

Universidad de las Ciencias Informáticas

Facultad 3



Título: Análisis y Diseño de una herramienta que permita generar reportes en la Hoja de Cálculo del OpenOffice a partir de los datos almacenados en los subsistemas Contabilidad y Costo y Procesos del sistema Cedrux.

Trabajo de Diploma para optar por el título de

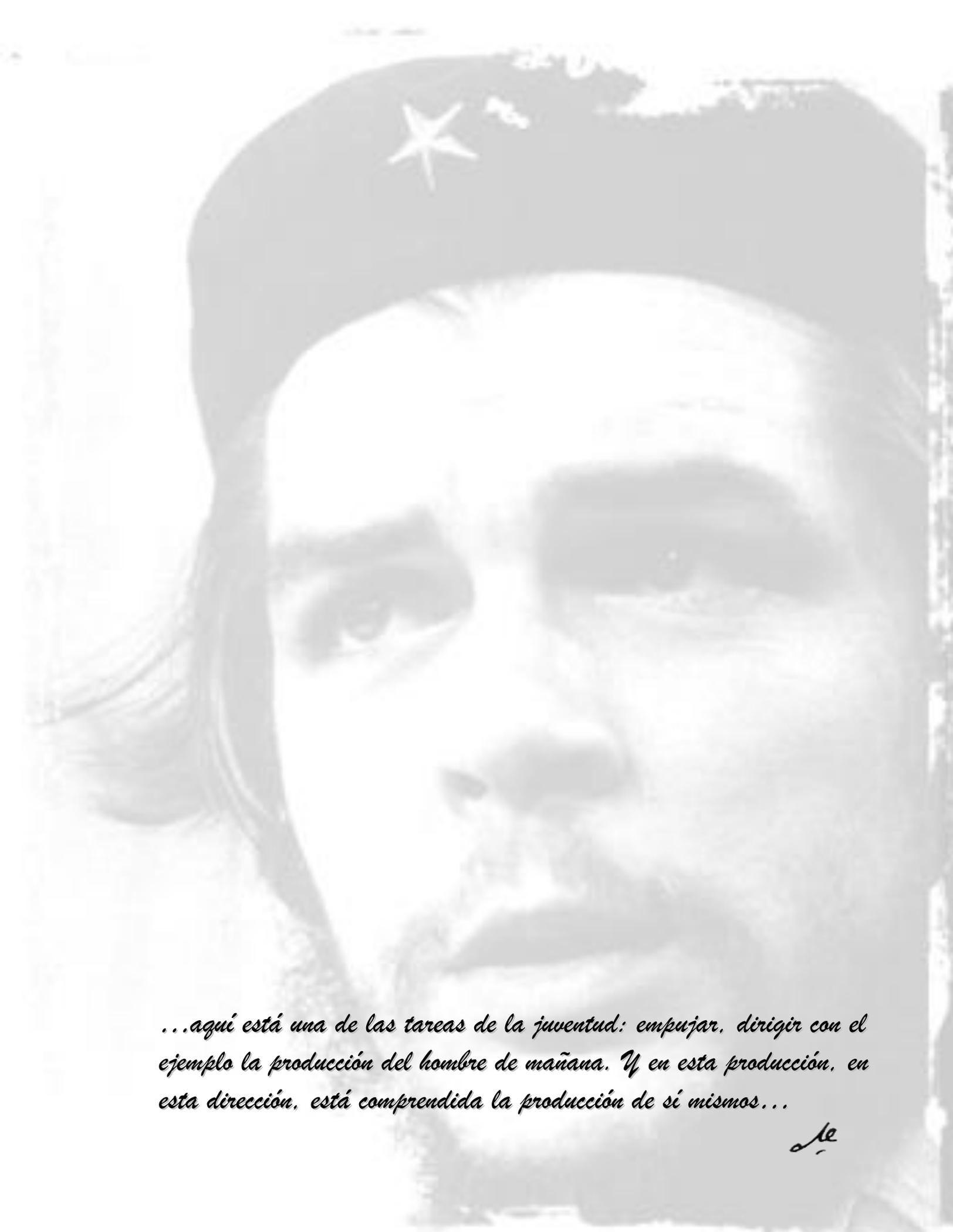
Ingeniero en Ciencias Informáticas

Autor(es): Aylin Chapman Cardoso.

Tutor(es): Ing. René R. Bauta Camejo.

Ing. Leydis Castellanos Cobas.

Junio, 2011



...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos...

Le

Declaración de Autoría

Declaro que soy la única autora del trabajo “Análisis y Diseño de una herramienta que permita generar reportes en la Hoja de Cálculo del OpenOffice a partir de los datos almacenados por los subsistemas Contabilidad y Costo y Procesos del sistema Cedrux” y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ing. Leydis Castellanos Cobas.

Tutora

Ing. René R. Bauta Camejo

Tutor

Aylin Chapman Cardoso

Autora

Agradecimientos

A mi familia: Por estar siempre pendiente de mí en todo momento, brindarme su apoyo cuando lo he necesitado y servirme de guía.

A mi antiguo grupo 4107 aunque ya quedemos pocos gracias por todo su apoyo, paciencia y por compartir tantas cosas lindas.

En especial a las que me han aguantado mi carácter durante estos cinco cursos, que se que eso es una dura tarea, a Mayelín, Zerelda, Yesmin, Yanet, Meylin, Dina, Betty y a Chabela que viene conmigo desde antes. Las considero excelentes amigas, juntas hemos compartido grandiosos momentos no las olvidaré.

A todas las amistades y las personas especiales que en el transcurso de estos cinco años me han demostrado que puedo contar con ellos para lo que necesite.

A mi piquete de fiestas, no voy a olvidar los buenos momentos que pasamos juntos.

A mis profesores: Que con su ejemplo y ayuda profesional me han encauzado en la vida y de esa forma lograr un crecimiento en cada una de las metas y tareas que enfrentamos juntos.

En especial a mis tutores: Ing. Leydis Castellanos Cobas y Ing. René R. Bauta Camejo por la confianza depositada en momentos trascendentales del proceso investigativo, por su amistad y exigencia.

A todas las personas que de una forma u otra, hicieron posible la realización de este trabajo.

A todos eternamente muchas gracias.

Dedicatoria

A mi madre y mi padre

Razón de ser de mi existencia y fuente de inspiración. Gracias por no perder la paciencia conmigo y por quererme tanto. Por darme siempre todo lo que ha estado a su alcance, por estar siempre ahí en el momento preciso y por tener el orgullo de ser su hija.

A mi hermano

Por contagiarme con su alegría y considerarme su paradigma. Sabes que te quiero con la vida y no tengo palabras para explicarlo.

A mi abuela

Ana Julia, por darme su amor y aliento para llegar a la meta final. Eres mi verdadero ejemplo a seguir en la vida por tu audacia y perseverancia ante las cosas.

A mis tíos

Tere, Tony y Norberto, por mantenerse al tanto de mis resultados en los estudios y prestarme su ayuda siempre que la he necesitado.

A mis primos

En especial a Julio Cesar que más que mi primo es mi hermano y mi ejemplo a seguir.

A la Revolución

Sin la cual hubiera sido una utopía la aspiración de obtener este título.

Resumen

Actualmente la Universidad se encuentra desarrollando un Sistema Integral de Gestión Empresarial llamado Cedrux, que cuenta con varios subsistemas entre los que se encuentran el de Contabilidad y Costos y Procesos. A través de las Hojas de Cálculo se quiere facilitar el manejo de los datos almacenados en estos subsistemas. Con estos programas se realizan la mayor parte de los cálculos en las empresas, los mismos se han convertido en un eslabón principal a la hora de trabajar con datos estadísticos porque es una herramienta que permite realizar una gran cantidad de funciones contables mediante varias operaciones matemáticas.

Para el desarrollo de este trabajo se propone como solución el análisis y diseño de una herramienta que permita generar reportes a través de la Hoja de Cálculo del OpenOffice utilizando los datos almacenados por los subsistemas Contabilidad y Costos y Procesos del sistema Cedrux.

La Hoja de Cálculo sobre la cual se quiere desarrollar el sistema es la Calc de OpenOffice, la misma posee muchas facilidades y funcionalidades. Esta herramienta que se quiere desarrollar es igual que la existente para el Excel lo que enfocada a otro Sistema Operativo (SO), en este caso para el SO Linux.

Se llega a esta propuesta mediante un estudio que se realizó sobre las Hojas de Cálculo, el cual permitió conocer la importancia que tienen las mismas para los sistemas contables.

PALABRAS CLAVE

Análisis, Cedrux, Contabilidad, Costos, Diseño, Hojas de Cálculo, Procesos, Subsistema

Índice

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN.....	6
1.2 HOJAS DE CÁLCULO	6
1.2.1 Editores de Hojas de Cálculo más utilizadas en el mundo.....	8
1.3 COMPLEMENTOS PARA HOJAS DE CÁLCULO	8
1.3.1 Complemento EZAnalyze.....	10
1.3.2 Complemento cubiFOR.....	11
1.3.3 MACRO OOO STATISTICS.	12
1.4 METODOLOGÍA DE DESARROLLO	13
1.5 TRAZABILIDAD DE REQUERIMIENTOS.....	13
1.5.1 Técnicas generales para la captura de requisitos	14
1.5.2 Técnicas generales de validación de requisitos.....	15
1.5.3 Actividades del proceso de definición de requisitos	16
1.5.4 Métricas de la calidad para la especificación de requisitos	16
1.5.5 Métricas para la validación del diseño.....	19
1.6 LENGUAJE DE MODELADO Y DESARROLLO	19
1.6.1 Lenguaje de Modelado de Clases	19
1.6.2 Lenguaje de Desarrollo.....	20
1.7 HERRAMIENTAS PARA EL DESARROLLO	21
1.7.1 Herramientas CASE (Computer Aided Software Engineering).....	21
1.7.2 Herramienta de desarrollo colaborativo	22
1.7.3 Entorno integrado de desarrollo	22
1.8 SISTEMA GESTOR DE BASE DE DATOS	23
1.9 PATRONES	23
1.9.1 ¿Qué es un Patrón?	23
1.9.2 Patrones de reutilización de especificación de requisitos.....	23
1.9.3 Patrones de especificación de requisitos.....	24
1.9.4 Patrones de diseño	25
1.10 CONCLUSIONES.....	27
CAPITULO 2. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN	28
2.1 INTRODUCCIÓN.....	28
2.2 ANÁLISIS.....	28
2.3 IDENTIFICACIÓN DE LOS REQUISITOS	28
2.4 MODELO CONCEPTUAL	29
2.5 REQUISITOS FUNCIONALES	30
2.6 REQUISITOS NO FUNCIONALES.....	30
2.6.1 Requisitos de Software	30
2.6.2 Requisitos de Hardware	30
2.6.3 Restricciones en el diseño y la implementación	31
2.6.4 Requisitos de Seguridad	31

2.6.5 <i>Requisitos de Usabilidad</i>	31
2.7 TRAZABILIDAD DE REQUISITOS: MATRIZ DE SEGUIMIENTO DE DEPENDENCIAS	31
2.8 ESPECIFICACIÓN DE REQUISITOS	32
2.9 DISEÑO	32
2.10 DESCRIPCIÓN DE LAS ENTRADAS	33
2.11 DISEÑO DE LA SOLUCIÓN EN TÉRMINOS DE COMPONENTES	33
2.12 DISEÑO DE CLASES	34
2.13 MODELO DE DATOS	36
2.14 PATRONES DE DISEÑO EMPLEADOS	37
2.15 DESCRIPCIÓN DE LAS CLASES Y FUNCIONALIDADES EL COMPONENTE	38
2.16 DESCRIPCIÓN GENERAL DEL FUNCIONAMIENTO DE COMPONENTE	39
2.17 CONCLUSIONES	39
CAPITULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	41
3.1 INTRODUCCIÓN	41
3.2 APLICACIÓN DE TÉCNICAS DE VALIDACIÓN DE REQUISITOS	41
3.2.1 <i>Prototipo de interfaz</i>	41
3.2.2 <i>La Revisión Técnica Formal (RTF)</i>	42
3.3 MÉTRICAS PARA LA VALIDACIÓN DEL DISEÑO DEL MODELO PROPUESTO	42
3.3.1 <i>Tamaño Operacional de Clases (TOC)</i>	42
3.3.2 <i>Relaciones entre clases (RC)</i>	46
3.4 CONCLUSIONES	48
CONCLUSIONES	49
RECOMENDACIONES	50
REFERENCIAS BIBLIOGRÁFICAS	51
BIBLIOGRAFÍA	52
ANEXOS	55
ANEXO 1. DESCRIPCIÓN DE LA CLASE CENTRO	55
ANEXO 2. DESCRIPCIÓN DE CLASE CONCEPTOS.	55
ANEXO 3. DESCRIPCIÓN DE LA CLASE CONTRATOS.	56
ANEXO 4. DESCRIPCIÓN DE LA CLASE CUENTA.	56
ANEXO 5. DESCRIPCIÓN DE LA CLASE EJERCICIO.....	57
ANEXO 6. DESCRIPCIÓN DE LA CLASE ELEMENTOSGATOS.	58
ANEXO 7. DESCRIPCIÓN DE LA CLASE ENTIDAD.	58
ANEXO 8. DESCRIPCIÓN DE LA CLASE FUNCION.	59
ANEXO 9. DESCRIPCIÓN DE LA CLASE PARAMETROS.	60
ANEXO 10. DESCRIPCIÓN DE LA CLASE PERIODO.....	61
ANEXO 11. DESCRIPCIÓN DE LA CLASE SUBSISTEMA.	61
ANEXO 12. DESCRIPCIÓN TEXTUAL DEL REQUISITO CONFIGURAR CONEXIÓN.	62
ANEXO 13. DESCRIPCIÓN TEXTUAL DEL REQUISITO CONECTAR A BASE DE DATOS.....	63
ANEXO 14. DESCRIPCIÓN TEXTUAL DEL REQUISITO CERRAR CONEXIÓN.	64
ANEXO 15. DESCRIPCIÓN TEXTUAL DEL REQUISITO INSERTAR FUNCIONES DE CONTABILIDAD.....	64
ANEXO 16. DESCRIPCIÓN DEL REQUISITO INSERTAR FUNCIONES DE COSTO Y PROCESO.	66
ANEXO 17. DESCRIPCIÓN DEL REQUISITO INSERTAR CODIFICADOR.	67

ANEXO 18. PROTOTIPO DE INTERFAZ DE USUARIO “INSERTAR FUNCIONES DE CONTABILIDAD”68

ANEXO 19. PROTOTIPO DE INTERFAZ DE USUARIO “INSERTAR FUNCIONES DE COSTO Y PROCESOS”69

ANEXO 20. PROTOTIPO DE INTERFAZ DE USUARIO “INSERTAR CODIFICADOR”70

ANEXO 21. ESPECIFICACIÓN DE LOS REQUISITOS.....71

GLOSARIO..... 73

Índice de figuras

Figura 1. Modelo Conceptual.....	29
Figura 2. Diagrama de Componentes.....	34
Figura 3. Diagrama de clases.....	35
Figura 5. Prototipo de interfaz gráfica de usuario Configurar Conexión.....	41
Figura 6. Prototipo de interfaz gráfica de usuario Conectarse a la Bases de Dato.....	42
Figura 7. Cantidad de Procedimientos por clases.....	43
Figura 8. Representación de la Responsabilidad.....	44
Figura 9. Representación de la Complejidad.....	45
Figura 10. Representación de la Reutilización.....	45
Figura 11. Representación del Acoplamiento de las Clases.....	47
Figura 12. Representación de la Complejidad de Mantenimiento de las clases.....	47
Figura 13. Representación de la Reutilización de las clases.....	48

Índice de tablas

Tabla 1. Matriz de seguimiento de dependencias: Requisitos funcionales.....	32
Tabla 2. Descripción de la clase Complementos.	39
Tabla 3. Cantidad de Procedimientos por clase.	43
Tabla 4. Criterio para calcular la Responsabilidad, la Complejidad implementación y la Reutilización.....	44
Tabla 5. Responsabilidad de las clases.	44
Tabla 6. Complejidad de las clases.	45
Tabla 7. Reutilización de las clases.	45
Tabla 8. Relaciones de uso por clases.....	46
Tabla 9. Acoplamiento de las clases.	46
Tabla 10. Complejidad de Mantenimiento de las clases.....	47
Tabla 11. Reutilización de las clases.	47

Introducción

En la actualidad debido al rápido desarrollo de las Ciencias y las Tecnologías todas las empresas y los sistemas educativos de todo el mundo se ven obligados a utilizar las Tecnologías de la Información puesto que la sociedad industrial no podrá vivir sin las computadoras debido a la sobrecarga de información y la necesidad de tener todos los datos digitalizados, ya que son de gran ayuda para analizar los datos, almacenar y recuperar información, simplificar expresiones, controlar experimentos, identificar moléculas, medir áreas de figuras específicas y llevar información estadística de procesos, así de esta forma se logra un incremento en la productividad porque con el uso de sistemas automatizados el personal de oficina está en mejor posición para contribuir a los objetivos de la organización.

Con el surgimiento de la Revolución Informática se producen profundos cambios estructurales en todas las naciones, a los que no se puede permanecer ajeno y en consecuencia a una impostergable modernización de los medios y herramientas con que se planifican, desarrollan y evalúan las diferentes actividades que se llevan a cabo en los institutos de enseñanza del país.

Los avances Tecnológicos han sido aprovechados ampliamente dentro del campo de la informática puesto que gracias al desarrollo de las computadoras se descentralizó el procesamiento de la información y fue posible crear redes de computadoras conectadas entre sí para el intercambio de datos. En un principio esas redes funcionaron de manera local y más tarde surgieron redes corporativas que conectaban instituciones o empresas.

Gran parte de la sociedad se ha desarrollado a través de los avances Tecnológicos, la Informática ha sido una rama imprescindible para la sociedad moderna, es decir es una parte primordial de la cotidianidad de las personas, porque con todos los programas que ofrece, permite una comunicación mucho más avanzada. Entre los tantos programas con que cuenta esta rama están los Paquetes de Ofimática que apoyan el trabajo en las oficinas y tienen un considerable uso en todos los sectores de la sociedad, destacando más su valor en los negocios porque dan la posibilidad de realizar trabajos sencillos con mayor rapidez y calidad como son las ediciones de documentos, animaciones, gráficos, imágenes, tablas, presentaciones, control de varias tablas, organigramas, entre otros.

Unos de los programas de toda esta familia son las Hojas de Cálculo. Las mismas son de gran importancia en los centros de trabajo porque permiten organizar datos numéricos de forma que se puedan

manipular en conjuntos clasificados por sus divisiones (filas y columnas) y mediante ellos se aplican fórmulas y operaciones diversas. Con esta herramienta se pueden transformar los datos a gráficos o hacer una presentación vistosa de los resultados de las operaciones.

La educación, las empresas, la ciencia y la ingeniería son las ramas que más las utilizan como una solución a sus problemas. El manejo de datos es una de sus principales utilidades en el estudio estadístico.

La Calc de OpenOffice es una planilla de cálculo libre de Sun Microsystems, esta aplicación es indispensable en la administración de cualquier empresa u organismo, nos da la posibilidad de generar cálculos en tiempos muy cortos y manejando grandes volúmenes de datos, con el agregado de poder visualizar estos valores a través de importantes gráficos, también se desempeña como una muy buena gestora de datos.

Actualmente en el país existen muchas empresas que a través de los sistemas de Planificación de Recursos Empresariales (ERP) llevan el control y gestión de sus procesos. Los mismos son de gran importancia debido a que son sistemas totalmente integrados, tienen gran capacidad para racionalizar los diferentes procesos y flujos de trabajo, la capacidad de compartir fácilmente los datos a través de diversos departamentos en una organización es muy efectiva, mejoran la eficiencia y los niveles de productividad, brindan un mejor seguimiento y previsión, reducen los costos y de esta forma ofrecen un mejor servicio al cliente.

Debido a que todos los sistemas ERP presentan un precio muy elevado la Universidad de las Ciencias Informáticas (UCI) se dio la tarea de crear un Sistema Integral de Gestión Empresarial llamado Cedrux, el mismo contiene varios subsistemas integrados como Configuración, Contabilidad, Planificación, Recursos Humanos, Activos Fijos Tangibles, Costo y Procesos y Logística. Cedrux se encarga de mejorar la eficiencia y los niveles de productividad, es un sistema totalmente integrado con la capacidad de realizar diferentes procesos.

Cedrux como sistema contable es capaz de gestionar activos de forma dinámica, tributa mediante operaciones al sistema contable de la empresa y estandariza el tratamiento de los Activos Fijos Tangibles (AFT) a nivel nacional. Incluye además temas como la multimoneda, la agrupación de activos y la generación de documentos, prestaciones del sistema que responden a necesidades del país, actualmente no satisfechas de manera conjunta por los sistemas existentes. A partir de la puesta en marcha del

sistema, la economía del país se verá favorecida y el control de los AFT será confiable, estable y legalmente correcto.

Actualmente los subsistemas Contabilidad y Costo y Procesos del sistema Cedrux no cuentan con una herramienta que permita manejar sus datos a través de la Hoja de Cálculo del OpenOffice. Dada esta situación y vista ya la importancia que tiene el uso de estos programas para los sistemas contables se decidió realizar el análisis para la implementación de una herramienta que permita a través de ellas interactuar con todas las funcionalidades que tienen estos subsistemas de Cedrux y a la vez hacer uso de las que ella brinda. Esta herramienta beneficiará tanto a los clientes como a los trabajadores y así el trabajo será más rápido y efectivo.

Dada la situación antes planteada se define como **Problema Científico**: ¿Cómo lograr el manejo de los datos almacenados por los subsistemas Contabilidad y Costo y Procesos del sistema Cedrux empleando las facilidades de la Hoja de Cálculo del OpenOffice?

Partiendo del problema planteado se define como **Objeto Estudio**: Proceso de Desarrollo de Software.

Campo Acción: Análisis y diseño de soluciones informáticas.

Como **Objetivo General**: Realizar el análisis y diseño de una herramienta que permita el manejo de los datos almacenados por los subsistemas Contabilidad y Costo y Procesos del sistema Cedrux empleando las facilidades de la Hoja de Cálculo del OpenOffice.

Objetivos Específicos:

- Realizar el marco teórico sobre complementos para Hojas de Cálculo.
- Realizar el análisis y el diseño de la solución.
- Validar el análisis y el diseño de la solución.

Estructura de la Investigación:

Para la realización del análisis y diseño de una herramienta fue necesario utilizar los siguientes métodos de investigación:

Método Científico

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones.

El método científico se puede clasificar en teóricos y empíricos, los cuales están dialécticamente relacionados.

Métodos Teóricos

Modelación: A través de este método se puede explicar el análisis que se le realizó a los procesos mediante diagramas y modelos de manera lógica.

Histórico – Lógico: Permite realizar el estudio de las hojas de cálculo y de los sistemas que lo implementan.

Analítico: Permite enfocarnos bien en el análisis que se realiza a la hora de buscar toda la información referente a la hojas de cálculos.

Métodos Empíricos

En este caso se utilizó la **Entrevista:** Se realizaron varias entrevistas a distintos especialistas y se buscaron consultantes que con su experiencia lograron realizar un buen desarrollo de la investigación.

Como **idea a defender** se plantea que si se realiza el análisis y diseño de una herramienta que permita generar reportes a través de la Hoja de Cálculo del OpenOffice, se podrá mejorar la manipulación de los datos almacenados por los subsistemas Contabilidad y Costo y Procesos del sistema CedruX.

Estructura del Documento

Capítulo 1. Fundamentación Teórica. Este capítulo le ofrece el análisis sobre los complementos para Hojas de Cálculo realizado durante la investigación. Se presenta información de los principales conceptos tratados en el trabajo.

Capítulo 2. Análisis y diseño de la solución. Está compuesto por la descripción y textual de los requisitos y la descripción del modelo conceptual. En este capítulo se realizará el diseño de varios

diagramas como el de componentes, el de modelado de datos y el de clases, también se analizarán los patrones de diseño empleados en la solución.

Capítulo 3. Validación de la solución propuesta. Contiene toda la validación del análisis y el diseño propuesto para la solución.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En este capítulo se abordan los principales conceptos a tratar relacionados con los complementos para Hojas de Cálculo, se realiza también un estudio de la metodología a utilizar, de las herramientas, los sistemas gestores de Base de Datos y el lenguaje de modelado con el que se realizará la propuesta del sistema.

1.2 Hojas de Cálculo

Las más usadas y sus beneficios

En el mundo de las computadoras personales, las Hojas de Cálculo tales como Microsoft Excel, Lotus 1-2-3, y Quattro Pro de Corel se han vuelto fáciles de usar como un procesador de texto. La mayoría de los ingenieros pueden introducir tablas de datos y programar las hojas para evaluar expresiones aritméticas con muy poco esfuerzo.

En el área de diseño, son ampliamente usadas para análisis de costos. Dado un costo total de equipo instalado, y el costo unitario de productos, subproductos y materia prima, así como las utilidades, las Hojas de Cálculo pueden calcular el capital total invertido, la hoja de costos, y varias medidas de beneficio, tales como el retorno de la inversión y el valor presente neto. Los resultados de las Hojas de Cálculo pueden expresarse en una gran variedad de gráficas, y para problemas más complejos puede ser ligada a un lenguaje de procedimiento. Por ejemplo Visual Basic for applications está disponible como lenguaje macro para Microsoft Excel. (1)

Actualmente los programas de Hojas de Cálculo son aplicados en una gran diversidad de áreas, desde sus aplicaciones iniciales en el análisis, control, planificación o evaluación de problemas económicos o empresariales, hasta las aplicaciones en las áreas de la Ingeniería y Medicina.

Son uno de los programas más utilizados a nivel mundial, y los mismos han pasado de ser sólo programas para realizar cálculos, a programas que han incorporado nuevas funciones como la graficación y manejo de Bases de Datos, convirtiéndose así en poderosas herramientas para el trabajo diario y profesional.

Actualmente cuando se habla de un programa de Hoja de Cálculo se refiere a programas muy diferentes a sus predecesores.

Los más utilizados en la actualidad son:

- Excel de Microsoft
- Calc de OpenOffice
- LOTUS 1-2-3 de IBM.

La Calc de OpenOffice permite realizar operaciones con números organizados en una cuadrícula. Es útil para realizar desde simples sumas hasta cálculos más complejos como de préstamos hipotecarios.

En la actualidad estos programas no se consiguen o se distribuyen por separado, sino que vienen inmersos en un conjunto de programas denominados como Suite de Ofimática o Suite de Productividad de Oficina.

Estas Suite de Ofimática por lo general incluyen un procesador de palabras, una Hoja de Cálculo, un manejador de base de datos y un programa de presentaciones; utilizando una interfaz de usuario homogénea. La Suite de Ofimática dominante en el mercado es Microsoft Office, la cual posee sus propios formatos cerrados (propietarios) de documentos para cada una de las aplicaciones. (2)

Las Suites de Ofimática más reconocidas son: (2)

- Microsoft Office
- OpenOffice
- Lotus Smartsuite
- StarOffice
- WordPerfect Office

La suite OpenOffice es una de las más utilizadas. El hecho de que la suite OpenOffice sea gratis, el cual es uno de sus principales atractivos en comparación con la suite Office de Microsoft, no quiere decir que el producto sea de baja calidad, ya que en las tareas más comunes se logran resultados muy similares.

Una de las ventajas de la suite OpenOffice respecto a la suite Office de Microsoft, es que OpenOffice funciona bajo el Sistema Operativo Windows y bajo el Sistema Operativo Linux, porque existen versiones para ambos sistemas operativos.

OpenOffice a diferencia de otros paquetes ofimáticos que existen en el mercado ha sido diseñado como una única aplicación, por tanto el entorno de trabajo de sus aplicaciones son familiarmente conocidas unas con respecto a otras.

Además, como la suite OpenOffice utiliza un formato de datos abierto, Open Data Format u ODF; lo que garantiza no tener que pagar derechos por la utilización del formato de datos, haciendo a su vez más atractivo para las organizaciones la selección de dicha suite como el estándar de trabajo.

1.2.1 Editores de Hojas de Cálculo más utilizadas en el mundo

- GS-Calc 8.4.0

Se utiliza para crear y editar Hojas de Cálculo fácilmente. Es un editor compatible con todos los formatos habituales como XML, ODS o DBF. Realiza todo tipo de gráficas: de barras, quesitos o lineares en dos.

- Abykus 3.0

Al igual que la anterior permite crear y editar. Es una alternativa al popular programa de Hojas de Cálculo Microsoft Excel, que cuenta con una serie de ventajas y prestaciones muy interesantes con respecto a sus competidores.

- Gnumeric 1.10.9

Excelente Hoja de Cálculo procedente de Linux. Es del entorno de escritorio Gnome para Linux que hoy se presenta en su versión para Windows.

1.3 Complementos para Hojas de Cálculo

Las aplicaciones que permiten manejar Hojas de Cálculo, como Microsoft Excel u OpenOffice Calc, ofrecen funciones que ayudan a realizar análisis estadísticos de los datos mediante promedios, modas, medianas, máximos, varianzas. En la actualidad existen varios complementos para las Hojas de Cálculo como son:

- El complemento EZAnalyze para Microsoft Excel.
- La macro OOO Statistics para OpenOffice Calc.
- El complemento cubiFOR para Microsoft Excel.

Las funciones de análisis estadístico que proporciona el complemento EZAnalyze son las siguientes: (3)

- Análisis de Varianza
 - Varianza de un factor
 - Varianza de dos factores con varias muestras por grupo
 - Varianza de dos factores con una sola muestra por grupo
- Correlación
- Covarianza
- Estadística descriptiva
- Suavización exponencial
- Prueba t para varianzas de dos muestras
- Análisis de Fourier
- Histograma
- Media móvil
- Generación de números aleatorios

- Jerarquía y percentil
- Regresión
- Muestreo
- Prueba t
 - Prueba t para medias de dos muestras emparejadas
 - Prueba t para dos muestras suponiendo varianzas iguales
 - Prueba t para dos muestras suponiendo varianzas desiguales
- Prueba z

1.3.1 Complemento EZAnalyze

EZAnalyze es un complemento de Microsoft Excel, que permite ampliar las prestaciones del programa para el análisis estadístico de datos y la creación de gráficos. (3)

Funciones que Ofrece: (3)

- Describe: proporciona porcentajes y estadísticos descriptivos.
- Disaggregate: permite clasificar los resultados en categorías.
- Graph: crea gráficos básicos.
- New Variable: permite crear nuevas variables a partir de las ya existentes (como suma o promedio, diferencias entre ellas).
- Advanced: proporciona opciones más avanzadas de estadística, como el cálculo de correlaciones entre las variables, el análisis de varianza o el análisis de regresión lineal, entre otros.
- Delete Extra Sheets: borra todas las hojas creadas por EZAnalyze que se han ido creando al analizar los datos. Por defecto, el nombre de estas hojas empieza por EZA.

- Help: da acceso a la ayuda online de EZAnalyze.
- About: contiene información sobre la versión de EZAnalyze que se está utilizando y sobre los términos del acuerdo de licencia.
- Options: da la posibilidad de desinstalar EZAnalyze y de cambiar la forma de desplegar las opciones de los menús (en cascada o de forma tradicional).
- Other Tools: contiene funciones que permiten fusionar datos y transponer datos.

1.3.2 Complemento cubiFOR

Es una herramienta para Microsoft Excel que se utiliza para cubicar, clasificar productos de madera y calcular biomasa. En la actualidad existe un interés creciente no sólo por el cálculo del volumen maderable sino por el de los distintos productos de madera, de la biomasa forestal, y por la cuantificación del CO₂ fijado por los bosques.

Objetivos principales: (4)

- Homogeneizar el sistema de cubicación en los aprovechamientos forestales de Castilla y León.
- Mejorar la precisión en la cubicación y el rango de aplicación de los actuales modelos.
- Reducir la toma de datos en campo durante el proceso del inventario forestal (se puede estimar el diámetro normal sin corteza y la altura individual del árbol) en algunas de las especies.
- Reducir el costoso proceso de apeo de árboles tipo para la construcción de tarifas de cubicación.
- Calcular la biomasa y el CO₂ almacenado.

Existen otro complementos como **Crystal Ball** y **@Risk** los mismos hacen mucho más sencilla la tarea de simulación que cuando esta se hace en una Hoja de Cálculo sola. En particular estos programas simplifican en gran medida los procesos de generación de variables aleatorias y el ensamblaje de los resultados estadísticos. Ambos complementos facilitan la captura y despliegue del resultado de la simulación.

El RISKOptimizer, por medio de la aplicación de poderosas técnicas de optimización de algoritmos genéticos y de la simulación Monte Carlo puede encontrar soluciones óptimas a problemas que no son solucionables para los optimizadores convencionales lineales y no lineales. El RISKOptimizer combina la tecnología de simulación de @RISK, el complemento de análisis de riesgos de Palisade, con Evolver, el solucionador de algoritmos genéticos de Palisade. Los usuarios familiarizados con el @RISK y tanto el Evolver o bien el Solver incorporado en el Excel podrían ser capaces de usar el RISKOptimizer con poca dificultad. (5)

El RISKOptimizer usa simulación (del @RISK) para lidiar con la incertidumbre presente en el modelo y usa algoritmos genéticos (del Evolver) para generar posibles valores para las celdas ajustables.

Modelando la Incertidumbre

El RISKOptimizer permite describir los posibles valores para cualquier elemento de una Hoja de Cálculo usando cualquiera de las funciones de distribución de probabilidad disponibles en el @RISK.

La simulación en una Hoja de Cálculo sin la ayuda de los paquetes complementarios sería una tarea lenta y tediosa, incluso para modelos simples. Para modelos más complicados, para distribuciones más complejas o para un mayor número de iteraciones, estos inconvenientes se acentúan.

1.3.3 Macro OOO Statistics.

Mediante el uso de macros se puede realizar distintas acciones como:

- Obtener de Internet los valores en bolsa de ciertas compañías.
- Crear una Hoja de Cálculo que se auto actualice cada vez que se abra.
- Abrir todos los documentos de Writer que existan en un determinado directorio y pegarlos uno detrás de otro.
- Convertir todos los documentos de Writer de un directorio a formato PDF.
- Crear una Base de Datos que permita gestionar los pedidos de los clientes.

En general, mediante el uso de macros en OpenOffice se pueden crear pequeñas y medianas aplicaciones ofimáticas, usando la potencia, versatilidad y facilidad del uso que ofrecen todos los componentes disponibles en la suite (Writer, Calc, Impress, Draw y el gestor de Bases de Datos integrado en OOO).

La Hoja de Cálculo que se seleccionó para desarrollar el sistema es la Calc de OpenOffice, no se escoge el Excel de Microsoft porque para la misma existe un complemento que trabaja con funciones, la herramienta a la que se le realizará el análisis y el diseño será distinta a las existentes porque tendrá funcionalidades específicas y también hará uso de las facilidades que brindan estos programas, es igual que la que existe para el Excel pero enfocada a la Calc.

1.4 Metodología de Desarrollo

Para el desarrollo de Cedrux se elaboró un Modelo de Desarrollo de software específico que responde a la dinámica del proyecto. Con la descripción de la misma los equipos de desarrollo poseen un modelo estandarizado, así como una definición clara y precisa de las responsabilidades y acciones a realizar en cada momento. Este Modelo está basado en las metodologías RUP y XP, tomando de estas las características más notables; de esta forma se obtiene como resultado el documento Modelo de Desarrollo orientado a componentes del Proyecto ERP-Cuba. Aquí se definen las características y necesidades de un Proyecto ERP, el organigrama organizativo de las líneas de desarrollo, los diferentes roles involucrados con las responsabilidades de cada uno, las actividades a realizar durante todo el procesos de desarrollo del software con cada uno de los roles involucrados en dichas actividades y los artefactos generados por cada uno de ellos. El flujo de actividades del Modelo de Desarrollo está compuesto por los roles y los artefactos generados por cada uno de ellos. Además de las métricas para la medición del avance del proyecto.

Teniendo en cuenta el Modelo de Desarrollo establecido y el alcance de esta investigación, los artefactos a generar durante la realización de esta investigación serán:

- Especificación de requisitos de software.
- Prototipo de interfaz de usuario.
- Modelo conceptual.

1.5 Trazabilidad de requerimientos

La gestión de requisitos es un proceso clave para cumplir dichos objetivos. Para realizar una exitosa gestión de requisitos la trazabilidad es una actividad clave.

Un proceso de trazabilidad se divide en dos subprocesos:

- Configuración de la trazabilidad
- Especificación y explotación de la información de trazabilidad.

La herramienta seleccionada para realizar el proceso de trazabilidad de los requisitos es la OSRMT, la misma es la que se usa en el proyecto debido al programa de mejora.

OSRMT: es una herramienta que permite la gestión de los requisitos y la completa trazabilidad. Es una herramienta de Software Libre pensada para asistir en todo el ciclo de vida del desarrollo del Software.

1.5.1 Técnicas generales para la captura de requisitos

La captura de requisitos es la actividad donde se extraen las necesidades del sistema. Durante la investigación realizada se encontraron una serie de técnicas usadas para realizar esta actividad, de ellas se seleccionaron cual o cuales se aplicarían.

- Entrevistas: Resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. (6) Es el método clásico, se prepara la entrevista en el puesto de trabajo, para después aplicarla al cliente, que es la persona que más conoce del tema que se está indagando.
- JAD (Joint Application Development): Esta técnica resulta una alternativa a las entrevistas. Está basada en cuatro principios fundamentales dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la entrevista se trabajará sobre lo que se generará. (6)
- Tormenta de ideas: Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso

(máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. (6)

- Mapas conceptuales: Son muy usados dentro de la ingeniería de Requisitos pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sugestivos y pueden llegar a ser ambiguos en casos complejos si no se acompaña de una descripción textual. (6)
- Bocetos: Esta técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno web. (6)
- Cuestionario y Listas de chequeo: Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. (6)
- Comparación de terminología: Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. (6)

Luego del estudio realizado de las mismas la técnica que se decidió utilizar para la captura de requisitos fue la Tormenta de ideas.

1.5.2 Técnicas generales de validación de requisitos

La validación de requisitos es el proceso que demuestra que la definición de los requisitos realmente detalla el sistema que necesita el cliente. Para realizar una perfecta validación es necesario utilizar algunas de las técnicas de validación existentes como:

- Reviews o Walk-throughs: Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante. (6)
- Auditorías: Esta técnica consiste en un chequeo de los resultados contra una (Listas de Chequeo) predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.

- Matrices de trazabilidad: Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. (6)
- Prototipos de interfaz: Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requisitos capturados le permite al usuario realizar su trabajo de manera eficiente y efectiva.
- Revisión Técnica Formal (RTF): Son reuniones del personal técnico (usuario final del sistema) con el objetivo de validar la especificación de requisitos.

Luego de un profundo estudio de las mismas se pudo definir que las técnicas a utilizar para la validación son la Revisión Técnica Formal (RTF) y los Prototipos de interfaz.

1.5.3 Actividades del proceso de definición de requisitos

En el proceso de IR (Ingeniería de Requisitos) son esenciales diversas actividades. Dependiendo del tamaño del proyecto y del modelo de proceso de software utilizado para el ciclo de desarrollo, las actividades de la IR varían tanto en número como en nombres. Aquí se muestran las principales actividades que se realizaron para este proceso:

- Análisis del Problema
- Evaluación y Negociación
- Especificación
- Validación
- Evolución

1.5.4 Métricas de la calidad para la especificación de requisitos

Las métricas tienen una gran importancia en las primeras etapas del ciclo de vida del desarrollo de un software para asegurar la calidad del producto final. Son escalas de unidades sobre las cuales puede medirse un atributo cuantificable.

Las métricas de calidad proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir, cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

Las métricas de calidad de los sistemas de información se utilizan para evaluar y controlar el proceso de desarrollo del software, de forma que permitan:

- Indicar la calidad del producto.
- Evaluar la productividad de los desarrolladores.
- Evaluar los beneficios en términos de productividad y calidad.
- Establecer una línea base para la estimación.

La especificación de requisitos es un factor principal para el buen funcionamiento del desarrollo de software. Algunas de las métricas de especificación son:

Especificación correcta

La corrección no se puede establecer a priori, sino que depende fundamentalmente del usuario final del sistema representado. Quien debe decidir si una especificación es correcta o no es el cliente que solicita el sistema. Por eso la corrección de una especificación debe ser verificada a través de la revisión y aceptación del usuario. (7)

Especificación no ambigua

El equipo de desarrollo que interviene en el proceso de especificación de requisitos suele tener varios puntos de vista. Por este motivo es difícil asegurar la ausencia de ambigüedad en la especificación. Aunque a priori parece un problema insalvable, existen formas de limitar los efectos negativos de la ambigüedad. (7)

Especificación completa

Una especificación es completa si, y sólo si, describe todos los requisitos relevantes para el usuario, incluyendo requisitos asociados con funcionalidad, actuación, restricciones de diseño, atributos o interfaces externas. (7)

Especificación consistente

Los mayores problemas relacionados con la consistencia son los que tienen que ver con las incoherencias lógicas entre distintos requisitos (requisitos incompatibles, incoherentes o mutuamente excluyentes), la repetición de la misma información a lo largo de distintos requisitos (requisitos repetitivos o redundantes) o la referencia en distintos requisitos a ítems que usan la misma palabra para designar conceptos del problema diferentes (incoherencia respecto al dominio del problema). (7)

Especificación organizada

La categorización de los requisitos por orden de importancia es una recomendable práctica que permite establecer prioridades a la hora de abordar el desarrollo. Esta categorización por el atributo importancia o prioridad es necesaria desde un punto de vista práctico.

Otra posible categorización que resulta interesante es la de la estabilidad de la especificación. El cambio de los requisitos de usuario es algo intrínseco al propio cambio en el problema. (7)

Especificación verificable

Se considera que una especificación es verificable si lo son cada uno de los requisitos constituyentes. A su vez, se considera que un requisito individual es verificable si existe un proceso acotado que permita determinar que el sistema construido satisface lo descrito en el propio requisito.

Una forma de conseguir que los requisitos sean verificables es describirlos con suficiente detalle, o teniendo en cuenta que una de las premisas que se debe cumplir es que sean probados una vez implementados. (7)

Especificación modificable

Se considera que una especificación es modificable si su estructura permite realizar cambios sobre los requisitos que contiene de forma sencilla, completa y consistente, manteniendo la estructura inicial del

conjunto. Esto implica que debe existir una buena organización de la información y que el acoplamiento entre requisitos sea el menor posible. (7)

Especificación trazable

Una especificación se considera trazable si el origen de cada requisito individual está claro y existe algún mecanismo que permita seguir el impacto de dicho requisito a lo largo del resto de actividades del ciclo productivo. (7)

1.5.5 Métricas para la validación del diseño

Las métricas de diseño proporcionarán una predicción de la calidad del diseño, también proporcionan una indicación general de la cantidad de esfuerzo de pruebas necesario para aplicarlo en un sistema OO.

Las métricas basadas en clases se dividen en cuatro categorías: tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaños para una clase Orientada a Objeto (OO) se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema OO en su totalidad. Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones. Las métricas para valores internos de clase examinan la cohesión y asuntos relacionados con el código, y las métricas orientadas a valores externos examinan el acoplamiento y la reutilización. Estas son unas de las métricas con las que se puede evaluar la calidad del diseño propuesto.

Tamaño operacional de clase (siglas: TOC): Se refiere al número de métodos pertenecientes a una clase. Está determinada por los atributos: Responsabilidad, Complejidad de implementación y la Reutilización, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado.

Relaciones entre clases (siglas: RC): Dado por el número de relaciones de uso de una clase. Está determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado.

1.6 Lenguaje de Modelado y Desarrollo

1.6.1 Lenguaje de Modelado de Clases

Unified Modling Lenguaje (UML)

UML es un lenguaje usado para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software en desarrollo, así como para modelado de negocios y almacenamiento de datos. UML es una notación, no un método, destinado a los sistemas de modelado que utilizan conceptos orientados a objetos.

En la actualidad está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Entre más complejo es el sistema que se desea crear más beneficios presenta su uso. Mediante UML es posible establecer una serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

1.6.2 Lenguaje de Desarrollo

Java

Java es un lenguaje de uso general, y su semántica es lo bastante completa para poder abarcar programas de todo tipo. No es necesariamente un lenguaje para Internet, sino que puede ser usado en el desarrollo de un sistema a ejecutarse simplemente en una computadora.

Puede ser ejecutado en múltiples plataformas. Es uno de los escasos lenguajes cuyos programas pueden ser transportados de sistema operativo, computadora o entorno, sin necesidad de cambiar el código. Java ha sido concebido, desde sus orígenes, como un lenguaje capaz de producir código totalmente transportable. (9)

Algunas Características

- Orientado a objetos
- Lenguaje simple
- Distribuido
- Interpretado y compilado a la vez
- Robusto

- Seguro
- Indiferente a la arquitectura
- Portable
- Alto rendimiento

A parte de todas las excelente características que tiene Java se decidió utilizarlo como lenguaje de desarrollo porque el mismo pose un plugin para OpenOffice, también que el centro cuenta con una política por la que se rige para hacer la selección, llamada “0120_6 Arquitectura Vista de Entorno de Desarrollo Tecnológico”.

1.7 Herramientas para el desarrollo

1.7.1 Herramientas CASE (Computer Aided Software Engineering)

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. (10)

Visual Paradigm

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (11)

Se utiliza el Visual Paradigm como herramienta CASE porque es una herramienta que soporta el ciclo de vida completo del desarrollo de software. Su mayor éxito consiste en la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. Además que permite la modelación de los procesos de negocio. Esta herramienta permite generar los artefactos Modelo

Conceptual, Modelado de Datos, Diagrama de clase, Diagrama de componentes y el Modelado de procesos. La Universidad de las Ciencias Informáticas cuenta con la licencia de esta herramienta, lo cual hace más acertada su selección.

1.7.2 Herramienta de desarrollo colaborativo

Tortoise-RapidSVN

El TortoiseSVN es una herramienta de gran alcance, es simplemente la mejor interfaz para Subversion y está disponible sólo para la plataforma Windows.

Adición de un repositorio para RapidSVN

A diferencia de lo que TortoiseSVN se integra perfectamente en el Explorador de Windows, RapidSVN le permite administrar los archivos desde una interfaz de usuario dedicado.

Este programa funciona en cualquier plataforma y se puede ejecutar en Linux, Windows, Mac OS / X, Solaris, etc. Se utiliza para el control de versiones, es decir a través de esta herramienta se sube toda la información terminada al repositorio y en caso de subir algo con un error se puede recuperar el anterior.

1.7.3 Entorno integrado de desarrollo

NetBeans

Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Es un entorno de desarrollo que está escrito en Java. Puede servir para cualquier otro lenguaje de programación. El NetBeans IDE es un producto libre y gratuito sin restricciones de uso tanto comercial como no comercial.

El mismo posee un plugin llamado API para OpenOffice, es muy fácil de utilizar el conjunto de asistentes que ayudan a crear extensiones de OpenOffice. Estas extensiones se pueden utilizar para integrar nuevas funcionalidades, adoptar la funcionalidad existente y crear aplicaciones Java que se puedan controlar remotamente.

1.8 Sistema Gestor de Base de Datos

PostgreSQL

Está ampliamente considerado como el sistema de Bases de Datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. Es usado para manejar grandes cantidades de información y está basado en el modelo relacional, aunque incorpora conceptos del modelado orientado a objeto.

Se pueden definir consultas anidadas, vistas, crear funciones por el usuario, no sólo en el lenguaje natural SQL, también en el lenguaje nativo PostgreSQL, Perl, PHP y Java. Es multiplataforma, soporta múltiples transacciones, integridad de datos, presenta una estabilidad muy alta, gran seguridad de los datos, soporta la réplica y procedimientos almacenados. Propone un tamaño ilimitado para las Bases de Datos, lo que da la medida de un gestor de Base de Datos robusto, y con grandes funcionalidades. (12)

Se decidió trabajar con PostgreSQL debido a que es el gestor de Bases de Datos con que trabaja el proyecto Cedrux regido por la política existente y por todas las ventajas y facilidades que brinda.

1.9 Patrones

1.9.1 ¿Qué es un Patrón?

Un patrón es un modelo que sirve de guía para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Capturan la experiencia existente y probada para promover buenas prácticas.

Ayudan a construir la experiencia colectiva de Ingeniería de Software, son una abstracción de problema – solución, se ocupan de problemas recurrentes, identifican y especifican abstracciones de niveles más altos que componentes o clases individuales y proporcionan vocabulario y entendimiento común.

1.9.2 Patrones de reutilización de especificación de requisitos

Reutilizar elementos del análisis es mucho más valioso que los de código. Pero en ocasiones implica algunos riesgos ya que se intenta forzar la realidad para que se ajuste dentro de los patrones identificados.

- Especificar: Este patrón aconseja describir cómo puede el usuario de un sistema seleccionar (especificar) una determinada información (para modificarla, eliminarla o consultarla) en un requisito separado y hacer referencia a dicho requisito cuando sea necesario.
- Priorizar: Este patrón sugiere que en el caso de que el usuario desee poder ordenar (priorizar) la información presentada por el sistema, se separen las posibles formas de ordenar, dicha información en un requisito aparte y se haga referencia desde a los que sea necesario. De forma similar al patrón Especificar.
- Presentación: Este patrón, recomienda limitarse a indicar qué datos debe solicitar o presentar el sistema sin entrar en detalles concretos de interfaz de usuario.

1.9.3 Patrones de especificación de requisitos

Recoger y especificar requisitos no es una actividad sencilla, requiere de técnicas que ayuden a mejorar los modelos clásicos de especificación.

- Clasificación: Este patrón permite clasificar las especificaciones de forma similar a como se clasifican los requisitos. Pueden ser: puramente textuales y se clasifican como reglas de negocio, restricciones tecnológicas o de diseño, lista de funcionalidades. También las hay gráficas basadas en modelo UML que recogen casos de uso, entidades del dominio o aspectos de navegación de interfaz de usuario. (13)
- Derivación: A partir de una determinada necesidad del cliente, se puede obtener una especificación escribiendo o modelando de manera formal lo que dicha necesidad sugiere. No sólo se identifica más rápido y mejor lo que dice el cliente, sino que se transmite mejor al equipo de desarrollo. (13)

Por ejemplo, una regla impuesta por el usuario puede derivarse en una precondition, una especificación; o un cambio en la navegación puede derivarse directamente en una nueva versión del modelo de interfaz de usuario.

1.9.4 Patrones de diseño

Un patrón de diseño ofrece una solución para un problema de diseño común, describiendo cómo una sociedad de clases (y objetos) trabajan juntos para resolver ese problema en un contexto particular.

El valor principal de los patrones de diseño se encuentra en el aprovechamiento de la experiencia de los expertos que los han identificado y documentado. (14)

GRASP (General Responsibility Assignment Software Patterns)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Principales patrones de GRASP:

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

Creador: Una clase B tiene la responsabilidad para crear una instancia de la clase A si alguna de las siguientes condiciones se cumple:

- B agrega objetos de A
- B contiene objetos de A
- B almacena objetos de A
- B usa objetos de A
- B tiene los datos necesarios para inicializar a A cuando ésta es creada.

Controlador: Asignar las responsabilidades para el manejo de mensajes de eventos del sistema a una clase:

- Representar al conjunto del sistema (Controlador *Facade*)
- Representar al conjunto del negocio u organización (Controlador *Facade*)
- Representar algo del mundo real que está activo, una persona que puede estar involucrada en la tarea (Controlador de roles).
- Representar un administrador artificioso para todos los eventos del sistema. (Controlador Caso de Uso).

Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Alta Cohesión: Asigna responsabilidades procurando que la cohesión sea lo más alta posible, es decir cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos.

Bajo Acoplamiento: Asigna responsabilidades de tal manera que el acoplamiento sea el menor posible es decir debe haber pocas dependencias entre las clases.

GOF (Gang of Four)

Estos patrones se refieren a cómo crear objetos cuando sus creaciones requieren tomar decisiones, además cómo instanciar clases o sobre que objetos, otro objeto delegará responsabilidades.

Fachada: Simplifica los accesos a un conjunto de objetos relacionados proporcionando un objeto que todos los objetos de fuera del conjunto utilizan para comunicarse con el conjunto. Define una interfase de más alto nivel que permite usar el sistema más fácil.

Instancia única (Singleton): Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Problema: se admite exactamente una instancia de una clase. Los objetos necesitan un único punto de acceso global.

Solución: Defina un método estático de la clase que devuelva el Singleton. (15)

1.10 Conclusiones

Para el desarrollo de este capítulo se realizó una profunda investigación sobre las Hojas de Cálculos más utilizadas en la actualidad y los complementos utilizados por las mismas. Se definieron las herramientas para el modelado del problema y para el desarrollo colaborativo, el lenguaje de modelado, el sistema gestor de Base de Datos, así como la metodología a utilizar para el desarrollo de la solución. Fueron definidas, además, las técnicas a utilizar para la captura y validación de requisitos y las métricas que serán aplicadas en la validación de la solución.

Capítulo 2. Análisis y diseño de la solución

2.1 Introducción

En el presente capítulo se realizará el análisis y el diseño de la solución propuesta, se desarrollaran los distintos diagramas que se generan en el flujo de actividades del Modelo de Desarrollo. Se podrán observar las características del sistema mediante la definición de los requisitos y el Modelo Conceptual.

2.2 Análisis

En esta fase se identifican las necesidades del cliente que no son más que los requisitos del software. Cuando se realiza un análisis para determinar los requisitos se está llevando a cabo un proceso de descubrimiento, refinamiento, modelado y especificación que le permite al desarrollador especificar la función y el rendimiento del software.

2.3 Identificación de los requisitos

Muchas veces lo que los clientes necesitan no es lo que los clientes consiguen, es por esta razón que antes de comenzar con el diseño y la implementación de un software es necesario ejecutar esta fase, la misma es muy importante porque mediante ella quedan claros los requisitos con los cuales se efectuarán el diseño y la posterior implementación del sistema.

Para realizar la captura de los requisitos se hizo necesario utilizar la técnica tormenta de ideas, porque mediante el estudio realizado de las otras técnicas esta era la más fácil de usar basándose en el poco tiempo con el que se contaba y la falta de experiencia por parte de los compañeros al frente de la tarea. También que era muy difícil lograr fijar un día para realizar una reunión formal con el director José Carlos ya que sus obligaciones y tareas no lo permitían. Esta técnica por ser informal no deja de ser efectiva.

La misma se puso en práctica al realizar una pequeña reunión con el director de política contable del país el compañero DrC. José Carlos del Toro Ríos el mismo es el encargado de chequear el proyecto y plantear las necesidades existentes, de ahí que varios compañeros aportaron sus opiniones, de las cuales se identificaron una serie de requisitos. Luego apoyándose en el sistema cubano de contabilidad Versat

fueron tomadas algunas ideas de sus funcionalidades para adaptarlas al sistema debido a que el mismo contiene un complemento instalado para trabajar con funciones.

Para la implementación en un futuro del sistema fue necesario priorizar los requisitos más significativos para el cliente teniendo en cuenta la complejidad de la implementación de los requisitos y el criterio del equipo de desarrollo.

2.4 Modelo Conceptual

El Modelo Conceptual es el artefacto más importante a crear durante el análisis ya que en él se realiza una representación de los conceptos que han sido utilizados a lo largo de todo el proceso de negocio. Explica los conceptos más significativos en el dominio del problema al igual que la descripción de los mismos. El Modelo Conceptual propuesto consta de 5 conceptos, cada uno con sus atributos y relaciones de dependencia, el mismo se muestra a continuación:

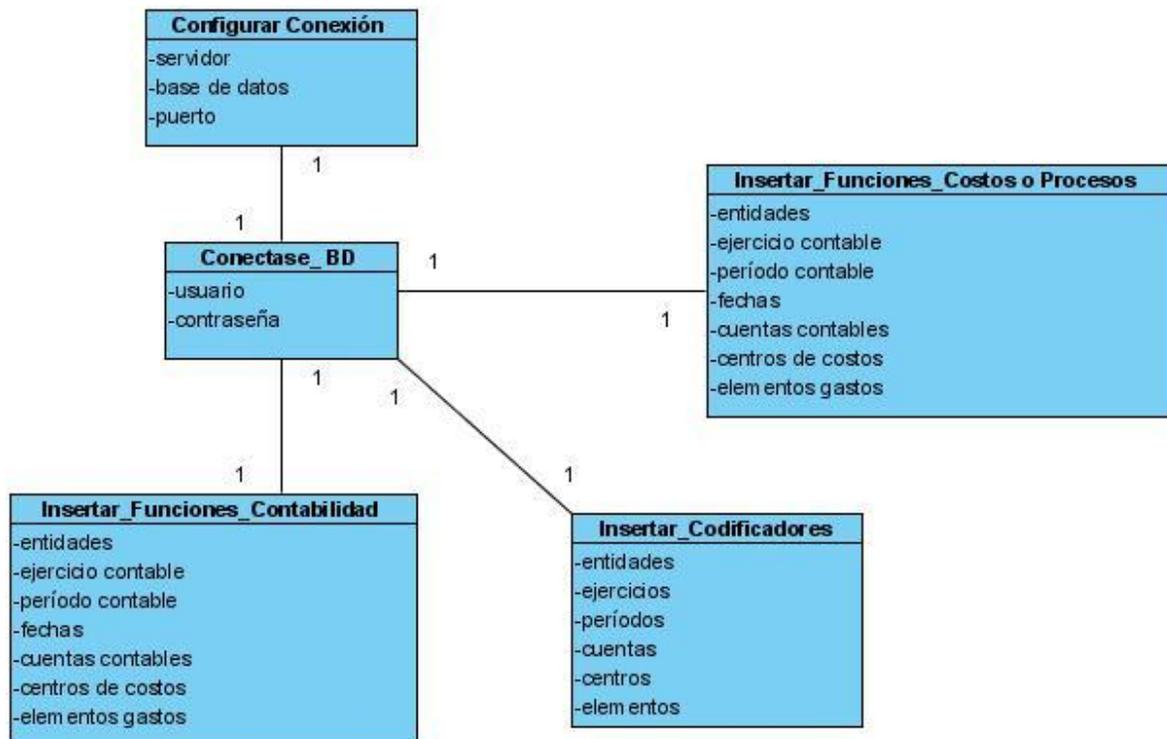


Figura 1. Modelo Conceptual.

2.5 Requisitos funcionales

Los requisitos funcionales no son más que funcionalidades del software. A través de ellos se define el alcance del sistema en cuanto a las acciones que deben realizar, y en cuanto a la transferencia de datos entre todas las diferentes funciones del sistema. Los requisitos capturados para darle solución al problema planteado son los siguientes:

R1- Configurar la conexión

R2- Conectar a Base de Datos

R3- Cerrar conexión

R4- Insertar funciones de Contabilidad

R5- Insertar funciones de Costo y Procesos

R6- Insertar Codificadores

2.6 Requisitos no funcionales

Los requisitos no funcionales son aquellos que definen lo que la herramienta de software debe tener en cuanto a apariencia, operabilidad, y mantenimiento.

2.6.1 Requisitos de Software

- Open Office, MS Office 2003 o superior, recomendado 2007.
- El gestor de Base de Datos que se debe utilizar es PostgreSQL 8.3.

2.6.2 Requisitos de Hardware

- Procesador Intel Pentium (recomendado Pentium II superiores o algún similar).
- 64 MB de RAM como mínimo (recomendado 128 MB).
- Tener en cuenta que mientras más prestaciones tenga el sistema operativo más memoria RAM requerirá el equipo para operar.

2.6.3 Restricciones en el diseño y la implementación

- Implementado con el lenguaje de programación Java.
- Desarrollado en Netbeans 6.9.
- Para el análisis y diseño del software se utiliza el lenguaje UML y para modelar la solución la herramienta Visual Paradigm 7.1.

2.6.4 Requisitos de Seguridad

- El complemento no modifica la información de la Base de Datos, solo consulta la misma en dependencia de los privilegios que tenga el usuario con el que se conecte.

2.6.5 Requisitos de Usabilidad

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

2.7 Trazabilidad de requisitos: Matriz de seguimiento de dependencias

La trazabilidad de los requisitos no es más que la habilidad de describir y seguir el ciclo de vida de un requisito para así lograr el éxito en el proyecto. Mediante esta matriz de trazabilidad se pueden ver relacionados todos los requisitos funcionales.

	Configurar Conexión	Conectarse a la BD	Cerrar Conexión	Insertar Funciones de Contabilidad	Insertar Funciones de Costos y Procesos	Insertar Codificadores
Configurar Conexión						
Conectarse a la BD	x					
Cerrar Conexión						

Insertar Funciones de Contabilidad		x				
Insertar Funciones de Costos y Procesos		x				
Insertar Codificadores		x				

Tabla 1. Matriz de seguimiento de dependencias: Requisitos funcionales.

2.8 Especificación de requisitos

La especificación no es más que una descripción completa del comportamiento del sistema que se va a desarrollar. A la hora de especificar un requisito lo que se registra son sus características y condiciones que debe cumplir cada requisito funcional. Este proceso se puede dividir en tres actividades:

- Captura de requisitos
- Definición de requisitos
- Validación de requisitos

La especificación textual se encuentra en el Anexo 22.

2.9 Diseño

El diseño no es más que el final de la fase de elaboración y el comienzo de las iteraciones de construcción. Todo esto lleva a cabo también la creación de un modelo de implementación. Este flujo de trabajo es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, es decir su función es ver que el sistema cumpla sus objetivos.

Los objetivos del diseño son:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento.

2.10 Descripción de las entradas

Los artefactos que se generaron mediante el análisis de la solución fueron:

Descripción de los Requisitos

Este artefacto se genera para lograr una descripción exacta de cómo va a funcionar cada requisito en el sistema, incluyendo validaciones que darán un aviso en caso de que no se estén realizando los pasos requeridos por el sistema. La descripción textual de estos requisitos se puede encontrar en los Anexos del 12 al 17.

Modelo Conceptual

Es una representación de conceptos del mundo real. El mismo se crea para aumentar la comprensión del problema y contribuir a esclarecer la terminología o nomenclatura del dominio.

Especificación de Requisitos

Al igual que la descripción de requisitos la especificación cuenta con 6 requisitos cada uno listados junto a los requisitos no funcionales.

2.11 Diseño de la solución en términos de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos componentes de código fuente, binarios o ejecutables. También puede contener paquetes utilizados para agrupar los elementos del modelo.

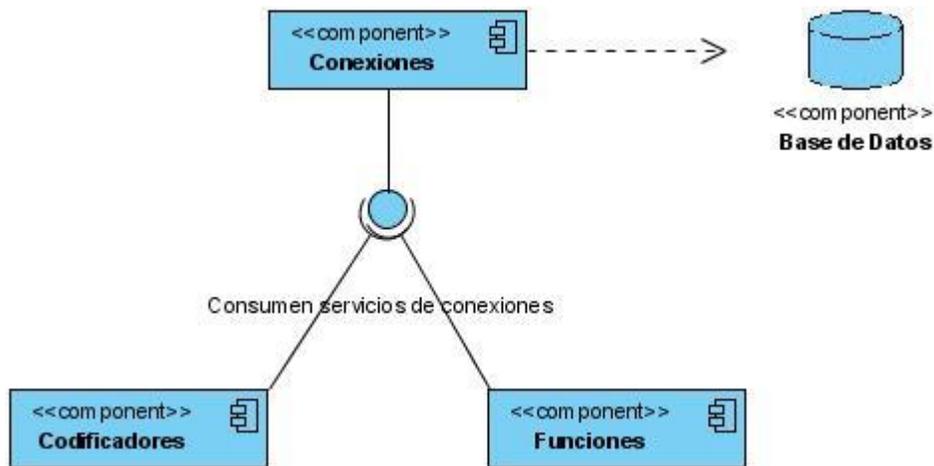


Figura 2. Diagrama de Componentes.

Cada componente está compuesto por varios requisitos. A continuación se explica de forma más detallada la funcionalidad de cada uno de estos componentes de acuerdo a los servicios que brindan y reciben.

Conexiones:

Este componente tiene a cargo la configuración de la conexión, la conexión a la Base de Datos y el cierre de la conexión, brindándole a los demás componentes del sistema una serie de servicios. El mismo es el que establece la conexión con la Base de Datos.

Funciones y Codificadores:

Estos componentes reciben servicios del componente Conexiones. Mediante estos servicios se podrán realizar todas las actividades que brindan cada uno de estos componentes porque para realizar las mismas es necesario que la conexión a la Base de Datos este creada.

Base de Datos:

Es donde se encuentra almacenada toda la información con la que trabajan estos componentes.

2.12 Diseño de clases

Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas con el objetivo de modelar los aspectos estáticos y definir una solución de diseño.

El diagrama de clases realizado para darle solución al problema planteado cuenta con 12 clases, cada una de ellas con sus atributos y métodos correspondientes y las relaciones que se establecen entre ellas. A continuación se muestra la representación del diagrama.

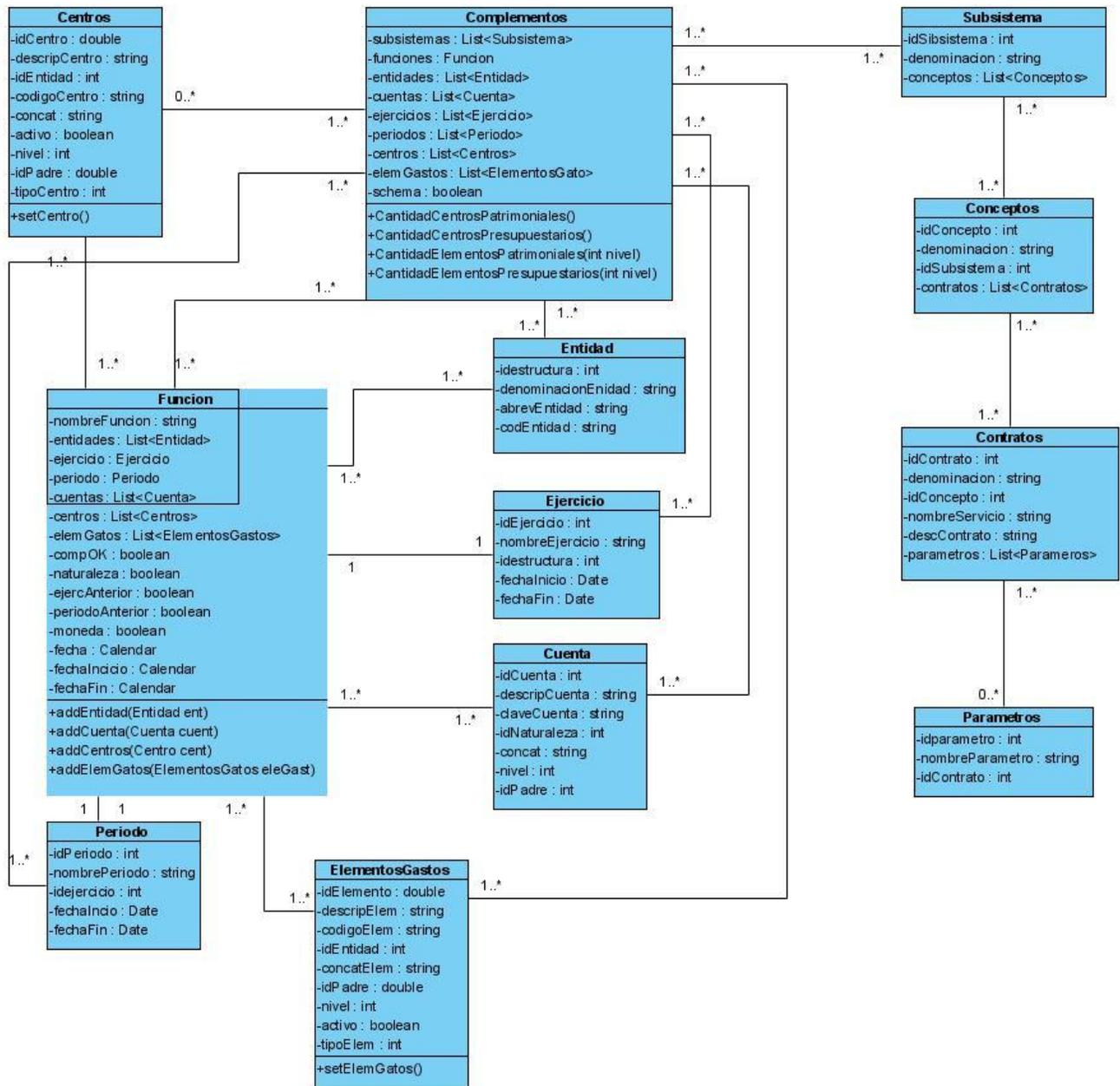


Figura 3. Diagrama de clases.

2.13 Modelo de datos

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo.

Los mismos se dividen en tres grupos:

- Modelos lógicos basados en objetos.
- Modelos físicos de datos.
- Modelos lógicos basados en registros.

Los modelos lógicos basados en registros se utilizan para describir datos en los niveles conceptuales y físicos, sin embargo los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a un nivel más alto de la implementación. De todos estos modelos el más utilizado es el modelo relacional porque se utiliza para exponer la solución.

En el Modelo Relacional se pueden representar los datos y las relaciones entre ellos, a través de una colección de tablas, en las cuales las filas equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características de cada registro localizado en la fila.

El modelo de datos propuesto para la solución cuenta con 4 tablas. El mismo se presenta a continuación.

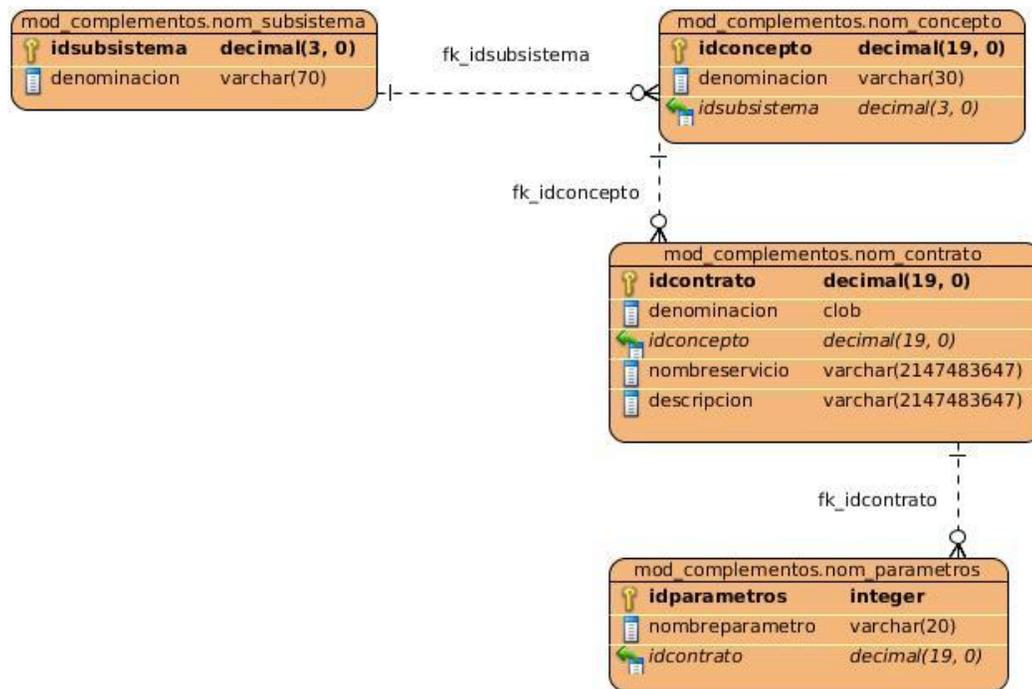


Figura 4. Modelo de datos de la Base de Datos.

2.14 Patrones de diseño empleados

Para la elaboración del diseño fue necesario utilizar una serie de patrones que de manera general constituyen soluciones simples a problemas comunes del diseño orientado a objeto. Los patrones que se emplearon para realizar el diseño de la aplicación fueron los GRASP y GOF:

- Instancia única (Singleton): este patrón se puede ver representado en la clase Complemento ya que esta clase tiene instancia con otras clases y así de esta manera puede utilizar los métodos de las otras clases sin necesidad de tenerlos implementados.
- Bajo Acoplamiento y Alta Cohesión: Estos patrones se ponen de ejemplo en la clase Concepto ya que como el mismo dice a través de esta clase le puede asignar responsabilidades a otras clases con las que esté relacionada tratando de que exista la menor cantidad de dependencias, pero al mismo tiempo deben existir.

- Experto: Se puede ver este patrón en la clase Subsistema, el mismo plantea que cada clase debe especializarse en lo que es, es decir la clase Subsistema por ejemplo solo debe devolver subsistemas.

2.15 Descripción de las clases y funcionalidades el componente

Nombre: Complemetos	
Tipo de clase:	
Atributo	Tipo
subsistemas	List<Subsistema>
funciones	Funcion
entidades	List<Entidad>
cuentas	List<Cuenta>
ejercicios	List<Ejercicio>
periodos	List<Periodo>
centros	List<Centros>
elemGastos	List<ElementosGasto>
schema	boolean
Para cada responsabilidad:	
Nombre:	Descripción:
CantidadCentrosPatrimoniales()	Busca en la clase Centros la cantidad de centros patrimoniales y eso es lo que devuelve.
CantidadCentrosPresupuestarios()	Busca en la clase elemGatos la cantidad de centros presupuestarios y eso es lo que devuelve.

CantidadElementosPatrimoniales(int nivel)	Da la cantidad de elementos patrimoniales buscándolo en la clase elemGastos pasándole como parámetro un entero.
CantidadElementosPresupuestarios(int nivel)	Da la cantidad de elementos presupuestarios buscándolo en la clase elemGastos pasándole como parámetro un entero.

Tabla 2. Descripción de la clase Complementos.

2.16 Descripción general del funcionamiento de componente

Complementos para CedruX es una herramienta que permite obtener información sin mayores complejidades de uso, expresando sus resultados en una gama de funciones tipo y predeterminadas. Interactúa con Open Calc, la Hoja de Cálculo del paquete de ofimática OpenOffice, utilizando sus facilidades para el diseño de información. La concepción de estas funciones está ligada en esta primera versión a los subsistemas de Contabilidad y Costo y Procesos del sistema CedruX. Estas funciones incorporan un valor agregado a la funcionabilidad y adaptabilidad del sistema a los requerimientos más disímiles de los distintos usuarios en materia de salida de información.

El usuario a través del Open Calc accederá a las opciones de conexión, estableciendo los parámetros necesarios para iniciar una conexión con la Base de Datos del sistema CedruX, para extraer la información necesaria para ejecutar las funciones. Una vez establecida la conexión el usuario podrá proceder a insertar funciones de Contabilidad y Costo y Procesos que arrojarán resultados tales como el saldo en determinado período y ejercicio, o en determinado centro de costo.

Además el usuario podrá también realizar un grupo de reportes de los denominados codificadores, es decir sobre las entidades, centros, ejercicios, períodos, elementos de gasto que contribuyen a lograr un mejor análisis de los estados financieros.

2.17 Conclusiones

En este capítulo se llega a una posible solución mediante el diseño de los diagramas de clase y componente. De forma general los artefactos que se generaron como resultado de la especificación de los

requisitos fueron: El Modelo Conceptual, la Especificación de requisitos y los Prototipos de interfaz de usuario. A partir de ellos se pudo llevar a cabo el análisis y el diseño de la herramienta.

Capítulo 3. Validación de la solución propuesta

3.1 Introducción

En este capítulo se validan los requisitos y el diseño a través de las métricas y las técnicas, de esta forma se asegura que los requisitos cumplen con las restricciones solicitadas por el cliente.

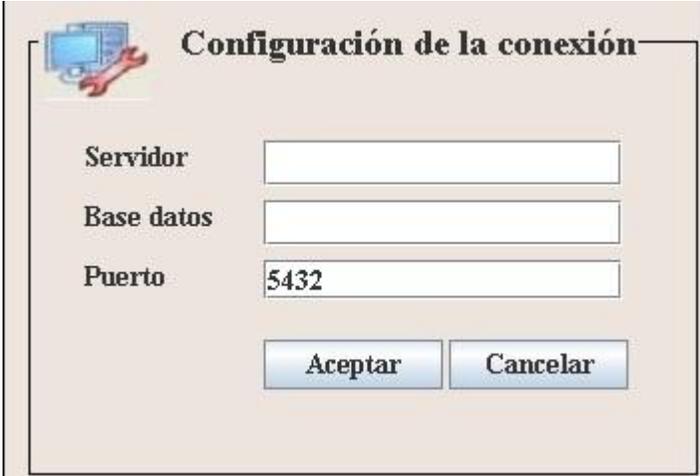
3.2 Aplicación de técnicas de validación de requisitos

Los requisitos funcionales se validan con el objetivo de verificar que estos hayan sido especificados sin ambigüedades, inconsistencias, omisiones y que el resultado obtenido se ajuste a los estándares establecidos por el proyecto. Para la validación de los requisitos capturados se utilizaron las siguientes técnicas.

3.2.1 Prototipo de interfaz

Los prototipos de interfaz de usuario son una representación no funcional del sistema, que se le presenta al cliente para validar que cumple con todas las necesidades que el mismo requiere. La utilización de esta técnica tiende a confundir al cliente, pues en ocasiones confunden los prototipos con el sistema final.

A continuación se muestran los prototipos más importantes:



Configuración de la conexión

Servidor

Base datos

Puerto

Figura 4. Prototipo de interfaz gráfica de usuario Configurar Conexión.

Autenticación

Servidor: 10.12.179.3

BD: erp2

puerto: 5432

Usuario:

Contraseña:

Configuración Aceptar Cancelar

Figura 5. Prototipo de interfaz gráfica de usuario Conectarse a la Bases de Dato.

El resto de los prototipos de interfaz de usuario están representados en los Anexos del 18 al 20.

3.2.2 La Revisión Técnica Formal (RTF)

Luego de haber terminado la especificación de cada uno de los requisitos se procede a revisarlos, cada revisión se lleva a cabo mediante una reunión y solo tiene éxito si esta es bien planificada, controlada y atendida por los especialistas. Las revisiones técnicas formales se realizaron correctamente donde se logró validar que cada requisito cumpliera con lo solicitado por el cliente.

3.3 Métricas para la validación del diseño del modelo propuesto

3.3.1 Tamaño Operacional de Clases (TOC)

Para realizar la validación del diseño propuesto para el sistema se utilizó la métrica Tamaño Operacional de Clases (**TOC**) donde se tuvo en cuenta la cantidad de procedimientos que tenía cada clase, a partir de dichos procedimientos y mediante un criterio se obtuvo la categoría (alta, media, baja) para la Responsabilidad, Complejidad y Reutilización. Para este diseño no fue necesario utilizar las otras métricas como la relacionada con la herencia entre clases porque las clases del sistema no presentan herencia lo que demuestra que el diseño es sencillo y sin mucha complicación.

Luego de efectuar los cálculos para un:

Total de clases	12
Promedio de procedimientos	0,833333333

Clase	Cantidad de Procedimientos
Complementos	4
Centro	1
Conceptos	0
Contratos	0
Cuenta	0
Ejercicio	0
ElementosGasto	1
Entidad	0
Funcion	4
Parametros	0
Periodos	0
Subsistema	0

Tabla 3. Cantidad de Procedimientos por clase.

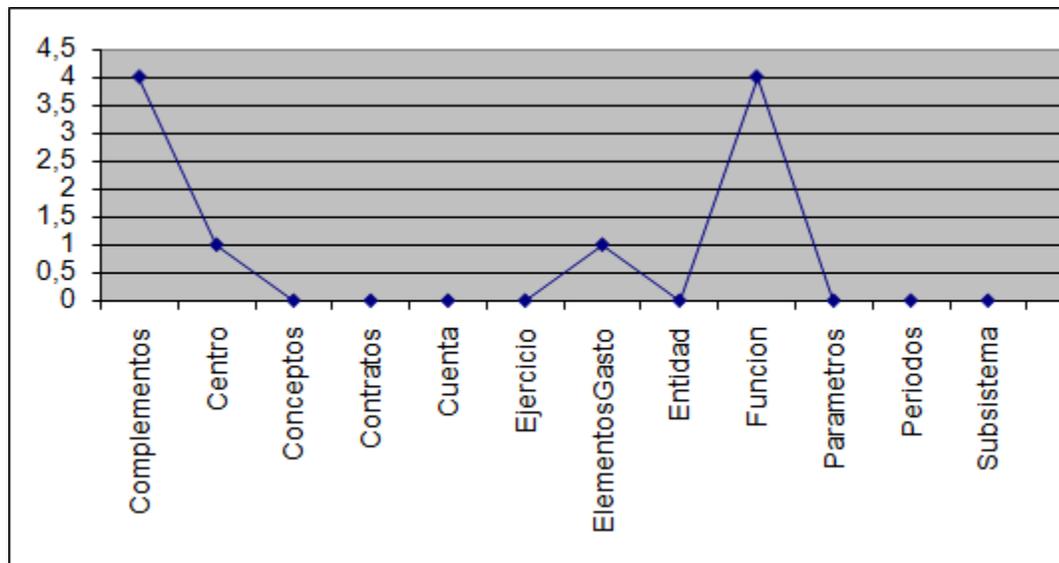


Figura 6. Cantidad de Procedimientos por clases.

	Categoría	Criterio
Responsabilidad	Baja	< =Prom.

	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Tabla 4. Criterio para calcular la Responsabilidad, la Complejidad implementación y la Reutilización.

A través de estas tablas para determinar la categoría se obtuvo las siguientes gráficas de porcentaje para la Responsabilidad, Complejidad y Reutilización:

Responsabilidad	Cantidad de clases
Baja	8
Media	2
Alta	2

Tabla 5. Responsabilidad de las clases.

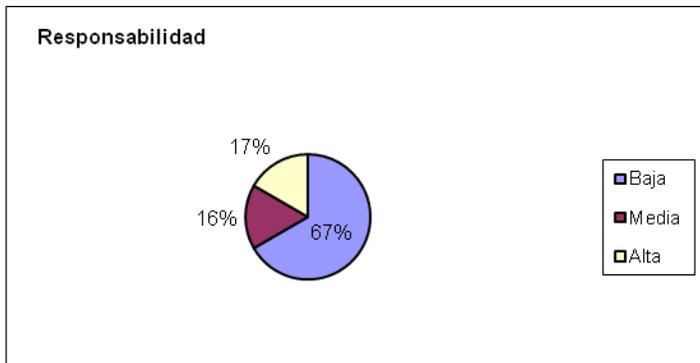


Figura 7. Representación de la Responsabilidad.

Complejidad	Cantidad de clases
Baja	8

Media	2
Alta	2

Tabla 6. Complejidad de las clases.

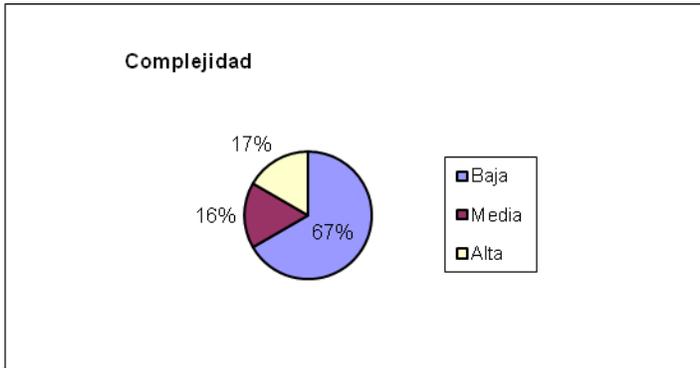


Figura 8. Representación de la Complejidad.

Como se pudo apreciar en las representaciones anteriores la Complejidad y la Responsabilidad son realmente bajas, lo que indica que el diseño es satisfactorio y la implementación no será de mucha complejidad.

Reutilización	Cantidad de clases
Alta	8
Media	2
Baja	2

Tabla 7. Reutilización de las clases.

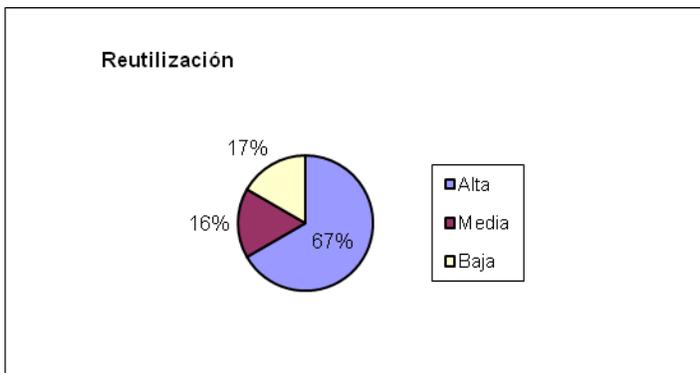


Figura 9. Representación de la Reutilización.

3.3.2 Relaciones entre clases (RC)

Para aplicar la métrica Relaciones entre Clases (RC) se tuvo en cuenta la cantidad de relaciones entre clases, a partir de estas relaciones y mediante un criterio se obtuvo la categoría (alta, media, baja y ninguna en caso del Acoplamiento) y (alta, media, baja) para la Complejidad de Mantenimiento y Reutilización.

Luego de efectuar los cálculos para un:

Total de clases	12
Promedio de asociaciones de uso	1.166666667

Clase	Cantidad de Relaciones de Uso
Complementos	0
Centros	2
Conceptos	1
Contratos	1
Cuenta	2
Ejercicio	1
ElementosGasto	2
Entidad	2
Funcion	0
Parametros	1
Periodos	1
Subsistema	1

Tabla 8. Relaciones de uso por clases.

Acoplamiento	Cantidad de clases
Ninguno	2
Bajo	6
Medio	4
Alto	0

Tabla 9. Acoplamiento de las clases.

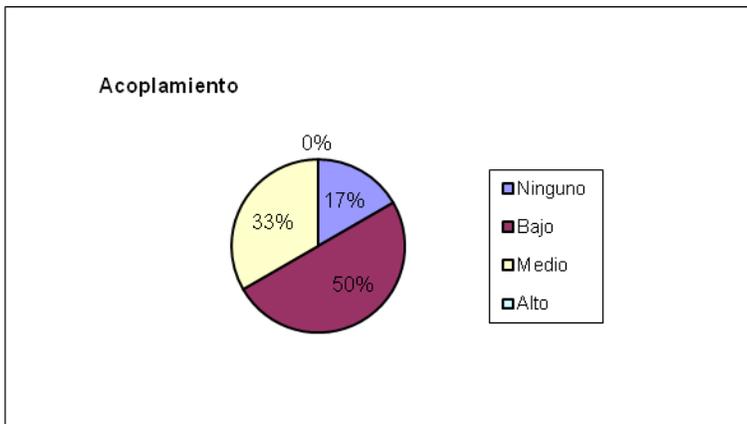


Figura 10. Representación del Acoplamiento de las Clases.

Complejidad de Mantenimiento	Cantidad de clases
Baja	8
Media	4
Alta	0

Tabla 10. Complejidad de Mantenimiento de las clases.

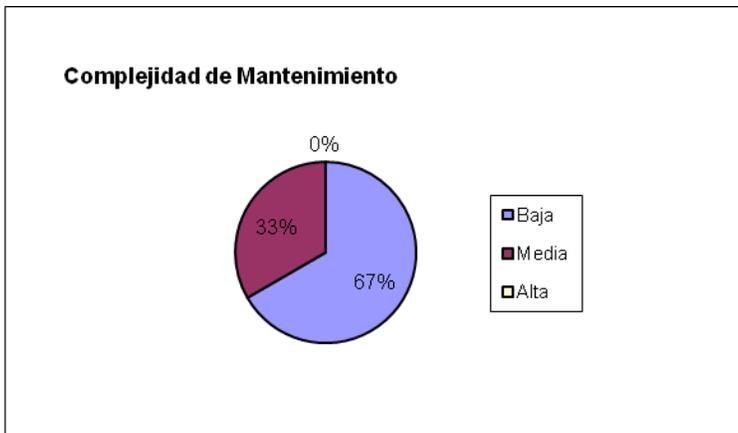


Figura 11. Representación de la Complejidad de Mantenimiento de las clases.

Reutilización	Cantidad de clases
Baja	0
Media	4
Alta	8

Tabla 11. Reutilización de las clases.

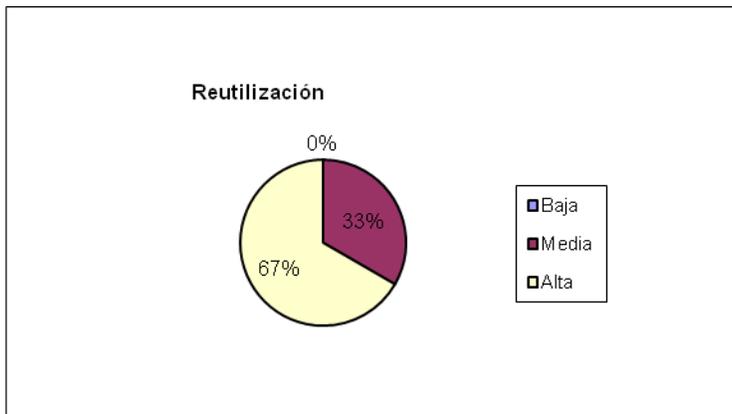


Figura 12. Representación de la Reutilización de las clases.

El cálculo de esta métrica al igual que la TOC demuestra que el diseño realizado para el sistema es de poca complejidad y que está entre los límites aceptables de calidad, porque como bien lo explican en las gráficas la Complejidad y el Acoplamiento son bajos.

3.4 Conclusiones

Las métricas y las técnicas utilizadas para la validación demostraron que los requisitos y el diseño están correctos y que cumplen con las necesidades del sistema, por lo que de esta forma se procederá a una buena implementación del sistema.

Conclusiones

Una vez concluido el presente trabajo se considera que se cumplió el objetivo general trazado, obteniéndose las siguientes conclusiones:

- Se realizó el estudio del estado del arte relacionado con la investigación que permitió definir conceptos fundamentales, herramientas y tecnologías relacionadas con la misma.
- Se realizó el análisis y diseño de la solución propuesta que permitió elaborar los artefactos necesarios para una posterior implementación de la misma.
- Se aplicaron las técnicas RTF y Prototipos para realizar la validación de los requisitos y las métricas TOC y RC que permitieron validar la calidad del diseño propuesto.

Recomendaciones

- Realizar el análisis y diseño de la funcionalidad para gestionar plantillas.
- Incluir funciones para los subsistemas de Planificación y Logística.

Referencias bibliográficas

1. **Camacho, Isaac Camacho.** *Desarrollo de software para predicción de propiedades termodinámicas de compuestos puros, mezclas binarias y multicomponentes en fase homogénea y heterogénea haciendo uso de microsoft excel y visual basic for applications.* Universidad de las Américas Puebla : s.n., 2003.
2. **Ramírez, Israel J.** *Las Hojas Electrónicas de Cálculo .* UNIVERSIDAD DE LOS ANDES : s.n., 2010.
3. **Leire Aldaz, Begoña Eguía y Leire Urcola.** *Análisis Estadístico de Datos.* 2009.
4. **F. Rodríguez, M. Broto, I. Lizarralde.** *cubiFOR: Complemento de Excel para cubicar, clasificar productos, calcular biomasa y CO2 en masas forestales de Castilla y León.* 2005.
5. *Optimización con simulación.* 2010.
6. **M.J. Escalona, N. Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo.*
7. *Conferencias de ingeniería de requisitos.* 2008.
8. *BPMN Business Process Modeling Notation.* 2006.
9. **López, Angel.** *Java la programación del futuro.* 1997.
10. **Informática, Sub-Jefatura de.** *Herramientas Case.* 2000.
11. Sitio de descargas de software. *Sitio de descargas de software.* [En línea] 2004-2010.
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
12. **González, Carlos D.** *Curso Base de Datos PostgreSQL, SQL avanzado y PHP. Curso Base de Datos PostgreSQL, SQL avanzado y PHP.* [En línea] Mayo de 2011.
<http://www.usabilidadweb.com.ar/postgre.php>.
13. **C, Larman.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 2004.
14. **José Sáez Martínez, Jesús García Molina, Pedro J. Jiménez García.** *Una Arquitectura para una Herramienta.* Universidad de Murcia : s.n., 1999.
15. **Rojas, M.C. Juan Carlos Olivares.** *Patrones de Diseño.* 2004.

Bibliografía

1. **José Sáez Martínez, Jesús García Molina, Pedro J. Jiménez García.** *Una Arquitectura para una Herramienta.* Universidad de Murcia : s.n., 1999.
2. **Rojas, M.C. Juan Carlos Olivares.** *Patrones de Diseño.* 2004.
3. **Gervás, Pablo.** *Captura de requisitos.* 2004.
4. **Cuaresma, Dra. María José Escalona y Romano, Dr. José Mariano González.** *Metodología y Técnicas en Proyectos de software para la web.* 2006-2007.
5. *Herramientas Case.*
6. **Ing. Miguel Ángel Sánchez Palmero, Ing. Olga Yarisbel Rojas Grass.** *LOS ACTIVOS FIJOS INTANGIBLES EN EL CONTEXTO DEL ERP CUBANO.* Ciudad de la Habana : s.n., 2009.
7. **Tomalá, Silvia Jazmín Cochea.** *Métricas de Calidad de los Sistemas de Información – aplicación en la Certificación de Calidad de un Sistema de un empresa del sector hidrocarburífero.* Guayaquil – Ecuador : s.n., 2009.
8. **Monzón, Antonio.** *Calidad de la Especificación: ¿Se pueden medir los Requisitos?*
9. **Altamirano, Ing. Alfonso Valdez.** *Comparativo de Entornos de Desarrollo Integrados.*
10. **Leire Aldaz, Begoña Eguía y Leire Urcola.** *TRATAMIENTO DE DATOS CON HOJA DE CÁLCULO.* 2009.
11. **Molpeceres, Alberto.** *Procesos de Desarrollo: RUP, XP, FDD.* 2002.
12. **F. Rodríguez, M. Broto, I. Lizarralde.** *cubiFOR: Complemento de Excel para cubicar, clasificar productos, calcular biomasa y CO2 en masas forestales de Castilla y León.* 2005.
13. **Systems, Popkin Software and.** *Modelado de Sistemas com UML.*
14. **Andrés, Jose Onofre Montesa.** *EL PROCESO DE DESARROLLO DE SOFTWARE.* Valencia : s.n., 2000.
15. **Giraldo, Luis y Zapata, Yuliana.** *HERRAMIENTAS DE DESARROLLO DE INGENIERIA DE SW PARA LINUX.* s.l. : Monitoria de Ingesoft, 2005.
16. **Ramírez, Prof. Israel J.** *Las Hojas Electrónicas de Cálculo.*

-
17. **M.J. Escalona, N. Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo.*
 18. **Víctor Anaya, Patricio Letelier.** *Trazabilidad de Requisitos Adaptada a las Necesidades del Proyecto: Un Caso de Estudio Usando Alternativamente RUP y XP.* Valencia-España : s.n., 2003.
 19. **López, Angél.** *Java la programación del futuro.* 1997.
 20. **Informática, Sub-Jefatura de.** *Herramientas Case.* 1999.
 21. **León, Rolando Alfredo Hernández y González, Sayda Coello.** *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA.* Ciudad de la Habana : s.n., 2002.
 22. **Leire Aldaz, Begoña Eguía y Leire Urcola.** *TRATAMIENTO DE DATOS CON HOJA DE CÁLCULO.*
 23. **Sanchez, María A. Mendoza.** *Metodologías De Desarrollo De Software.* Perú S.A.C. : s.n., 2004.
 24. **Méndez, Alejandra Virrueta.** *INVESTIGACIÓN DOCUMENTAL METODOLOGÍAS DE DESARROLLO DE SOFTWARE.* 2010.
 25. **Robles, M.C. Esperanza Aguillón.** *Métricas para la Gestión de Proyectos de Software.*
 26. **Dans, Enrique.** *HOJAS DE CALCULO: LAS GRANDES DESCONOCIDAS.*
 27. **Javier Berrocal, José Manuel García Alonso, Juan Manuel Murillo Rodríguez.** *Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio.* Universidad de Extremadura : s.n., 2009.
 28. *RISKOptimizer.* s.l. : Palisade Corporation, 2009.
 29. *WEBMASTERS: CONTROL DE VERSIONES USANDO SUBVERSION.* **GALÍNDEZ, RODRIGO.** 2008.
 30. **Berndtsson, Mikael, y otros.** *Thesis Projects.* 2008.
 31. **Mora, Francisco.** *UML: Lenguaje Unificado de Modelaje.* Universidad de Alicante : s.n., 2002-2003.
 32. **Rivera, Javier Fernández.** *Modelo de Datos.*
 33. **Rojas, M.C. Juan Carlos Olivares.** *Patrones de Diseño.*
 34. **Diego, Javier Nieto y Fernández, Pablo Ramos.** *El Patrón Fachada.* Universidad de Salamanca : s.n.
 35. **Pereira, Leisniel Ignacio Cabrera y Gómez, Mailyn Hernández.** *Análisis y Diseño del Módulo de Cobros y Pagos del sistema integral de gestión CEDRUX.* Ciudad de la Habana : s.n., 2009.

36. **Figueredo, Rolando Avila y Rodón, Damián Cervantes.** *ANÁLISIS Y DISEÑO DEL SISTEMA DE VENTA Y ADMISIÓN DE LA PLATAFORMA DE SERVICIOS POSTALES.* 2007.
37. *Procesos de desarrollo y gestión de proyectos de software.* 2009.
38. **Arregui, Juan José Olmedilla.** *Revisión Sistemática de Métricas de Diseño Orientado a Objetos.* Madrid : s.n., 2005.
39. **Dra. María José Escalona Cuaresma, Dr. José Mariano González Romano.** *Metodología y Técnicas en Proyectos software para la Web.* Sevilla : s.n., 2006.
40. Jazztel. [Online] 2011. <http://www.anexom.es/servicios-en-la-red/ofimatica/las-mejores-extensiones-de-openoffice-para-calc-i/>.
41. **Alan.** [Online] Octubre 2008. <http://www.pcdigital.org/como-usar-solver-en-openoffice/>.

Anexos

Anexo 1. Descripción de la clase Centro.

Nombre: Centro	
Tipo de clase:	
Atributo	Tipo
idCentro	double
descripCentro	String
idEntidad	int
codigoCentro	String
concat	String
activo	boolean
nivel	int
idPadre	double
tipoCentro	int
Para cada responsabilidad:	
Nombre:	Descripción:
setCentro()	Cambia los atributos de esta clase.

Anexo 2. Descripción de clase Conceptos.

Nombre: Conceptos	
Tipo de clase:	
Atributo	Tipo
idConcepto	int

denominacion	String
idSubsistema	int
contratos	List<Contratos>
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 3. Descripción de la clase Contratos.

Nombre: Contratos	
Tipo de clase:	
Atributo	Tipo
idContrato	int
denominacion	String
idConcepto	int
nombreServicio	String
descContrato	String
parametros	List<Parametros>
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 4. Descripción de la clase Cuenta.

Nombre: Cuenta	
Tipo de clase:	

Atributo	Tipo
idCuenta	int
descripCuenta	String
claveCuenta	String
idNaturaleza	int
concat	String
nivel	int
idPadre	int
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 5. Descripción de la clase Ejercicio.

Nombre: Ejercicio	
Tipo de clase:	
Atributo	Tipo
idEjercicio	int
nombreEjercicio	String
idestructura	int
fechaInicio	Date
fechaFin	Date
Para cada responsabilidad:	
Nombre:	Descripción:

--	--

Anexo 6. Descripción de la clase ElementosGatos.

Nombre: ElementosGatos	
Tipo de clase:	
Atributo	Tipo
idElemento	double
descripElem	String
codigoElem	String
idEtnidad	int
concatElem	String
idPadre	double
nivel	int
activo	boolean
tipoElem	int
Para cada responsabilidad:	
Nombre:	Descripción:
setElemGastos()	Cambia los atributos de esta clase.

Anexo 7. Descripción de la clase Entidad.

Nombre: Entidad	
Tipo de clase:	
Atributo	Tipo
idestructura	int

denominacionEntidad	String
abrevEntidad	String
codEntidad	String
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 8. Descripción de la clase Funcion.

Nombre: Funcion	
Tipo de clase:	
Atributo	Tipo
nombreFuncion	String
entidades	List<Entidad>
ejercicio	Ejercicio
periodo	Periodo
cuentas	List<Cuenta>
centros	List<Centros>
elemGastos	List<ElementosGasto>
compOK	boolean
naturaleza	boolean
ejercAnterior	boolean
periodoAnterior	boolean
moneda	boolean

fecha	Calendar
fechaInicio	Calendar
fehcaFin	Calendar
Para cada responsabilidad:	
Nombre:	Descripción:
addEntidad(Entidad ent)	Adiciona una entidad pasándolo por parámetro.
addCuenta(Cuenta cuent)	Adiciona una cuenta pasándolo por parámetro.
addCentros(Centros cent)	Adiciona un centro pasándolo por parámetro.
addElemntoGasto(ElementosGasto eleGast)	Adiciona un elemento gasto pasándolo por parámetro.

Anexo 9. Descripción de la clase Parametros.

Nombre: Parametros	
Tipo de clase:	
Atributo	Tipo
idParametro	int
nombreParametro	String
idContrato	int
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 10. Descripción de la clase Periodo.

Nombre: Periodo	
Tipo de clase:	
Atributo	Tipo
idPeriodo	int
nombrePeriodo	String
idejercicio	int
fechaInicio	Date
fechaFin	Date
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 11. Descripción de la clase Subsistema.

Nombre: Subsistema	
Tipo de clase:	
Atributo	Tipo
idSubsistema	int
denominacion	String
concepto	List<Conceptos>
Para cada responsabilidad:	
Nombre:	Descripción:

Anexo 12. Descripción textual del requisito Configurar Conexión.

Precondiciones	N/A
Flujo de eventos	
Flujo básico Configurar la Conexión	
1	Se establecen los parámetros: Servidor Base de Datos Puerto
2	El sistema valida los datos introducidos.
3	Si no hay campos vacíos.
4	Se presiona el botón Aceptar.
5	Concluye el requisito.
Pos-condiciones	
1	N/A
2	
Flujos alternativos	
Flujo alternativo 3.a Campos Vacíos	
1	Se presiona el botón Aceptar.
2	Se muestra un mensaje de error que se deben llenar los campos.
3	Se vuelve al paso 2 del flujo básico.
4	
Pos-condiciones	
1	N/A
Flujo alternativo *.a Se Cancela Operación	
1	Concluye el requisito.
Pos-condiciones	
1	No se realiza la conexión a la base de datos.
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.
Conceptos	Configurar la Conexión
	Visibles en la interfaz: Servidor Base de Datos
	Puerto Utilizados internamente: N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.
Asuntos pendientes	N/A

Anexo 13. Descripción textual del requisito Conectar a Base de Datos.

Precondiciones	Es necesario haber configurado la conexión.
Flujo de eventos	
Flujo básico Conectar a Base de Datos	
1	Se establecen los parámetros: Usuario Contraseña
2	El sistema valida los datos introducidos.
3	Si no hay campos vacíos.
4	Se presiona el botón Aceptar.
5	Concluye el requisito.
Pos-condiciones	
1	Se establece la conexión según los parámetros especificados.
2	Se cargan en los combobox de la cinta de opciones las Entidades, Cuentas, Ejercicios y Periodos
Flujos alternativos	
Flujo alternativo 3.a Campos Vacíos	
1	Se presiona el botón Aceptar.
2	Se muestra un mensaje de error que se deben llenar los campos.
3	Se vuelve al paso 2 del flujo básico.
4	
Pos-condiciones	
1	N/A
Flujo alternativo 2.a Datos Inválidos	
1	Se introducen datos inválidos.
2	Se presiona el botón Aceptar.
3	El sistema muestra un mensaje de error.
4	Se vuelve al paso 2 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo *.a Se Cancela Operación	
1	Concluye el requisito.
Pos-condiciones	
1	No se realiza la conexión a la base de datos.
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.

Conceptos	Conectar a la Base de Datos	Visibles en la interfaz: Usuario Contraseña Puerto Utilizados internamente: N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.	
Asuntos pendientes	N/A	

Anexo 14. Descripción textual del requisito Cerrar Conexión.

Precondiciones	N/A	
Flujo de eventos	Flujo básico Cerrar Conexión	
1	Se cierra la conexión a la Base de Datos.	
5		
Pos-condiciones		
1	N/A	
2		
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.	
Conceptos	Cerrar Conexión	Visibles en la interfaz: N/A Puerto Utilizados internamente: N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.	
Asuntos pendientes	N/A	

Anexo 15. Descripción textual del requisito Insertar Funciones de Contabilidad.

Precondiciones	Es necesario haber establecido una conexión a la base de datos.	
Flujo de eventos	Flujo básico Insertar Funciones de Contabilidad	
6	Se selecciona la función.	
7	Se presiona el botón Siguiente.	

8	Se introducen todos los parámetros de la función seleccionada a través del Asistente. Entidades Ejercicio Contable Período Contable Fecha Cuentas Contables Centros de Costos Elementos de Gastos
9	Se presiona el botón Finalizar.
10	Concluye el requisito.
Pos-condiciones	
3	Se inserta la función seleccionada en la celda activa.
4	
Flujos alternativos	
Flujo alternativo *.a Se cancela la operación.	
5	Concluye el requisito.
6	
Pos-condiciones	
1	N/A
Flujo alternativo 1.a No se marca una función.	
1	No se selecciona ninguna función.
2	Se presiona el botón Siguiente.
3	El sistema muestra un mensaje de error indicando que tiene que marcar una función.
4	Se vuelve al paso 1 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 3.a No se marca ninguna entidad.	
1	No se selecciona ninguna entidad.
2	Se presiona el botón Siguiente.
3	El sistema muestra un mensaje de error indicando que tiene seleccionar una entidad.
4	Se vuelve al paso 3 del flujo básico.
Pos-condiciones	
2	N/A
3	
Validaciones	
2	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.

Conceptos	Saldo Ejercicio	Visibles en la interfaz: <<Nombre del atributo 1>>
		Utilizados internamente: N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.	
Asuntos pendientes	N/A	

Anexo 16. Descripción del requisito Insertar funciones de Costo y Proceso.

Precondiciones	Es necesario haber establecido una conexión a la base de datos.
Flujo de eventos	
Flujo básico Insertar Funciones de Costo y Procesos	
11	Se selecciona la función.
12	Se presiona el botón Siguiente.
13	Se introducen todos los parámetros de la función seleccionada a través del Asistente. Entidades Ejercicio Contable Período Contable Fecha Cuentas Contables Centros de Costos Elementos de Gastos
14	Se presiona el botón Finalizar.
15	Concluye el requisito.
Pos-condiciones	
5	Se inserta la función seleccionada en la celda activa.
6	
Flujos alternativos	
Flujo alternativo *.a Se cancela la operación.	
7	Concluye el requisito.
8	
Pos-condiciones	
1	N/A
Flujo alternativo 1.a No se marca una función.	
1	No se selecciona ninguna función.
2	Se presiona el botón Siguiente.

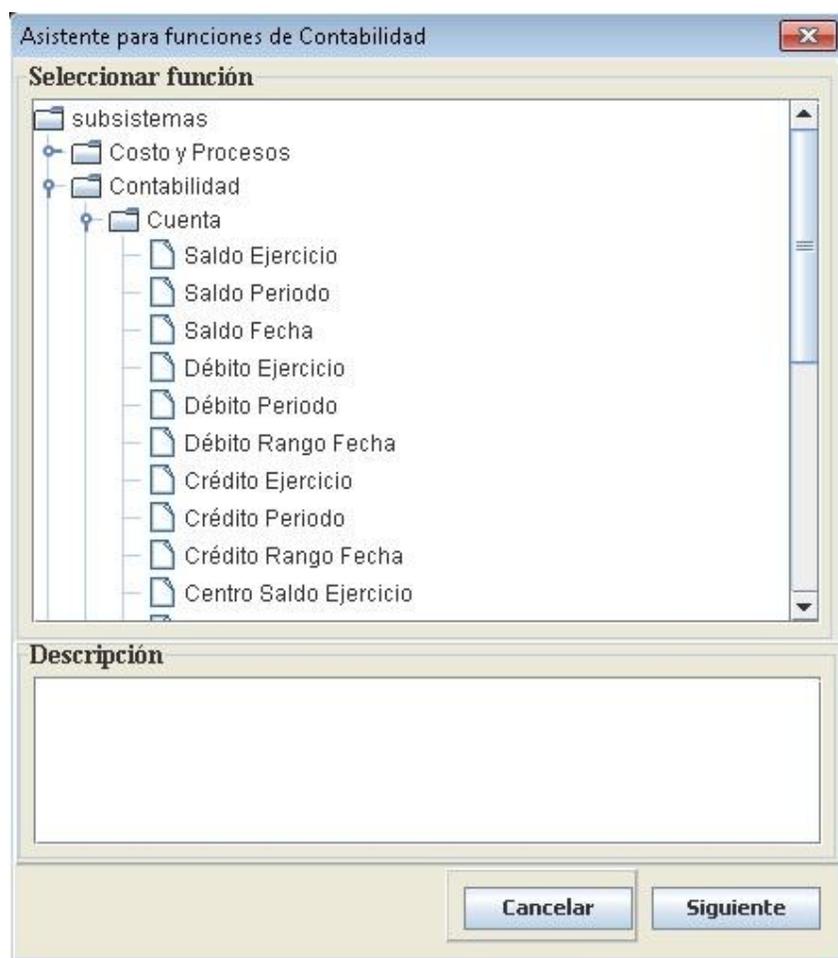
3	El sistema muestra un mensaje de error indicando que tiene que marcar una función.	
4	Se vuelve al paso 1 del flujo básico.	
Pos-condiciones		
1	N/A	
Flujo alternativo 3.a No se marca ninguna entidad.		
1	No se selecciona ninguna entidad.	
2	Se presiona el botón Siguiente.	
3	El sistema muestra un mensaje de error indicando que tiene seleccionar una entidad.	
4	Se vuelve al paso 3 del flujo básico.	
Pos-condiciones		
4	N/A	
5		
Validaciones		
3	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.	
Conceptos	Saldo	Visibles en la interfaz:
	Ejercicio	<<Nombre del atributo 1>>
		Utilizados internamente:
		N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.	
Asuntos pendientes	N/A	

Anexo 17. Descripción del requisito Insertar Codificador.

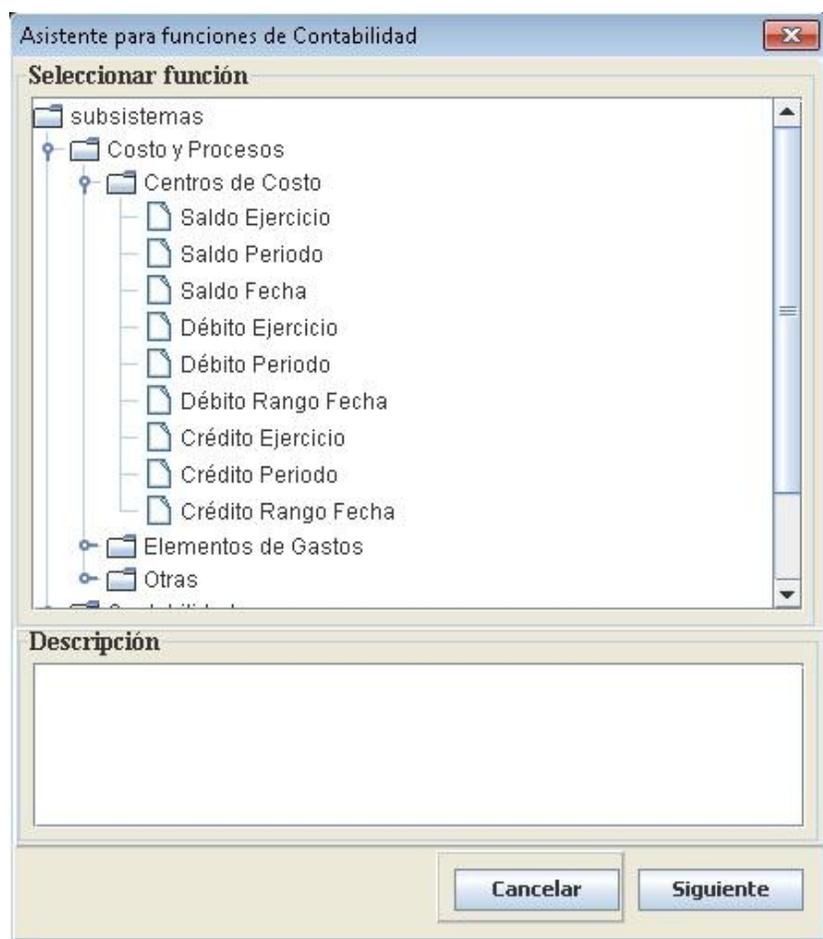
Precondiciones	Es necesario haber establecido la conexión a una Base de Datos.
Flujo de eventos	
Flujo básico Insertar Codificador	
16	Se definen los filtros según el codificador que se desee insertar. Entidades Ejercicios Periodos Cuentas Centros Elementos
17	Se presiona insertar.

18	Se concluye el requisito.	
19		
20		
Pos-condiciones		
7	Se cargan en la hoja Excel seleccionada y a partir de la celda activa los codificadores seleccionados.	
8		
Flujos alternativos		
Flujo alternativo *.a Se cancela la operación.		
9	Se cancela la operación.	
10		
Pos-condiciones		
6	N/A	
Validaciones		
4	Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.	
Conceptos	Insertar Codificador	Visibles en la interfaz: Entidades Ejercicios Periodos Cuentas Centros Elementos Utilizados internamente: N/A
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.	
Asuntos pendientes	N/A	

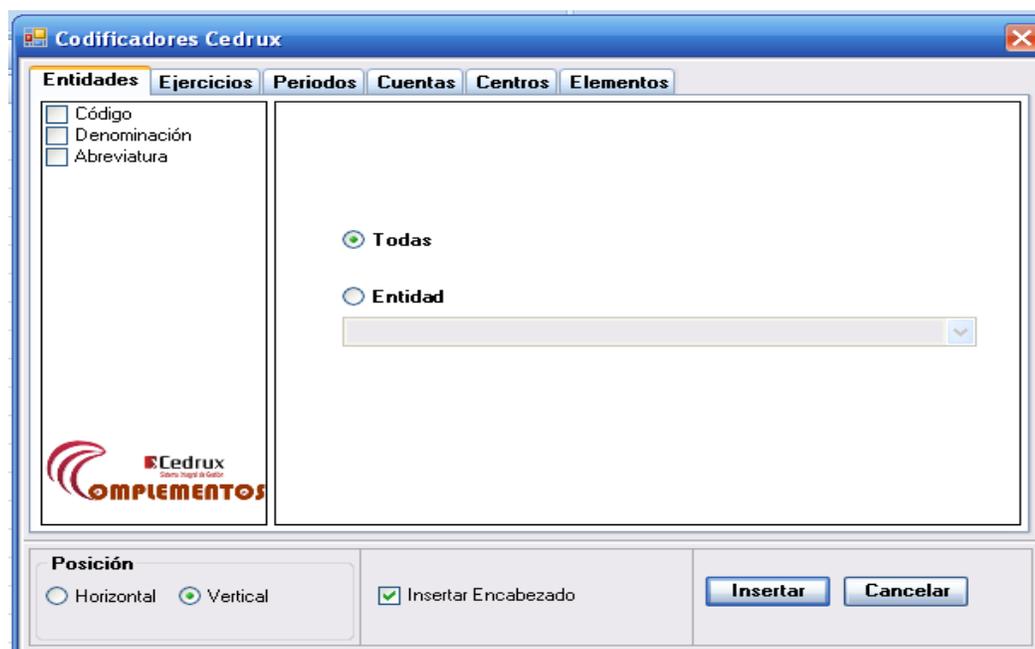
Anexo 18. Prototipo de interfaz de usuario “Insertar Funciones de Contabilidad”.



Anexo 19. Prototipo de interfaz de usuario “Insertar Funciones de Costo y Procesos”.



Anexo 20. Prototipo de interfaz de usuario “Insertar Codificador”.



Anexo 21. Especificación de los Requisitos.

No	Funcionalidad	Descripción	Complejidad	Prioridad
1	Configurar la conexión	Se configura la conexión a la base de datos siguiendo una serie de parámetros para la configuración.	Alto	Requerido
2	Conectar a Base de Datos	Se realiza la conexión con la base de datos siguiendo una serie de parámetros para la conexión.	Alto	Requerido
3	Cerrar conexión	Se cierra la conexión.	Media	Opcional
4	Insertar funciones de Contabilidad	Se insertan las funciones de contabilidad cuando se han establecidos la configuración y la conexión a la base de datos.	Media	Deseado
5	Insertar funciones de Costo y Procesos	Se insertan las funciones de costos y procesos cuando se han establecidos la configuración y la conexión a la base de datos.	Media	Deseado

6	Insertar Codificadores	Se insertan los codificadores cuando se han establecidos la configuración y la conexión a la base de datos.	Media	Requerido
---	------------------------	---	-------	-----------

Glosario

AFT: Activos Fijos Tangibles.

BPD: Business Process Diagram.

BPMN: Business Process Modeling Notation.

CASE: Computer Aided Software Engineering.

ERP: Planificación de Recursos Empresariales.

GOF: Gang of Four.

GRASP: General Responsibility Assignment Software Patterns.

IR: Ingeniería de Requisitos.

RTF: Revisión Técnica Formal.

TOC: Tamaño Operacional de Clases.

UML: Unified Modeling Language