso Emitir Libro Diario General

El usuario inicializa el proceso haciendo la petición a la IU Libro Mayor, el Gestor de Reportes hace la solicitud de los datos de los asientos al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Libro Mayor una vez realizados los cálculos correspondientes.

IU Libro Mayor
Diario General

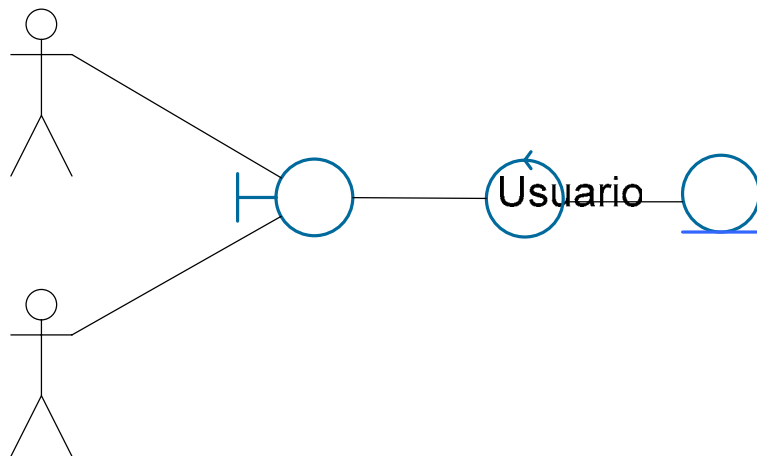




Figura 3.26 Diagrama de Análisis para el Caso de Uso Emitir Libro Mayor

Diagrama de Análisis para el Caso de Uso Emitir Mayor Analítico

El usuario inicializa el proceso haciendo la petición a la IU Mayor Analítico, el Gestor de Reportes hace la solicitud de los datos de los asientos al SMBD, estos son almacenados temporalmente por la entidad Asiento y se muestran al usuario en la IU Mayor Analítico una vez realizados los cálculos correspondientes.

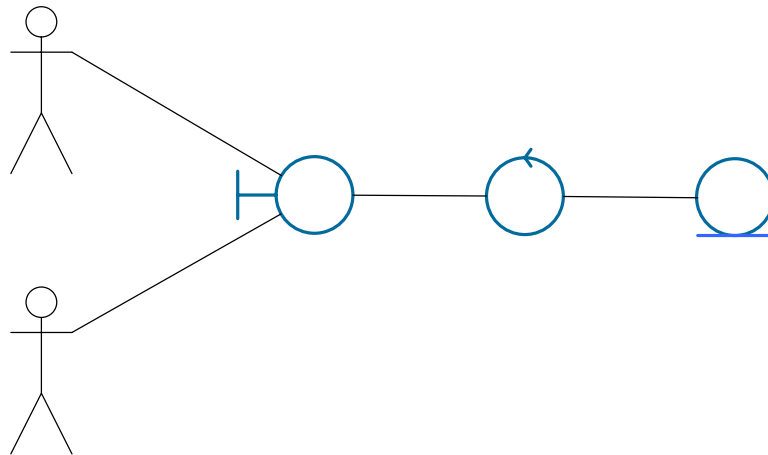


Figura 3.27 Diagrama de Análisis para el Caso de Uso Emitir Mayor Analítico

3.6.3 Paquetes del Análisis

Los paquetes de análisis son usados como medio para organizar los artefactos del modelo de análisis en partes manejables, pueden constar de clases del análisis, de realizaciones de casos de uso y de otros paquetes del análisis. Una forma directa de identificar los paquetes del análisis es asignar la mayor parte de un cierto número de



casos de uso a un paquete concreto y después realizar la funcionalidad correspondiente dentro de ese paquete.

Los paquetes de análisis deben ser altamente cohesivos, lo que significa que los elementos que agrupan deben cumplir funciones relacionadas. Además, los paquetes deben ser débilmente acoplados, limitando las dependencias entre éstos. Estas dos características permiten que los paquetes puedan ser desarrollados en forma separada y además puedan distribuirse como módulos de funcionalidad independiente.

A continuación se muestran los paquetes de análisis.

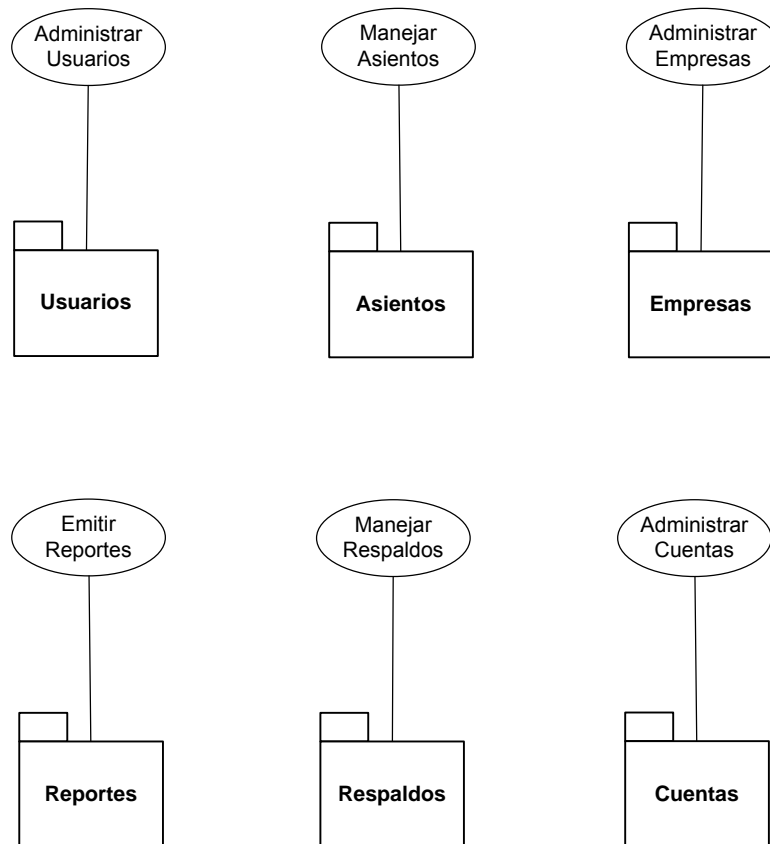


Figura 3.28 Paquetes de Análisis



3.7 EVALUACION DE LA FASE DE INICIO

En el transcurso de la fase de inicio se logró obtener una visión global del sistema propuesto a través del modelo de dominio, y así establecer lo que debe cubrir la arquitectura. Además se identificaron en detalle los riesgos críticos del sistema y se determinaron acciones para disminuirlos.

Adicionalmente se realizó el análisis de los diferentes casos de uso identificados en esta fase, obteniendo también una visión sólida de los paquetes de análisis que conformarán el sistema, teniendo la base de la arquitectura ya estructurada para la siguiente fase.

Como conclusión de esta fase, se tiene una primera versión del modelo de casos de uso, el modelo de análisis y el modelo de diseño los cuales se trabajarán con más detalle en la fase de elaboración que es la fase siguiente en este proceso.



CAPÍTULO IV. FASE DE ELABORACIÓN

4.1 INTRODUCCIÓN

Este capítulo tiene como objetivo recopilar la mayor parte de los requisitos que aun quedan pendientes, alcanzando así la línea base de la arquitectura. En esta fase se refinaran y transformaran los modelos de la fase de inicio en otra serie de modelos que vayan perfeccionando una solución más cercana al mundo real.

Al concluir esta fase se tiene la información necesaria de todos los requisitos funcionales y la elaboración de un modelo de análisis y de diseño, lo suficientemente sólidos como para llevar a cabo la construcción de la arquitectura base del sistema.

4.2 PLANIFICACIÓN DE LA FASE DE ELABORACIÓN

En esta fase se desarrollaron los flujos de trabajos Requisitos, Análisis, Diseño e Implementación, tomando como guía los modelos de casos de uso y de análisis, construidos en la fase de inicio.

En el flujo de trabajo Requisitos, se buscaron nuevos requerimientos, se evaluaron y refinaron los requisitos de la fase anterior, y se adaptó el modelo de casos de uso, considerando la aparición de nuevos casos de uso.



En el flujo de trabajo de Diseño, se refinó la arquitectura que sustentará al sistema y un diseño conceptual bastante sólido para proceder a la codificación del software.

Finalmente se llevará a cabo el flujo de trabajo de Implementación, donde se procede a transformar un modelo de diseño en código ejecutable.

4.3 FLUJO DE TRABAJO DE REQUISITOS

La captura de requisitos en la fase de elaboración se lleva a cabo con la finalidad de refinar los ya obtenidos y contemplar nuevos requisitos que fueron pasados por alto en la fase de inicio, sin embargo, en esta fase no se identificaron nuevos requisitos debido a que los requerimientos necesarios fueron identificados en la fase de inicio.

4.3.1 Modelo de Caso de Uso

Debido a que no se encontraron nuevos requisitos, tampoco existirán nuevos casos de usos ni actores, por lo que se seguirá la misma ruta de análisis iniciada en la fase anterior.



4.4 FLUJO DE TRABAJO DE ANÁLISIS

En la fase de inicio se construyó un bosquejo de la arquitectura, en esta fase se extiende el análisis hasta el punto en que pueda construirse la línea principal de la arquitectura ejecutable.

4.4.1 Identificación de Clases de Análisis

Las clases de análisis de los casos de uso identificados en la fase anterior no presentaron ningún cambio en esta fase.

4.4.2 Diagramas de Colaboración

Las interacciones de las clases de los diferentes casos de uso, desarrollados en la fase anterior, se representarán a través de diagramas de colaboración para ilustrar su ejecución y observar adecuadamente la interacción de un objeto con respecto de los demás.

4.4.2.1 Diagrama de Colaboración para el Caso de Uso Administrar Usuarios

Diagrama de Colaboración para el Caso de Uso Crear Usuario

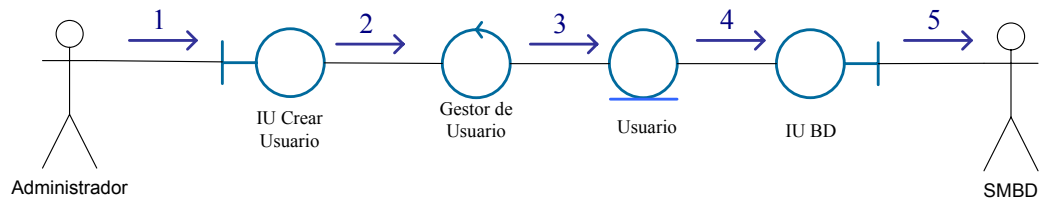


Figura 4.1 Diagrama de Colaboración para el Caso de Uso Crear Usuario

- Ingresa los datos del nuevo usuario.
- Envía la información.
- Valida y envía la información.
- Almacena temporalmente y envía para agregación.
- Solicita el registro de la información especificada.

Diagrama de Colaboración para el Caso de Uso Eliminar Usuario

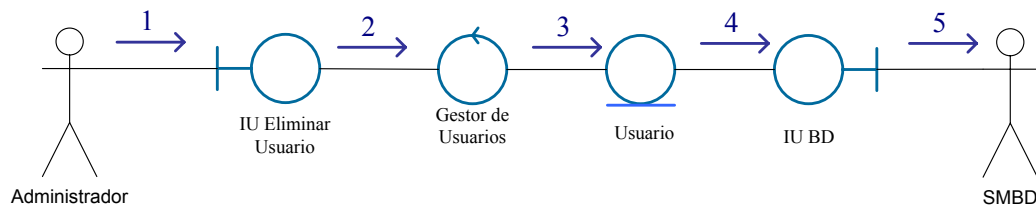


Figura 4.2 Diagrama de Colaboración para el Caso de Uso Eliminar Usuario

- Selecciona el usuario a eliminar.
- Envía la información.
- Valida y envía la información.
- Almacena temporalmente y envía para eliminación.
- Solicita la eliminación de la información especificada.



4.4.2.2 Diagrama de Colaboración para el Caso de Uso Manejar Respaldos

Diagrama de Colaboración para el Caso de Uso Exportar Archivo de Respaldo

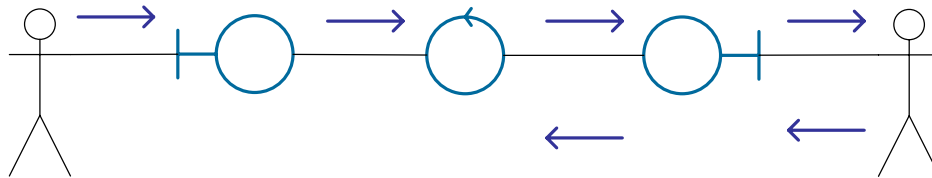


Figura 4.3 Diagrama de Colaboración para el Caso de Uso Exportar Archivo de Respaldo

- Solicita creación de respaldo, especificando el destino.
- Envía la solicitud y la información de destino.
- Solicita la data que se desea exportar.
- Realiza la petición al SMBD de los registros contenedores de la data deseada. **1**
- Retorna los registros contenedores de la data deseada. **2**
- Entrega la data al gestor para su exportación.

Diagrama de Colaboración para el Caso de Uso Importar Archivo de Respaldo

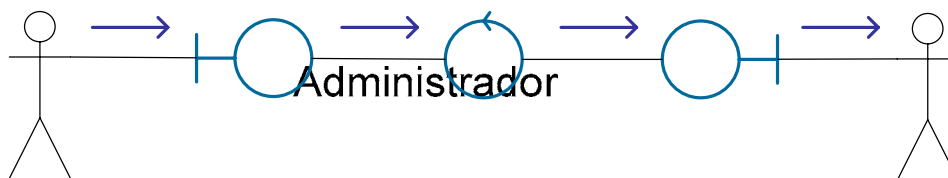


Figura 4.4 Diagrama de Colaboración para el Caso de Uso Importar Archivo de Respaldo

- Solicita recuperación desde respaldo, especificando el origen.

Gesto
Respa



- Envía la solicitud y la información de origen.
- Envía la data que se desea importar.
- Realiza la petición al SMBD de la actualización de los registros contenedores de la data.

4.4.2.3 Diagrama de Colaboración para el Caso de Uso Administrar Empresas

Diagrama de Colaboración para el Caso de Uso Crear Empresa

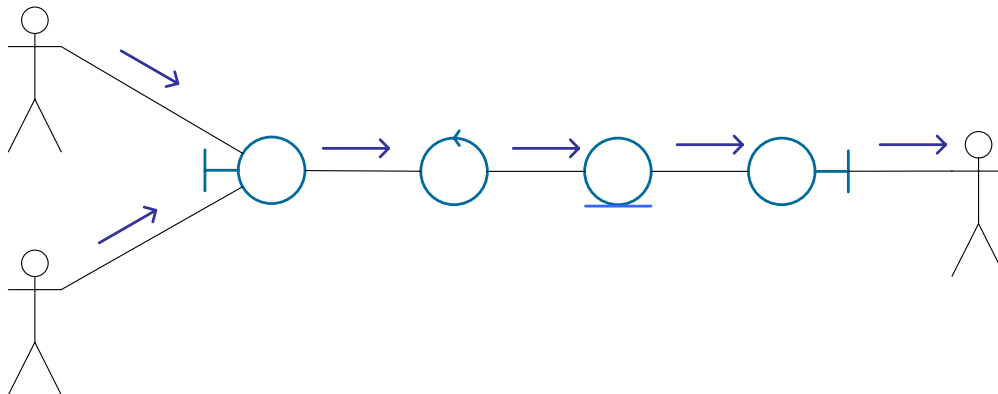


Figura 4.5 Diagrama de Colaboración para el Caso de Uso Crear Empresa

- Ingresar los datos de la nueva empresa.
- Enviar la información.
- Validar y enviar la información.
- Almacenar temporalmente y enviar para agregación.
- Solicitar el registro de la información especificada.



Diagrama de Colaboración para el Caso de Uso Modificar Empresa

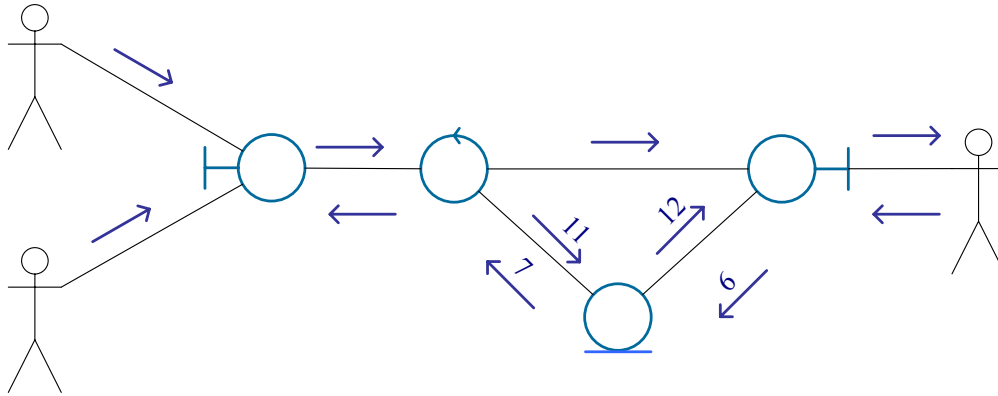


Figura 4.6 Diagrama de Colaboración para el Caso de Uso Modificar Empresa

1,9

- Selecciona la empresa a modificar.
- Envía el RIF de la empresa que se desea modificar.
- Solicita la información de la empresa que se desea modificar.
- Hace la solicitud de Administrador contenedores de la data de la empresa.
- Retorna los registros contenedores de la data de la empresa.
- Envía la información para su almacenaje temporal en memoria.
- Envía la información ya organizada.
- Envía la información de la empresa a la interfaz.
- Se especifican las modificaciones a realizar en la empresa.
- Se envían las modificaciones al gestor.
- La información se valida y envía.
- Se envía la información con las modificaciones realizadas.
- Se solicita al SMBD la modificación de los registros necesarios.

2,10

8

Gestor de Empresa

Administrador

1,9

UI Modificar Empresa

Usuario



Diagrama de Colaboración para el Caso de Uso Eliminar Empresa

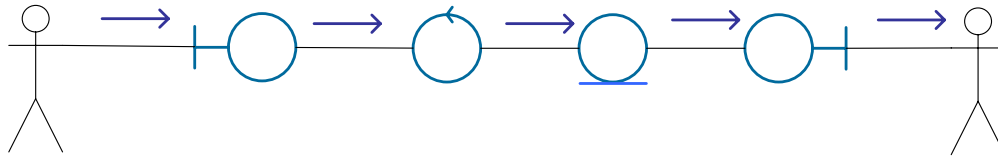


Figura 4.7 Diagrama de Colaboración para el Caso de Uso Eliminar Empresa

- Selecciona la empresa a eliminar.
- Envía la información.
- Valida y envía la información. **1**
- Almacena temporalmente y envía para eliminación.
- Solicita la eliminación de la información especificada.

2

IU Eliminar
Empresa

Gestor d
Empresa

4.4.2.4 Diagrama de Colaboración para el Caso de Uso Administrar Cuentas

Administrador

Diagrama de Colaboración para el Caso de Uso Crear Cuenta

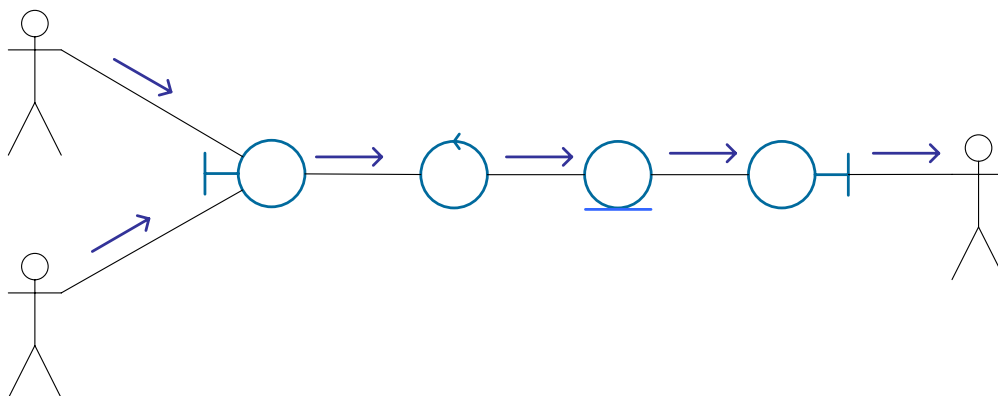


Figura 4.8 Diagrama de Colaboración para el Caso de Uso Crear Cuenta



- Ingresa los datos de la nueva cuenta.
- Envía la información.
- Valida y envía la información.
- Almacena temporalmente y envía para agregación.
- Solicita el registro de la información especificada.

Diagrama de Colaboración para el Caso de Uso Modificar Cuenta

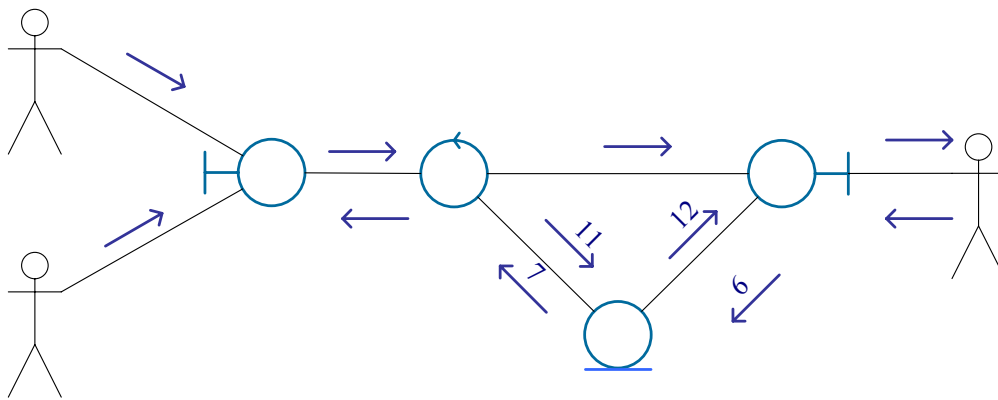


Figura 4.9 Diagrama de Colaboración para el Caso de Uso Modificar Cuenta

- Selecciona la cuenta a modificar.
- Envía el código de la cuenta que se desea modificar.
- Solicita la información de la cuenta que se desea modificar.
- Hace la solicitud de los registros contenedores de la data de la cuenta.
- Retorna los registros contenedores de la data de la cuenta.
- Envía la información para su almacenaje temporal en memoria.
- Envía la información ya organizada.
- Envía la información de la cuenta a la interfaz.
- Se especifican las modificaciones a realizar en la cuenta.
- Se envían las modificaciones al administrador.

2,10

Administrador

1,9

8

IU Modificar
Cuenta

Gestor de
Cuenta



- La información se valida y envía.
- Se envía la información con las modificaciones realizadas.
- Se solicita al SMBD la modificación de los registros necesarios.

Diagrama de Colaboración para el Caso de Uso Eliminar Cuenta

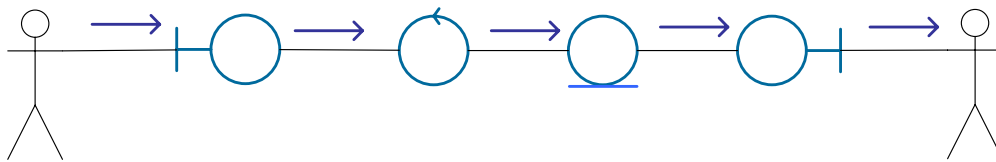


Figura 4.10 Diagrama de Colaboración para el Caso de Uso Eliminar Cuenta

- Selecciona la cuenta a eliminar.
- Envía la información.
- Valida y envía la información.
- Almacena temporalmente y envía para eliminación.
- Solicita la eliminación de la información especificada.

1

2

4.4.2.5 Diagrama de Colaboración para el Caso de Uso Manejar Asientos

Diagrama de Colaboración para el Caso de Uso Crear Asiento

III Eliminar
Cuenta

Gestor de
Cuentas

Administrador

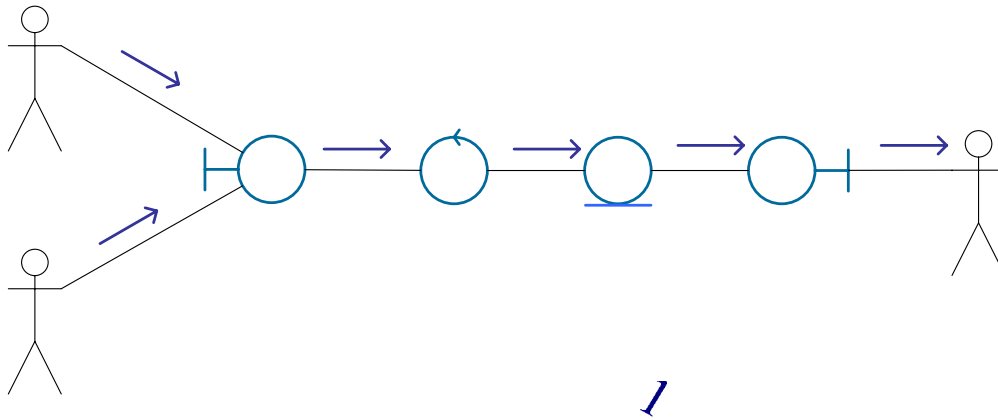


Figura 4.11 Diagrama de Colaboración para el Caso de Uso Crear Asiento

- Ingresar los datos del nuevo asiento.
- Envía la información al Administrador
- Valida y envía la información.
- Almacena temporalmente y envía para agregación.
- Solicita el registro de la información especificada.

2

IU Crear Asiento

Gestor de Asientos

Diagrama de Colaboración para el Caso de Uso Modificar Asiento

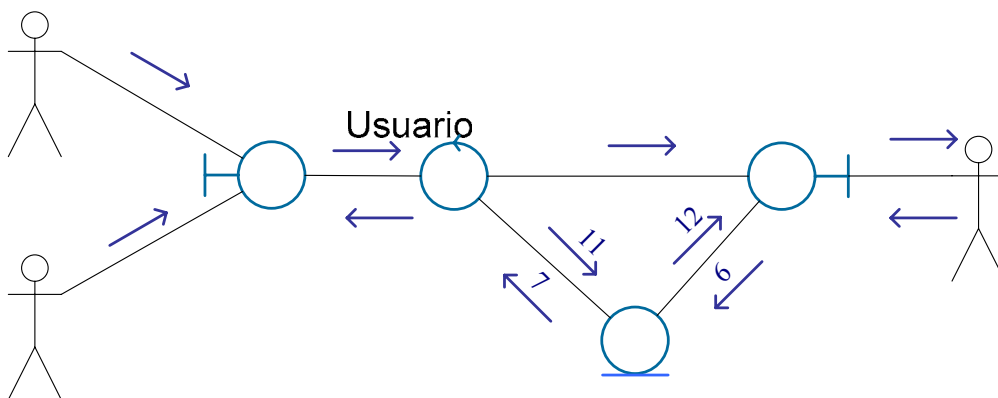


Figura 4.12 Diagrama de Colaboración para el Caso de Uso Modificar Asiento



- Selecciona el asiento a modificar.
- Envía el código del asiento que se desea modificar.
- Solicita la información del asiento que se desea modificar.
- Hace la solicitud de los registros contenedores de la data del asiento.
- Retorna los registros contenedores de la data del asiento.
- Envía la información para su almacenaje temporal en memoria.
- Envía la información ya organizada.
- Envía la información del asiento a la interfaz.
- Se especifican las modificaciones a realizar en el asiento.
- Se envían las modificaciones al gestor.
- La información se valida y envía.
- Se envía la información con las modificaciones realizadas.
- Se solicita al SMBD la modificación de los registros necesarios.

Diagrama de Colaboración para el Caso de Uso Eliminar Asiento

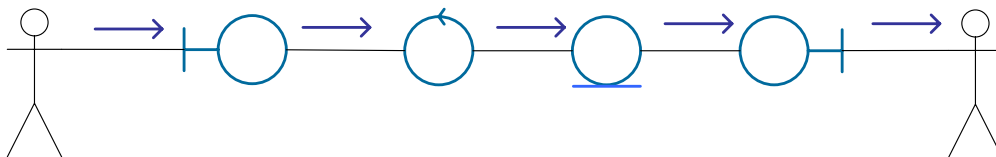


Figura 4.13 Diagrama de Colaboración para el Caso de Uso Eliminar Asiento

- Selecciona el asiento a eliminar.
- Envía la información.
- Valida y envía la información.
- Almacena temporalmente y envía para eliminación.
- Solicita la eliminación de la información especificada.



4.4.2.6 Diagrama de Colaboración para el Caso de Uso Emitir Reportes

Diagrama de Colaboración para el Caso de Uso Emitir Balance General

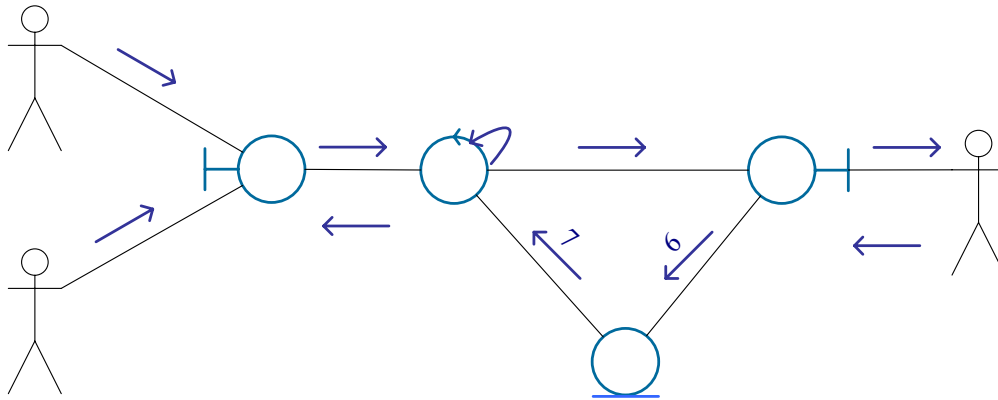


Figura 4.14 Diagrama de Colaboración para el Caso de Uso Emitir Balance General

- Se solicita el balance general especificando el intervalo de tiempo.
- Se realiza la solicitud al gestor. 1
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros contenedores de la data de los asientos necesarios. 2
- Retorna los registros contenedores de la data de los asientos necesarios.
- La información de Administrador enviada para su almacenamiento temporal en memoria. 9
- Los asientos solicitados se envían al gestor. 1
- El gestor realiza los cálculos necesarios para la presentación del balance general.
- La información del balance general es enviada a la interfaz para que sea vista por el usuario.

Usuario

Gestor
Reporte



Diagrama de Colaboración para el Caso de Uso Emitir Balance de Comprobación

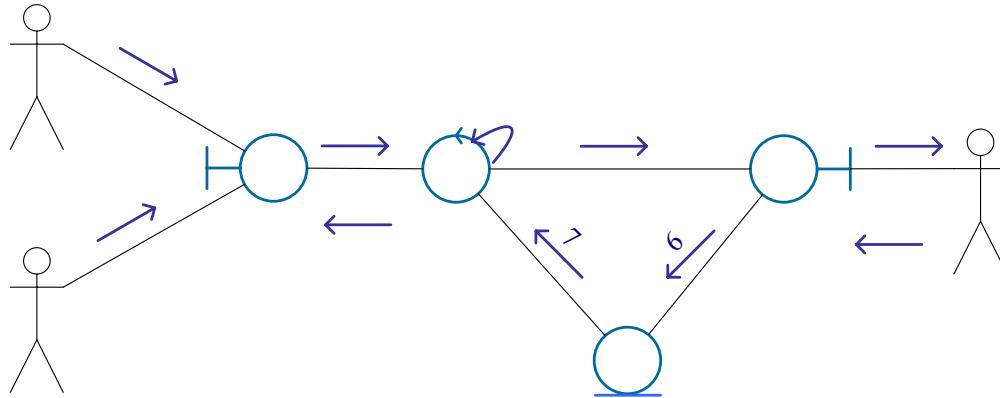


Figura 4.15 Diagrama de Colaboración para el Caso de Uso Emitir Balance de Comprobación

- Se solicita el balance de comprobación especificando el intervalo de tiempo.
- Se realiza la solicitud al gestor.
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros contenedores de la data de los asientos necesarios.
- Retorna los registros de la data de los asientos necesarios.
- La información de los asientos es enviada para su almacenamiento temporal en memoria.
- Los asientos solicitados se envían al gestor.
- El gestor realiza los cálculos necesarios para la presentación del balance de comprobación.
- La información del balance de comprobación es enviada a la interfaz para que sea vista por el usuario.

Usuario

Administrador

IU Balance
De Comprobación

Gestor
Reporte



Diagrama de Colaboración para el Caso de Uso Emitir Estados de Resultado

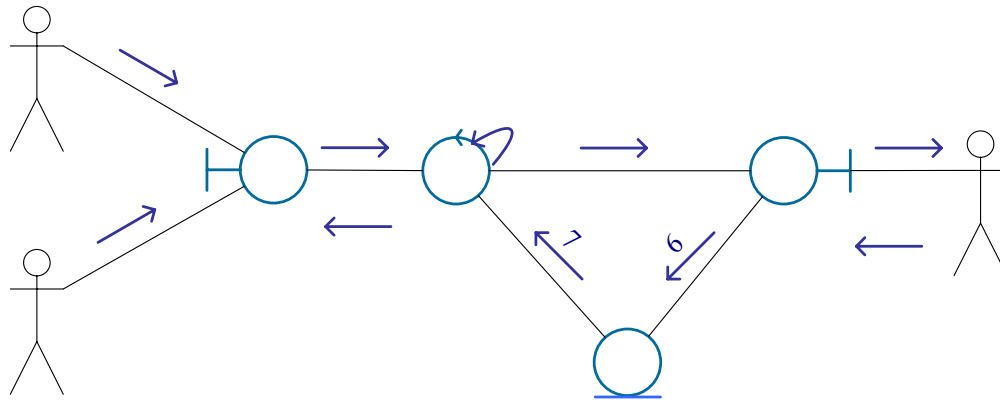


Figura 4.16 Diagrama de Colaboración para el Caso de Uso Emitir Estados de Resultados

- Se solicita el estado de resultados especificando el intervalo de tiempo.
- Se realiza la solicitud al gestor.
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros de la data de los asientos necesarios.
- Retorna los registros contenedores de la data de los asientos necesarios.
- La información de los asientos es enviada para su almacenamiento temporal en memoria.
- Los asientos solicitados se envían al gestor.
- El gestor realiza los cálculos necesarios para la presentación estado de resultados.
- La información del estado de resultados es enviada a la interfaz para que sea vista por el usuario.

Administrador

Estado de Resultados

Usuario

2

9

Gestor de Reportes

Diagrama de Colaboración para el Caso de Uso Emitir Libro Diario General

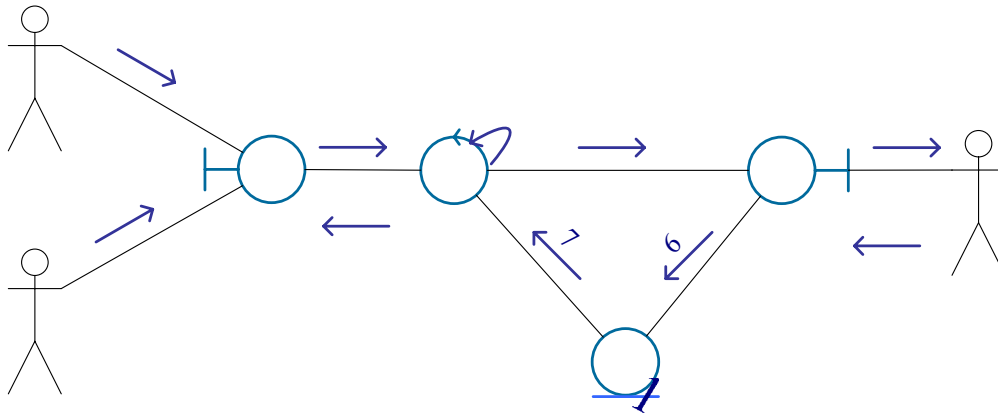


Figura 4.17 Diagrama de Colaboración para el Caso de Uso Emitir Libro Diario General

- Se solicita el Libro Diario General especificando el intervalo de tiempo. 2
- Se realiza la solicitud. 9
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros contenedores de la data de los asientos necesarios.
- Retorna los registros contenedores de la data de los asientos necesarios.
- La información de los asientos es enviada para su almacenamiento temporal en memoria.
- Los asientos solicitados se envían al gestor.
- El gestor realiza los cálculos necesarios para la presentación del Libro Diario General.
- La información del Libro Diario General es enviada a la interfaz para que sea vista por el usuario.

Gestor de
Reportes



Diagrama de Colaboración para el Caso de Uso Emitir Libro Mayor

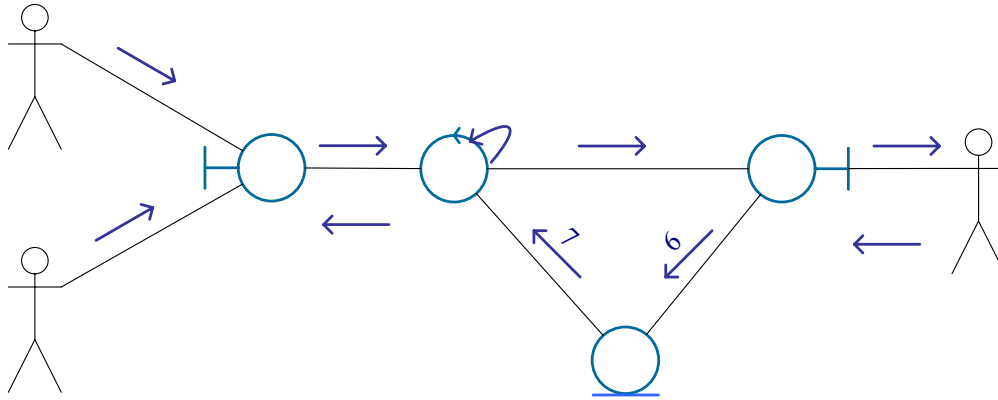


Figura 4.18 Diagrama de Colaboración para el Caso de Uso Emitir Libro Mayor

- Se solicita el Libro Mayor especificando el intervalo de tiempo.
- Se realiza la solicitud al gestor.
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros contenedores de la data de los asientos necesarios.
- Retorna los registros contenedores de la data de los asientos necesarios.
- La información de los asientos es enviada para su almacenamiento temporal en memoria.
- Los asientos solicitados se envían al gestor.
- El gestor realiza los cálculos necesarios para la presentación del Libro Mayor.
- La información del Libro Mayor es enviada a la interfaz para que sea vista por el usuario.

Usuario

Administrador

Gestor
Libro Mayor

Gestor
Reporte



Diagrama de Colaboración para el Caso de Uso Emitir Mayor Analítico

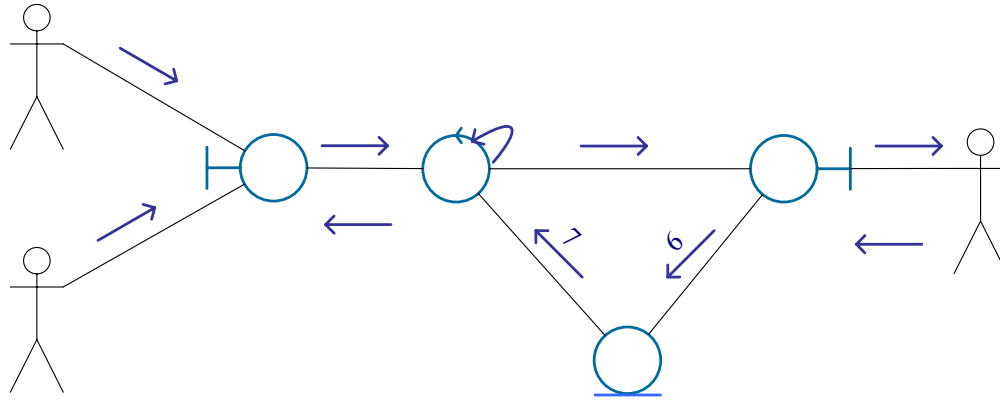


Figura 4.19 Diagrama de Colaboración para el Caso de Uso Emitir Mayor Analítico

- Se solicita el Mayor Analítico especificando el intervalo de tiempo.
- Se realiza la solicitud al gestor. 2
- Se solicitan los asientos ubicados en el intervalo antes indicado.
- Se solicitan los registros contenedores de la data de los asientos necesarios.
- Retorna los registros contenedores de la data de los asientos necesarios. 9
- La información de los asientos es enviada para su almacenamiento temporal en memoria.
- Los asientos solicitados se envían al gestor.
- El gestor realiza los cálculos necesarios para la presentación del Mayor Analítico.
- La información del Mayor Analítico es enviada a la interfaz para que sea vista por el usuario.

Usuario

Administrador

Gestor de Mayor Analítico

Gestor de Reporte



4.5 FLUJO DE TRABAJO DE DISEÑO

En el flujo de trabajo de diseño se modelará el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones, que se le suponen. Aquí se toma como entrada el análisis de los requisitos previamente realizado. En el diseño se realizarán actividades delicadas y cuidadosas que darán como resultado los componentes que se implementarán en la siguiente fase. Entre ellos tenemos, el diseño de la base de datos y conclusión del diseño de la arquitectura.

4.5.1 Clases de Diseño del software J-Contab

En este apartado presentaremos las clases de diseño que conformarán las diferentes construcciones del sistema J-Contab, para esto empezaremos realizando un mapeo de las Clases de Análisis obtenidas en la fase anterior con las Clases de Diseño que tendremos en el software.

Como podremos observar a continuación casi todas las clases de análisis tienen un equivalente con su clase de diseño, a excepción de ciertas clases de análisis cuya funcionalidad era integrable en alguna otra clase que intervenga en el mismo proceso. El mapeo de clases está representado en la siguiente tabla:



Tabla 4.1 Clases de Diseño. 1/2

Clase de Análisis	Clase de Diseño
IU Crear Usuario	DialogoNuevoUsuario
IU Eliminar Usuario	DialogoAdministrarUsuario
IU Exportar Respaldo	DialogoExportarData
IU Importar Respaldo	DialogoImportarData
IU Crear Empresa	DialogoNuevaEmpresa
IU Modificar Empresa	DialogoModificarEmpresa
IU Eliminar Empresa	DialogoAdministrarEmpresa
IU Crear Cuenta	DialogoNuevaCuenta
IU Modificar Cuenta	DialogoModificarCuenta
IU Eliminar Cuenta	DialogoAdministrarCuenta
IU Crear Asiento	DialogoNuevoAsiento
IU Modificar Asiento	DialogoModificarAsiento
IU Eliminar Asiento	JPanelPrincipalAsiento
IU Balance General	JPanelBalanceGeneral
IU Balance de Comprobación	JPanelBalanceDeComprobacion
IU Estado de Resultados	JPanelEstadoDeResultados
IU Libro Diario General	JPanelLibroDiarioGeneral
IU Libro Mayor	JPanelLibroMayor
IU Mayor Analítico	JPanelMayorAnalitico
IU BDD	ConectorSql
Gestor de Usuarios	DialogoAdministrarUsuarios
Gestor de Respaldos	DialogoRespaldos
Gestor de Empresas	DialogoAdministrarEmpresas
Gestor de Cuentas	DialogoAdministrarCuentas
Gestor de Asientos	DialogoAdministrarAsientos



Tabla 4.1 Clases de Diseño. 2/2

Clase de Análisis	Clase de Diseño
Gestor de Reportes	GestorDeReportes
Usuario	NodoUsuario
Empresa	NodoEmpresa
Cuenta	NodoCuenta
Asiento	NodoAsiento

4.5.1.1 Diagrama de clases del software J-Contab

El diagrama de clases describe la estructura del sistema mostrando sus clases y las relaciones entre ellas.

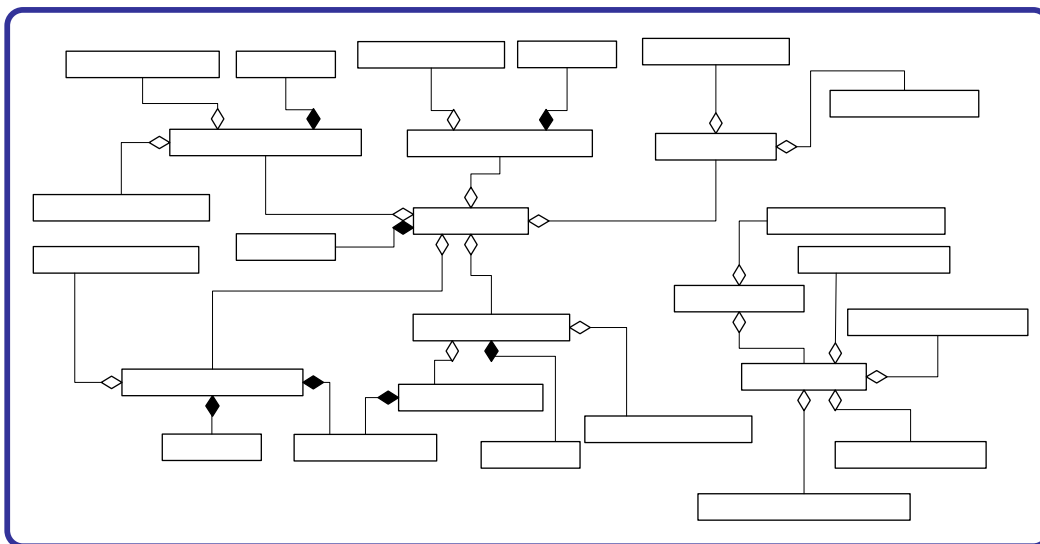


Figura 4.20: Diagrama de clases del software J-Contab.



4.5.2 Diagramas de secuencia

El Diagrama de secuencia muestra cronológicamente el paso de mensajes entre los objetos de las clases de diseño, se observan las instancias de los actores y la interacción de éstos con los objetos de diseño.

4.5.2.1 Diagrama de Secuencia para el Caso de Uso Administrar Usuarios

Diagrama de Secuencia para el Caso de Uso Crear Usuario

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoNuevoUsuario, luego de introducir y confirmar la información esta clase solicita el método CrearUsuario(Información) de la clase DialogoAdministrarUsuarios, a este se le envía la información introducida por el Administrador. Posteriormente esta clase valida la información y solicita el método NuevoUsuario(Información) de la clase NodoUsuario, al cual se le envía la información ya validada, acto seguido esta clase invocara al método CrearUsuario(NodoUsuario) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

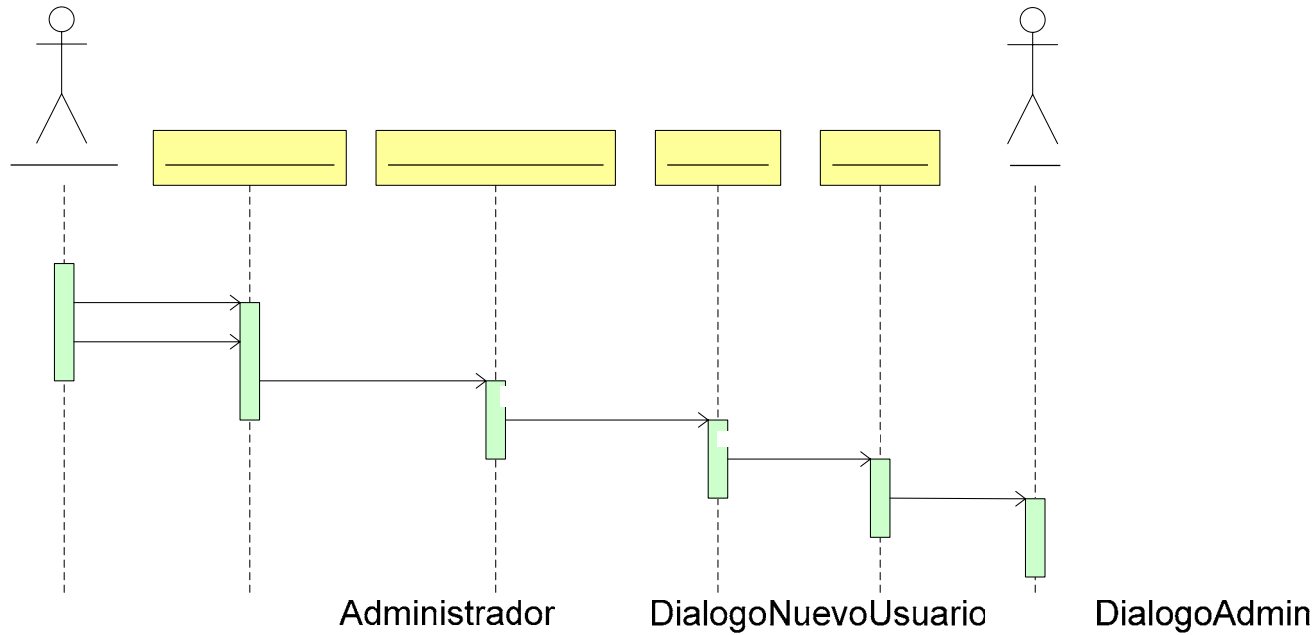


Figura 4.21 Diagrama de Secuencia para el Caso de Uso Crear Usuario

Diagrama de Secuencia para el Caso de Uso Eliminar Usuario

El Administrador inicia el caso de uso invocando el método `Cargar()` de la clase `DialogoAdministrarUsuario`, luego de introducir y confirmar la información esta clase valida dicha información y solicita el método `EliminarUsuario(Usuario)` de la clase `NodoUsuario`, al cual se le envía la información ya validada, acto seguido esta clase invocara al método `EliminarUsuario(NodoUsuario)` de la clase `ConectorSql`, quien realizará la(s) consulta(s) al SMBD.

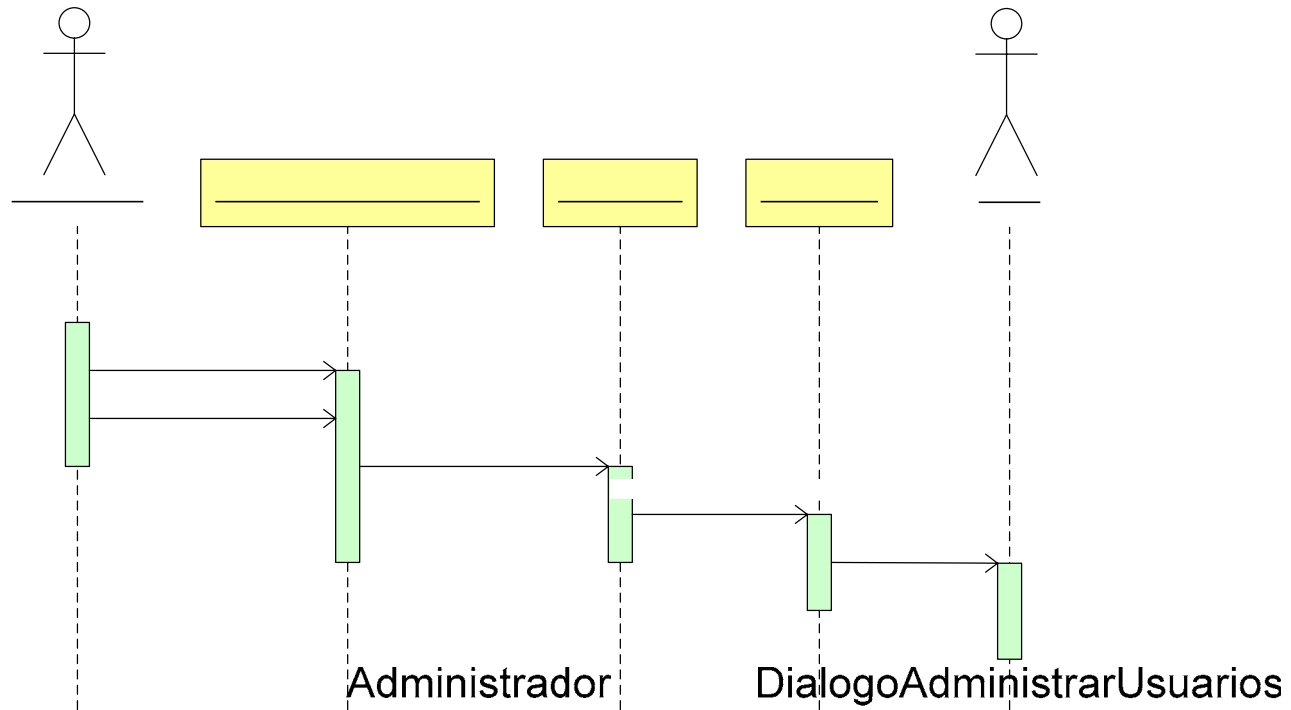


Figura 4.22 Diagrama de Secuencia para el Caso de Uso Eliminar Usuario

Cargar()

4.5.2.2 Diagrama de Secuencia para el Caso de Uso Manejar Respaldo

Seleccionar Usuario a Eliminar

Diagrama de Secuencia para el Caso de Uso Exportar Archivo de Respaldo

EliminarUsua

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoExportarData, luego de introducir y confirmar la información esta clase solicita el método ExportarData(Empresa) de la clase DialogoRespaldos, a este se le envía la información introducida por el Administrador. La clase DialgoRespaldos valida la información y solicita el método SolicitarData(Empresa) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD. El SMBD entrega a la clase



ConectorSql los datos de la(s) consulta(s) realizada y esta la envía a la clase DialogoRespaldo para su almacenamiento en un archivo de texto.

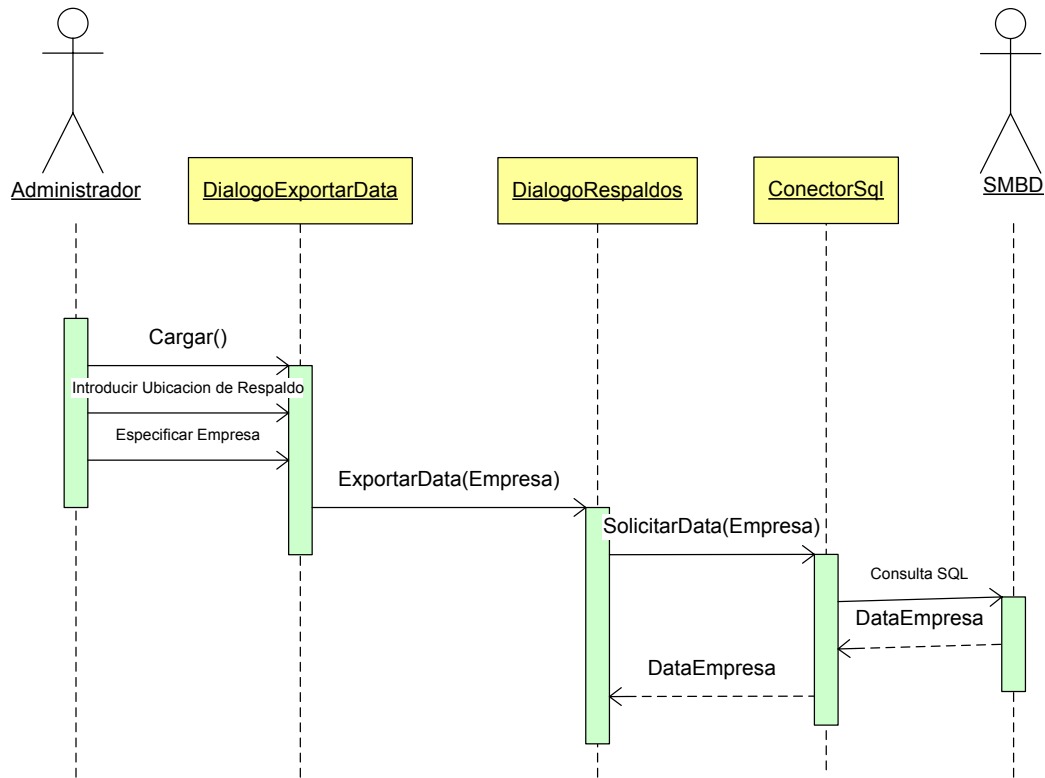


Figura 4.23 Diagrama de Secuencia para el Caso de Uso Exportar Archivo de Respaldo

Diagrama de Secuencia para el Caso de Uso Importar Archivo de Respaldo

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoImportarData, luego de indicar la ruta al archivo que contiene el respaldo y confirmar la información esta clase solicita el método ImportarData(UbicRespaldo) de la clase DialogoRespaldos, a este se le envía la información introducida por el Administrador. La clase DialogoRespaldos valida la información y solicita el método



SolicitarActualizacionData(UbicRespaldo) de la clase ConectorSql, quien realizará la(s) petición de actualización(es) al SMBD.

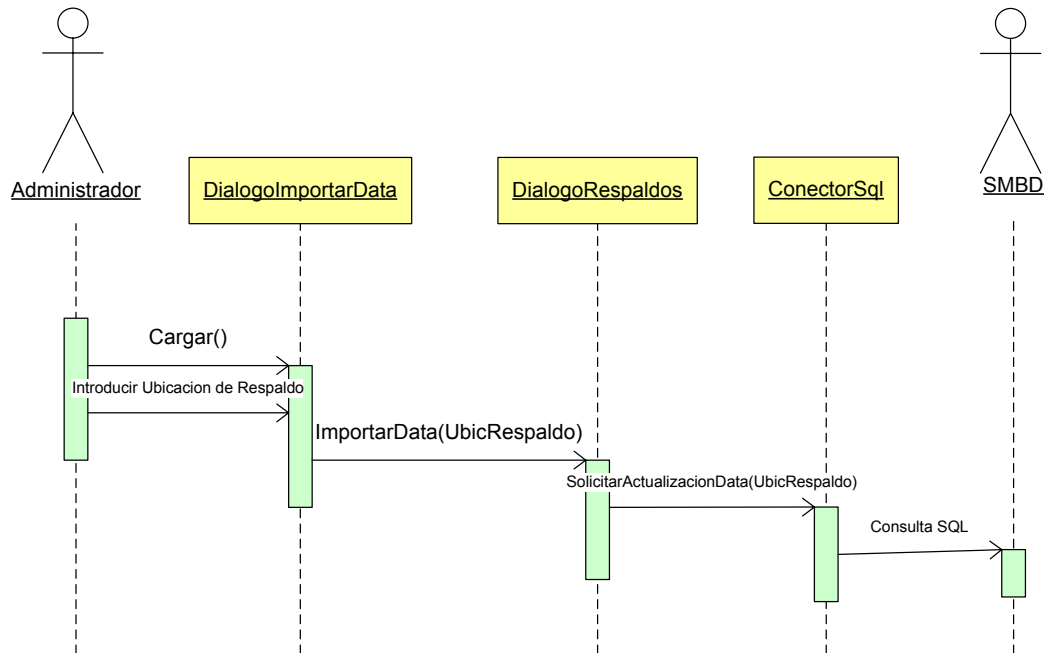


Figura 4.24 Diagrama de Secuencia para el Caso de Uso Importar Archivo de Respaldo

4.5.2.3 Diagrama de Secuencia para el Caso de Uso Administrar Empresas

Diagrama de Secuencia para el Caso de Uso Crear Empresa

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoNuevaEmpresa, luego de introducir y confirmar la información esta clase solicita el método CrearEmpresa(Información) de la clase DialogoAdministrarEmpresa, a este se le envía la información introducida por el



Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método NuevaEmpresa(Información) de la clase NodoEmpresa, al cual se le envía la información ya validada, acto seguido esta clase invocara al método CrearEmpresa(NodoEmpresa) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

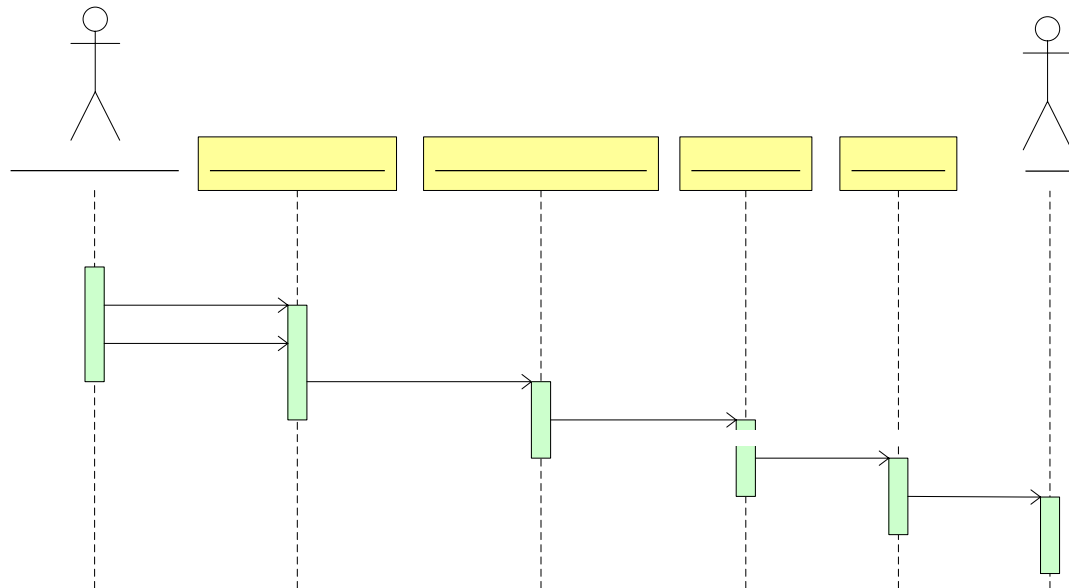


Figura 4.25 Diagrama de Secuencia para el Caso de Uso Crear Empresa

Diagrama de Secuencia para el Caso de Uso Modificar Empresa

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoModificarEmpresa, luego de introducir y confirmar la información esta clase solicita el método CargarEmpresa(Empresa) de la clase DialogoAdministrarEmpresa, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el

Administrador/Usuario DialogoNuevaEmpresa

DialogoAc

Introducir Informacion Empresa

CrearEmpresa(Informacio



método CargarEmpresa(Empresa) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información actual de la empresa a modificar es mostrada al Administrador/Usuario quien hará las modificaciones respectivas y confirmara los cambios. Una vez realizada esta acción la clase DialogoModificarEmpresa solicita el método ModificarEmpresa(NuevaInformación) de la clase DialogoAdministrarEmpresa, a este se le envía la información introducida por el Administrador/Usuario. Esta clase valida la información y solicita el método ModificarEmpresa(NuevaImformación) de la clase NodoEmpresa, que a su vez solicita a la clase ConectorSql la actualización de la información en el SMBD, a través del Método ModificarEmpresa(NuevaImformacion).

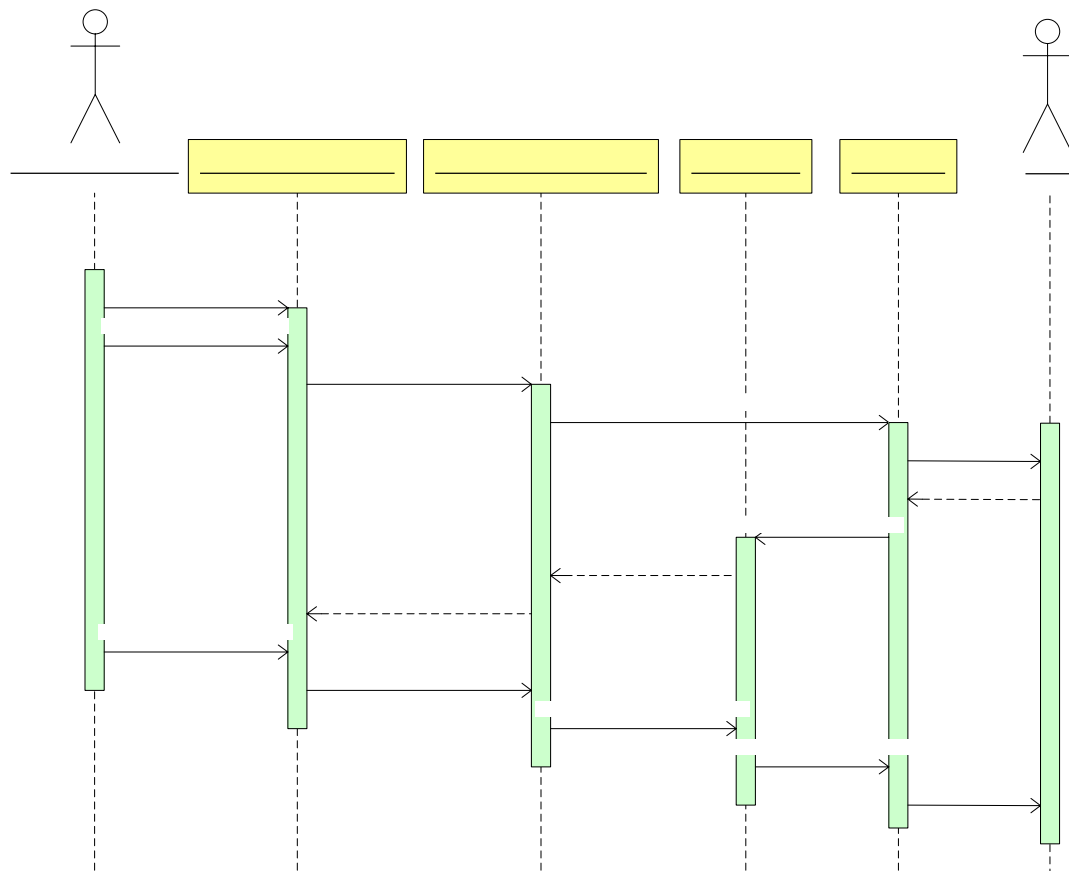


Figura 4.26 Diagrama de Secuencia para el Caso de Uso Modificar Empresa



Diagrama de Secuencia para el Caso de Uso Eliminar Empresa

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoAdministrarEmpresa, luego de introducir y confirmar la información esta clase valida dicha información y solicita el método EliminarEmpresa(Empresa) de la clase NodoEmpresa, al cual se le envía la información ya validada, acto seguido esta clase invocara al método EliminarEmpresa(NodoEmpresa) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

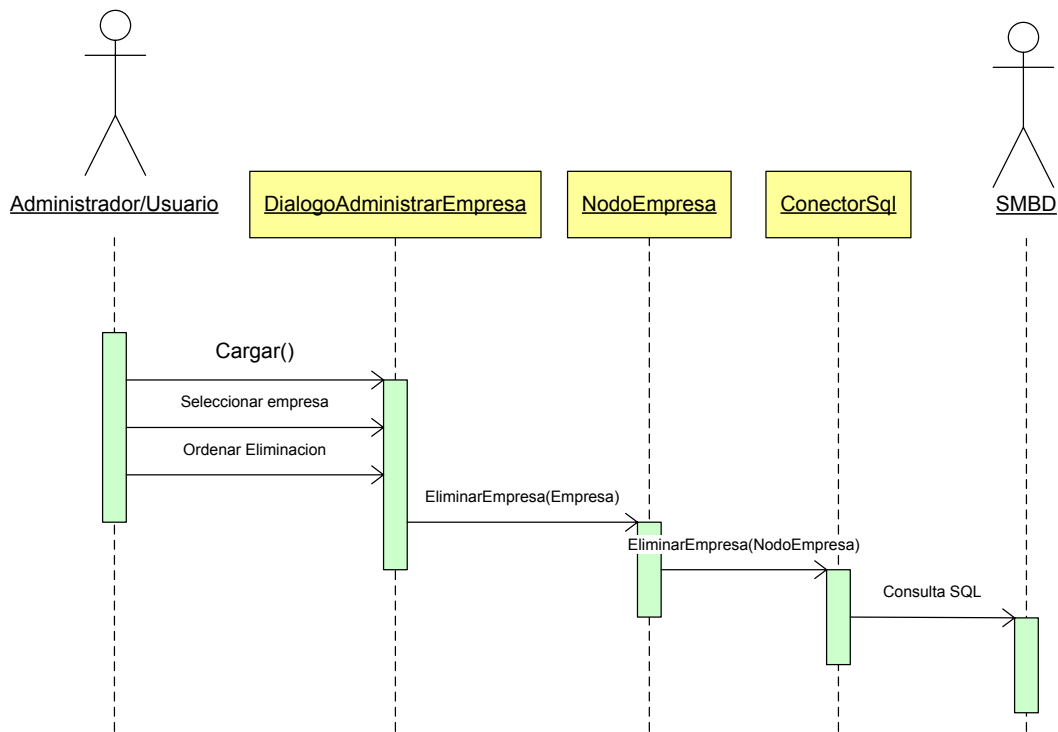


Figura 4.27 Diagrama de Secuencia para el Caso de Uso Eliminar Empresa

4.5.2.4 Diagrama de Secuencia para el Caso de Uso Administrar Cuentas



Diagrama de Secuencia para el Caso de Uso Crear Cuenta

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoNuevaCuenta, luego de introducir y confirmar la información esta clase solicita el método CrearCuenta(Informacion) de la clase DialogoAdministrarCuenta, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método NuevaCuenta(Información) de la clase NodoCuenta, al cual se le envía la información ya validada, acto seguido esta clase invocara al método CrearCuenta(NodoCuenta) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

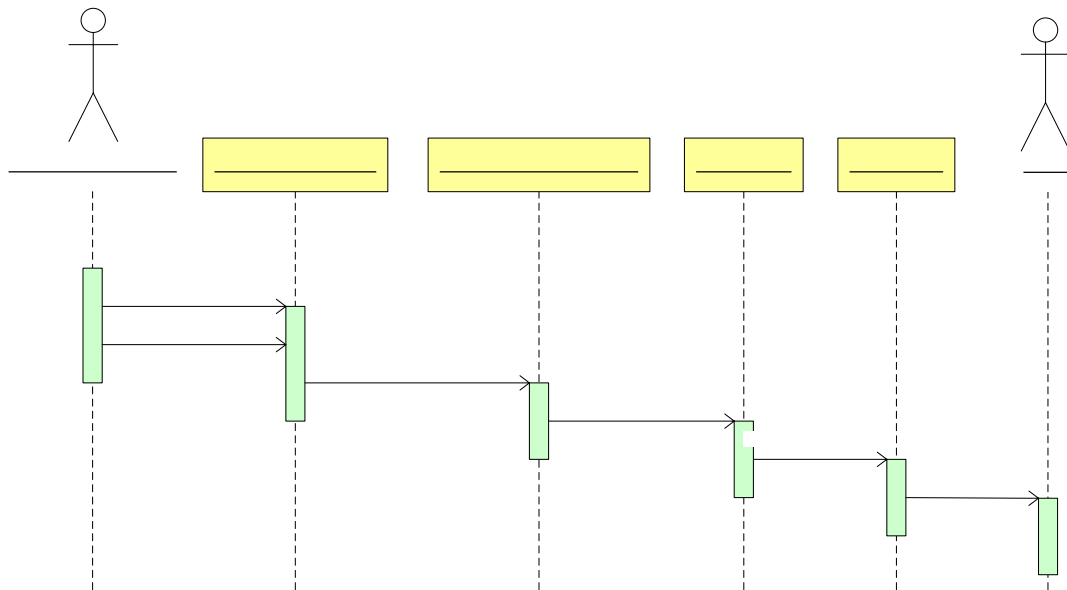


Figura 4.28 Diagrama de Secuencia para el Caso de Uso Crear Cuenta



Diagrama de Secuencia para el Caso de Uso Modificar Cuenta

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoModificarCuenta, luego de introducir y confirmar la información esta clase solicita el método CargarCuenta(Cuenta) de la clase DialogoAdministrarCuenta, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método CargarCuenta(Cuenta) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información actual de la Cuenta a modificar es mostrada al Administrador/Usuario quien hará las modificaciones respectivas y confirmara los cambios. Una vez realizada esta acción la clase DialogoModificarCuenta solicita el método ModificarCuenta(NuevaInformacion) de la clase DialogoAdministrarCuenta, a este se le envía la información introducida por el Administrador/Usuario. Esta clase valida la información y solicita el método ModificarCuenta(NuevaInformacion) de la clase NodoCuenta, que a su vez solicita a la clase ConectorSql la actualización de la información en el SMBD, a través del Método ModificarCuenta(NuevaInformacion).

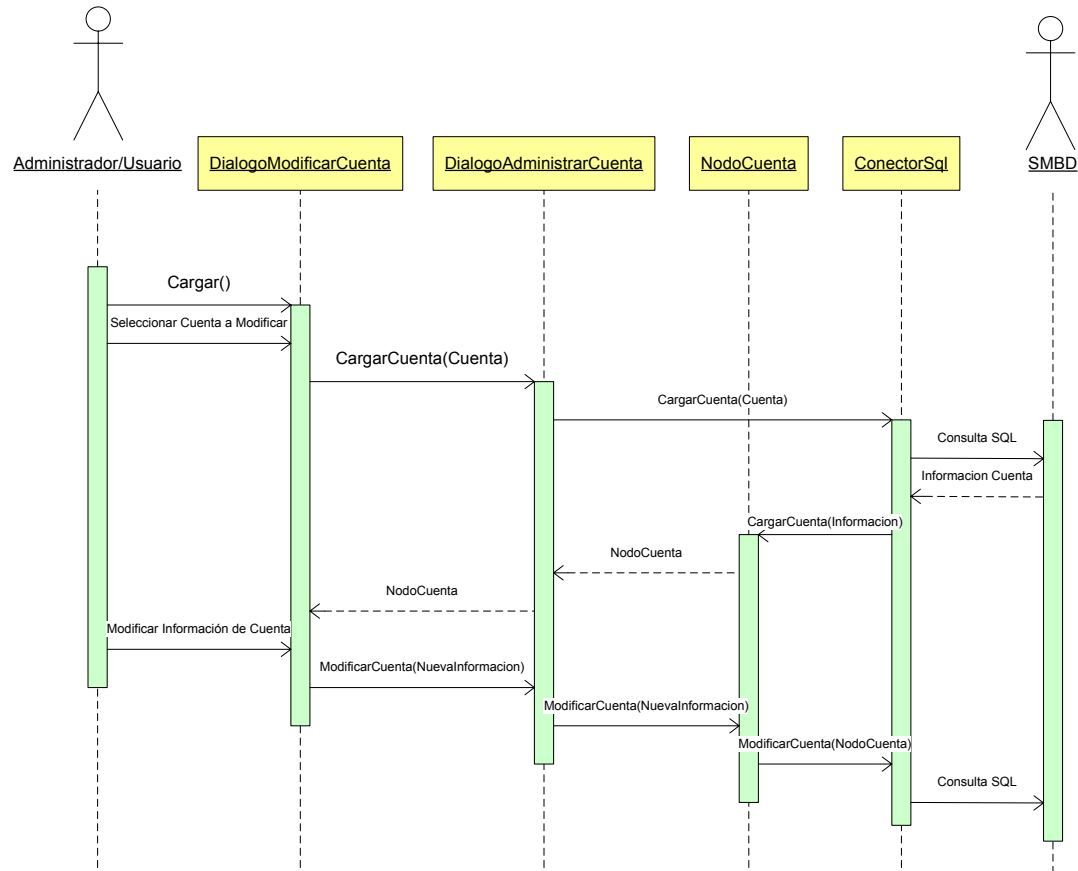


Figura 4.29 Diagrama de Secuencia para el Caso de Uso Modificar Cuenta

Diagrama de Secuencia para el Caso de Uso Eliminar Cuenta

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoAdministrarCuenta, luego de introducir y confirmar la información esta clase valida dicha información y solicita el método EliminarCuenta(Cuenta) de la clase NodoCuenta, al cual se le envía la información ya validada, acto seguido esta clase invocara al método EliminarCuenta(NodoCuenta) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

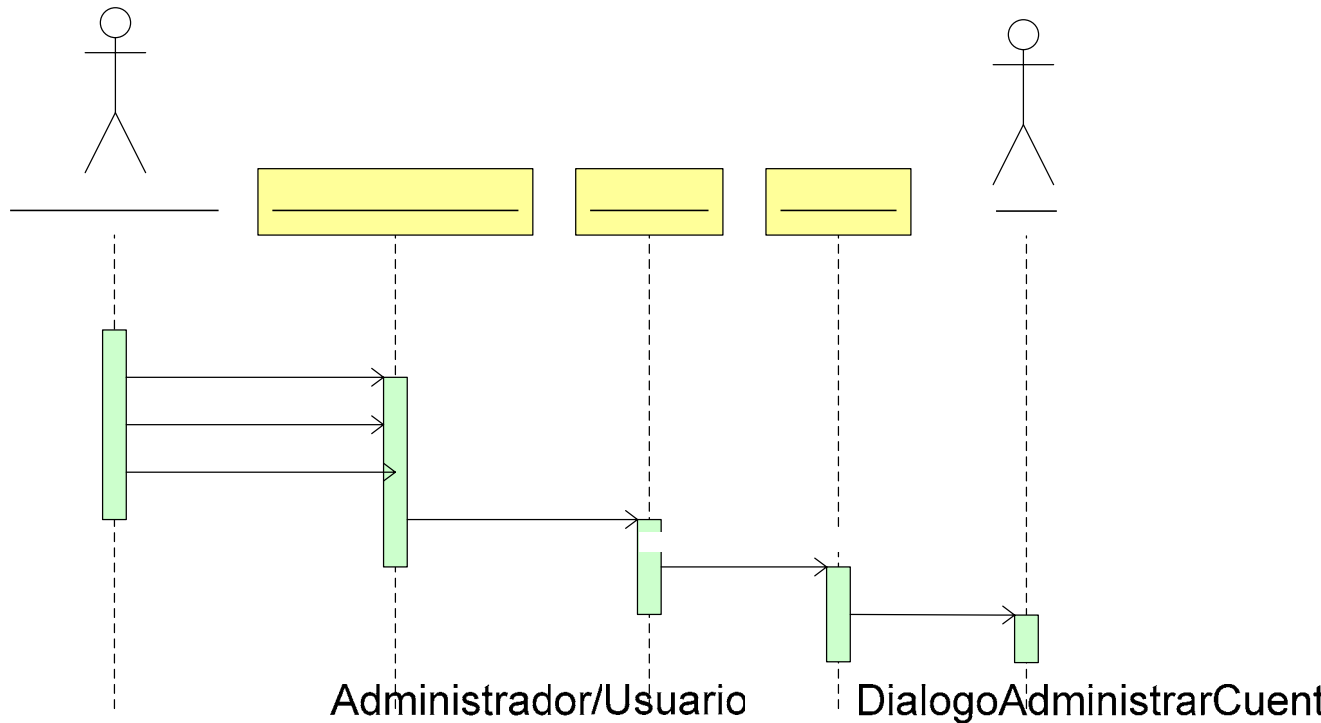


Figura 4.30 Diagrama de Secuencia para el Caso de Uso Eliminar Cuenta

4.5.2.5 Diagrama de Secuencia para el Caso de Uso Manejar Asientos

Cargar()

Seleccionar cuenta

Diagrama de Secuencia para el Caso de Uso Crear Asiento

Ordenar Eliminacion

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoNuevoAsiento, luego de introducir y confirmar la información esta clase solicita el método CrearAsiento(Informacion) de la clase DialogoAdministrarAsiento, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método NuevoAsiento(Información) de la clase NodoAsiento, al cual se le envía la

EliminarC



información ya validada, acto seguido esta clase invocara al método CrearAsiento(NodoAsiento) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

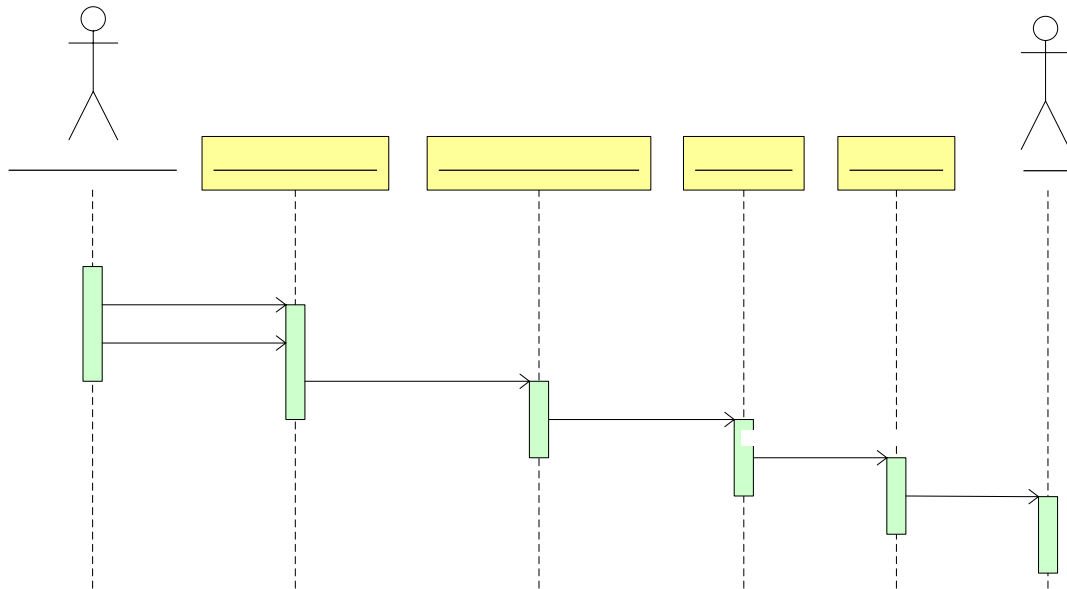


Figura 4.31 Diagrama de Secuencia para el Caso de Uso Crear Asiento

Diagrama de Secuencia para el Caso de Uso Modificar Asiento

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase DialogoModificarAsiento, luego de introducir y confirmar la información esta clase solicita el método CargarAsiento() de la clase DialogoAdministrarAsiento, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método CargarAsiento(Asiento) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información actual del Asiento a modificar es mostrada al

DialogoA

CrearAsiento(Informacion



Administrador/Usuario quien hará las modificaciones respectivas y confirmara los cambios. Una vez realizada esta acción la clase DialogoModificarAsiento solicita el método ModificarAsiento(NuevaInformacion) de la clase DialogoAdministrarAsiento, a este se le envía la información introducida por el Administrador/Usuario. Esta clase valida la información y solicita el método ModificarAsiento(NuevaImformacion) de la clase NodoAsiento, que a su vez solicita a la clase ConectorSql la actualización de la información en el SMBD, a través del Método ModificarAsiento(NuevaImformacion).

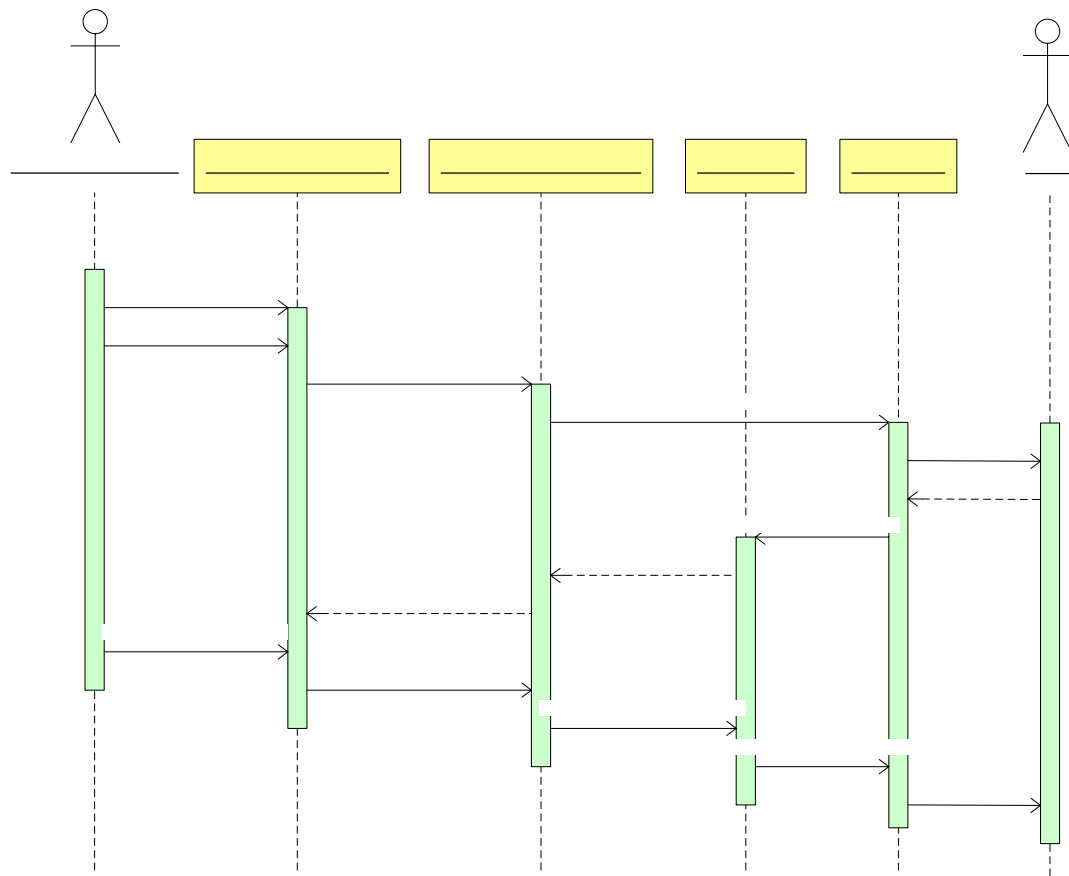


Figura 4.32 Diagrama de Secuencia para el Caso de Uso Modificar Asiento

Administrador/Usuario DialogoModificarAsiento DialogoAdministrarAsiento

DialogoAsiento



Diagrama de Secuencia para el Caso de Uso Eliminar Asiento

El Administrador inicia el caso de uso invocando el método Cargar() de la clase DialogoAdministrarAsiento, luego de introducir y confirmar la información esta clase valida dicha información y solicita el método EliminarAsiento(Asiento) de la clase NodoAsiento, al cual se le envía la información ya validada, acto seguido esta clase invocara al método EliminarAsiento(NodoAsiento) de la clase ConectorSql, quien realizará la(s) consulta(s) al SMBD.

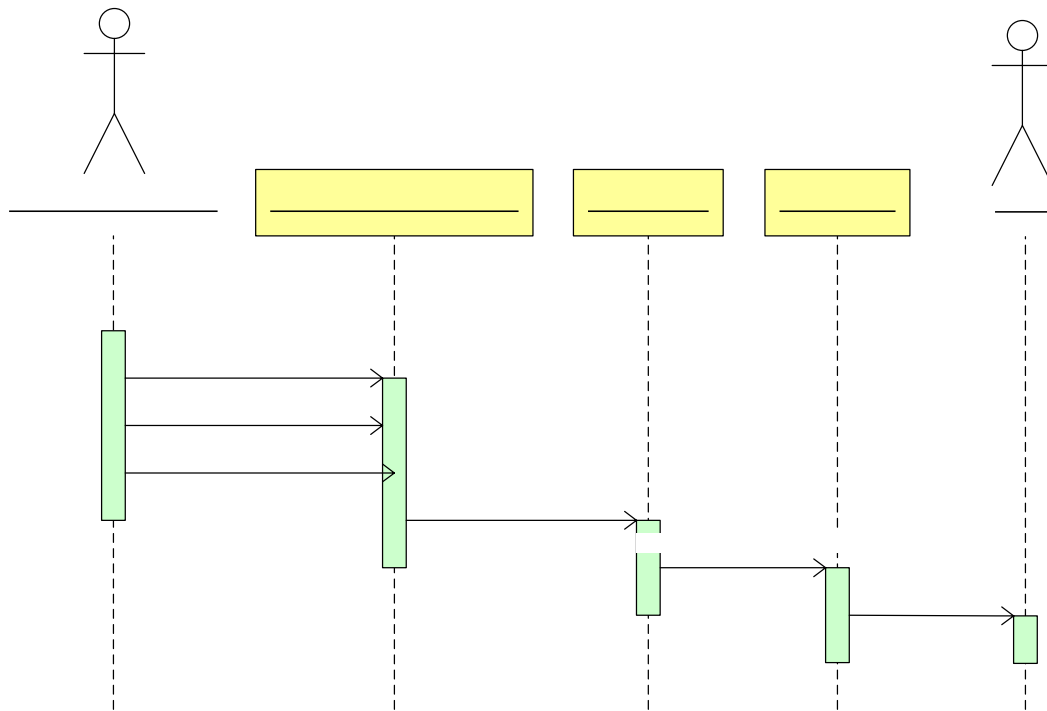


Figura 4.33 Diagrama de Secuencia para el Caso de Uso Eliminar Asiento



4.5.2.6 Diagrama de Secuencia para el Caso de Uso Emitir Reportes

Diagrama de Secuencia para el Caso de Uso Emitir Balance General

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase JPanelBalanceGeneral, luego de introducir y confirmar la información esta clase solicita el método CrearBalanceGeneral(Fecha) de la clase GestorDeReporte, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método CargarAsiento(Fecha) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información de los Asientos solicitados es almacenada temporalmente por la clase NodoAsiento por medio del método CargarAsiento(Asiento en Rango), seguidamente la clase GestorDeReportes haciendo uso del método CalcularBalanceGeneral(NodoAsiento) realiza los cálculos correspondientes con la información de los Asientos y solicita el método CargarBalanceGeneral() de la clase JPanelBalanceGeneral() para generar el reporte final.

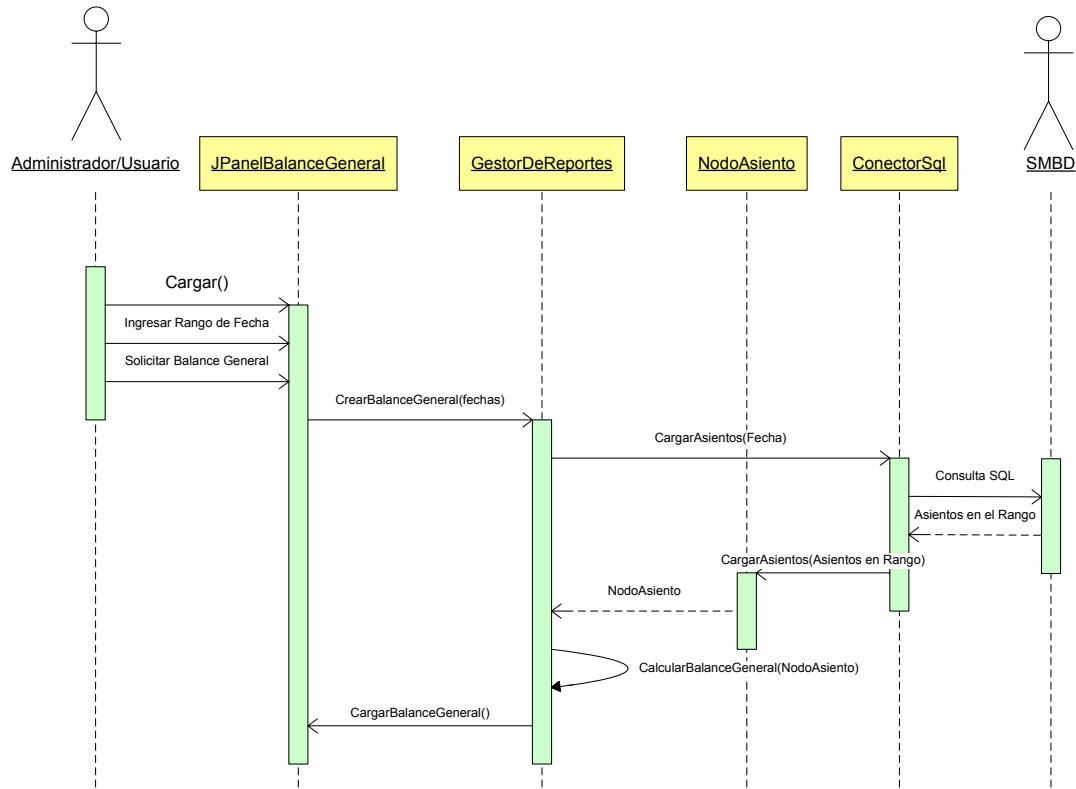


Figura 4.34 Diagrama de Secuencia para el Caso de Uso Emitir Balance General

Diagrama de Secuencia para el Caso de Uso Emitir Balance de Comprobación

El Administrador/Usuario inicia el caso de uso invocando el método `Cargar()` de la clase `JPanelBalanceDeComprobación`, luego de introducir y confirmar la información esta clase solicita el método `CrearBalanceDeComprobación(Fecha)` de la clase `GestorDeReporte`, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método `CargarAsiento(Fecha)` de la clase `ConectorSql` quien realizará la(s) consulta(s) al SMBD. La información de los Asientos solicitados es almacenada temporalmente por la clase `NodoAsiento` por medio del método



CargarAsiento(Asiento en Rango), seguidamente la clase GestorDeReportes haciendo uso del método CalcularBalanceDeComprobación(NodoAsiento) realiza los cálculos correspondientes con la información de los Asientos y solicita el método CargarBalanceDeComprobación() de la clase JPanelBalanceDeComprobación() para generar el reporte final.

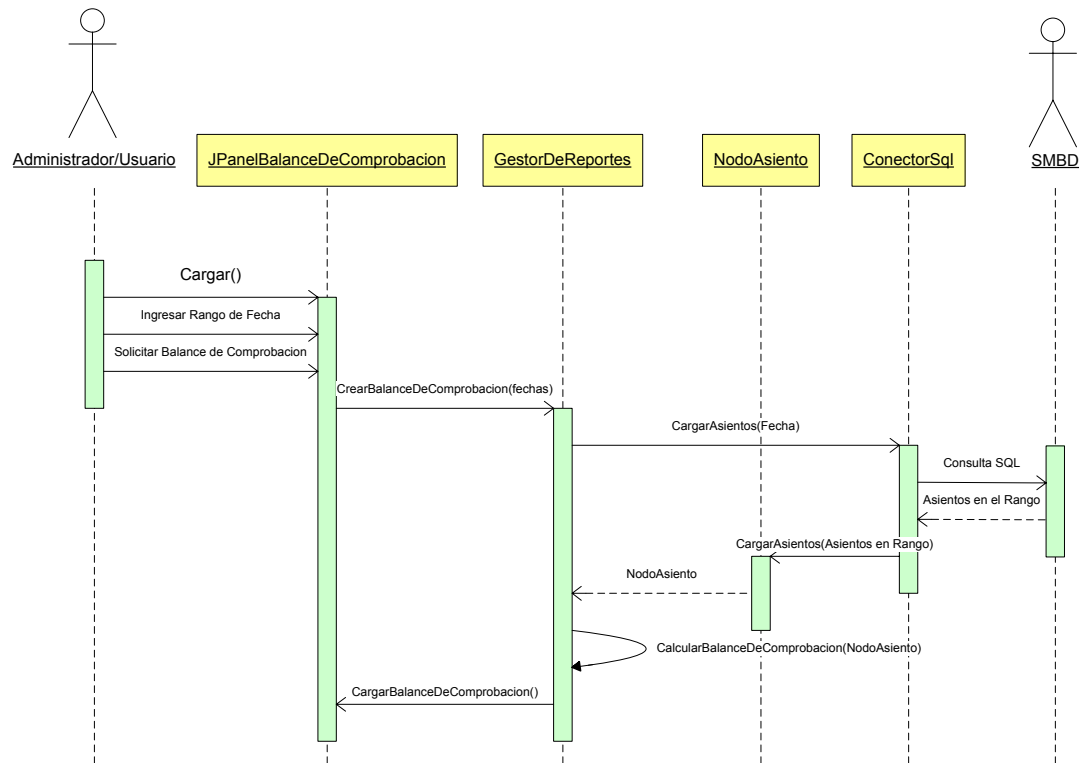


Figura 4.35 Diagrama de Secuencia para el Caso de Uso Emitir Balance de Comprobación

Diagrama de Secuencia para el Caso de Uso Emitir Estado de Resultados

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase JPanelEstadoDeResultados, luego de introducir y confirmar la información



esta clase solicita el método `CrearEstadoDeResultados(Fecha)` de la clase `GestorDeReporte`, a este se le envía la información introducida por el `Administrador/Usuario`. Posteriormente esta clase valida la información y solicita el método `CargarAsiento(Fecha)` de la clase `ConectorSql` quien realizará la(s) consulta(s) al `SMBD`. La información de los `Asientos` solicitados es almacenada temporalmente por la clase `NodoAsiento` por medio del método `CargarAsiento(Asiento en Rango)`, seguidamente la clase `GestorDeReportes` haciendo uso del método `CalcularEstadoDeResultados(NodoAsiento)` realiza los cálculos correspondientes con la información de los `Asientos` y solicita el método `CargarEstadoDeResultados()` de la clase `JPanelEstadoDeResultados()` para generar el reporte final.

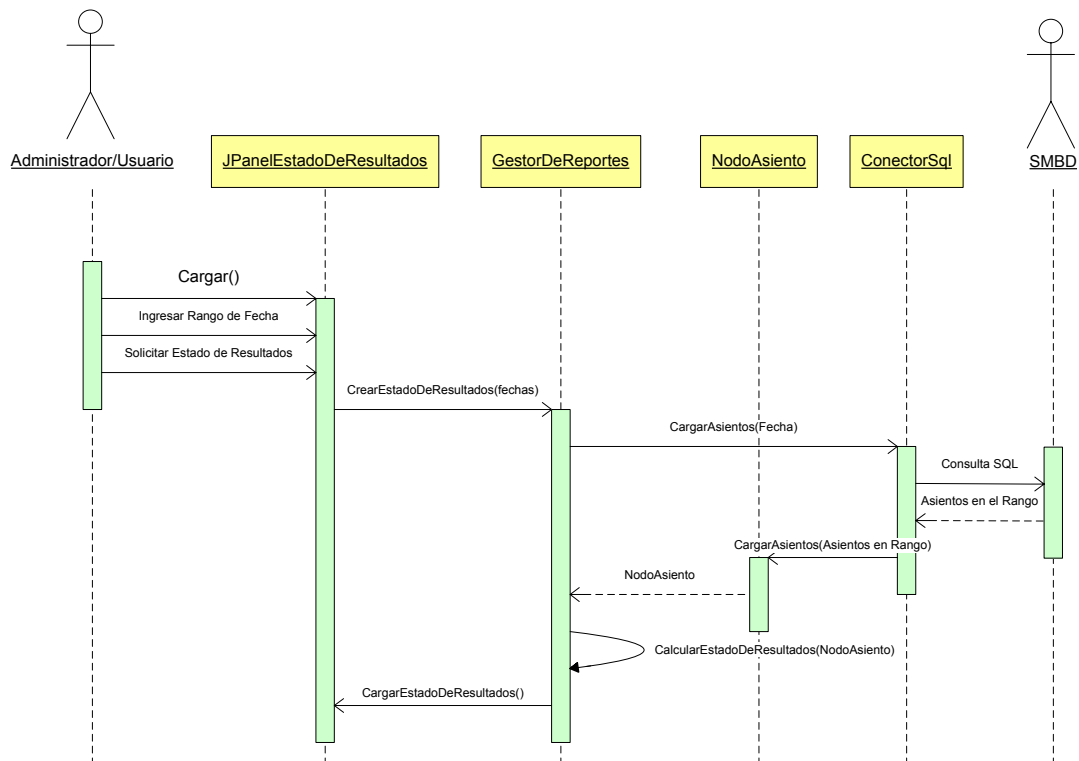


Figura 4.36 Diagrama de Secuencia para el Caso de Uso Emitir Estado de Resultados



Diagrama de Secuencia para el Caso de Uso Emitir Libro Diario General

El Administrador/Usuario inicia el caso de uso invocando el método `Cargar()` de la clase `JPanelLibroDiarioGeneral`, luego de introducir y confirmar la información esta clase solicita el método `CrearLibroDiarioGeneral(Fecha)` de la clase `GestorDeReporte`, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método `CargarAsiento(Fecha)` de la clase `ConectorSql` quien realizará la(s) consulta(s) al SMBD. La información de los Asientos solicitados es almacenada temporalmente por la clase `NodoAsiento` por medio del método `CargarAsiento(Asiento en Rango)`, seguidamente la clase `GestorDeReportes` haciendo uso del método `CalcularLibroDiarioGeneral(NodoAsiento)` realiza los cálculos correspondientes con la información de los Asientos y solicita el método `CargarLibroDiarioGeneral()` de la clase `JPanelLibroDiarioGeneral()` para generar el reporte final.

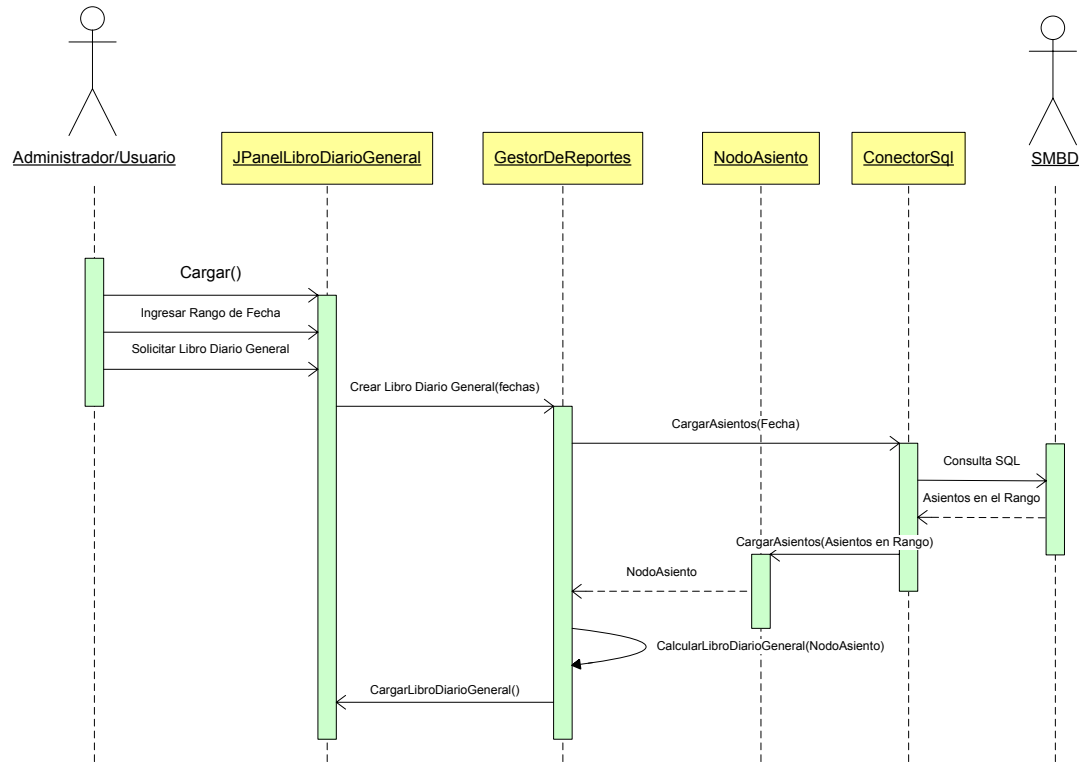


Figura 4.37 Diagrama de Secuencia para el Caso de Uso Emitir Libro Diario General

Diagrama de Secuencia para el Caso de Uso Emitir Libro Mayor

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase JPanellibroMayor, luego de introducir y confirmar la información esta clase solicita el método CrearLibroMayor(Fecha) de la clase GestorDeReporte, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método CargarAsiento(Fecha) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información de los Asientos solicitados es almacenada temporalmente por la clase NodoAsiento por medio del método CargarAsiento(Asiento en Rango), seguidamente la clase



GestorDeReportes haciendo uso del método CalcularLibroMayor(NodoAsiento) realiza los cálculos correspondientes con la información de los Asientos y solicita el método CargarLibroMayor() de la clase JPanelLibroMayor() para generar el reporte final.

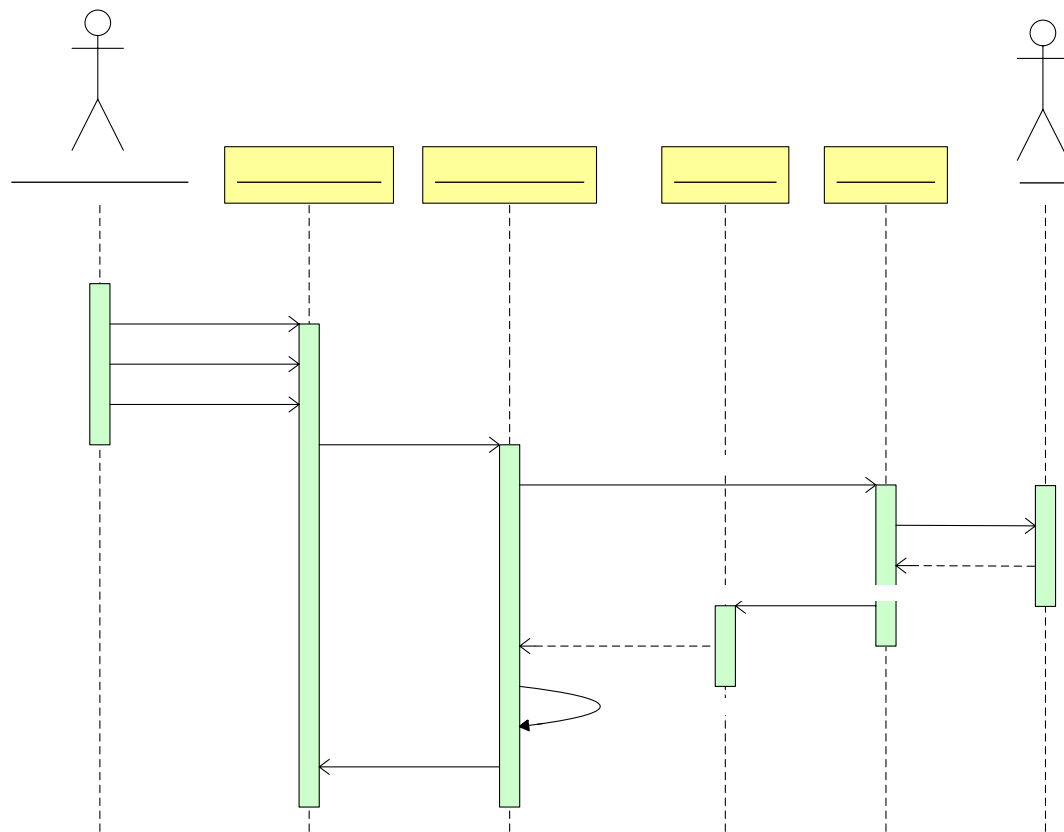


Figura 4.38 Diagrama de Secuencia para el Caso de Uso Emitir Libro Mayor

Diagrama de Secuencia para el Caso de Uso Emitir Mayor Analítico

Cargar()

El Administrador/Usuario inicia el caso de uso invocando el método Cargar() de la clase JPanelMayorAnalítico, luego de introducir y confirmar la información esta

Ingresar Rango de Fecha

Solicitar Libro Mayor

CrearLibroMayorfechas)



clase solicita el método CrearMayorAnalítico(Fecha) de la clase GestorDeReporte, a este se le envía la información introducida por el Administrador/Usuario. Posteriormente esta clase valida la información y solicita el método CargarAsiento(Fecha) de la clase ConectorSql quien realizará la(s) consulta(s) al SMBD. La información de los Asientos solicitados es almacenada temporalmente por la clase NodoAsiento por medio del método CargarAsiento(Asiento en Rango), seguidamente la clase GestorDeReportes haciendo uso del método CalcularMayorAnalítico(NodoAsiento) realiza los cálculos correspondientes con la información de los Asientos y solicita el método CargarMayorAnalítico() de la clase JPanelMayorAnalítico() para generar el reporte final.

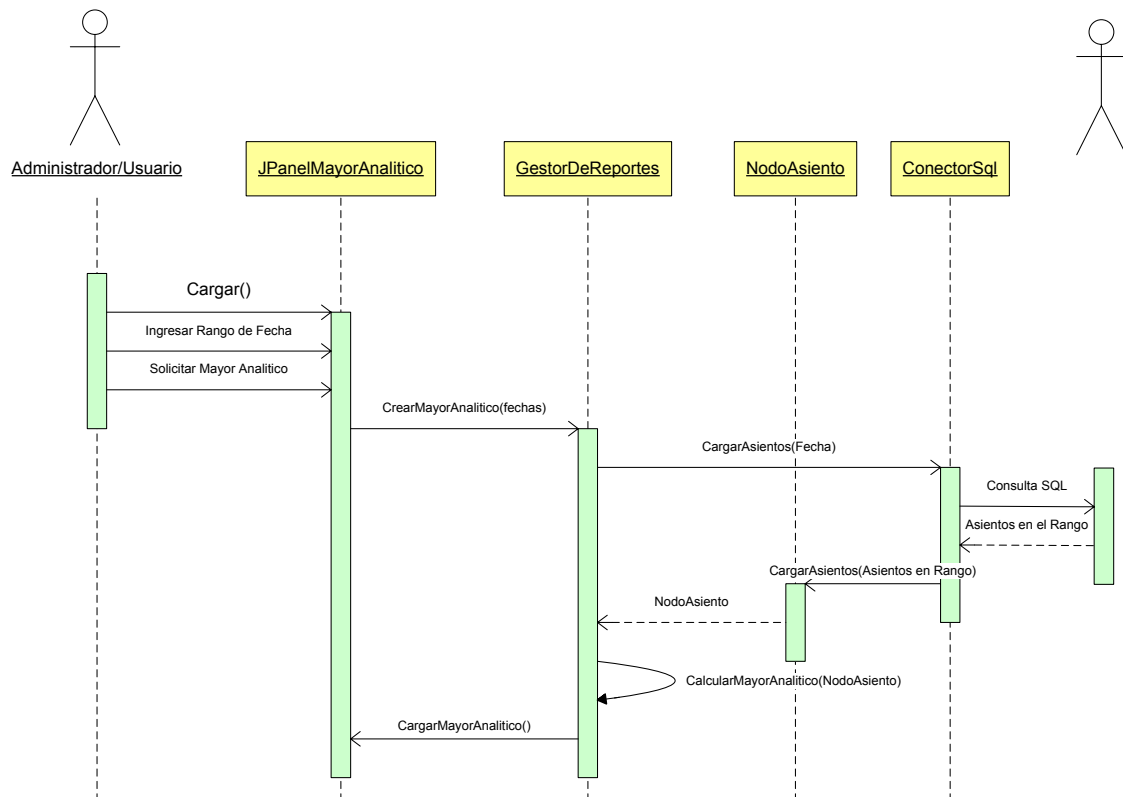


Figura 4.39 Diagrama de Secuencia para el Caso de Uso Emitir Mayor Analítico



4.5.3 Identificación de paquetes de diseño

El sistema J-Contab identifica sus paquetes de diseño a partir del modelo de casos de uso. El caso de uso: Administrar usuarios, esta contenido en el paquete de diseño denominado Usuarios. Este paquete contiene los casos de usos encargados de todo lo relacionado con los usuarios. De igual manera, cada caso de uso del sistema esta contenido en un paquete de diseño como se muestra en la siguiente imagen.

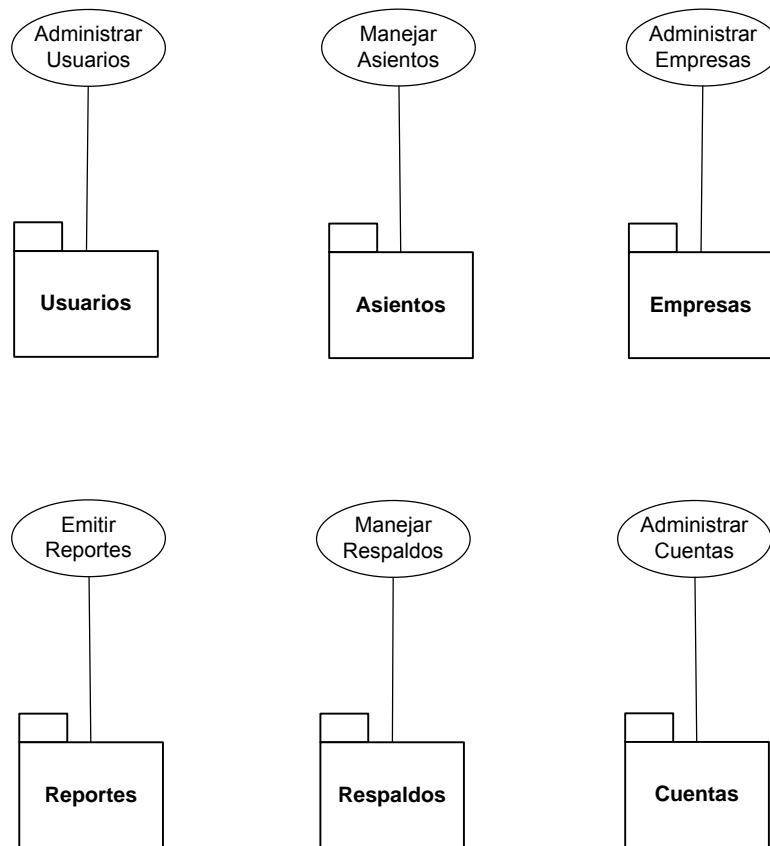


Figura 4.40 Paquete de Diseño Usuarios



4.5.4 Diseño de la Arquitectura

En el diseño de la arquitectura, se muestra la capa específica y general de la aplicación, que están compuestas de los paquetes de análisis realizados en la fase anterior. En la capa intermedia y la capa de software del sistema se muestra los subsistemas relacionados con software prefabricado, que ejecute y de soporte a las funcionalidades requeridas por el sistema, tales como sistemas operativos, manejadores de bases de datos y componentes reutilizables.

La capa específica de la aplicación esta conformada por los paquetes de Usuarios, Respaldos, Empresas, Cuentas, Asientos y Reportes; y por su parte la capa general de la aplicación estará formada por los paquetes MySQL JDBC Driver y Java Foundation Classes (JFC), que son los paquetes que harán posible la conexión entre la aplicación y el Sistema Manejador de Base de Datos y los paquetes que conforman la librería de clases del lenguaje de programación Java. La capa intermedia está integrada por el Java Virtual Machina (JVM) que es el software de máquina virtual que hace posible la ejecución de cualquier aplicación java en el equipo en donde esté instalado.

En la capa de software del sistema tenemos al sistema operativo que es el software encargado de coordinar, administrar y gestionar todos los recursos del sistema, para así dar soporte a las actividades realizadas por los demás subsistemas y el sistema manejador de base de datos MySQL, que será el encargado de manejar o atender de forma transparente las solicitudes realizadas sobre la base de datos.

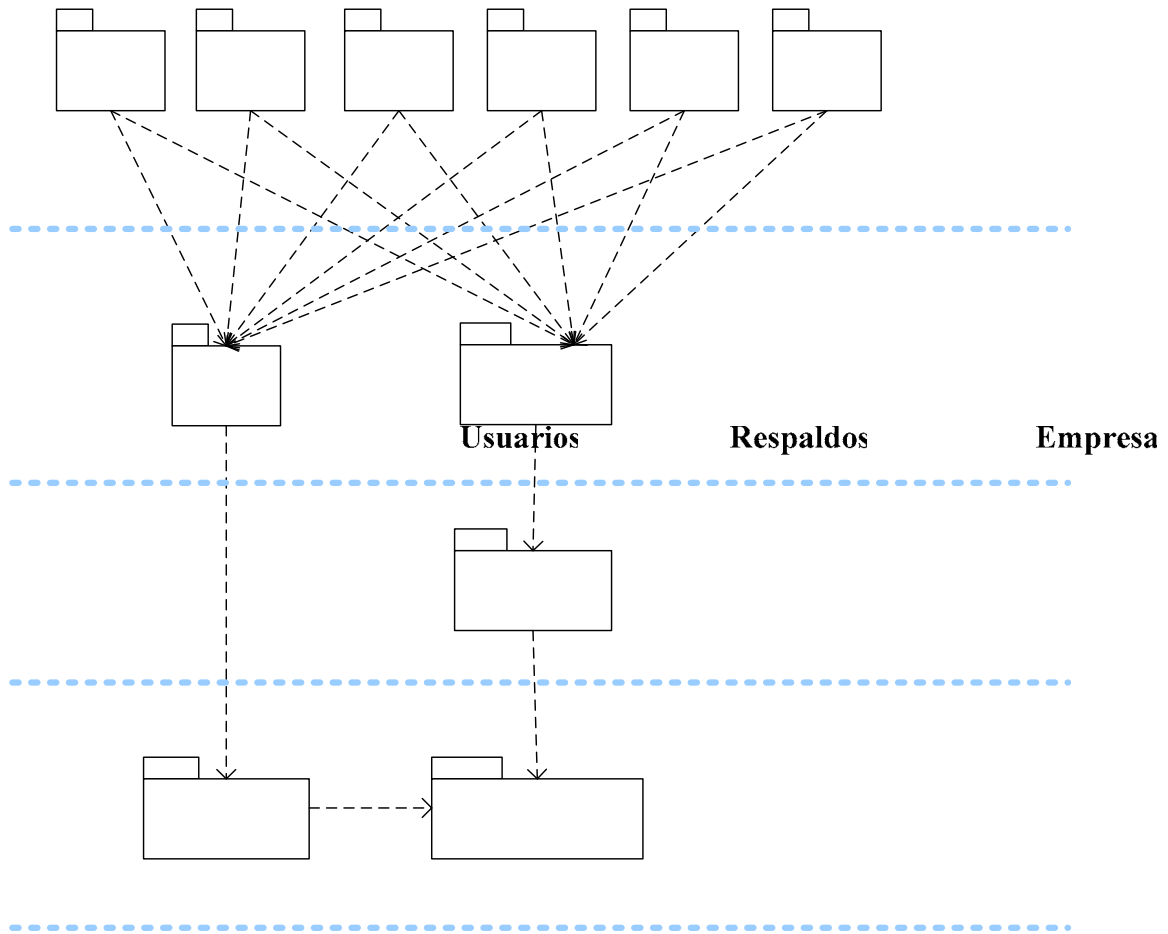


Figura 4.41 Vista Lógica de las Capas de la Arquitectura del Sistema
MySQL JDBC
Driver

4.5.5 Diseño de la Base de Datos del Sistema

Como se puede observar en los modelos de gestión diseñados en la sesión anterior, se le hacía un llamado a la base de datos del sistema, bien sea para consultar o ingresar o eliminar algo en ella. Para el diseño de la base de datos, se crea un esquema conceptual mediante un modelo de datos de alto nivel.

Java
Cla

Java
Environ



4.5.5.1 Identificación de la Tablas de la Base de Datos del Sistema

En el estudio y definición de las tablas del sistema fueron identificadas 4 tablas, las cuales serán mostradas a continuación. Los campos claves de las tablas son señalados mediante un subrayado.

Tabla Usuarios:

En la tabla Usuarios, se almacena toda la información relacionada con los usuarios del sistema tales como el nombre del usuario, nombre de inicio de sesión del usuario, privilegios y contraseña.

Esquema:

Usuario: varchar (45).

Nombre: varchar (45).

Contraseña: varchar (45).

Tipo: varchar (1).

Tabla Empresas:

En la tabla Empresas, se almacena toda la información relacionada con las empresas registradas en el sistema, entre ellos, nombre de la empresa, dirección, Rif, etc.

Esquema:

Rif: varchar (45).

Nombre: varchar (100).

Dirección: varchar (200).

Teléfono: varchar (12).

**Tabla PCuentas:**

En la tabla PCuentas, se almacena toda la información relacionada con las cuentas contables de una empresa registrada en el sistema.

Esquema:

Código: varchar (12).

Descripción: varchar (45).

GoM: varchar (1).

Nivel: smallint.

Tabla LDiario:

En la tabla LDiario, se almacenan los asientos contables de una empresa registrada en el sistema.

Esquema:

Rif: varchar(45)

CodigoAsiento: int.

CodigoCuenta: varchar (12).

DescripciónCuenta: varchar (45).

DescripciónAsiento: varchar (45).

Debe/Haber: varchar (1).

Monto: real.

Mes: smallint.

Año: smallint.



4.5.5.2 Modelo Físico de Datos de la Base de Datos Local

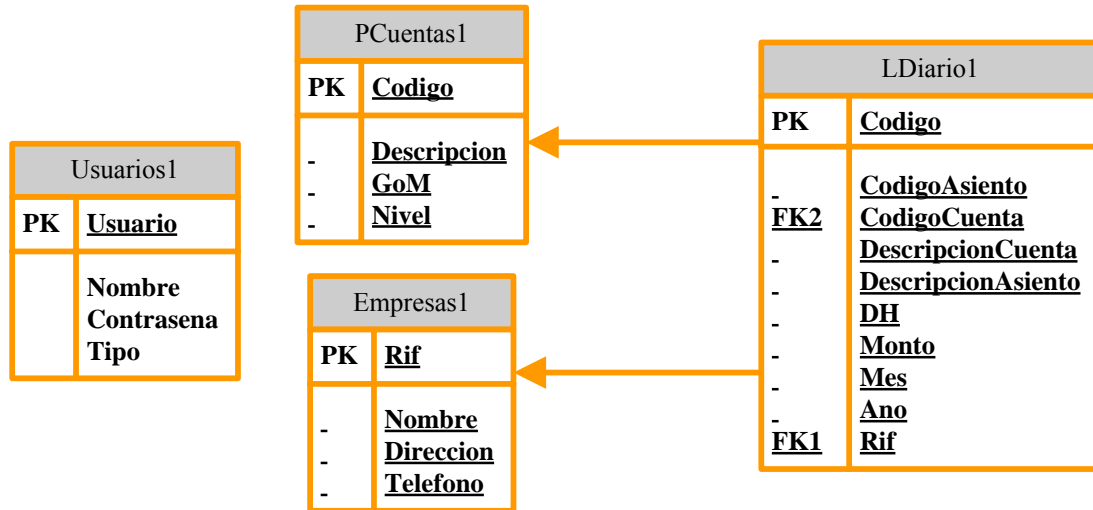


Figura 4.42 Modelo Físico de la Base de Datos Local

4.6 EVALUACIÓN DE LA FASE DE ELABORACIÓN

Durante el desarrollo de la fase de elaboración, se lograron obtener los resultados esperados, tomando en cuenta que la toma de requisitos en la fase anterior fue altamente efectiva, no encontrándose ningún nuevo requisito y por consiguiente manteniéndose el análisis realizado en dicha fase, esto permitió continuar con el proceso, permitiendo el desarrollo de los modelos de colaboración y secuencia, obteniendo finalmente el modelo final de la arquitectura, cumpliendo con los requisitos de ser sólida y confiable, además de cumplir con los demás requisitos funcionales. Se realizó el diseño de la base de datos, obteniendo un diseño eficiente sin redundancias y soportando hasta la Forma Bormal de Boyce-Codd.



Capítulo IV - Fase de Elaboración

Tomando en cuenta los avances realizados en esta fase, se dejan las partes de implementación y pruebas para la siguiente fase, ya teniendo los basamentos de arquitectura y secuencia obtenidos aquí.



CAPÍTULO V. FASE DE CONSTRUCCIÓN

5.1 INTRODUCCIÓN

En la fase anterior se estableció la línea base de la arquitectura y se mantuvieron los casos de uso iniciales. En la fase de construcción se realizó la codificación de los subsistemas que fueron diseñados en la fase de elaboración. También serán realizadas pruebas para así poder obtener una versión beta del sistema.

5.2 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN

En la fase de construcción se llevó a cabo los flujos de trabajos de implementación y prueba, tomando como guía los diseños elaborados en la fase anterior.

5.3 FLUJO DE TRABAJO DE IMPLEMENTACIÓN

En este flujo de trabajo se implementan los archivos de código fuente, archivos de código binario y ejecutable correspondientes al sistema, esto con la finalidad de obtener la versión beta de la aplicación.



En esta etapa se integran todos los componentes del subsistema de software obtenidos durante la etapa de diseño y se realizan las pruebas de unidad de los componentes más importantes, los cuales se expresan como pequeños módulos ejecutables del sistema.

5.3.1 Implementación de los Subsistemas.

Al realizar la identificación de los componentes ejecutables del sistema J-Contab han sido implementados en esta fase los casos de uso: Administrar Usuarios, Manejar Respaldos, Administrar Empresas, Administrar Cuentas, Manejar Asientos y Emitir Reportes.

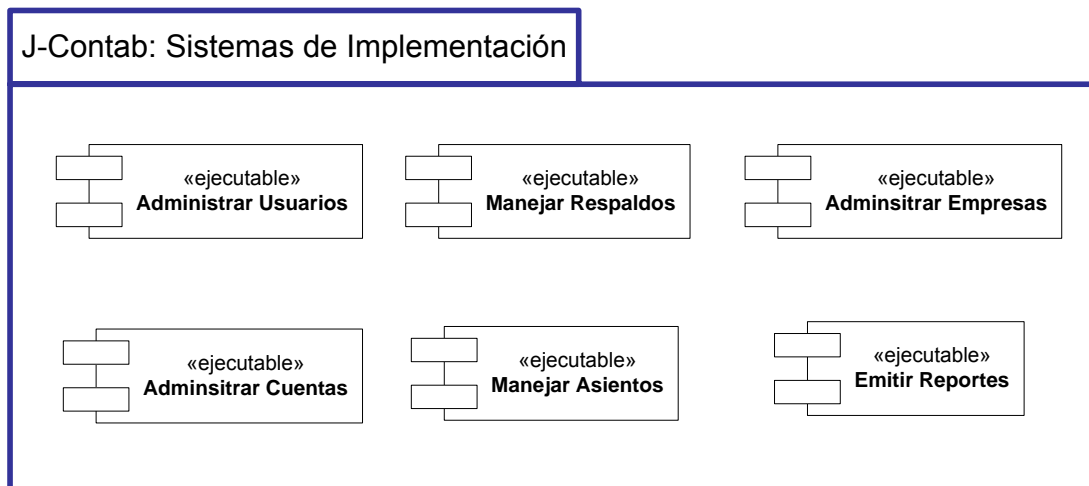


Figura 5.1 Subsistemas de Implementación.

Administrar Usuarios.



El subsistema de implementación Administrar Usuarios contiene los archivos necesarios para crear y eliminar usuarios del sistema.

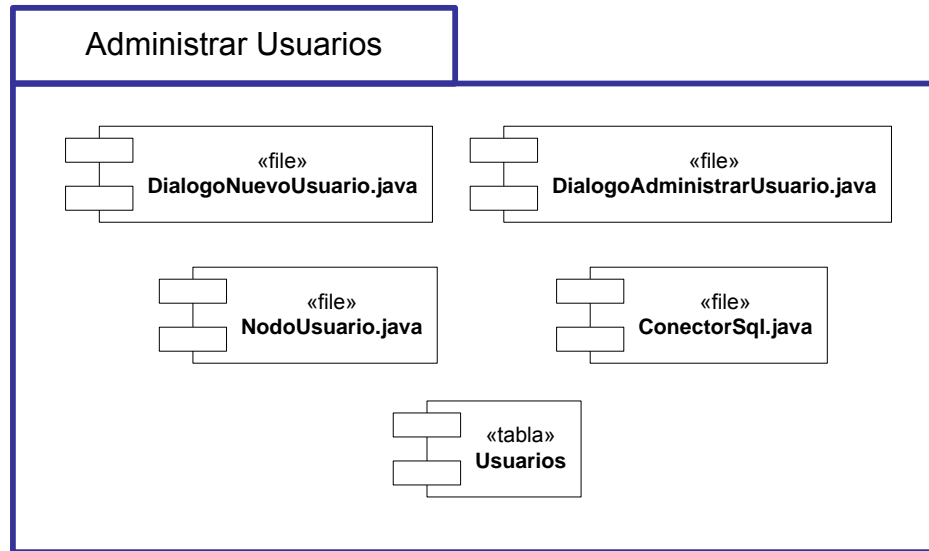


Figura 5.2 Subsistema Administrar Usuarios para la implementación.

Manejar Respaldos.

El subsistema de implementación Manejar Respaldos contiene los archivos necesarios para exportar e importar archivos de respaldo de las empresas cargadas en el sistema.

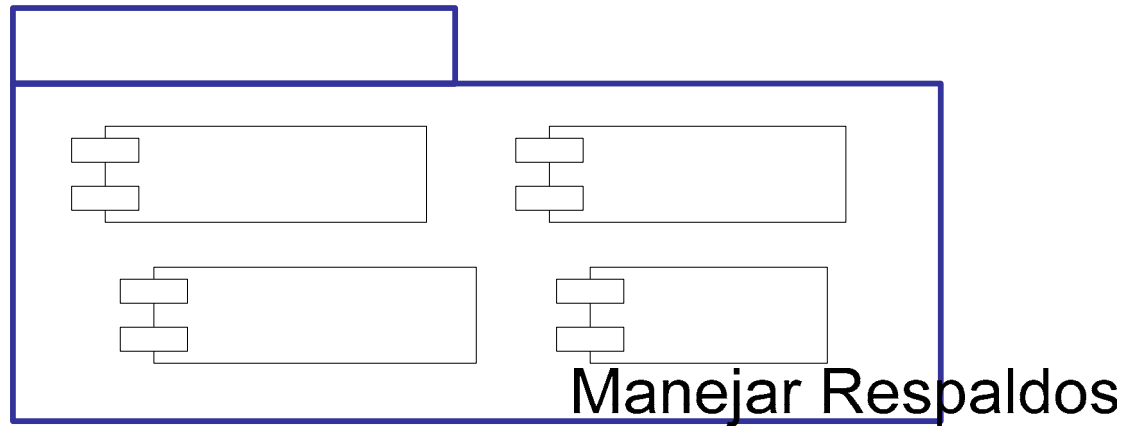


Figura 5.3 Subsistema Manejar Respaldos para la implementación.

Administrar Empresas.

«file»
DialogoImportarData.java

El subsistema de implementación Administrar Empresas contiene los archivos necesarios para crear, modificar y eliminar empresas del sistema.

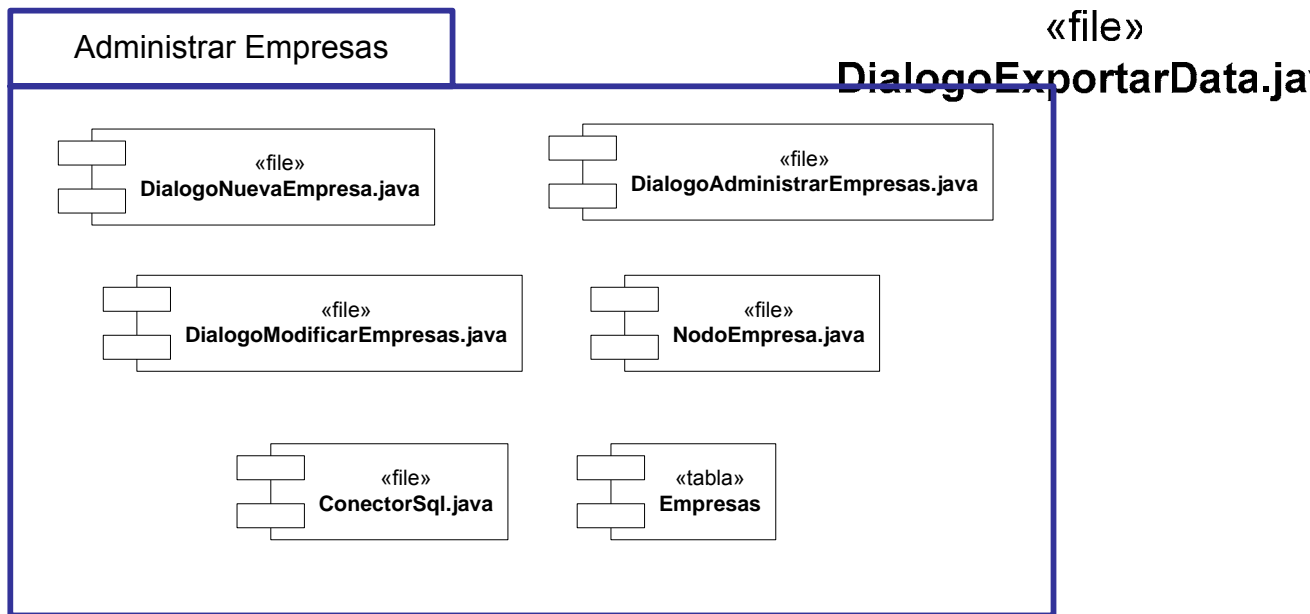


Figura 5.4 Subsistema Administrar Empresas para la implementación.



Administrar Cuentas.

El subsistema de implementación Administrar Cuentas contiene los archivos necesarios para crear, modificar y eliminar cuentas contables de las empresas existentes en el sistema.

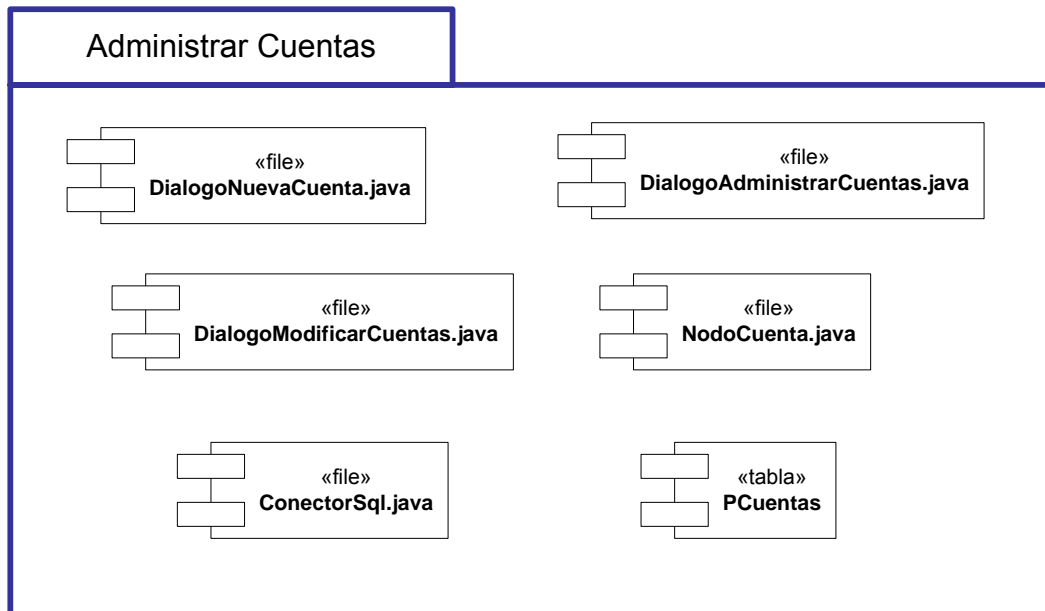


Figura 5.5 Subsistema Administrar Cuentas para la implementación.

Manejar Asientos.

El subsistema de implementación Manejar Asiento contiene los archivos necesarios para crear, modificar y eliminar asientos en el sistema.

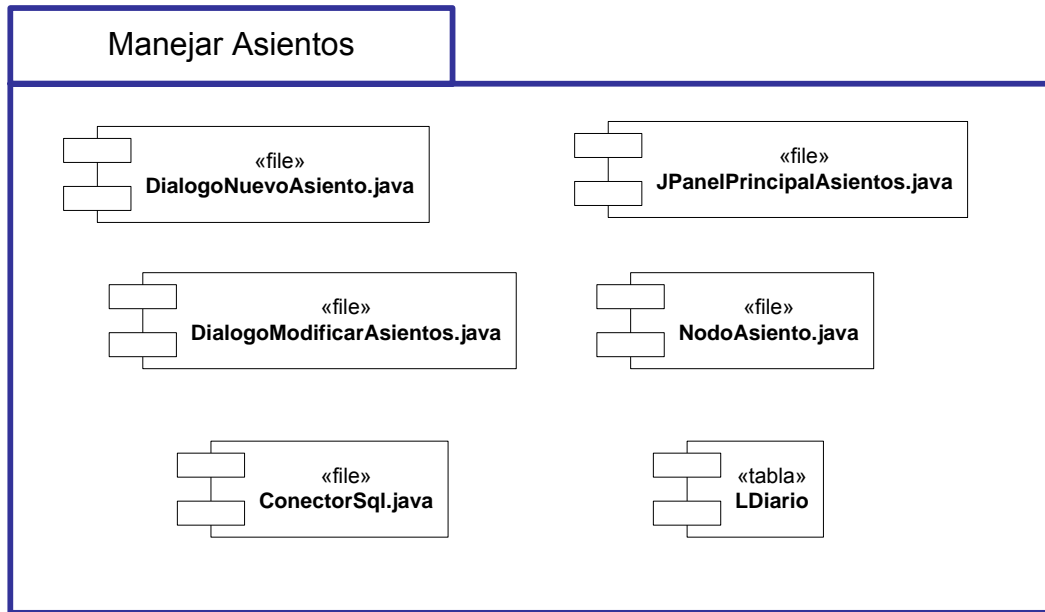


Figura 5.6 Subsistema Manejar Asientos para la implementación.

Emitir Reportes.

El subsistema de implementación Emitir Reporte contiene los archivos necesarios para generar los diferentes reportes disponibles en el sistema.

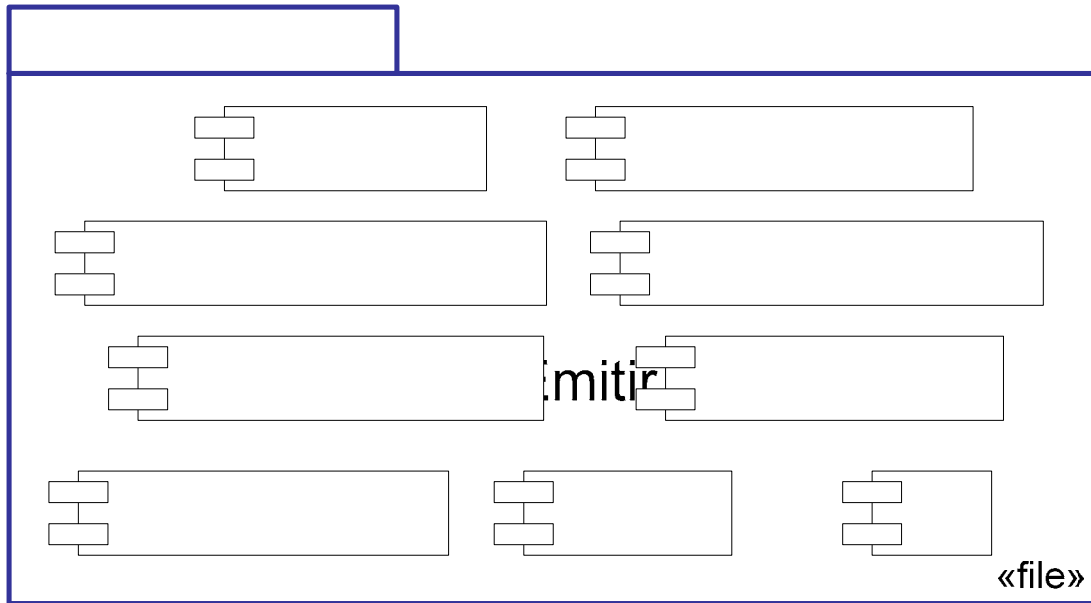


Figura 5.7 Subsistema Emitir Reportes para la implementación.

5.3.2 Integración del sistema

Tomando los casos de uso analizados e implementados, la integración del sistema estará establecida por dichos casos de uso, formando así el plan de integración del sistema.

Construcción 1.

En la primera construcción se implementaron las interfaces y gestores asociados a los casos de uso Administrar Usuarios y Mantener Resaldos. Estos subsistemas están compuestos por los ejecutables Usuarios y Resaldos así como también por los archivos DialogoNuevoUsuario, DialogoAdministrarUsuario, NodoUsuario, DialogoResaldos, DialogoExportarData, DialogoImportarData y ConectorSQL.

«file»
JPanelPrincipalBalancedeComprobacion.java
 «file»
JPanelPrincipalLibroDiarioGeneral.java
 «file»
JPanelPrincipalMayorAnalitico.java

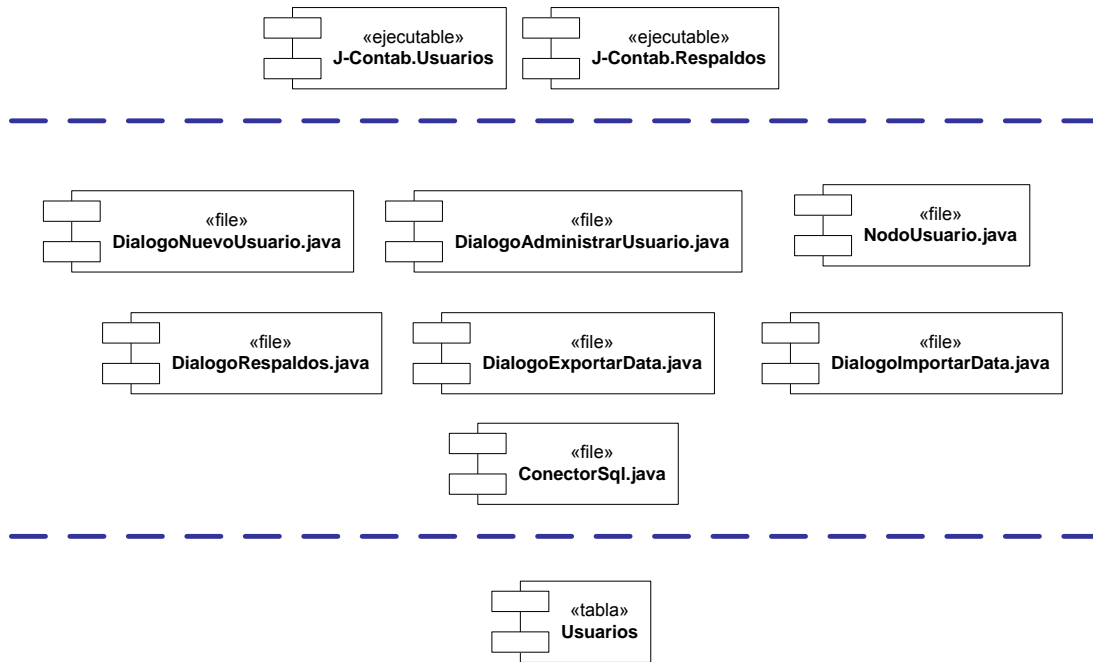


Figura 5.8 Componentes implementados en la primera construcción.

Construcción 2.

En la segunda construcción se implementaron las interfaces y gestores asociados a los casos de uso Administrar Empresas y Administrar Cuentas. Estos subsistemas están compuestos por los ejecutables Empresas y Cuentas así como también por los archivos DialogoNuevaEmpresa, DialogoAdministrarEmpresas, DialogoModificarEmpresas, NodoEmpresa, DialogoNuevaCuenta, DialogoAdministrarCuentas, DialogoModificarCuentas, NodoCuenta y ConectorSQL.

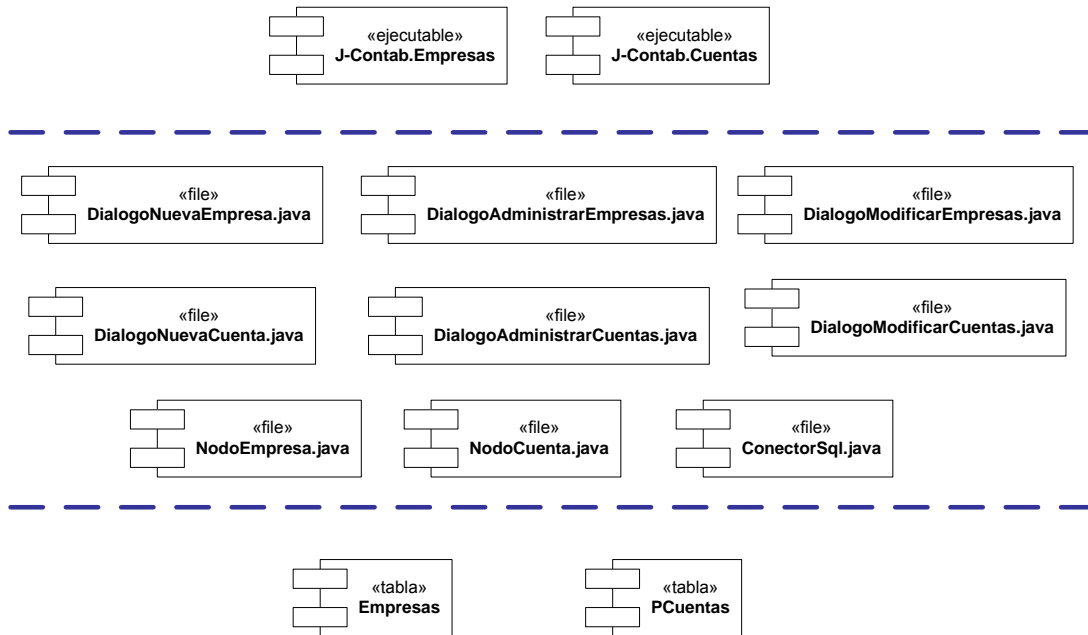


Figura 5.9 Componentes implementados en la segunda construcción.

Construcción 3.

En la tercera construcción se implementaron las interfaces y gestores asociados a los casos de uso Manejar Asientos y Emitir Reportes. Estos subsistemas están compuestos por los ejecutables Asientos y Reportes así como también por los archivos DialogoNuevoAsiento, JPanelPrincipalAsientos, DialogoModificarAsientos, NodoAsiento, GestorDeReportes, JPanelPrincipalBalanceGeneral, JPanelPrincipalBalanceDeComprobación, JPanelPrincipalEstadoDeResultados, JPanelPrincipalLibroDiarioGeneral, JPanelPrincipalLibroMayor, JPanelPrincipalMayorAnalítico y ConectorSQL.

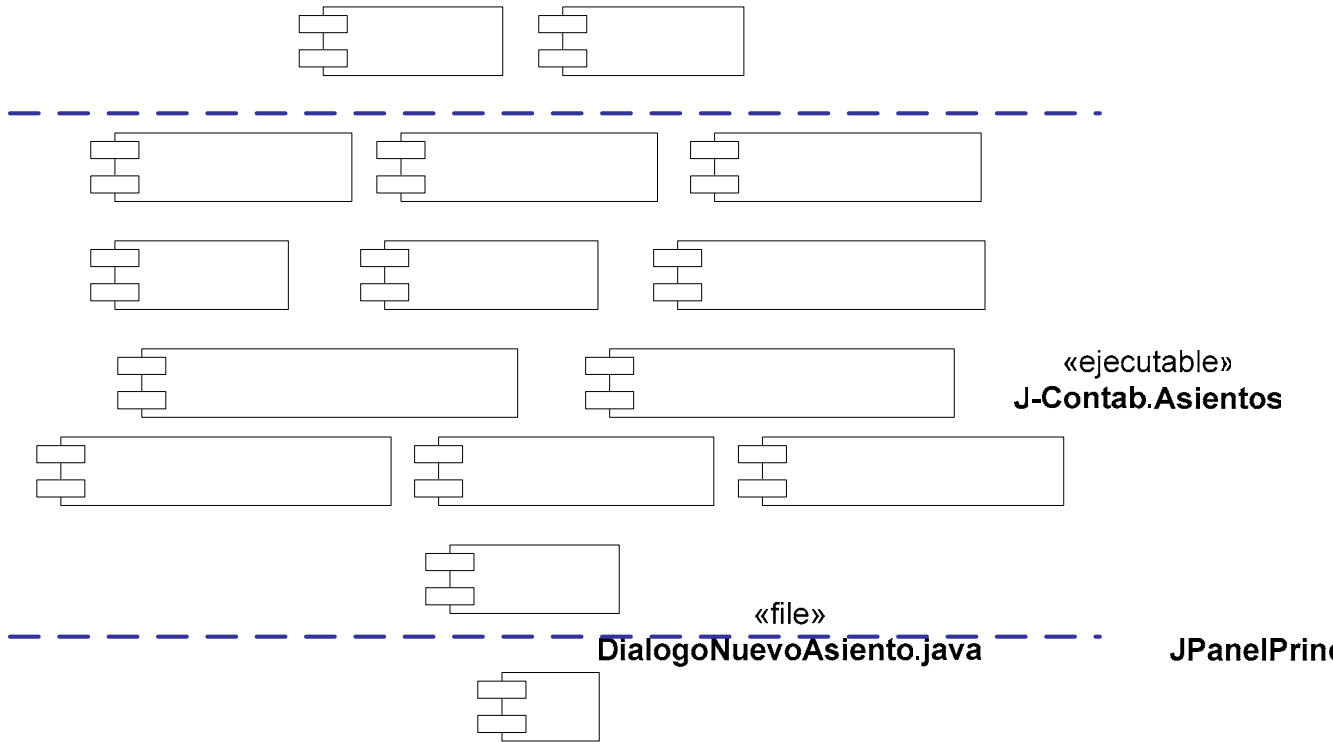


Figura 5.10 Componentes implementados en la tercera construcción.

5.4 PRUEBAS

El objetivo principal de llevar a cabo pruebas del sistema, es asegurarse de que todos los subsistemas de todos los niveles hasta las capas específicas de la aplicación funcionen y no presenten ningún problema.

«file»
JPanelPrincipalBalancedeComprobacion.java
 «file»
JPanelPrincipalLibroDiarioGeneral.java **JPanel**

5.4.1 Partición Equivalente

Conc...



Una partición equivalente es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

Un caso de prueba ideal de manejo simple descubre una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos de caracteres) que, de otra manera, requeriría la ejecución de muchos casos antes de que se observe el error general. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse.

5.4.1.1 Identificación de las clases de equivalencia

Sólo números.

Sólo caracteres.

Caracteres y números.

Ningún carácter.

5.4.1.2 Grupo de Tipos de Entrada de Datos

- Usuario: Hace referencia al campo de entrada del identificador de usuario, recibe caracteres alfanuméricos. Al procesar la entrada, la cadena de caracteres se convierte a minúscula.
- Contraseña: Hace referencia al campo de entrada de la contraseña asociada al identificador de usuario.



Capítulo V - Fase de Construcción

- Tipo: Hace referencia al campo de selección del tipo de privilegio que tendrá un usuario determinado.
- Nombre: Hace referencia al campo de entrada del nombre del usuario/empresa, debe contener al menos un carácter alfanumérico.
- Rif: Hace referencia al campo de entrada del rif de una empresa.
- Dirección: Hace referencia al campo de entrada de la dirección de una empresa.
- Teléfono: Hace referencia al campo de entrada del teléfono de una empresa, solo recibe caracteres numéricos.
- Código cuenta: Hace referencia al campo de entrada del código de una cuenta contable, solo recibe caracteres numéricos.
- Descripción: Hace referencia al campo de entrada de la descripción de una cuenta/asiento contable, Al procesar la entrada, la cadena de caracteres se convierte a mayúscula.
- Documento: Hace referencia al campo de entrada documento.
- Debe/haber: Hace referencia al campo de selección de la naturaleza del asiento pudiendo ser un ingreso o egreso a una determinada cuenta.
- Monto: Hace referencia al campo de entrada del monto de un asiento.



5.4.2 Aplicación de Casos de Prueba

Tabla 5.1 Aplicación de los casos de prueba (1/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
1	1234567		X	1
1	ABCdefg		X	2
1	01ab02C	X		3
1	“ ”		X	4
2	1234567		X	1
2	ABCdefg		X	2
2	01ab02C	X		3
2	“ ”		X	4

Tabla 5.2 Aplicación de los casos de prueba (2/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
3	1234567		X	1
3	ABCdefg		X	2
3	01ab02C		X	3
3	“ ”	X		4
4	1234567		X	1
4	ABCdefg		X	2
4	01ab02C	X		3
4	“ ”		X	4



Tabla 5.3 Aplicación de los casos de prueba (3/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
5	1234567		X	1
5	ABCdefg		X	2
5	01ab02C	X		3
5	“ ”		X	4
6	1234567		X	1
6	ABCdefg		X	2
6	01ab02C	X		3
6	“ ”		X	4

Tabla 5.4 Aplicación de los casos de prueba (4/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
7	1234567	X		1
7	ABCdefg		X	2
7	01ab02C		X	3
7	“ ”		X	4
8	1234567	X		1
8	ABCdefg		X	2
8	01ab02C		X	3
8	“ ”		X	4



Tabla 5.5 Aplicación de los casos de prueba (5/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
9	1234567		X	1
9	ABCdefg		X	2
9	01ab02C	X		3
9	“ ”		X	4
10	1234567		X	1
10	ABCdefg		X	2
10	01ab02C	X		3
10	“ ”		X	4

Tabla 5.6 Aplicación de los casos de prueba (6/6)

GRUPO	CASO DE PRUEBA	VÁLIDA	NO VÁLIDA	CLASE DE EQUIVALENCIA
11	1234567		X	1
11	ABCdefg		X	2
11	01ab02C		X	3
11	“ ”	X		4
12	1234567	X		1
12	ABCdefg		X	2
12	01ab02C		X	3
12	“ ”		X	4



5.4.3 Identificación de los Casos de Prueba

Serán identificados los casos de prueba, estos especifican una forma de probar los escenarios y los casos de uso propuestos. Los casos de prueba serán organizados según la estructura del sistema.

5.4.3.1 Caso de Prueba para el caso de uso Administrar Usuarios

Tabla 5.7 Caso de Prueba para el caso de uso Administrar Usuario

Caso de Prueba	Crear Usuario
Entrada	Usuario: jmalave, Nombre: Jesús Malavé, Contraseña: 123.
Resultado	Los campos son validados, el usuario es agregado al sistema.
Condiciones	Solo puede ser creado por un administrador existente.
Procedimiento	Ingresar usuario, nombre, contraseña en los campos respectivos. Seleccionar el tipo de usuario. Presionar el “Botón Aceptar”



5.4.3.2 Caso de Prueba para el caso de uso Administrar Empresas

Tabla 5.8 Caso de Prueba para el caso de uso Administrar Empresas

Caso de Prueba	Crear Empresa
Entrada	Rif: J-29600142-1, Nombre: JEMARO C.A, Dirección: Carúpano, Teléfono: 0294-1234567.
Resultado	Los campos son validados, la empresa es agregada al sistema.
Condiciones	El usuario debe estar autenticado.
Procedimiento	Ingresar la información de la Empresa. Presionar el “Botón Aceptar”

5.4.3.3 Caso de Prueba para el caso de uso Administrar Cuentas

Tabla 5.9 Caso de Prueba para el caso de uso Administrar Cuenta

Caso de Prueba	Crear Cuenta
Entrada	Código: 10101001, Descripción: Caja General
Resultado	Los campos son validados, la nueva cuenta es agregada al plan de cuentas de la empresa respectiva.
Condiciones	Haber seleccionado previamente una empresa.
Procedimiento	Ingresar los datos de la cuenta. Presionar el “Botón Aceptar”



5.4.3.4 Caso de Prueba para el caso de uso Manejar Asientos

Tabla 5.10 Caso de Prueba para el caso de uso Crear Asientos

Caso de Prueba	Crear Asientos
Entrada	Descripción: Pago factura 100, Documento: Cheque BancoX N° 02020103, Monto: 1390
Resultado	Los campos son validados, el nuevo asiento es agregado al libro diario de la empresa respectiva.
Condiciones	Haber seleccionado previamente una empresa.
Procedimiento	Ingresar los datos del asiento. Presionar el “Botón Aceptar”

5.5 RESUMEN DE LA FASE DE CONSTRUCCIÓN.

En la fase de construcción se ha finalizado con los modelos de análisis, diseño, implementación y prueba. Los casos de uso desarrollados cumplieron con su funcionalidad en la construcción e implementación de sus interfaces y gestores y sobre sus pruebas individuales. Las pruebas de integración del sistema realizadas fueron exitosas. Además se realizó el diseño de la interfaz de usuario para la aplicación, obteniendo así la versión operativa del software ejecutable. Se cumplieron a cabalidad los objetivos para esta fase.



CONCLUSIONES

El sistema contable J-Contab, es el resultado final del proceso de investigación desarrollado a través de la metodología del Proceso Unificado a lo largo de tres fases típicas de dicho proceso, con la finalidad de sistematizar y simplificar las tareas de contabilidad, mejorando la capacidad de procesamiento, disponibilidad y control de los datos contables de una organización.

En lo referido a la descripción de las normas y procedimientos contables asociados a una empresa o actividad productiva, esta permitió obtener un conocimiento claro y específico de las labores básicas llevadas a cabo por un contador, información imprescindible tomada como base para el desarrollo de este proyecto.

La fase de inicio permitió la recopilación de los requisitos del sistema, los cuales permitieron obtener la primera versión de la arquitectura de software del sistema contable J-Contab.

Durante la fase de elaboración del Proceso Unificado, se obtuvo la arquitectura final de software del sistema contable J-Contab.

La fase de construcción del Proceso Unificado permitió el desarrollo del sistema J-Contab, mediante las bases obtenidas en la fase de elaboración, donde se obtuvo como resultado una versión “beta” funcional del sistema.

Se realizó de manera eficiente y clara el diseño de los módulos que conforman el sistema, logrando una arquitectura estable por la cual es sustentado.



RECOMENDACIONES

Trabajar modularmente, para facilitar el desarrollo de componentes funcionales y reutilizables.

Seleccionar algoritmos para el manejo y tratamiento de datos que proporcionen el menor tiempo de procesamiento y respuesta.

Se recomienda a los estudiantes de Ingeniería en Computación, el desarrollo de aplicaciones utilizando herramientas de programación de software libre, tomando como marco de trabajo el Proceso Unificado de Desarrollo de Software.

Se recomienda a los estudiantes de Ingeniería en Computación, el desarrollo de aplicaciones utilizando lenguajes de programación que permitan la portabilidad y la fácil implantación de dichas aplicaciones en entornos productivos bajo el esquema Cliente/Servidor.

Leer el manual de usuario del sistema para tener un conocimiento previo de las funcionalidades que brinda el sistema.



BIBLIOGRAFÍA

[1] Millan L., Garelli L. “Desarrollo de un Software que permita la automatización de las actividades asociadas al Departamento de Admisión y Control de Estudios de la extensión región centro/sur del Núcleo de Anzoátegui de la Universidad de Oriente”. Tesis. (2007).

[2] Otero J. “Desarrollo de una Aplicación para el Control Administrativo de una Organización Farmacéutica”. Tesis. (2004).

[3] Rodríguez C., Farias M. “Diseño de un Sistema de Información para la atención de usuarios de la Biblioteca Pública Central- Julián Temístocles Maza”. Tesis.

[4] Caguaripano H. “Diseño De Un Sistema De Información De Inventario En El Departamento IT (Información Tecnología) de una Empresa Petrolera. Puerto La Cruz, Sector Venecia”. Tesis.

[5] Templeman J., Olsen A. “Microsoft Visual C++ .NET Aprenda Ya”. Primera Edición. Editorial McGraw – Hill. Estados Unidos. (2002).

[6] Jacobson B., Rumbaugh J. “El Proceso Unificado de Desarrollo de Software”. Editorial Pearson Educación, S.A., Madrid. (2000).

[7] Rumbaugh J., Jacobson I. Y Booch G. “El Proceso Unificado de Desarrollo de Software”. Primera Edición. Addison-Wesley. Madrid, España. (2000).



- [8] Elmasri R., Navathe S. “Sistemas de Bases de Datos. Conceptos Fundamentales”. Editorial Pearson Educación. Segunda Edición. México. (2000).
- [9] Brito J. “Contabilidad Básica e Intermedia”. Ediciones Centro de Contadores. Sexta Edición. Venezuela. (2001).
- [10] Wales J. y Sanger L. Enciclopedia Libre, “Software Libre”. http://es.wikipedia.org/wiki/Software_libre, (2006).
- [11] Stallman R. “Software Libre para una Sociedad Libre”. Primera Edición. Editorial Traficante de sueños. Madrid. (2004).



ANEXO A: MANUAL DE USUARIO

A.1 DESCRIPCION DE J-CONTAB

J-Contab es un software contable destinado a sistematizar y simplificar las tareas básicas de contabilidad. J-Contab registra y procesa las transacciones históricas que se generan en una empresa o actividad productiva para ser consultada oportunamente y así facilitar la toma de decisiones y cumplir con obligaciones legales. Para ello solo hay que ingresar la información requerida, como los ingresos y egresos, y hacer que el programa realice los cálculos necesarios.

A.2 PANTALLA PRINCIPAL

J-Contab le permite al usuario manipular fácilmente el sistema a través de su interfaz de usuario basada en menús que le ofrecen todas las opciones del sistema. Su interfaz principal está formado por un menú principal que contiene todas las alternativas disponibles para el usuario.

Para iniciar el sistema debe autenticarse introduciendo un nombre de usuario y contraseña validos.



Figura A.1: Validar Usuario.

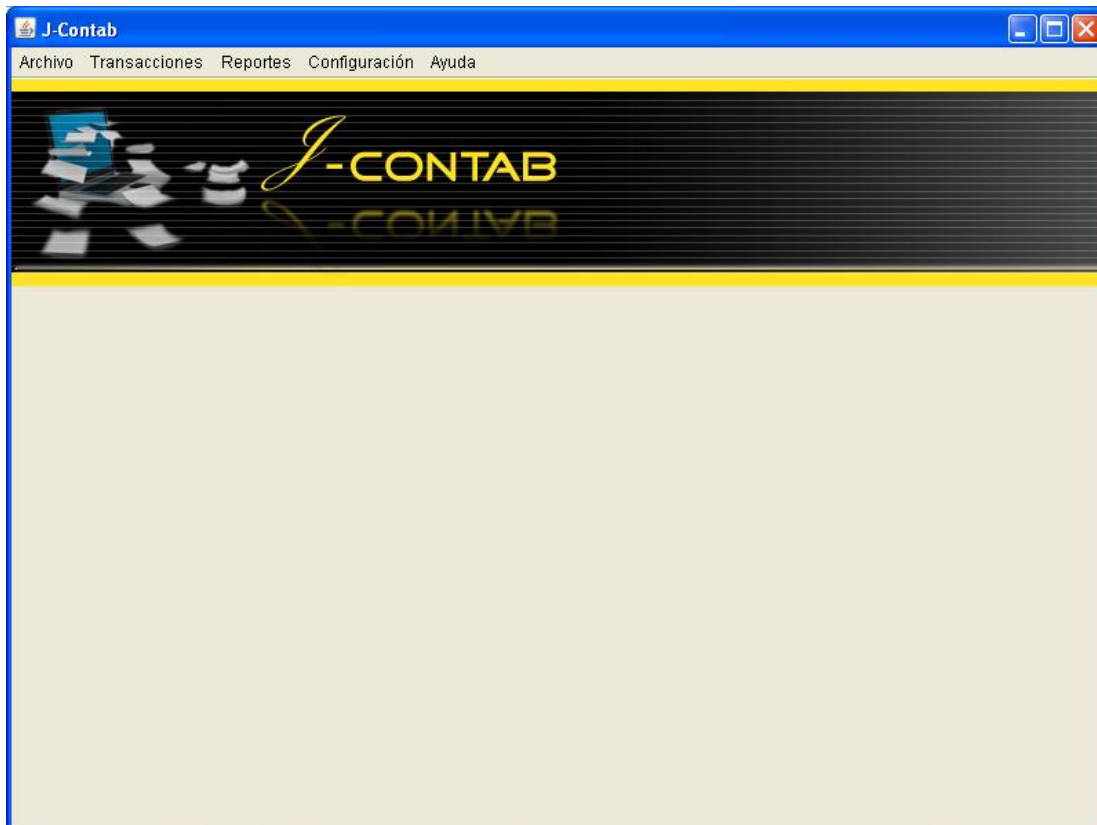


Figura A.2: Pantalla Principal



A.3 MENÚ

El menú principal del sistema consta de cinco opciones que agrupan las tareas que puede llevar a cabo de acuerdo a sus funciones. A través del menú principal se puede acceder a todas las opciones que brinda el sistema además de poder consultar la ayuda y conocer acerca del sistema. En este menú se encuentra:

- Archivo
- Transacciones
- Reportes
- Configuración
- Ayuda

A.3.1 Menú Archivo

Mediante este menú se puede acceder a las opciones:

- Seleccionar Empresa
- Cerrar Empresa
- Exportar Data
- Importar Data
- Salir

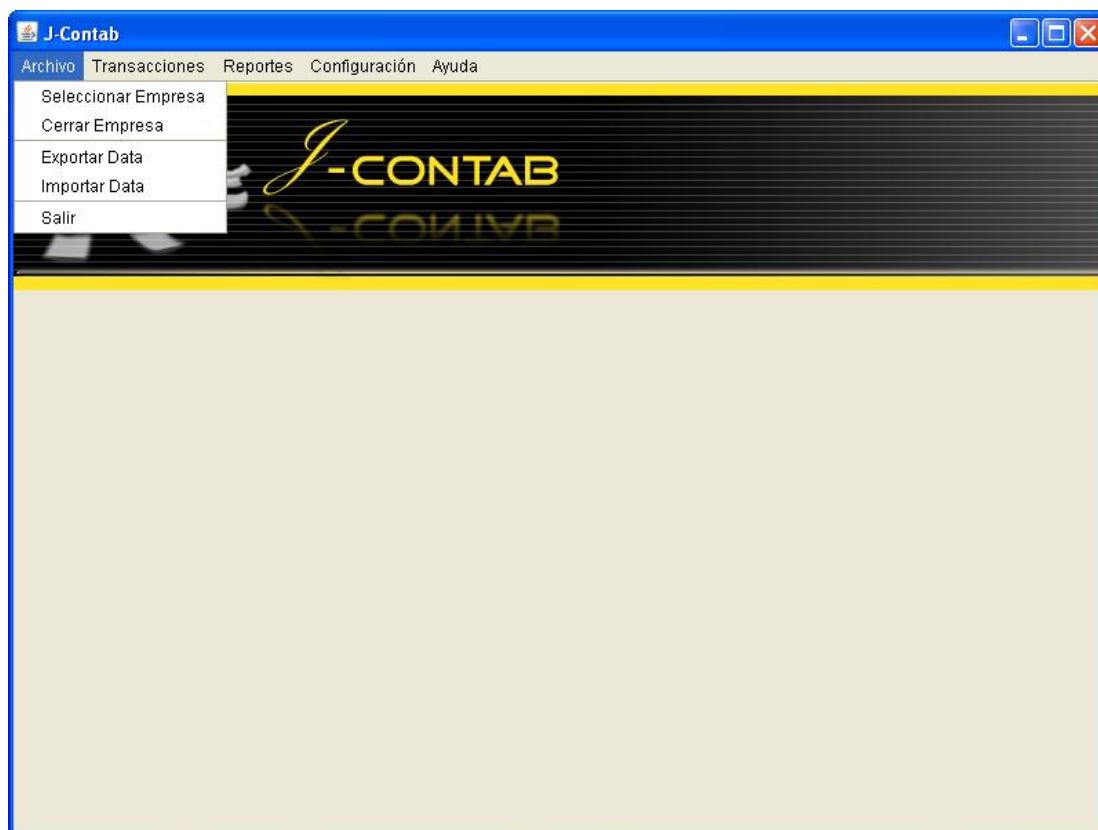


Figura A.3: Menú Archivo

A.3.1.1 Opción Seleccionar Empresa

Permite seleccionar una de las empresas cargadas en el sistema, una vez seleccionada se pueden realizar las diversas operaciones permitidas, tales como procesar comprobantes, modificar cuentas, emitir reportes, etc.

Para seleccionar una empresa, elija la opción **Seleccionar Empresa** del menú archivo.

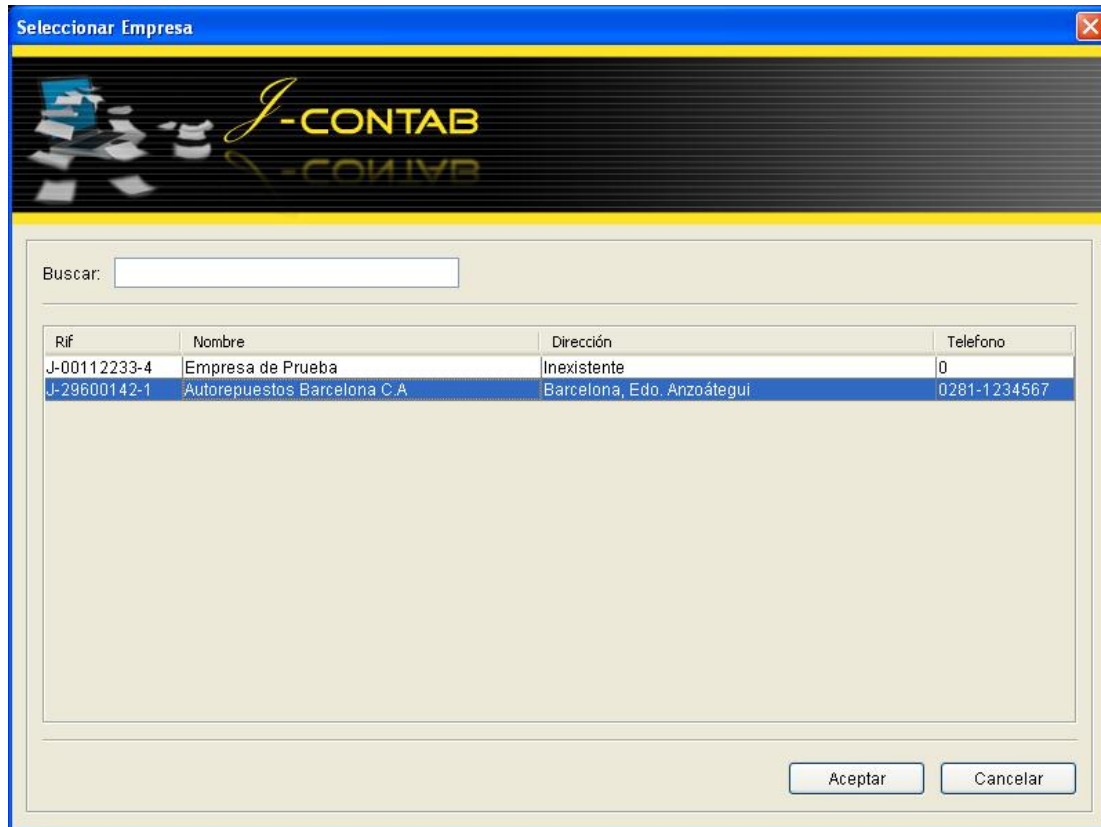


Figura A.4: Seleccionar Empresa

Elija una de las empresas disponibles y presione el botón aceptar.

A.3.1.2 Opción Cerrar Empresa

Cierra la empresa con la que actualmente se esta trabajando.

A.3.1.3 Opción Exportar Data

Permite exportar la data de las empresas existente en el sistema, a un archivo de respaldo.



Para exportar la data de las empresas, elija la opción **Exportar Data** del menú archivo. Seguidamente seleccione la ruta donde desea guardar el archivo y presione el botón guardar.

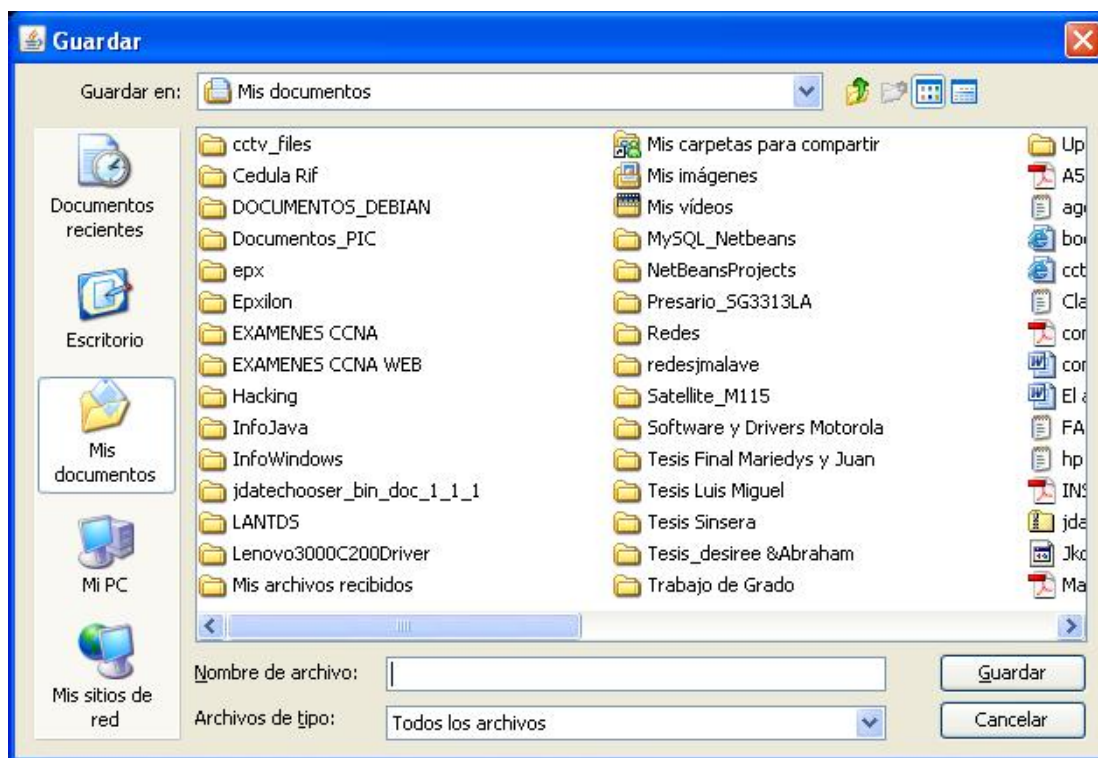


Figura A.5: Guardar

A.3.1.4 Opción Importar Data

Permite importar la data de las empresas al sistema, desde un archivo de respaldo.

Para importar la data de las empresas desde un archivo, elija la opción **Importar Data** del menú archivo. Seleccione la ruta desde donde desea abrir el archivo y presione el botón abrir.

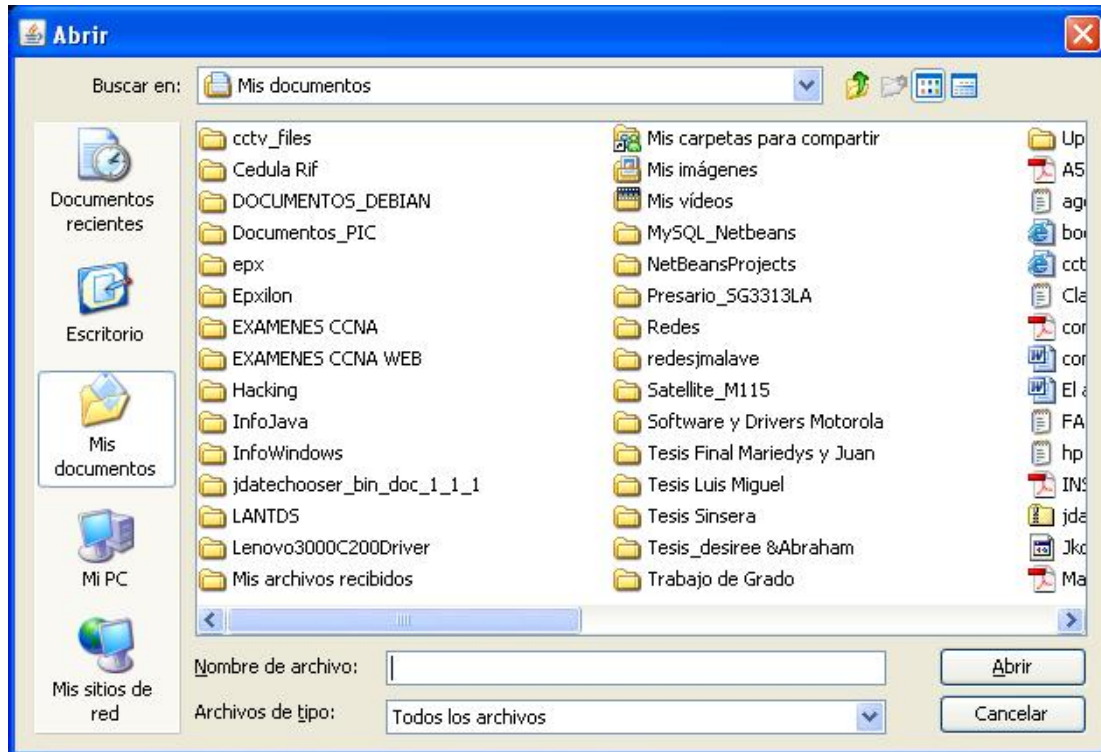


Figura A.6: Abrir

A.3.1.5 Opción Salir

Cierra la aplicación J-Contab y todos sus componentes.

A.3.2 Menú Transacciones

Mediante este menú se puede acceder a la opción Procesar Comprobantes.

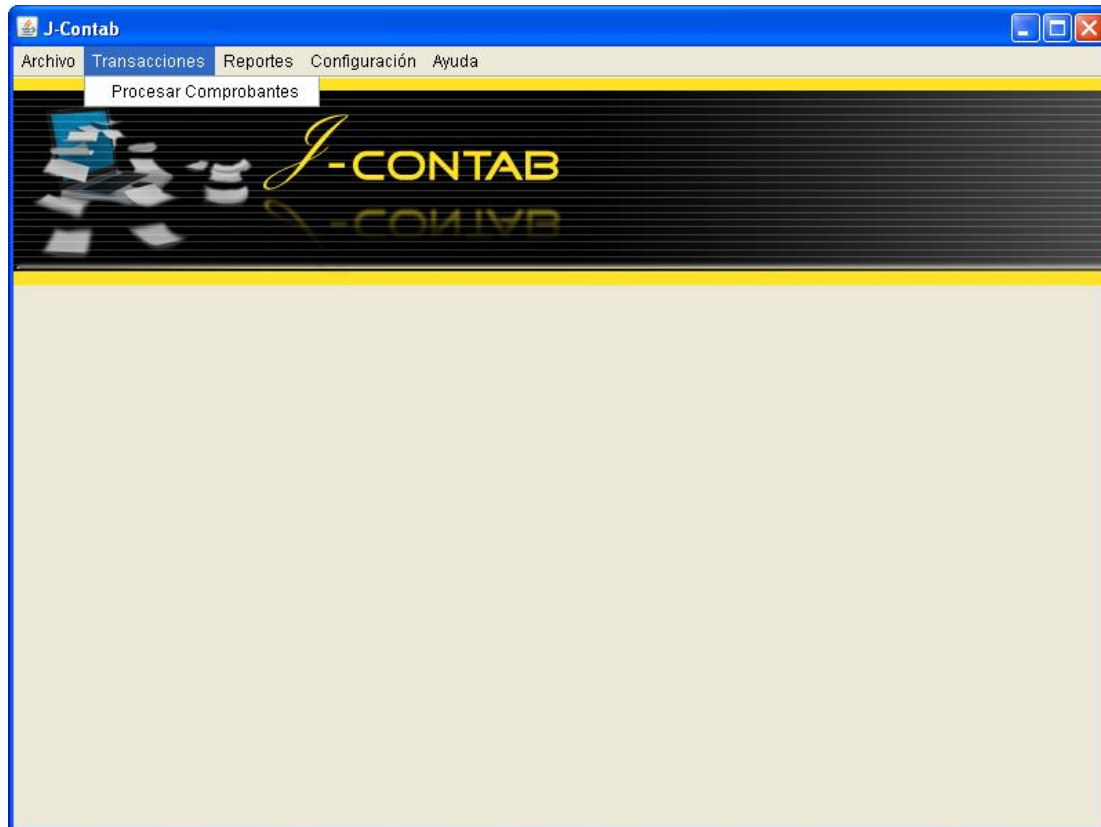


Figura A.7: Menú Transacciones

A.3.2.1 Opción Procesar Comprobantes

Permite ingresar, modificar o eliminar los asientos contables de una empresa existente en el sistema.

Para procesar los comprobantes de una empresa, elija la opción **Procesar Comprobantes** del menú Transacciones.

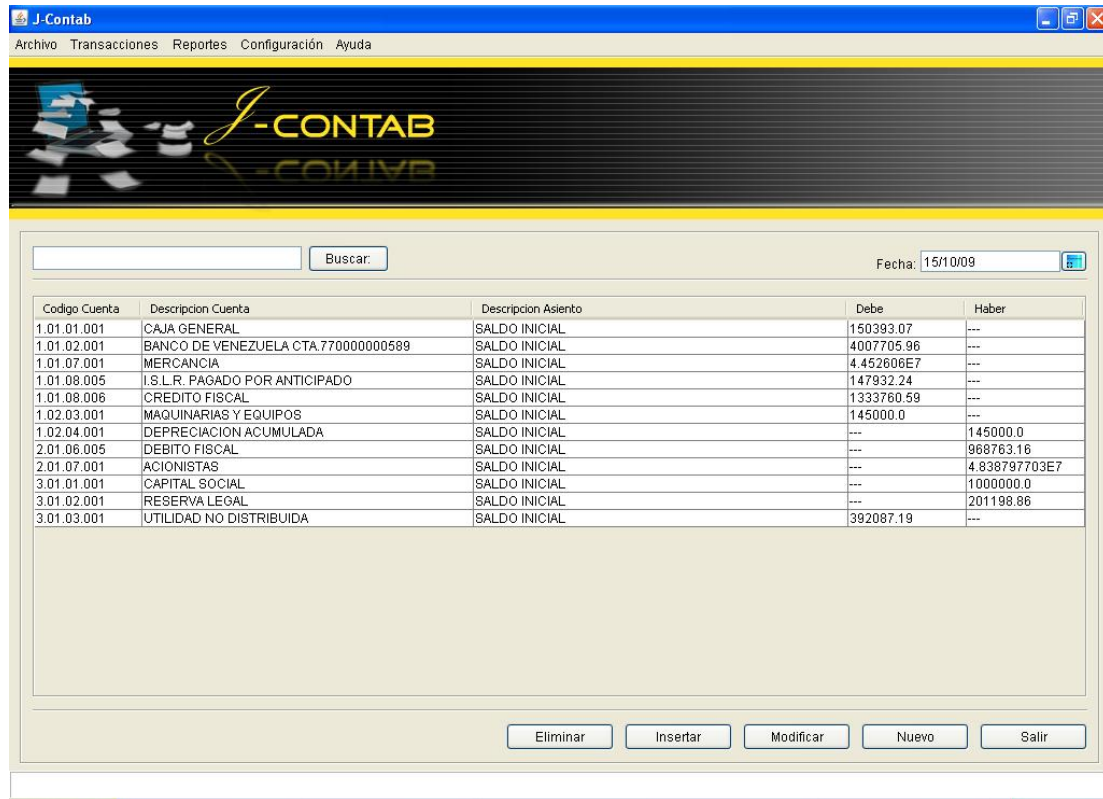


Figura A.8: Procesar Comprobantes

1. Para agregar un nuevo asiento contable, presione el botón **Nuevo**, ingrese los datos solicitados y presione el botón aceptar.



Nuevo Asiento

J-CONTAB

Codigo Cuenta:

Descripcion Cuenta:

Descripcion Asiento:

Monto: Debe ▼

Aceptar Cancelar

Figura A.9: Nuevo Asiento

2. Para modificar un asiento contable, presione el botón **Modificar**, modifique los datos actuales y presione el botón aceptar.



Modificar Asiento

J-CONTAB

Codigo Cuenta:

Descripcion Cuenta:

Descripcion Asiento:

Monto: Debe

Aceptar Cancelar

Figura A.10: Modificar Asiento

3. Para insertar un asiento contable en una posición específica de la lista de asientos, presione el botón **Insertar**, ingrese los datos solicitados y presione el botón aceptar.



Insertar Asiento

Codigo Cuenta:

Descripcion Cuenta:

Descripcion Asiento:

Monto: Debe

Insertar en fila N°:

Aceptar Cancelar

Figura A.11: Insertar Asiento

4. Para eliminar un asiento contable, seleccione el asiento en la lista de asientos contables, presione el botón **Eliminar**, confirme la operación presionando el botón aceptar del dialogo de confirmación.
5. Presione el botón **Salir** para volver a la pantalla principal.

A.3.3 Menú Reportes

Mediante este menú se puede acceder a las opciones:

- Libro Diario



- Libro Mayor
- Mayor Analítico
- Balance General
- Estado de Resultados
- Balance de Comprobación

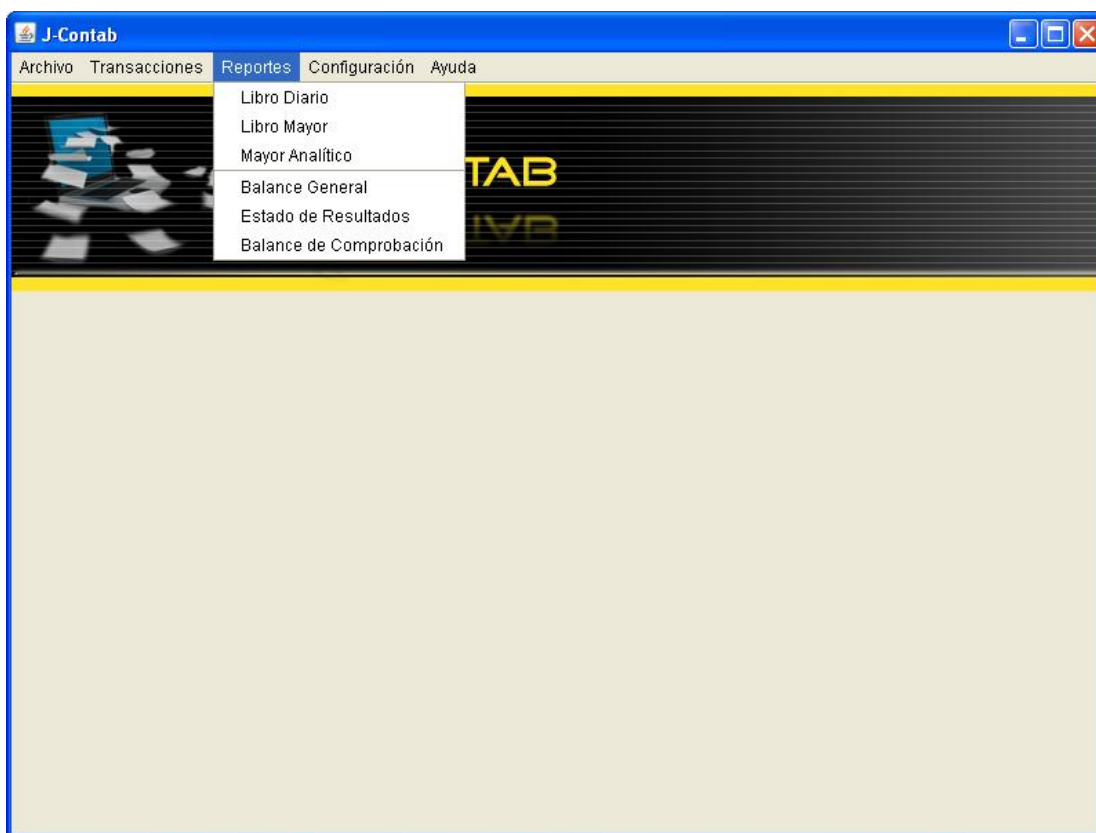


Figura A.12: Menú Reportes

A.3.3.1 Opción Libro Diario

Muestra el libro diario de una empresa existente en el sistema y previamente seleccionada.



A.3.3.2 Opción Libro Mayor

Muestra el libro mayor de una empresa existente en el sistema y previamente seleccionada.

A.3.3.3 Opción Mayor Analítico

Muestra todos los movimientos de una cuenta contable específica, perteneciente a una empresa existente en el sistema y previamente seleccionada.

A.3.3.4 Opción Balance General

Muestra el balance general de una empresa existente en el sistema y previamente seleccionada.

A.3.3.5 Opción Estado de Resultados

Muestra el estado de ganancias y pérdidas de una empresa existente en el sistema y previamente seleccionada.

A.3.3.6 Opción Balance de Comprobación

Muestra el balance de comprobación de saldos de una empresa existente en el sistema y previamente seleccionada.



A.3.4 Menú Configuración

Mediante este menú se puede acceder a las opciones:

- Usuarios
- Empresas
- Cuentas

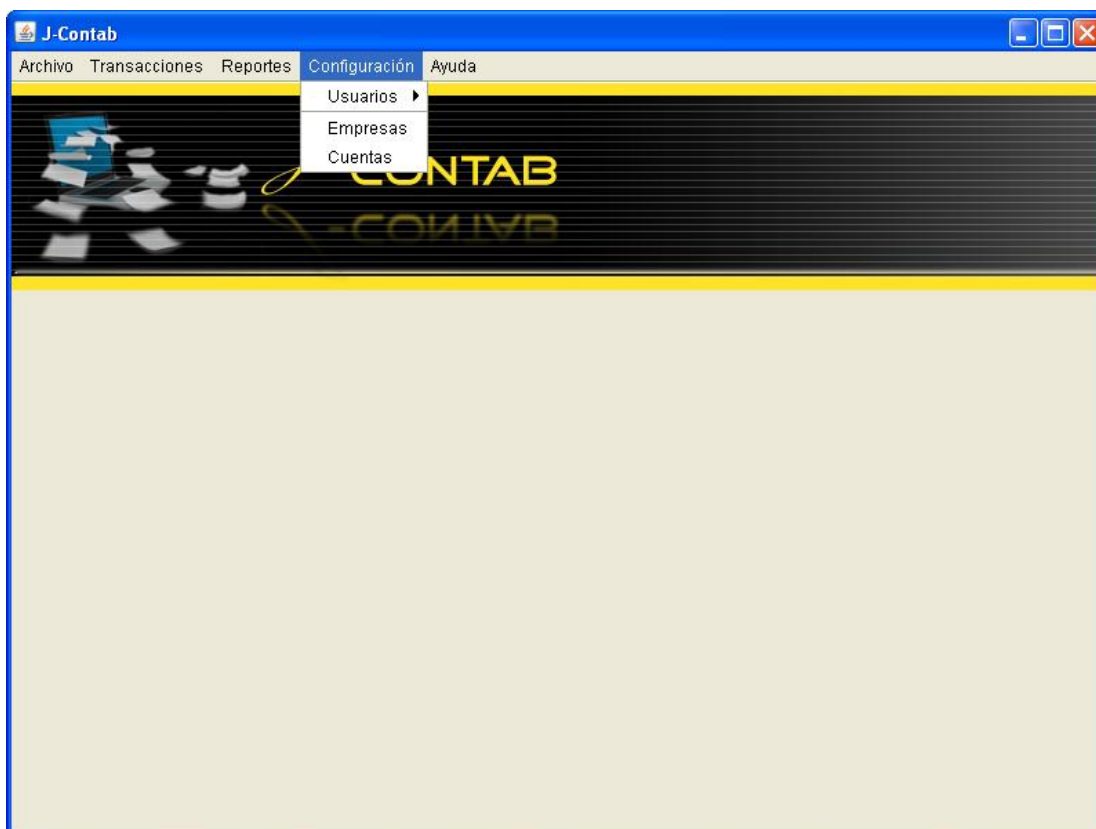


Figura A.13: Menú Configuración

A.3.4.1 Submenú Usuarios



Este submenú permite realizar operaciones sobre los usuarios del sistema tales como ingresar usuario, eliminar y cambiar contraseña.

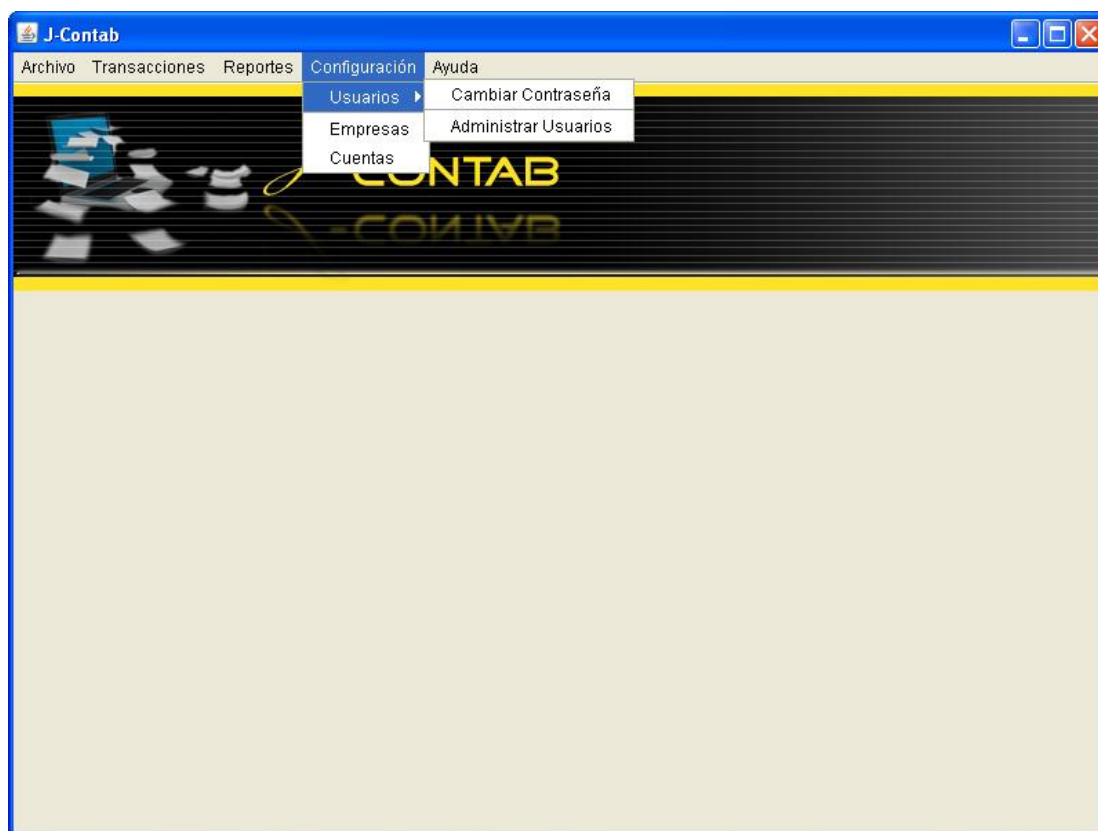


Figura A.14: Submenú Usuarios

1. Para cambiar la contraseña del usuario actual, elija la opción **Cambiar Contraseña**, ingrese los datos solicitados y presione el botón Aceptar.



Cambiar Contraseña

J-CONTAB

Contraseña actual:

Nueva contraseña:

Confirmar:

Aceptar Cancelar

Figura A.15: Cambiar Contraseña

2. Para agregar o eliminar un usuario, elija la opción **Administrar Usuarios**.

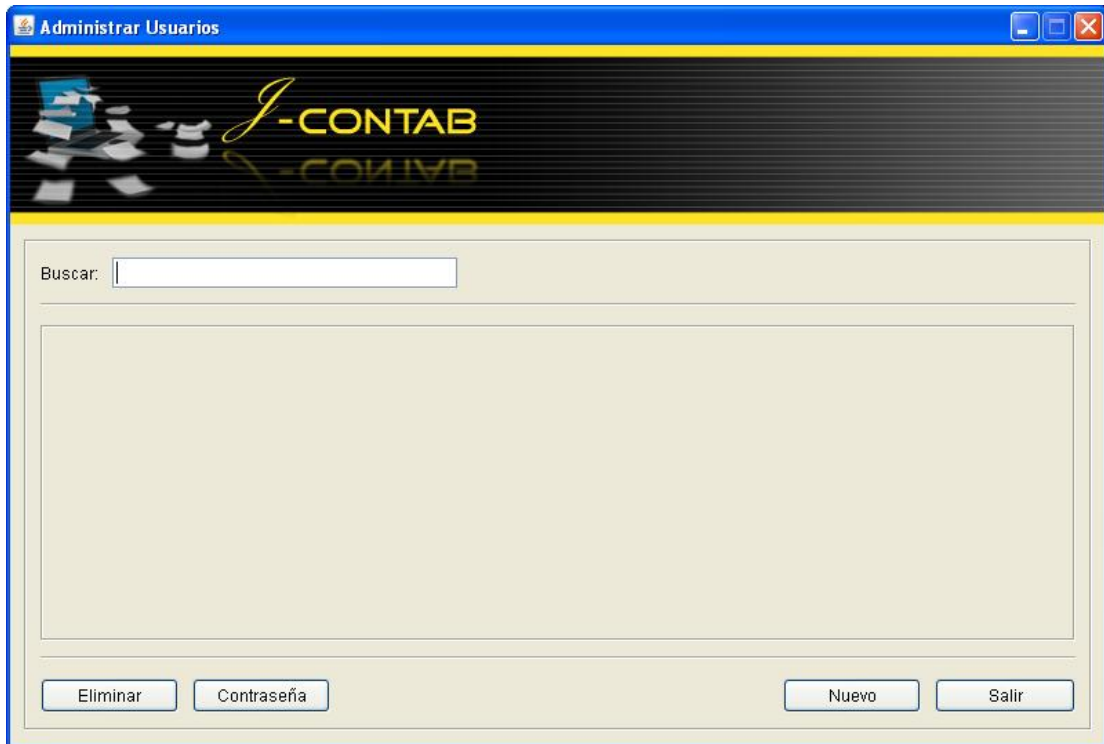


Figura A.16: Administrar Usuarios

A.3.4.2 Opción Empresas

Permite ingresar, modificar o eliminar las empresas en el sistema.

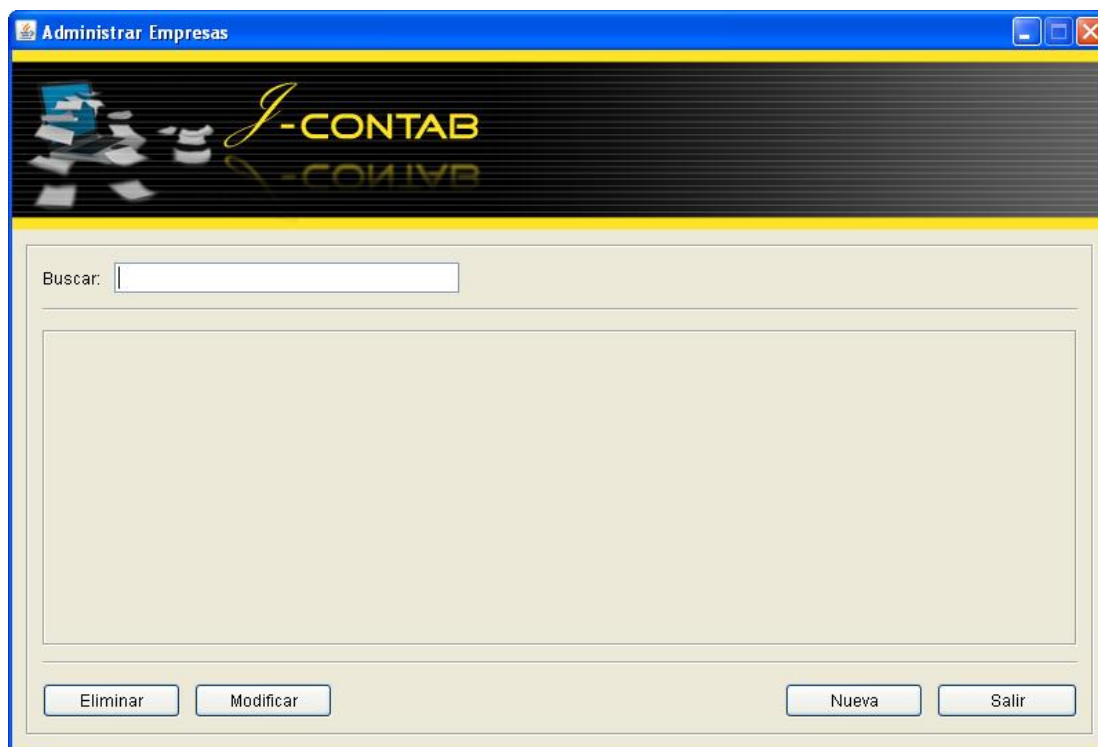


Figura A.17: Administrar Empresas

1. Para agregar una nueva empresa, presione el botón **Nueva**, ingrese los datos solicitados y presione el botón aceptar.



Nueva Empresa

Rif:

Nombre:

Dirección:

Telefono:

Importar... Aceptar Cancelar

Figura A.18: Nueva Empresa

2. Para modificar los datos de una empresa existente, presione el botón **Modificar**, modifique los datos actuales y presione el botón aceptar.



Modificar Empresa

J-CONTAB

Rif:

Nombre:

Dirección:

Telefono:

Aceptar Cancelar

Figura A.19: Modificar Empresa

3. Para eliminar una empresa, seleccione la empresa en la lista de empresas, presione el botón **Eliminar**, confirme la operación presionando el botón aceptar del dialogo de confirmación.
4. Presione el botón **Salir** para volver a la pantalla principal.

A.3.4.3 Opción Cuentas

Permite ingresar, modificar o eliminar las cuentas contables de una empresa en el sistema.

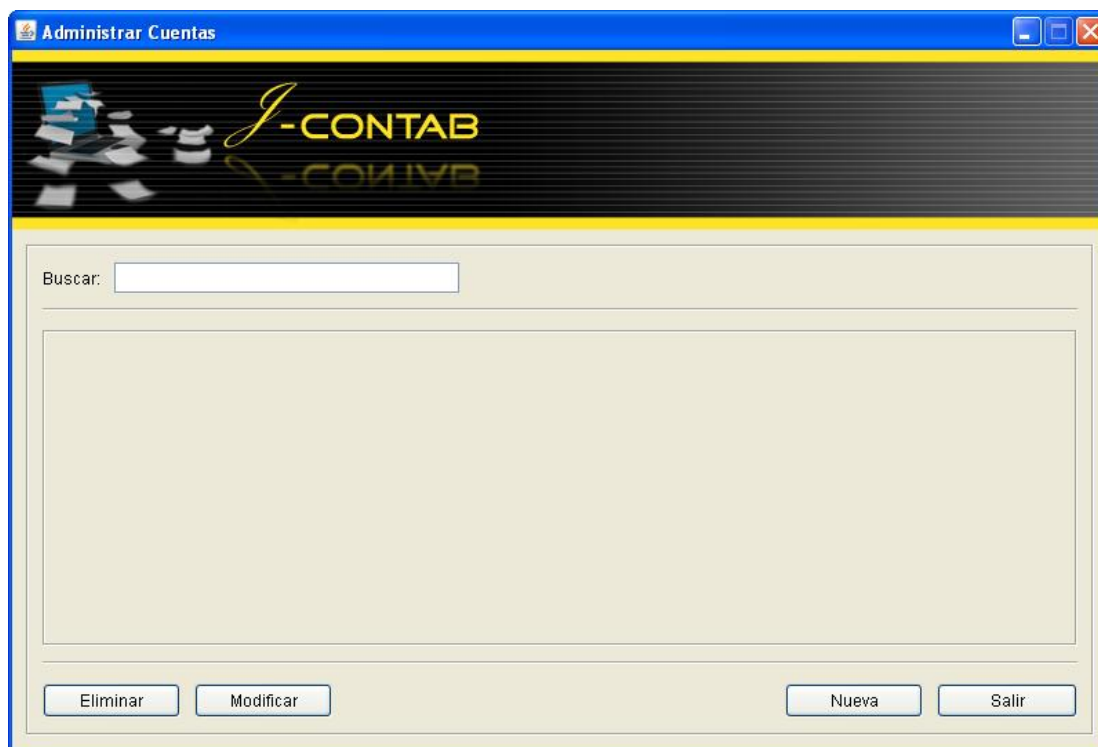


Figura A.20: Administrar Cuentas

1. Para agregar una nueva cuenta contable, presione el botón **Nueva**, ingrese los datos solicitados y presione el botón aceptar.



Nueva Cuenta

J-CONTAB
COMIVR

Codigo Cuenta:

Descripcion:

Aceptar Cancelar

Figura A.21: Nueva Cuenta

2. Para modificar una cuenta contable, presione el botón **Modificar**, modifique los datos actuales y presione el botón aceptar.

Modificar Cuenta

J-CONTAB
COMIVR

Codigo Cuenta:

Descripcion:

Aceptar Cancelar

Figura A.22: Modificar Cuenta



3. Para eliminar una cuenta contable, seleccione la cuenta en la lista de cuentas contables, presione el botón **Eliminar**, confirme la operación presionando el botón aceptar del dialogo de confirmación.

4. Presione el botón **Salir** para volver a la pantalla principal.

A.3.5 Menú Ayuda

Mediante este menú se puede acceder a las opciones:

- Ayuda
- Acerca de J-Contab

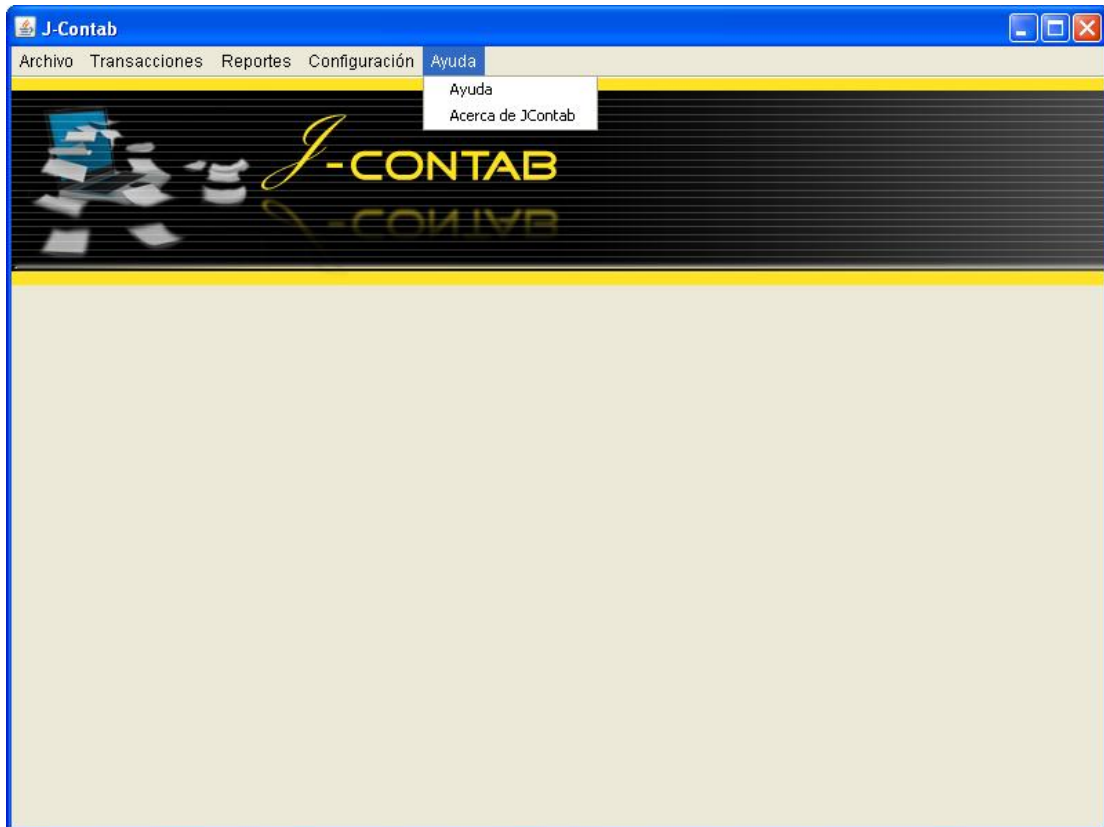


Figura A.23: Menú Ayuda

A.3.5.1 Opción Ayuda

Muestra la ayuda del sistema.

A.3.5.2 Opción Acerca de J-Contab

Muestra información acerca del sistema, donde puede conocer el nombre, desarrolladores y licencia.



A.4 REQUISITOS OPERACIONALES DE HARDWARE Y SOFTWARE

Requisitos mínimos de hardware

- 256 Mb de memoria RAM.
- Teclado y mouse estándar.
- Monitor a color de 16 bits con resolución de 800 x 600 píxeles.

Requisitos mínimos de software

Linux Kernel 2.4, Windows XP, Mac OS.

Java Runtime Environment 5 (JRE).

MySQL 5.0.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	DESARROLLO DE UN SOFTWARE QUE PERMITA LA AUTOMATIZACIÓN DEL REGISTRO Y PROCESAMIENTO DE LAS OPERACIONES CONTABLES HISTÓRICAS QUE SE GENERAN EN UNA EMPRESA O ACTIVIDAD PRODUCTIVA.
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
MALAVE C., JESUS M.	CVLAC: 16.841.728 E MAIL: jesus.malave@epxilon.net
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

CONTABILIDAD SOFTWARE CONTABLE LIBROS UML RUP
ESTRUCTURA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

RESUMEN (ABSTRACT):

El presente proyecto consistió en el desarrollo de un software que permitirá la automatización del registro y procesamiento de las operaciones contables históricas que se generan en una empresa o actividad productiva. La sistematización y automatización de la contabilidad tiene como finalidad liberar al contador de esta fase del proceso, permitiéndole dedicar más tiempo a labores de mayor importancia, como el análisis e interpretación de la información. El software se elaboró utilizando tecnologías de software libre, rigiéndose por el decreto 3390, el cual dispone que “La Administración Pública Nacional deberá emplear prioritariamente Software Libre desarrollado con Estándares Abiertos, en todos sus Sistemas, Proyectos y Servicios Informáticos”. El desarrollo del proyecto estuvo dirigido por el Proceso Unificado de Desarrollo de Software que permite dividir el proyecto en fases y éstas en flujos de trabajos, y el lenguaje de modelado UML. Para construir el software se utilizó Java como lenguaje de programación y como manejador de bases de datos MySQL.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Dorta, Pedro.	ROL	CA	AS X	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				
Carrasquero, Manuel.	ROL	CA	AS	TU	JU X
	CVLAC:				
	E_MAIL				
	E_MAIL				
Saettone, Mónica.	ROL	CA	AS	TU	JU X
	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	10	23
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
Tesis.doc	application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J K L
M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x
y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente, Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario

Malavé Caraballo., Jesús Miguel.

AUTOR

Dorta, Pedro.

TUTOR

Carrasquero, Manuel.

JURADO

Saettone, Mónica.

JURADO

POR LA SUBCOMISION DE TESIS