

Universidad de las Ciencias Informáticas
Facultad 3



**Título: Desarrollo de la versión 1.1 del componente
Estados financieros del subsistema contabilidad
de XEDRO-ERP.**



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Alberto Mesa Martínez.

Tutores: Ing. Yordi Chaveco Bustamante.

Ing. Yadini Pérez López.

La Habana, 2014

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Alberto Mesa Martínez.

Firma del Autor.

Ing. Yadini Pérez López.

Firma del Tutor.

Ing. Yordi Chaveco Bustamante.

Firma del Tutor.

DATOS DE CONTACTO

Ing. Yordi Chaveco Bustamante:

Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas.

Correo electrónico: ybustamante@uci.cu

Ing. Yadini Pérez López:

Ingeniera en Ciencias Informáticas, Universidad de las Ciencias Informáticas.

Correo electrónico: yadini@uci.cu.

AGRADECIMIENTOS

A mi familia por todo el apoyo y dedicación que me han brindado durante toda mi vida y por estar siempre presentes, en los buenos y malos momentos.

A todos mis compañeros por permanecer junto a mí en la primera línea de batalla y sin retroceder.

A mis tutores por apoyarme y guiarme durante todos estos meses.

A todos los especialistas del proyecto por brindarme su ayuda en los momentos que la necesité.

A todos, muchas gracias.

DEDICATORIA

El presente trabajo lo dedico a las personas más importantes de mi vida, por ser mi fuente de inspiración, mis guías y ejemplo a seguir:

A mis queridos padres.

A mi hermanita.

RESUMEN

Como resultado del proceso de informatización que se ha estado llevando a cabo en diversos sectores del país, se ha propiciado, por parte de las entidades cubanas, el uso de herramientas informáticas. En tal sentido, la Universidad de las Ciencias Informáticas (UCI) desempeña un rol fundamental en la producción de soluciones y servicios. En la UCI se encuentra en desarrollo el sistema XEDRO-ERP, un paquete de soluciones integrales de gestión para las empresas cubanas. Dicho sistema posee un componente para la gestión de los estados financieros de las entidades presupuestadas y patrimoniales, el cual cuenta con un conjunto de opciones de configuración que son insuficientes para la personalización de los estados financieros establecidos en las nuevas resoluciones del Ministerio de Finanzas y Precios (MFP).

Con el objetivo de solucionar las dificultades detectadas se decide realizar la implementación de una nueva versión del componente Estados financieros, que cuente con las opciones de configuración necesarias para la personalización de los estados financieros. Para ello se realizó un estudio de distintos sistemas en Cuba y el mundo que permiten la obtención de estados financieros. Se seleccionó el modelo de desarrollo a emplear, junto con los lenguajes, tecnologías y herramientas para el diseño e implementación del producto. Aparejado a ello se obtuvieron los artefactos definidos para cada una de las disciplinas del modelo de desarrollo. Se realizó la implementación del componente cumpliendo con el análisis y el diseño realizado. Por último se realizaron pruebas de caja negra y preexperimento para validar la solución propuesta.

PALABRAS CLAVE: Estados financieros, opciones de configuración, personalización de estados financieros.

ÍNDICE	
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1. Introducción	4
1.2. Estados financieros	4
1.3. Marco legal	5
1.4. Sistemas informáticos	7
1.4.1 Versat Sarasola.....	8
1.4.2 Rodas XXI.....	8
1.4.3 OpenBravo.....	9
1.4.4 SAP ERP	10
1.4.5 Microsoft Dynamics AX	10
1.5. Modelo de desarrollo.....	11
1.5.1 Disciplina Modelado del negocio	11
1.5.2 Disciplina Requisitos	12
1.5.3 Disciplina Análisis y diseño	15
1.5.4 Disciplina Implementación.....	20
1.5.5 Disciplina Pruebas internas	20
1.6. Lenguajes, tecnologías y herramientas	22
1.6.1 Lenguajes de programación	22
1.6.2 Tecnologías	22
1.6.3 Lenguaje de modelado.....	23
1.6.4 Herramienta de modelado.....	23
1.6.5 Herramientas de programación	23
1.6.6 Herramienta para el control de versiones.....	24
1.6.7 Servidor web	24
1.6.8 Navegador web	24
1.6.9 Marco de trabajo	25
1.7. Conclusiones parciales	25
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	26
2.1. Introducción	26
2.2. Propuesta del sistema.....	26
2.3. Modelación del proceso de negocio	26

2.2.1.	Diagrama de proceso de negocio.....	27
2.2.2.	Descripción del proceso de negocio.....	28
2.2.3.	Modelo Conceptual	29
2.4.	Definición de los requisitos de software.....	31
2.4.1.	Requisitos funcionales:	31
2.4.2.	Descripción de requisitos	33
2.4.3.	Prototipos de interfaz de usuario.....	34
2.5.	Diseño de la solución	35
2.5.1.	Modelo de datos.....	35
2.5.2.	Diagrama de clase del diseño	38
2.5.3.	Patrones de diseño utilizados:.....	40
2.5.4.	Validación del diseño	41
2.6.	Conclusiones parciales	47
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN.....		48
3.1.	Introducción	48
3.2.	Implementación.....	48
3.2.1.	Diagrama de componentes	48
3.2.2.	Estándares de codificación	49
3.2.3.	Normas de comentarios	51
3.3.	Pruebas.....	52
3.3.1.	Diseño de casos de prueba.....	52
3.3.2.	Resultados de las pruebas.....	54
3.4.	Validación de la solución.....	54
3.5.	Conclusiones parciales	57
CONCLUSIONES.....		58
RECOMENDACIONES		59
BIBLIOGRAFÍA		60
GLOSARIO DE TÉRMINOS.....		64
ANEXOS.....		66
	Anexo 1. Modelo de datos	66
	Anexo 2. Juegos de datos a probar	66
ÍNDICE DE FIGURAS		
	Figura 1. Diagrama Proceso de Negocio: Obtención de estados financieros.....	27

Figura 2. Mapa conceptual.....	29
Figura 3. Prototipo de interfaz: Adicionar grupo.....	35
Figura 4. Sección del modelo de datos.....	36
Figura 5. Diagrama de clase del diseño: Gestionar grupo.	39
Figura 6. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.....	42
Figura 7. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.....	43
Figura 8. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.....	43
Figura 9. Representación en % de los resultados de la evaluación de la métrica RC en el atributo.....	45
Figura 10. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.....	45
Figura 11. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Mantenimiento.....	46
Figura 12. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.....	46
Figura 13. Diagrama de componentes.....	48
Figura 14. Notación <i>PascalCasing*</i> aplicada en el nombre de la clase EstadosFinancieroController.....	49
Figura 15. Notación <i>PascalCasing*</i> aplicada en el nombre de la clase NomEstadofModel.....	49
Figura 16. Notación <i>PascalCasing*</i> aplicada en el nombre de la clase NomEstadof.....	50
Figura 17. Notación <i>PascalCasing*</i> aplicada en el nombre de la clase BaseNomEstadof.....	50
Figura 18. Notación <i>CamelCasin *</i> aplicada en el nombre de la función insertar estado financiero.....	50
Figura 19. Notación <i>CamelCasing*</i> aplicada en el nombre de la función adicionar concepto. ...	50
Figura 20. Notación <i>CamelCasing*</i> aplicada en los nombres de las variables.....	51
Figura 21. Comentarios de la clase NomEstadofModel.....	51
Figura 22. Comentarios de la función insertar estado financiero.....	52
Figura 23. Índice de personalización de estados financieros por versión del componente.....	56

ÍNDICE DE TABLAS

Tabla 1. Métrica Tamaño Operacional de Clase.....	18
---	----

Tabla 2. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.	18
Tabla 3. Relaciones entre clases.....	19
Tabla 4. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.....	20
Tabla 5. Descripción del proceso de negocio: Obtención de estados financieros.....	29
Tabla 6. Concepto estado financiero.....	30
Tabla 7. Concepto cuenta.....	31
Tabla 8. Especificación de requisitos: Adicionar grupo.....	34
Tabla 9. Descripción de la tabla nom_estadof.....	36
Tabla 10. Descripción de la tabla nom_conceptooperadores.....	37
Tabla 11. Descripción de la tabla dat_grupoconcepto.....	38
Tabla 12. Resultados de la aplicación de la métrica TOC para cada clase del componente.....	42
Tabla 13. Resultados de la aplicación de la métrica RC para cada clase del componente.....	44
Tabla 14. Opciones de configuración por componente.....	56

INTRODUCCIÓN

Como resultado del proceso de informatización que se ha estado llevando a cabo en diversos sectores del país, se ha propiciado, por parte de las entidades cubanas, el uso de herramientas informáticas para la gestión de sus procesos. En tal sentido, la UCI desempeña un rol fundamental, desde su surgimiento, en la producción de soluciones y servicios informáticos. En la universidad se encuentra el Centro de Informatización de la Gestión de Entidades (CEIGE), que como su nombre lo indica, se especializa en el desarrollo de sistemas para la gestión empresarial. Entre los sistemas en desarrollo en el centro, se encuentra XEDRO-ERP, un paquete de soluciones integrales de gestión para las entidades del país, que incorpora funcionalidades generales de los procesos y las particularidades de la economía cubana. Dentro de dicho sistema, en el subsistema Contabilidad se encuentra implementado el componente Estados financieros, el cual además de permitir la gestión de estados financieros, permite llevar a cabo la gestión de grupos y conceptos contables, entre otras funcionalidades.

A partir de la actualización del modelo económico cubano, el MFP, por las resoluciones 434/2012, 359/2013 y 369/2013, definió quince estados financieros para las entidades presupuestadas y patrimoniales. Por otra parte, la Oficina Nacional de Estadísticas e Información (ONEI), con el objetivo de consolidar la información a nivel nacional, estableció un mecanismo a partir del cual todos los sistemas contables del país deben entregar en un formato de archivo determinado, en este caso XML o MSET¹, sus estados financieros.

Actualmente XEDRO-ERP no brinda la posibilidad de obtener los estados financieros establecidos en las nuevas resoluciones del MFP debido a que el componente Estados financieros predefine las columnas que conformarán al reporte final. A su vez, dicho componente permite la inserción y eliminación de grupos pero no brinda la posibilidad de modificarlos, entorpeciendo el trabajo ya que obliga al usuario a no cometer ningún tipo de errores durante la gestión de los mismos. Es posible asociar conceptos a los grupos previamente insertados, al igual que es válido asociar cuentas a estos conceptos, pero no es posible asociar a los mismos, elementos de gasto ni otros conceptos. Tal situación dificulta la obtención de información, tanto de la situación y rendimiento financiero, como de los resultados operacionales del período y estado de inversiones y gastos de las empresas cubanas. Además,

¹ MSET: Archivo de texto que representa los datos en forma de tabla con las columnas separadas por un espacio.

el componente no permite exportar los estados financieros configurados en ninguno de los formatos establecido por la ONEI, obstaculizando el proceso de consolidación y toma de decisiones, por parte de este organismo. Paralelamente a estos aspectos existen errores de integridad referencial, debido a que el componente permite eliminar los estados financieros, pero no elimina las distintas asociaciones con otras tablas en la base de datos, incrementando el volumen de la misma y por tanto aumentando los tiempos de respuesta del sistema.

Dada la situación existente se define el siguiente **problema a resolver**: La personalización de los estados financieros durante el proceso de obtención de los mismos, se ve afectada por las limitadas opciones de configuración existentes en el componente Estados financieros del sistema XEDRO-ERP. Definiéndose como **objeto de estudio**: el proceso de obtención de estados financieros. Objeto de estudio que se encuentra delimitado por el siguiente **campo de acción**: el proceso de obtención de estados financieros en el subsistema Contabilidad de XEDRO-ERP.

Para dar solución al problema planteado se define entonces como **objetivo general**: Permitir la personalización de los estados financieros definidos por el MFP, mediante el desarrollo de la versión 1.1 del componente Estados financieros, para satisfacer las necesidades de información de las entidades cubanas. Y para dar cumplimiento al mismo se definieron los siguientes **objetivos específicos**:

1. Fundamentar la investigación mediante la elaboración del Marco Teórico para sustentar los conceptos, la propuesta de desarrollo del componente y las herramientas y tecnologías a utilizar.
2. Realizar el diseño de la nueva versión del componente Estados financieros, validando que el mismo responda a las necesidades de las entidades cubanas.
3. Implementar el componente Estados financieros, de manera que se ajuste al diseño propuesto.
4. Aplicar pruebas de caja negra y preexperimento para validar la propuesta de solución.

Planteando como **idea a defender**: El desarrollo de la versión 1.1 del componente Estados financieros, permitirá la personalización de los estados financieros definidos en las nuevas resoluciones del MFP.

Con el desarrollo del presente trabajo de diploma el **resultado esperado** es: obtener la versión 1.1 del componente Estados financieros en el subsistema Contabilidad de XEDRO-ERP, de manera tal que sea posible la personalización de los estados financieros definidos en las nuevas resoluciones del MFP.

El contenido del trabajo se divide en tres capítulos:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se mencionan aspectos referentes a los estados financieros y al marco legal bajo el que se sustenta este trabajo. Se lleva a cabo el estudio del estado del arte, a través del cual se realiza una investigación de varios sistemas que permiten la obtención de estados financieros. Se describen el modelo de desarrollo, los lenguajes, las tecnologías y las herramientas que se utilizarán durante la fase de desarrollo.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

En este capítulo se describe la propuesta de solución así como el proceso a informatizar. En consecuencia, se obtendrá un conjunto de artefactos definidos por el modelo de desarrollo seleccionado. Igualmente se detallan los distintos patrones utilizados para el diseño del componente y se muestran los resultados obtenidos tras la validación de dicho diseño a través de métricas.

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN

El punto de partida de este capítulo se basa en los resultados obtenidos del diseño del componente. Se describen aspectos relacionados con la implementación de la solución, como su integración con el sistema y los estándares de codificación empleados. Por último, se reflejan los resultados obtenidos de las pruebas de caja negra y el preexperimento realizado durante la validación de la propuesta de solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se hace un estudio de varios sistemas contables que permiten la gestión de estados financieros, donde se realiza un análisis de sus principales características en busca de elementos que puedan ser reutilizados en la propuesta de solución. Se mencionan varios aspectos relacionados con los estados financieros y se describen las tecnologías, los lenguajes y las herramientas a utilizar para la construcción del producto, así como el modelo de desarrollo a utilizar.

1.2. Estados financieros

A continuación se mencionan aspectos propios de los estados financieros como su definición, objetivos y los elementos que los conforman.

➤ Definición

Los estados financieros constituyen una representación estructurada de la situación financiera y del rendimiento financiero de la entidad (MFP, 2006).

➤ Objetivos

El objetivo de los estados financieros, con propósitos de información general, es suministrar información acerca de la situación financiera, del rendimiento financiero y de los flujos de efectivo de la entidad. Los estados financieros también muestran los resultados de la gestión realizada por los administradores con los recursos que se les han confiado (MFP, 2006).

➤ Elementos de los estados financieros

Los estados financieros reflejan los efectos financieros de las transacciones y otros sucesos, agrupándolos en grandes categorías, de acuerdo con sus características económicas (López, 2010). Estas grandes categorías son los elementos de los estados financieros que se muestran a continuación:

- **Activos:** Es un recurso controlado por la entidad como resultado de sucesos pasados, del que la misma espera obtener, en el futuro, beneficios económicos (López, 2010).

- **Pasivos:** Es una obligación actual de la entidad, surgida a raíz de sucesos pasados, al vencimiento de la cual, y para saldarla, la empresa espera desprenderse de recursos que incorporan beneficios económicos (López, 2010).
- **Patrimonio neto o capital contable:** Es la parte residual de los Activos de la entidad, una vez deducidos todos los Pasivos (López, 2010).
- **Ingresos:** Son los incrementos en los beneficios económicos, producidos a lo largo del periodo contable, en forma de entradas o incrementos de valor de los Activos, o bien como decrementos de las obligaciones, que dan como resultado aumentos del Patrimonio Neto o Capital Contable, y no están relacionados con las aportaciones al Patrimonio Neto o Capital Contable (López, 2010).
- **Gastos:** Son los decrementos en los beneficios económicos, producidos a lo largo del periodo contable, en forma de salidas o disminuciones del valor de los Activos, o bien como surgimiento de obligaciones, que dan como resultado decrementos en el Patrimonio Neto o Capital Contable, y no están relacionados con las distribuciones de estos beneficios (López, 2010).

1.3. Marco legal

Como consecuencia de la actualización del modelo económico cubano se establecieron una serie de resoluciones concernientes a la elaboración y obtención de estados financieros en las entidades cubanas. En dichas resoluciones se establecen proformas de uso obligatorio tanto para las entidades patrimoniales o presupuestadas, como para los órganos y organismos de la Administración Central del Estado.

➤ Resolución No. 434/2012

Establece las proformas e indicaciones metodológicas de los estados financieros de las unidades de registro del Sistema de Tesorería en los órganos y organismos de la Administración Central del Estado a través de la Norma Específica de Contabilidad Gubernamental No. 102 “Proformas de estados financieros de la Unidad de Registro de Tesorería OACE²”, con el propósito de homogeneizar la información y posibilitar los procesos de agregación de la información a diferentes niveles de dirección (MFP, 2012).

² Órgano de la Administración Central del Estado

Norma Específica de Contabilidad Gubernamental No. 102 - Proformas de Estados Financieros de la Unidad de Registro de Tesorería OACE (NECG - 102):

- Objetivo:

El objetivo de esta Norma es establecer las proformas para la presentación de los estados financieros de las unidades de registro de tesorería en los órganos y organismos de la Administración Central del Estado (MFP, 2012).

- Definición de proformas:

- a) Proforma EF-URT-OACE 5930 - Estado de Situación.
- b) Proforma EF-URT-OACE 5931 - Estado de Gastos del Presupuesto Central.
- c) Proforma EF-URT-OACE 5932 - Estado de Movimiento de Tesorería.

➤ Resolución 359/2013

Establece las proformas e indicaciones metodológicas de los estados financieros para la actividad presupuestada a través de la Norma Específica de Contabilidad de la Actividad Presupuestada No. 2 “Proformas de Estados Financieros” (MFP, 2012).

Norma Específica de Contabilidad de la Actividad Presupuestada No. 2 - “Proformas de Estados Financieros” (NECAP - 2):

- Objetivo:

El objetivo de esta norma es establecer las proformas para la presentación de los estados financieros de las unidades presupuestadas (MFP, 2012).

- Definición de proformas:

- a) Proforma EFUP 5910 - 03 Estado de Situación.
- b) Proforma EFUP 5911 - 03 Estado de Rendimiento Financiero.
- c) Proforma EFUP 5912 - 03 Estado de Inversiones.
- d) Proforma EFUP 5913 - 03 Estado de Disponibilidad.
- e) Proforma EFUP 5914 - 03 Estado de Gastos por Partidas - Actividad Presupuestada.
- f) Proforma EFUP 5915 - 03 Estado de Gastos por Partidas - Actividad Autofinanciada.

➤ Resolución 369/2013

Establece las proformas e indicaciones metodológicas de los estados financieros para la actividad empresarial, unidades presupuestadas de tratamiento especial y el sector cooperativo agropecuario y no agropecuario a través de la Norma Específica de Contabilidad No. 5 “Proformas de Estados Financieros para la actividad empresarial, unidades presupuestadas de tratamiento especial y el sector cooperativo agropecuario y no agropecuario” (MFP, 2012).

Norma Específica de Contabilidad No. 5 - “Proformas de Estados Financieros para la actividad empresarial, unidades presupuestadas de tratamiento especial y el sector cooperativo agropecuario y no agropecuario” (NEC - 5):

- Objetivo:

El objetivo de esta Norma es establecer las proformas para la presentación de los estados financieros de la actividad empresarial, unidades presupuestadas de tratamiento especial y el sector cooperativo agropecuario y no agropecuario (MFP, 2012).

- Definición de proformas:

- a) Proforma EFE 5920 - 02 Estado de Situación.
- b) Proforma EFE 5921 - 02 Estado de Rendimiento Financiero.
- c) Proforma EFE 5922 - 02 Estado de Rendimiento Financiero - Actividad de Seguro y Reaseguro.
- d) Proforma EFE 5923 - 02 Estado de Rendimiento Financiero para la Actividad Agropecuaria que no poseen otras actividades.
- e) Proforma EFE 5924 - 02 Estado de Gastos por Elementos.
- f) Proforma EFE 5925 - 02 Estado de Inversiones.

1.4. Sistemas informáticos

En general existe una gran diversidad de sistemas contables que se encuentran en explotación a nivel mundial y en Cuba. Se destacan entre ellos los ERP³ por su amplia variedad y alcance en la gestión de las diferentes áreas de los negocios, entre los cuales se seleccionaron Versat Sarasola y Rodas XXI por ser soluciones desarrolladas por empresas cubanas y ampliamente difundidas a nivel nacional. Además se seleccionaron para el estudio Microsoft Dynamics AX,

³ **ERP:** (por sus siglas en inglés, Enterprise Resource Planning) o sistemas de planificación de recursos empresariales.

Openbravo y SAP ERP por pertenecer a grandes industrias de software que llevan varios años trabajando en el campo de la informática.

1.4.1 Versat Sarasola

Sistema Económico Integrado, conformado por 12 subsistemas o módulos, diseñado para ser utilizado por el sector empresarial cubano. Desarrollado por un grupo de especialistas económicos e informáticos pertenecientes a la Empresa de Servicios Técnicos Industriales del Grupo Azucarero nacional (TEICO). Es un sistema concebido sobre una plataforma de trabajo cliente-servidor y de uso multiusuario (Cabrera Gonzáles, y otros, 2006).

Referente a la obtención de estados financieros el sistema trabaja en combinación con el paquete de Office. A través de Microsoft Excel permite configurar los estados financieros de las entidades, ofreciendo una solución práctica y sencilla. Al mismo tiempo permite al usuario acceder a la información contenida en los distintos módulos del sistema. En este caso el usuario puede acceder a las cuentas y elementos de gasto contenidos en el subsistema de contabilidad e insertarlos en el reporte a conveniencia. Además de hacer uso de las funciones que ofrece Excel para el cálculo, Versat incluye e integra sus propias funciones para llevar a cabo las operaciones definidas en los estados financieros (Cabrera Gonzáles, y otros, 2006). Debido a la flexibilidad de Microsoft Excel para la configuración de reportes, es posible configurar los estados financieros listados en las nuevas resoluciones del MFP.

Inconveniente: Versat Sarasola es una aplicación de escritorio implementada en Delphi y propietaria. Es compatible solamente con el sistema operativo Windows y utiliza como gestor de base de datos SQLServer de Microsoft. La misma, para la configuración de los distintos estados financieros, necesita Microsoft Office y no permite la obtención de éstos en ninguno de los formatos definidos por la ONEI.

1.4.2 Rodas XXI

Sistema Integral Económico Administrativo desarrollado por la empresa cubana CITMATEL (Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados) que posibilita automatizar el funcionamiento de cualquier empresa o unidad presupuestada. Es un sistema multiempresa que cuenta actualmente con ocho módulos (CITMATEL, 2002).

El sistema brinda la posibilidad de configurar a gusto del usuario informes generados en Excel, actualizables de forma automática y en cualquier momento. A través de los cuales pueden

obtenerse desde estados financieros, hasta informes de indicadores económico-financieros, y cualquier otro informe que pueda obtenerse a partir de la información contable contenida en el módulo contabilidad (CITMATEL, 2002).

Inconveniente: El sistema requiere como prestaciones mínimas de software en las estaciones de trabajo: Sistema Operativo (SO) Microsoft Windows XP, 2000 ó 2003 y en el servidor: SO Microsoft Windows XP, 2000 ó 2003 y Microsoft SQL Server 2000. Necesita, al igual que el sistema Versat, Microsoft Office para la configuración de sus estados financieros, y no permite obtener éstos en los formatos establecido por la ONEI.

1.4.3 OpenBravo

Es una aplicación de Código abierto de gestión empresarial del tipo ERP destinada a empresas de pequeño y mediano tamaño, multiplataforma y completamente web. Fue concebido con una arquitectura cliente/servidor y desarrollada siguiendo el modelo MVC⁴. Fue implementado en Java, con soporte para bases de datos PostgreSQL u Oracle y necesita para su ejecución Apache y Tomcat (Acosta Romero, y otros, 2007).

Openbravo ofrece dos herramientas personalizables para construir los informes correspondientes a los estados financieros: Balance sheet and P&L structure (balance general y estructura ganancia y perdida), y User Defined Accounting Report (Reportes contables definidos por el usuario). Ambas herramientas permiten la creación de estados financieros jerárquicamente. La diferencia entre estas dos herramientas es que con la primera, el usuario define las distintas categorías en las que se divide el estado financiero y por cada una de ellas selecciona los nodos, que contendrían las cuentas a mostrar en el reporte. Además, esta herramienta permite mostrar las cuentas asociadas a los nodos con el nivel de detalle que desee el usuario, es decir en el reporte final es posible visualizar la cuenta seleccionada o un desglose de las cuentas que componen a esta. En la segunda herramienta se deben seleccionar cada una de las filas a mostrar en el reporte. Por otra parte en Balance sheet and P&L structure el usuario puede obtener un reporte entre dos fechas determinadas, y exportarlo en formato pdf. En el caso de User Defined Accounting Report solo se pueden obtener reportes del mes, del trimestre o del año (OpenBravo, 2007).

⁴ **MVC**:(por sus siglas en inglés Model, View, Controller) o modelo, vista, controlador

Inconvenientes: El sistema no permite configurar los estados financieros según las nuevas resoluciones establecidas por el MFP, debido a que no cumple con el formato de presentación establecido en las mismas, es decir, no permite definir todos los elementos correspondientes a la cabecera y base del reporte, ni todas las columnas que lo conformarán. Además que no es posible exportar los reportes configurados a MSET o XML.

1.4.4 SAP ERP

Sistema desarrollado por SAP AG compañía alemana de software para aplicaciones de negocios. SAP está compuesto por una serie de áreas funcionales o módulos que responden de forma completa y en tiempo real a los procesos operativos de las organizaciones (SAP AG, 1999).

A través de la herramienta Balance sheet (balance general o estado de situación) permite la creación del balance general y el estado de pérdida y ganancia para un período y año fiscal definido por el usuario. Esta herramienta utiliza FSV⁵ definidas durante la configuración. Estas FSV reflejan diferentes vistas del balance general, por ejemplo, el formato U.S. GAAP, el formato British GAAP, etc. El balance general y el estado de ganancias y pérdida pueden verse en comparación con periodos anteriores o valores planificados. Para ejecutar un reporte es necesario especificar el FSV, el año fiscal, el periodo de reporte y el nivel de resumen seleccionado. Sin embargo, es práctico especificar el catálogo de cuentas, el código de la compañía, y el área de negocio (SAP AG, 1999).

Inconveniente: Es software privativo, sumamente costoso y requiere Windows NT, Microsoft Internet Explorer 5.x o superior, lector de pantalla y software de aumento de imagen en pantalla para su funcionamiento correcto. Solamente permite la configuración del Balance General y el Estado de Pérdidas y Ganancias, pero no según el formato establecido en las resoluciones del MFP, además no permite exportar los reportes en los formatos establecidos por la ONEI.

1.4.5 Microsoft Dynamics AX

Microsoft Dynamics AX es una solución integrada, creada por la empresa Microsoft que permite interrelacionar datos entre cada una de las áreas de la empresa. Es una solución empresarial para empresas globales que soportan procesos de negocio operacionales y específicos de la

⁵ **FSV** (por sus siglas en inglés, Financial Statement Versions) o versiones de estados financieros

industria, junto con una funcionalidad integral y centralizada de ERP para la administración financiera y de recursos humanos (Microsoft, 2014).

Concerniente a los estados financieros, Microsoft Dynamics AX permite un diseño personalizado tanto de las filas como de las columnas del estado financiero en cuestión. No solo permite configurar las filas según las necesidades del usuario sino que es posible importar plantillas de definición de filas ya predefinidas para la entidad, el país o la región al que pertenezca la organización, así como crear plantillas del estado financiero y exportarlas para su uso posterior. Permite diseñar las filas de acuerdo al tipo especificado por el usuario, por ejemplo, filas de tipo elemento que correspondería, en este caso, a las distintas cuentas a mostrar en el reporte, o de tipo grupo que corresponden a las filas cabeceras para un grupo de cuentas o como subtotal. Posterior a la configuración del estado financiero, es posible generar e imprimir un reporte de estados financieros y exportar los resultados en varios tipos de formatos (Microsoft, 2014).

Inconveniente: El sistema sólo se ejecuta sobre la plataforma de Windows y es software propietario, por lo que priva el acceso al código fuente del programa y niega el derecho de poder copiarlo y modificarlo a las necesidades requeridas. No permite configurar toda la información concerniente a la cabecera y base del reporte según lo establecido en las resoluciones del MFP y no permite exportar a MSET o XML los estados financieros configurados.

1.5. Modelo de desarrollo

Para la construcción del componente, el modelo de desarrollo de software a emplear será el propuesto por la Subdirección de Producción de CEIGE. En el cual se establecen las fases que conformaran el ciclo de vida de los proyectos del centro, así como los distintos artefactos a generar durante cada una de ellas.

1.5.1 Disciplina Modelado del negocio

Destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito (CEIGE, 2013).

Los artefactos generados en esta disciplina son los siguientes:

- Descripción de proceso de negocio.

- Modelo conceptual.

Notación para Modelado de Procesos de Negocio

El modelado de procesos de negocio demanda la representación de los elementos que intervienen en los procesos. Para ello se propone emplear la Notación para Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés). BPMN es una notación gráfica que describe los pasos en un proceso de negocio. La notación fue específicamente diseñada para coordinar la secuencia de procesos y mensajes que fluyen entre los participantes del proceso en un conjunto relacionado de actividades (OMG, 1997).

Patrones de flujo de trabajo candidatos:

- **Sequence** (Secuencia): es definido como una serie ordenada de actividades o enrutamiento secuencial (White, 2014).
- **Parallel Split** (División paralela): es definido como un mecanismo que permite la realización concurrente de actividades, en vez de en serie (White, 2014).
- **Synchronization** (Sincronización): combina dos caminos que fueron generados por una división paralela (White, 2014).
- **Exclusive Choice** (Selección exclusiva): es definido como una ubicación en el proceso donde el flujo se divide en dos o más caminos exclusivos. Donde solo uno de los caminos alternativos se puede elegir para que continúe el proceso (White, 2014).
- **Simple Merge** (Combinación sencilla): es definido como una ubicación en el proceso donde un conjunto de caminos alternativos se unen en uno solo (White, 2014).
- **Multiple Choice** (Selección múltiple): este patrón defiere del patrón selección exclusiva en que éste permite la selección de uno a todos los caminos alternativos al mismo tiempo (White, 2014).

1.5.2 Disciplina Requisitos

Se desarrolla un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del sistema. Se especifican los requisitos funcionales y no funcionales (CEIGE, 2013).

Los artefactos generados en esta disciplina son los siguientes:

- Descripción de requisitos de software.

Elicitación de requisitos

Esta actividad se realiza con el fin de conocer el dominio del problema e identificar las necesidades reales de clientes y usuarios, para esto son consultadas distintas fuentes de información como clientes, usuarios y expertos en el dominio (Toro, 2000).

Técnicas de elicitación candidatas:

- **Entrevistas:** son utilizadas para obtener información sobre el negocio o sistema a automatizar y a partir de éstos es que se definen los requisitos (Toro, 2000). Para realizar la entrevista se debe estudiar el dominio del problema, con el objetivo de conocer los conceptos fundamentales para el cliente y que este vea que se entienden sus planteamientos. Además, se deben seleccionar las personas a entrevistar, definir el objetivo de las entrevistas y planificarlas teniendo en cuenta adecuarse al horario de los entrevistados (Escalona, 2002).
- **Tormenta de ideas:** se basa en una reunión de pocas personas, por lo general no más de diez, donde los participantes exponen sus ideas de forma libre y espontánea. Se utiliza con el objetivo de generar ideas que brinden una visión a grandes rasgos de las necesidades del sistema, pero generalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros (Escalona, 2002).
- **Sketches y Storyboards:** comúnmente es usada por los diseñadores gráficos de aplicaciones en el entorno web. La misma consiste en representar sobre papel de forma muy bosquejada las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard) (Escalona, 2002).
- **Observación y Análisis de Tareas:** consiste en que un observador estudia a los futuros usuarios en su ambiente de trabajo. Apunta todo aquello que es susceptible de mejora, para luego generar una serie de requisitos tentativos. El uso de esta técnica implica la visita del analista a las áreas donde se realizan los procesos (Moreno, 2009).
- **Arqueología de documentos:** se fundamenta en determinar posibles requisitos inspeccionando la documentación utilizada por la empresa; por ejemplo, manuales de

procedimientos, publicaciones, en investigaciones relacionadas con el proceso. Esta técnica sirve como complemento de las demás técnicas, a través de ella se obtiene información que de otra manera sería muy difícil de conseguir (Moreno, 2009).

- **Sistemas Existentes:** consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el proceso que se intenta informatizar. Pueden ser analizadas las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada. Esto puede ser útil para descubrir información importante a tener en cuenta, información que tal vez el cliente haya fallado en comunicar. También es útil analizar las distintas salidas que los sistemas producen (listados, consultas, reportes), porque siempre pueden surgir nuevas ideas sobre la base de éstas. Esta técnica requiere cierto grado de investigación, así como contactos con otros profesionales que hayan desarrollado sistemas parecidos (Moreno, 2009).

Validación de requisitos

Determina si los requisitos definidos son los que realmente quiere el cliente. Examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencia, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto (Pressman, 2005).

Técnicas candidatas para la validación de los requisitos:

- **Revisiones:** es la fórmula más empleada para validación. Un grupo de personas (incluyendo usuarios) se ocupan de revisar el documento de requisitos y los modelos. Consta de tres fases: búsqueda de problemas, reunión y acuerdos, (Moreno, 2009).
- **Prototipos:** permite al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario, permitiendo descubrir con rapidez si el usuario se encuentra satisfecho, o no, con los requisitos, (Moreno, 2009).
- **Auditorías:** la revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada, (Moreno, 2009).

Patrones de requisitos candidatos:

Según (Moreno, 2009), existen patrones de casos de usos que pueden ser aplicados a los requisitos. Entre ellos se encuentran:

- **El nombre revela la intención:** este patrón nombra los casos de uso utilizando un verbo activo o frase que represente la meta del actor primario (Larman, 2001).
- **Preciso y legible:** este patrón tiene como objetivo garantizar que cada caso de uso sea escrito lo suficientemente legible a fin de que los clientes los lean, los evalúen y precisen lo suficiente a fin de que los implementadores entiendan qué están construyendo (Larman, 2001).

1.5.3 Disciplina Análisis y diseño

Durante esta disciplina es modelado el sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima fase. Los artefactos generados en esta etapa son más formales y específicos de una implementación (CEIGE, 2013).

Los artefactos generados en esta disciplina son los siguientes:

- Modelo de datos.
- Diagrama de clase del diseño.
- Diagrama de componentes.

Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces (Gamma, y otros, 1995).

Patrones GRASP: Patrones de Software para la Asignación General de Responsabilidad. En este caso las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. GRASP destaca 5 patrones principales:

- **Experto:** asigna responsabilidades a la clase que se encarga de gestionar la información referente a un objeto determinado y es capaz de cumplir con la actividad asignada conservando el encapsulamiento y promoviendo clases sencillas, fáciles de comprender y mantener (Larman, 1999).

- **Creador:** tiene la responsabilidad de asignar la creación de una clase a otra. Esta asignación solo puede realizarse cuando agrega o contiene una instancia de ella, contiene instancias de sus objetos y contiene los datos de inicialización que serán transmitidos a la nueva clase (Larman, 1999).
- **Controlador:** gestiona los eventos de entrada generado por actores externos, asignando responsabilidades a otras clases de manera independiente que permitan el bajo acoplamiento y la alta cohesión (Larman, 1999).
- **Bajo acoplamiento:** es la medida en que cada una de las clases realiza actividades independientes, además de poseer un conocimiento de las actividades que realizan las otras clases del sistema permitiendo la reutilización (Larman, 1999).
- **Alta cohesión:** la cohesión es la relación que existe entre las clases y la medida en que éstas realizan labores únicas dentro del sistema, pero que están estrechamente relacionadas entre sí (Larman, 1999).

Patrones GOF: los patrones Gang Of Four (GOF), son patrones de diseño. Están divididos fundamentalmente en tres grandes grupos: estructurales, creacionales y de comportamiento.

- **Facade** (Fachada): patrón estructural que provee una interfaz unificada para un conjunto de interfaces en un subsistema. Fachada define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar (Gamma, y otros, 1995).
- **Chain of responsibility** (Cadena de Responsabilidad): patrón de comportamiento que se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición. Encadena los objetos receptores y pasa la solicitud a la cadena hasta que un objeto la maneje (Gamma, y otros, 1995).
- **Singleton:** es un patrón creacional que garantiza que exista una instancia única para una clase y proporciona un punto de acceso global a ella (Gamma, y otros, 1995).

Patrón arquitectónico

Patrón Modelo Vista Controlador (MVC): es un patrón arquitectónico usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas, donde se requiere una mejor separación de los conceptos para que el desarrollo esté estructurado de una mejor manera (Gómez Baryolo, 2012).

- **Modelo:** esta capa es la representación específica de la información con la cual el sistema opera. Se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas.
- **Vista:** esta capa presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** éste responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y notifica a la vista.

Métricas para la validación del diseño

Las métricas de software son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Permiten averiguar cuán bien están definidas las clases y el sistema, lo cual tiene un impacto directo en el mantenimiento del mismo, tanto por la comprensión de lo desarrollado como por la dificultad de modificarlo con éxito. Posibilitan al ingeniero evaluar el software al inicio del proceso, haciendo cambios que reducirán la complejidad y mejorarán la viabilidad, a largo plazo, del producto final. También permiten descubrir y corregir problemas potenciales antes de que se conviertan en defectos catastróficos. Estas métricas tienen como propósito entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo llevado a cabo al nivel del proyecto (Pressman, 2005).

Métrica Tamaño Operacional de las Clases (TOC): La métrica TOC está dada por la cantidad de funcionalidades contenidas en las clases, a partir de las cuales se determina la afectación que ejerce en el diseño. La misma comprende los siguientes atributos de calidad:

- **Responsabilidad:** responsabilidad que posee una clase en un marco conceptual correspondiente al modelado de la solución propuesta (García Ramírez, y otros, 2013).
- **Complejidad de implementación:** grado de dificultad que tiene implementar un diseño de clases determinado (García Ramírez, y otros, 2013).
- **Reutilización:** significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software (García Ramírez, y otros, 2013).

A continuación se muestra una serie de tablas encaminadas a un mejor entendimiento de la utilización de esta métrica.

Atributo de Calidad	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento en la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Tabla 1. Métrica Tamaño Operacional de Clase.

Atributo	Categoría	Criterio
Responsabilidad	Baja	(\leq Promedio)
	Media	(Entre Promedio y $2 \times$ Promedio)
	Alta	($> 2 \times$ Promedio)
Complejidad de implementación	Baja	(\leq Promedio)
	Media	(Entre Promedio y $2 \times$ Promedio)
	Alta	($> 2 \times$ Promedio)
Reutilización	Baja	($> 2 \times$ Promedio)
	Media	(Entre Promedio y $2 \times$ Promedio)
	Alta	(\leq Promedio)

Tabla 2. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.

Promedio: sumatoria de la cantidad de operaciones por clase entre la cantidad de clases.

Métrica de Relaciones entre Clases (RC): la métrica RC está dada por la cantidad de relaciones existentes entre las clases contenidas en el diseño, a partir de las cuales se determina la afectación que éstas ejercen dentro de la eficiencia del sistema. Los indicadores medidos por esta métrica son los siguientes:

- **Acoplamiento:** dependencia o interconexión de una clase o estructura de clase respecto a otras (García Ramírez, y otros, 2013).
- **Cantidad de pruebas:** número o grado de esfuerzo necesario para realizar las pruebas de calidad al producto (componente) diseñado (García Ramírez, y otros, 2013).

- **Complejidad del mantenimiento:** nivel de esfuerzo necesario para sustentar, mejorar o corregir el diseño de software propuesto. Puede influir significativamente en los costes y la planificación del proyecto (García Ramírez, y otros, 2013).
- **Reutilización:** significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software (García Ramírez, y otros, 2013).

Esta métrica evalúa los siguientes atributos de calidad:

Atributo de Calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento en la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución del grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 3. Relaciones entre clases.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	(>2)
Complejidad de mantenimiento	Baja	(<=Promedio)
	Media	(Entre Promedio y 2*Promedio)
	Alta	(>2* Promedio)
Reutilización	Baja	(>2* Promedio)
	Media	(Entre Promedio y 2* Promedio)

	Alta	(\leq Promedio)
Cantidad de pruebas	Baja	(\leq Promedio)
	Media	(Entre Promedio y 2^* Promedio)
	Alta	($>2^*$ Promedio)

Tabla 4. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.

Promedio: sumatoria de la cantidad de relaciones de uso por clase entre la cantidad de clases.

1.5.4 Disciplina Implementación

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes de software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes (CEIGE, 2013).

Los artefactos generados en esta disciplina son los siguientes:

- Ficheros de código.
- Prototipos de interfaz funcionales.

1.5.5 Disciplina Pruebas internas

Durante esta fase se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple (CEIGE, 2013).

Los artefactos generados en esta disciplina son los siguientes:

- Diseño de casos de pruebas.
- Documento de no conformidades.

Los métodos de pruebas independientemente del nivel en que se enmarque la prueba, ayudan a encontrar buenos conjuntos de casos de prueba para detectar diferentes tipos de errores. Existen dos enfoques alternativos según indica (Ramírez, 2012), descritos como: pruebas de Caja Blanca y Caja Negra. En la primera mencionada se cuenta con el código, pues se conoce

el mismo y se tratan de ejecutar cada uno de sus elementos. Por otra parte, en las pruebas de caja negra se cuenta con la interfaz y se maneja con cada uno de los elementos que la componen para llevar a cabo la validación.

Técnicas para realizar pruebas de caja negra:

- **Técnica de la Partición de Equivalencia:** se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (González, y otros, 2010).
- **Técnica del Análisis de Valores Límites:** es una técnica de diseño de casos de prueba que completa a la técnica anterior, llevando a la elección de casos de prueba en los extremos de la clase de equivalencia (González, y otros, 2010).

Niveles de prueba

A la hora de evaluar dinámicamente un sistema, se debe comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Las pruebas se aplican en distintos niveles de trabajo, según lo indicado por (González, y otros, 2010), dentro de éstos se distinguen:

- **Pruebas de Integración:** los componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.
- **Pruebas del Sistema:** son usualmente conducidas para asegurar que todos los módulos trabajan como sistema sin error. Es similar a la prueba de integración pero con un alcance mucho más amplio.
- **Pruebas de Aceptación:** son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.
- **Pruebas de Unidad:** conocidas también como unitarias son pruebas individuales a las unidades separadas de un sistema de software.

1.6. Lenguajes, tecnologías y herramientas

En el presente epígrafe se hace una descripción de los principales lenguajes a utilizar, así como las principales herramientas para el diseño e implementación del producto.

1.6.1 Lenguajes de programación

- **HTML5** (Lenguaje de Marcas de Hipertexto) es un conjunto de etiquetas, complementadas por extensiones que permiten dar formato a un archivo para que pueda ser visualizado en forma de página web (HTML.net, 2012).
- **XML 1.0** (Lenguaje de Etiquetado Extensible) es un formato que permite la lectura de datos a través de diferentes aplicaciones y además es utilizado para estructurar, almacenar e intercambiar información (Colectivo editorial W3C Consortium, 2008).
- **JavaScript 1.6** es un lenguaje de programación multiplataforma, interpretado y orientado a objetos lo que permite a los desarrolladores añadir y crear interactividad en el desarrollo y diseño de sitios web así como la validación de datos (Mozilla Developer Network, 2011)
- **PHP 5.3.10** (Preprocesador de hipertexto) es un lenguaje de programación que se ejecuta en el servidor, gratuito e independiente de la plataforma de desarrollo, por lo que puede ser utilizado en cualquier marco de trabajo o IDE de desarrollo que lo soporte (The PHP Group, 2011).
- **PL/pgSQL** es un lenguaje de procedimientos almacenados para Postgres que está provisto por el gestor de base de datos PostgreSQL, y permite además ejecutar comandos SQL⁶ mediante un lenguaje de sentencias imperativas y uso de funciones (Martínez, 2009).
- **DQL** es el lenguaje de consulta de Doctrine que proporciona capacidades de consulta sobre los objetos del modelo (Doctrine-Project.org, 2010).

1.6.2 Tecnologías

- **JSON** (Notación de Objetos de JavaScript) es un formato de intercambio de datos que está constituido por una colección de pares de nombre/valor. Actualmente se ha convertido en un estándar en el desarrollo de aplicaciones web donde en ocasiones sustituye a XML para permitir la integración de servicios en el navegador del usuario (Colectivo de JSON, 2011).

⁶ **SQL** (por sus siglas en inglés, Structured Query Language) o Lenguaje de consulta estructurado.

- **AJAX** (JavaScript y XML asíncronos) es una técnica de desarrollo web para crear aplicaciones interactivas, la cual se ejecuta en el navegador del usuario, y mantiene comunicación asíncrona con el servidor, para de esta forma poder realizar cambios sobre la misma página sin necesidad de recargarla (ProgramacionWeb.net, 2005).

1.6.3 Lenguaje de modelado

UML (Lenguaje Unificado de Modelado) “es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software “(Jacobson, 2000).

1.6.4 Herramienta de modelado

Visual Paradigm para UML 8.0 es una herramienta que emplea UML y permite representar el ciclo de vida completo del desarrollo de software. Posibilita la representación de varios tipos de diagramas, entre otras opciones. Dentro de sus características fundamentales se encuentran que soporta la opción de usar BPMN (Visual Paradigm, 2007).

1.6.5 Herramientas de programación

- **NetBeans IDE 7.2** es un producto creado bajo licencias de software libre, es gratuito y no tiene restricciones de uso. Es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris, de código abierto, escrito completamente en Java. Es una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente aplicaciones, utilizando un número importante de módulos para extenderlo a otros lenguajes como PHP (Zend y Symfony), JavaScript, Ajax, Groovy y Grails, y C/C++. Entre sus características se encuentra la de poseer un sistema de proyectos basado en Ant⁷, control de versiones y refactoring. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación (Oracle Corporation, 2012).
- **PostgreSQL 8.3** es un sistema de gestión de bases de datos, objeto-relacional, basado en el proyecto POSTGRES de la Universidad de Berkeley. Publicado bajo la licencia BSD según plantea (The FreeBSD Foundation, 2012). Tiene amplio soporte por una comunidad mundial. Presenta características orientadas a objetos y herencia entre tablas, además de

⁷ **Ant:** Es una herramienta Open-Source utilizada en la compilación y creación de programas Java, Está escrita en XML y Java.

tener integridad referencial y permitir replicación asíncrona y acceso encriptado vía SSL⁸ (Martínez, 2009).

- **PgAdmin III** es una aplicación gráfica para administrar las bases de datos PostgreSQL, con licencia Open Source. Se puede usar en varios sistemas operativos como Linux, Solaris, Mac OS X y Windows. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados (Martínez, 2009).

1.6.6 Herramienta para el control de versiones

Subversion 1.6.6 es un controlador de versiones empleado en la administración de archivos utilizados en el desarrollo de software que permite seguir los cambios de los archivos y directorios a través de copias y renombrados. Presenta integración con el servidor web Apache y existen varias interfaces de Subversion, ya sean programas individuales o interfaces que lo integran en entornos de desarrollo (CENDITEL, 2011).

1.6.7 Servidor web

Apache 2.0 según (Observatorio Tecnológico, 2008) es un servidor web de software libre desarrollado por la Apache Software Foundation cuyo objetivo es servir o suministrar páginas web a los clientes web o navegadores que las solicitan. Es una tecnología gratuita y de código abierto, lo cual le brinda transparencia al software de manera que se pueda conocer lo que se está instalando como servidor. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Es altamente configurable en la creación y gestión de logs.

1.6.8 Navegador web

Mozilla Firefox 2.0 o superior es un navegador libre y de código abierto. Es usado para visualizar páginas web. Incluye corrector ortográfico, búsqueda progresiva y marcadores dinámicos. Además se pueden añadir funciones a través de complementos desarrollados por terceros. Es multiplataforma, realiza la navegación por pestañas, presenta compatibilidad para

⁸ **Secure Sockets Layer:** protocolo criptográfico que proporciona comunicación segura en Internet.

múltiples extensiones. Utiliza el sistema SSL para proteger la comunicación con los servidores web, utilizando fuerte criptografía cuando se utiliza el protocolo HTTPS⁹. (Mozilla Firefox, 2009).

1.6.9 Marco de trabajo

El marco de trabajo **Sauxe 1.5.4** posee un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo, (Baryolo, et al., 2008). Sauxe utiliza **ExtJS 2.2** para implementar la capa de presentación, se apoya en ZendExt, una extensión de **Zend Framework 1.9.7** para el desarrollo de la lógica del negocio, y para la gestión de los datos que maneja **Doctrine 1.2.2**. Utiliza como estilo arquitectónico el Modelo-Vista-Controlador. También tiene como propósito insertar la inversión de controles para la integración de servicios (Baryolo,2012). Además como herramienta de seguridad emplea el sistema **Acaxia** (Sistema de Gestión Integral de Seguridad), el cual está desarrollado sobre software libre e incorpora procesos tan importantes como la administración de conexiones y de perfiles.

1.7. Conclusiones parciales

Una vez finalizado el presente capítulo se arribaron a las siguientes conclusiones parciales:

- Los sistemas contables estudiados no se consideran una alternativa aplicable debido a que muchos de éstos son dependientes de tecnologías privativas o no se ajustan a las características y necesidades de información de las entidades cubanas.
- El modelo de desarrollo, las tecnologías y herramientas definidas, posibilitarán desarrollar una solución que permitirá la obtención de estados financieros según las nuevas resoluciones del MFP.

⁹ **HTTPS:** (Hypertext Transfer Protocol Secure) Protocolo de transferencia de hipertexto seguro.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1. Introducción

En el presente capítulo se describen las características bajo las cuales operará la nueva versión del componente a implementar, en correspondencia con el modelo de desarrollo seleccionado. Se obtendrán un conjunto de artefactos correspondientes a la modelación de los procesos del negocio, como el mapa de procesos del negocio y el modelo conceptual. Además, comprenderá aspectos relacionados con la disciplina de requisitos. Por último se establecerá el diseño de la solución con su respectiva validación.

2.2. Propuesta del sistema

El sistema deberá ser capaz de realizar el proceso de obtención de estados financieros. Para ello permitirá que el usuario pueda autenticarse y posteriormente acceder a las funcionalidades que le están permitidas según su rol. El componente posibilitará la configuración de las columnas del reporte, así como la creación de grupos de manera jerárquica. A cada grupo será posible asociar conceptos desglosados por cuentas, por elementos de gasto o por otros conceptos registrados en el sistema. Posterior a la configuración del reporte, se le brindará la posibilidad al usuario de visualizarlo en modo vista previa y si así lo desea podrá exportar el mismo a formato MSET. La selección del formato, estuvo condicionada por el hecho de que los archivos XML precisaban cierta información que en el sistema XEDRO-ERP no se trataba y los ficheros MSET resultaban mucho más sencillos de generar.

El componente ofrecerá servicios a los componentes Análisis financiero y Agregación, que como parte de sus funcionalidades necesitan los estados financieros configurados para llevar a cabo sus respectivos procesos.

2.3. Modelación del proceso de negocio

El modelado del negocio tiene como objetivo describir los procesos existentes u observados, con el propósito de comprenderlos. Los modelos de procesos deben estar expresados de forma que permitan que éstos sean analizados, automatizados o mejorados (Layola, 2006). En tal sentido en el presente epígrafe se muestran la descripción del proceso de negocio, el diagrama de proceso de negocio y el modelo conceptual.

2.3.1. Diagrama de proceso de negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente que se llevan a cabo para lograr como resultado un negocio definido. Los procesos describen cómo es realizado el trabajo en la organización y se caracterizan por ser observables, medibles, mejorables y repetitivos. A continuación se muestra cómo se realiza el proceso Obtención de estados financieros, ver Figura 1.

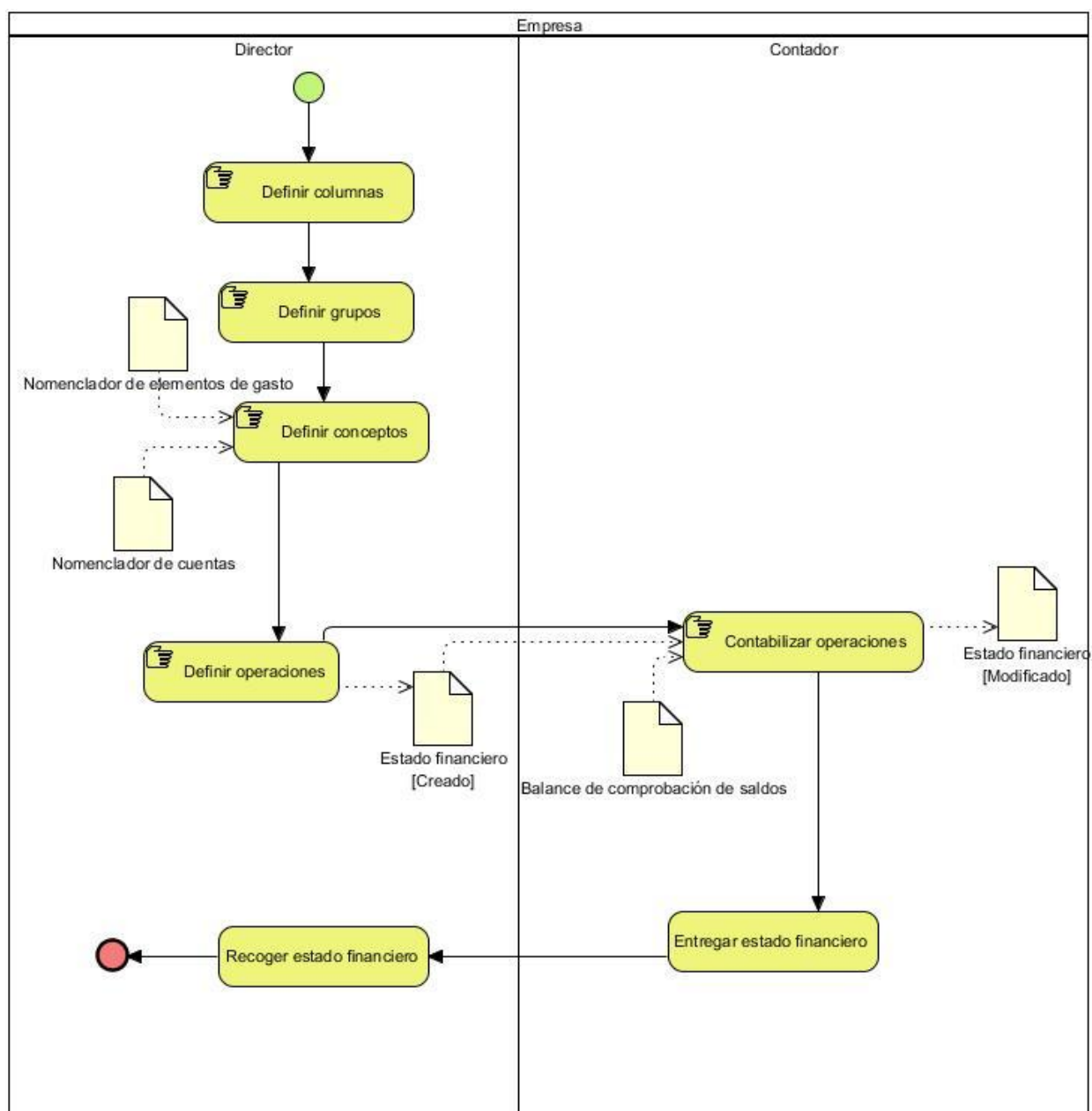


Figura 1. Diagrama Proceso de Negocio: Obtención de estados financieros.

2.3.2. Descripción del proceso de negocio

A continuación se muestra la descripción del proceso de negocio, obtención de estados financieros, donde se explica detalladamente cómo se ejecuta el mismo, ver Tabla 5.

Objetivo	Obtener los estados financieros de una organización
Evento(s) que lo genera(n)	N/A
Pre condiciones	Debe existir una entidad.
Marco legal	Resolución No. 434-2012 Ministerio de Finanzas y Precios. Resolución No. 359-2013 Ministerio de Finanzas y Precios. Resolución No. 369-2013 Ministerio de Finanzas y Precios.
Clientes internos	N/A
Clientes externos	Órganos de Finanzas.
Entradas	Nomenclador de cuentas, nomenclador de elementos del gasto, balance de comprobación de saldos.
Flujo de eventos	
Flujo básico obtener estados financieros	
	Definir columnas: Se definen las columnas que van a formar parte del Estado Financiero a obtener de acuerdo a las resoluciones del MFP.
	Definir conceptos contables: Se definen los conceptos que van a formar parte del reporte.
	Definir operaciones: Se definen las operaciones de cálculo a llevar a cabo entre los conceptos definidos.
	Contabilizar operaciones: Se llevan a cabo las operaciones de cálculo definidas entre los saldos de los conceptos.
	Entregar estado financiero: Se entregan a la máxima dirección de la empresa el estado financiero obtenido.
	Recoger estado financiero: La dirección de la empresa recoge el estado financiero obtenido.
Pos-condiciones	
	Se obtienen los estados financieros.
Salidas	
	Estado financiero
Flujos paralelos	
	N/A
Pos-condiciones	
	N/A
Salidas	

N/A
Flujos alternos
N/A
Pos-condiciones
N/A
Salidas
N/A
Asuntos pendientes
N/A

Tabla 5. Descripción del proceso de negocio: Obtención de estados financieros.

2.3.3. Modelo Conceptual

El modelo conceptual define los conceptos significativos en el dominio del problema. Muestra los distintos conceptos junto a sus atributos y asociaciones. A continuación se muestra la representación conceptual del proceso obtención de estados financieros, ver Figura 2.

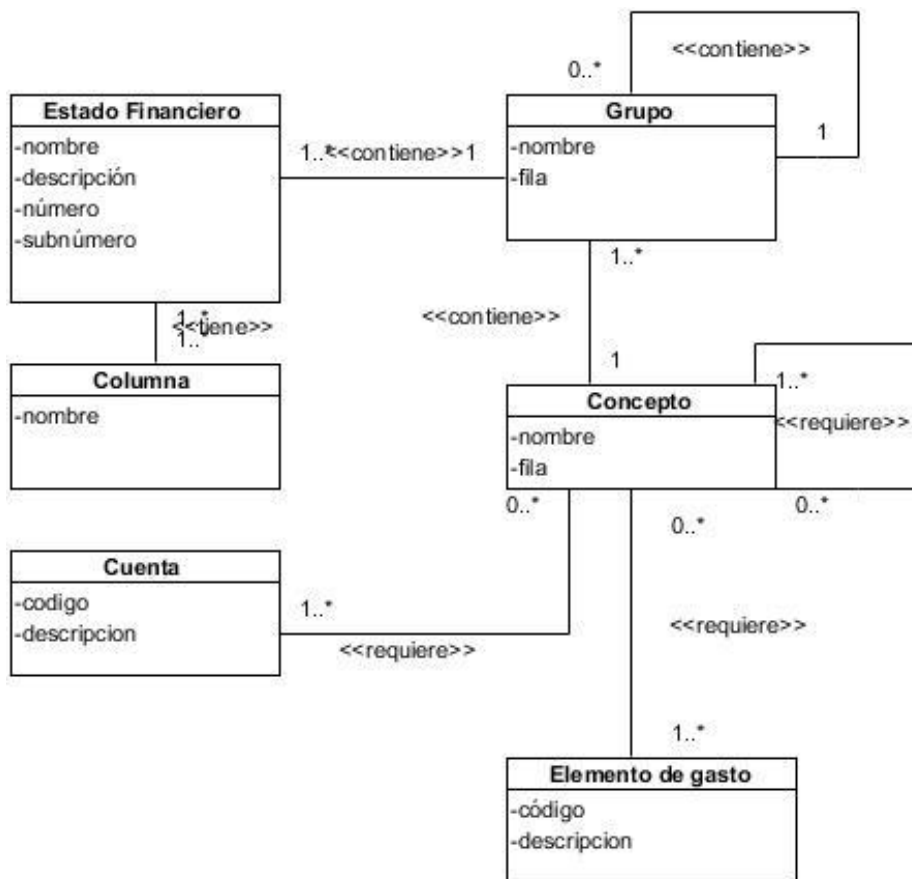


Figura 2. Mapa conceptual.

A continuación se muestra la descripción concerniente a los conceptos estado financiero (ver Tabla 6) y cuenta (ver Tabla 7). La descripción del resto de los conceptos se encuentra en el expediente de proyecto del centro CEIGE, además de los restantes artefactos entregados del componente.

Descripción						
Representa informes que utilizan las instituciones para reportar la situación económica y financiera y los cambios que experimenta la misma a una fecha o período determinado.						
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
nombre	Nombre del estado financiero que se desea emitir.	Alfanumérico	No	Sí	Clases válidas Cualquier combinación de letras y números.	Clases no válidas No admite caracteres especiales.
descripción	Descripción del estado financiero.	Alfanumérico	Sí	Sí	Cualquier combinación de letras y números.	No admite caracteres especiales.
número	Número del modelo al que pertenece el estado financiero.	Numérico	No	Sí	Solo números.	No admite la combinación de letras y caracteres especiales.
sub-número	Sub-número del modelo al que pertenece el estado financiero.	Numérico	Sí	No	Solo números.	No admite la combinación de letras y caracteres especiales.

Tabla 6. Concepto estado financiero.

Descripción						
Representa las cuentas o subcuentas que son afectadas por las operaciones contables que se llevan a cabo en la Entidad.						
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
código	Define un número que representa la cuenta o subcuenta.	Numérico	No	Sí	Clases válidas Cualquier combinación de números que esté dentro del rango de	Clases no válidas Combinación de letras o caracteres especiales.

descripción	Nombre que recibe la cuenta.	Alfanumérico	dígitos configurada.		Cualquier combinación de letras y números.	No admite caracteres especiales.
			No	Sí		

Tabla 7. Concepto cuenta.

2.4. Definición de los requisitos de software

El propósito de la definición de requisitos es especificar las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento entre el equipo de desarrollo y el cliente, y especificando las necesidades reales de forma que satisfaga sus expectativas (Colectivo de autores: UCID, 2009).

2.4.1. Requisitos funcionales:

Los requisitos funcionales representan el comportamiento que tendrá el sistema. Describen las funcionalidades que debe cumplir o que se espera que éste provea. Deben ser lo más completos, claros y concisos posible. Además pueden definirse a partir de reglas del negocio o la propia interacción de los usuarios (S. Pressman, 2001).

Para la obtención de los requisitos funcionales del componente Estados financieros se aplicaron en conjunto las técnicas de **tormentas de ideas** y **entrevistas** con el objetivo de obtener una visión general de las funcionalidades que debía brindar el componente. A través de la técnica **Sketches y Storyboards** se confeccionaron distintos bocetos en los que se modelaban las distintas interfaces y el funcionamiento básico que debería tener el componente.

A continuación se muestran los requisitos funcionales que debe cumplir la propuesta de solución:

RF1 Gestionar estado financiero

RF1.1. Adicionar estado financiero.

RF1.2. Modificar estado financiero.

RF1.3. Eliminar estado financiero.

RF1.4 Listar estado financiero.

RF1.5 Generar vista previa del estado financiero.

RF1.6 Exportar estado financiero.

RF2 Gestionar grupo.

RF2.1. Adicionar grupo.

RF2.2. Modificar grupo.

RF2.3. Eliminar grupo.

RF2.4 Listar grupo.

RF3 Gestionar concepto.

RF3.1. Adicionar concepto.

RF3.2. Modificar concepto.

RF3.3. Eliminar concepto.

RF3.4 Listar concepto.

RF4 Gestionar cuenta a concepto.

RF4.1. Adicionar cuenta a concepto.

RF4.2. Eliminar cuenta a concepto.

RF4.3. Listar cuenta a concepto.

RF5 Gestionar elemento de gasto a concepto.

RF5.1. Adicionar elemento de gasto a concepto.

RF5.2. Eliminar elemento de gasto a concepto.

RF5.3. Listar elemento de gasto a concepto.

RF6 Gestionar columna.

RF6.1. Adicionar columna.

RF6.2. Eliminar columna.

RF6.3. Listar columnas.

RF7 Gestionar asociación de concepto a concepto.

RF7.1 Adicionar asociación de concepto a concepto.

RF7.2 Eliminar asociación de concepto a concepto.

RF7.3 Listar asociación de concepto a concepto.

Debido a las nuevas especificaciones del MFP y la ONEI se adicionaron un total de 12 requisitos funcionales. Tales son los casos de los requisitos: Exportar estado financiero, Modificar grupo, Modificar concepto y los requisitos pertenecientes a las agrupaciones Gestionar columna, Gestionar elemento de gasto a concepto y Gestionar asociación de concepto a concepto.

En la elaboración de los requisitos funcionales se utilizaron los patrones:

- **El nombre revela la intención:** donde los requisitos fueron nombrados de forma tal que se pusiera de manifiesto en éstos, las funcionalidades a informatizar.
- **Preciso y legible:** se aplicó de forma tal que los requisitos definidos comunicaran información suficiente para su adecuada comprensión.

2.4.2. Descripción de requisitos

A continuación se muestra la especificación del requisito Adicionar grupo de la descripción de requisito Gestionar grupo, ver Tabla 8. Las descripciones restantes se encuentran en el expediente de proyecto de CEIGE, además de los restantes artefactos entregados del componente.

Precondiciones	El usuario debe haberse autenticado en el sistema y debe tener permisos para realizar esta operación. Los datos generales del estado financiero deben haber sido introducidos correctamente.
Flujo de eventos	
Flujo básico Adicionar grupo	
1.	Si se desea insertar un nuevo grupo se selecciona la raíz del árbol.
2.	Se introducen los datos correspondientes al grupo: Nombre. Se selecciona la Operación. Fila. Si se desea se puede desmarcar la opción Visible.
3.	Si se desea se puede marcar la opción Subtotal.
4.	El sistema valida (ver validación 1) los datos introducidos.
5.	Si los datos son correctos el sistema los registra.
6.	El sistema confirma el registro de los datos.
7.	Concluye el requisito.
Pos-condiciones	
1	Se registró en el sistema un nuevo grupo para el estado financiero.
Flujos alternativos	
Flujo alternativo 1.a Se selecciona un nodo del árbol para insertar un subgrupo	
1	Volver al paso 2 del flujo básico
Pos-condiciones	

1	N/A
Flujo alternativo *.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se registran los datos.
Flujo alternativo 3.a Se selecciona la opción subtotal	
1	El sistema permite seleccionar un nodo del árbol grupos.
2	Volver al paso 3 del flujo básico.
Pos-condiciones	
1	Se registró en el sistema un nuevo grupo para el estado financiero.
Flujo alternativo 4.a Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A.
Flujo alternativo 4.b Información incompleta	
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A.
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual de contabilidad general: CGS-ERP-N-CG-i2201.
Conceptos	Grupo
	Atributos del concepto que se utilizan en el requisito: Visibles en la interfaz: Nombre Operación Fila Subtotal Utilizados internamente: N/A.
Requisitos especiales	N/A.
Asuntos pendientes	N/A.

Tabla 8. Especificación de requisitos: Adicionar grupo.

2.4.3. Prototipos de interfaz de usuario

Los prototipos de interfaz de usuario brindan la posibilidad de modelar cómo serán las futuras interfaces del sistema, permite además realizar la validación de los requisitos funcionales en cuanto a las funcionalidades o condiciones de ejecución que deben cumplirse. En la Figura 3 se muestra la interfaz Adicionar grupo, el resto de las interfaces se encuentran en el expediente de proyecto junto al resto de los artefactos generados.

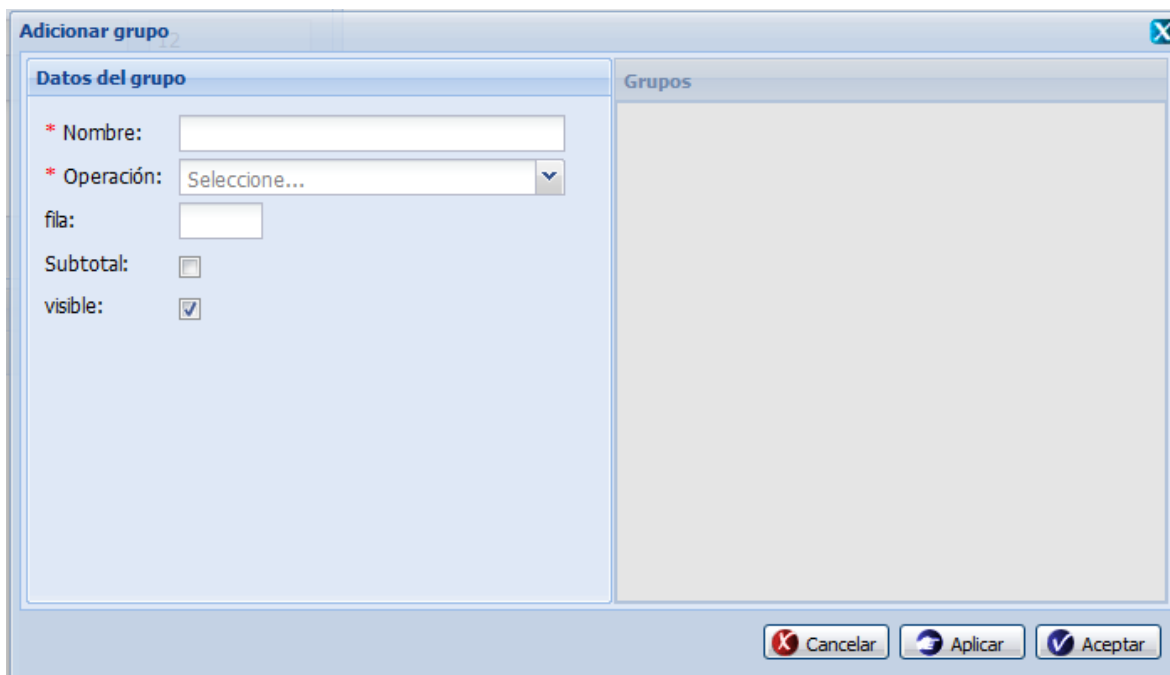


Figura 3. Prototipo de interfaz: Adicionar grupo.

2.5. Diseño de la solución

En este epígrafe se muestran un conjunto de artefactos que serán de gran valor para la disciplina de implementación del componente como son: el modelo de datos, el diagrama de clases del diseño, junto a los patrones utilizados en el mismo.

2.5.1. Modelo de datos

El modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí. Para el desarrollo de la nueva versión del componente se hizo necesario agregar nuevas tablas y modificar las ya existentes. Del diseño de la base de datos anterior se mantuvieron sin cambios tres tablas: `nom_columna`, `dat_columnaestadof` y `datgrupooperadores`. Recibieron cambios un total de cuatro tablas: `nomestadof`, `dat_concepto`, `dat_grupoconcepto` y `dat_desglose`. Además se adicionaron las tablas `dat_conceptooperadores` y `dat_estadofconceptocolumna`. A continuación se muestra una sección del modelo de datos referente al componente Estados financieros, ver Figura 4. Para visualizar el modelo de datos en su totalidad ver Anexo 1.

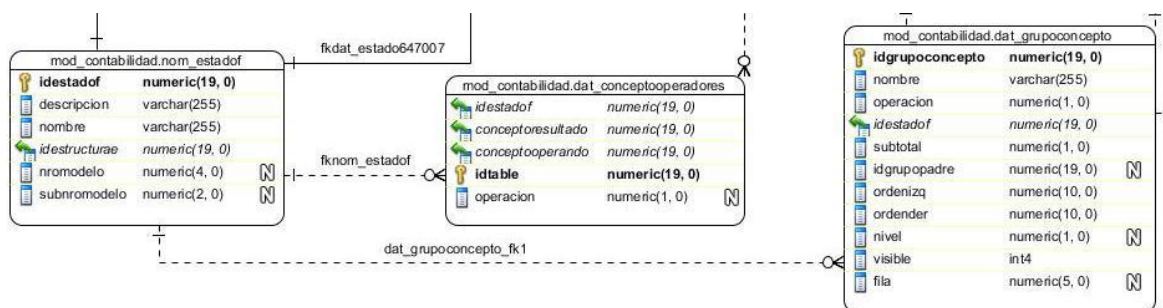


Figura 4. Sección del modelo de datos.

Descripción de las tablas utilizadas

Para una mejor comprensión del modelo antes mostrado se hace necesario realizar una descripción de las tablas utilizadas en su diseño. A continuación se muestra la descripción de las tablas nom_estadof (ver Tabla 9), dat_conceptooperadores (ver Tabla 10) y dat_grupoconcepto (ver Tabla 11). El resto de las descripciones se encuentran en el expediente de proyecto de CEIGE, junto al resto de los artefactos generados.

Nombre	nom_estadof	
Descripción de la tabla	Almacena los datos correspondientes a los estados financieros.	
Atributo	Tipo	Descripción
idestadof	numeric	Llave primaria de la tabla.
descripcion	varchar	Descripción del estado financiero.
nombre	varchar	Nombre del estado financiero.
idestructurae	numeric	Llave foránea, identificador de la entidad a la cual pertenece el estado financiero.
nromodelo	numeric	Número del modelo del estado financiero.
subnromodelo	numeric	Sub-número del modelo del estado financiero.
Tablas relacionadas	Descripción de la relación	
dat_columnaestadof	uno a muchos	
dat_conceptooperadores	uno a muchos	
dat_grupoconcepto	uno a muchos	

Tabla 9. Descripción de la tabla nom_estadof.

Nombre	nom_conceptooperadores
Descripción de la tabla	Almacena los datos correspondientes a las relaciones establecidas entre conceptos.

Atributo	Tipo	Descripción
idtable	numeric	Llave primaria de la tabla.
idestadof	numeric	Llave foránea, identificador del estado financiero al que pertenece el concepto operando.
conceptoresultado	numeric	Llave foránea, identificador del concepto resultado.
conceptooperando	numeric	Llave foránea, identificador del concepto operando.
operación	numeric	Operación del concepto operando.
Tablas relacionadas	Descripción de la relación	
nom_estadof	uno a uno	
dat_concepto	uno a uno	

Tabla 10. Descripción de la tabla nom_conceptooperadores.

Nombre	dat_grupoconcepto	
Descripción de la tabla	Almacena los datos correspondientes a los grupos.	
Atributo	Tipo	Descripción
idgrupoconcepto	numeric	Llave primaria de la tabla.
nombre	varchar	Nombre del grupo.
operación	numeric	Operación a llevar a cabo con el grupo, 1 en caso de suma y -1 en caso de resta.
idestadof	numeric	Llave foránea, identificador del estado financiero al que pertenece el concepto operando.
subtotal	numeric	Toma valor 1 en caso de que el grupo sea subtotal de otros grupos, en caso contrario toma valor 0.
idgrupopadre	numeric	Identificador del grupo padre.
ordender	numeric	Orden derecho del grupo en el árbol.
ordenizq	numeric	Orden izquierdo del grupo en el árbol.
nivel	numeric	Nivel del grupo en el árbol.
fila	numeric	Fila en la que se desea mostrar al grupo.
visible	Numeric	Define si un grupo determinado se muestra o no. Toma valor 1 en caso que se desee mostrar el grupo y 0 en caso contrario.
Tablas relacionadas	Descripción de la relación	

dat_grupooperadores	uno a muchos
dat_concepto	uno a muchos

Tabla 11. Descripción de la tabla dat_grupoconcepto.

2.5.2. Diagrama de clase del diseño

El diagrama de clases es uno de los principales artefactos generados durante la disciplina análisis y diseño del modelo de desarrollo, ya que sirve para visualizar las distintas relaciones estructurales y de herencia entre las distintas clases que componen a un sistema. Incluye además definiciones para atributos y operaciones. Para la realización del diseño del componente Estados financieros se elaboraron un total de 7 diagramas de clases. A continuación se muestra el diagrama de clase del diseño Gestionar grupo, ver Figura 5. El resto de los diagramas se encuentran en el expediente de proyecto de CEIGE, junto con el resto de los artefactos generados.

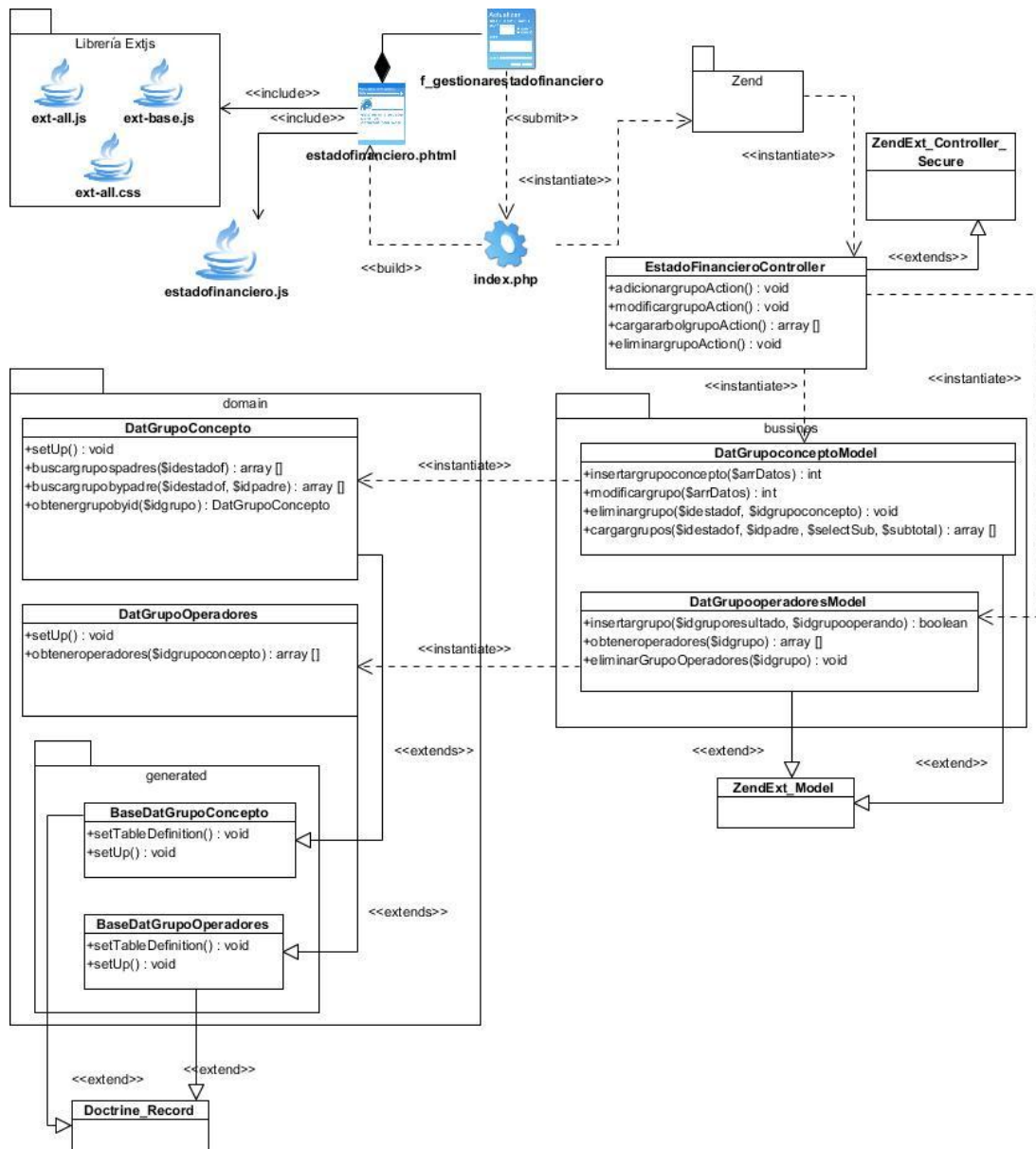


Figura 5. Diagrama de clase del diseño: Gestionar grupo.

Para la definición del diseño de clases, se tomaron como base una serie de patrones entre los que se encuentra el patrón arquitectónico MVC a través del cual se dividen la interfaz de usuario, del control de la lógica del negocio y del acceso a los datos. La interfaz de usuario está compuesta por el archivo `estadofinanciero.phtml` que incluye tanto las librerías de Extjs como el fichero `estadofinanciero.js` donde se implementan las interfaces del componente. Además tendrá contenido un formulario que será el encargado de enviar la información entrada por el usuario, hacia la clase `EstadosFinancieroController`, mediante una instancia única del

controlador frontal de Zend. La clase controladora antes mencionada actúa como intermediaria entre la interfaz y la lógica de negocio. En el diagrama anterior se muestran además las clases necesarias para llevar a cabo la lógica del negocio y el acceso a datos, estas responsabilidades corresponderían a las clases contenidas en los paquetes bussines y domain respectivamente.

2.5.3. Patrones de diseño utilizados:

El diseño de la solución fue elaborado siguiendo patrones que constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. En este caso se emplearon los patrones GRASP. Para contribuir a que el sistema sea más robusto y flexible se utilizaron los siguientes:

- **Experto:** Se pone en práctica con el uso de clases que poseen responsabilidades específicas a cumplir, de acuerdo con la información que manejan. El componente cuenta con clases controladoras, modelos y de entidad que poseen funciones concretas de acuerdo con la información que gestionan. Específicamente: la clase `DatGrupoConceptoModel` será la responsable de efectuar las operaciones que conciernen a las funciones: adicionar, eliminar, listar y modificar los grupos dentro de la base de datos, asumiendo toda la lógica de negocio para cada una de estas funciones. Sobre este mismo principio se realizó el diseño de las restantes funcionalidades.
- **Alta Cohesión:** Este patrón fue utilizado de manera general en el diseño del sistema donde las clases fueron agrupadas según los requerimientos, siempre siguiendo la premisa de que cada clase debe implementar las operaciones que estén sobre la misma área funcional.
- **Creador:** El patrón de creación hizo posible que el diseño pudiera soportar bajo acoplamiento, encapsulamiento y reutilización, así como mayor claridad. La clase controladora es responsable de crear una instancia de las modelos, y éstas a su vez de las entidades. Además este patrón se adapta a las clases del paquete domain, quienes son las encargadas de crear los objetos de tipo `Doctrine_Query`, para permitir el acceso a la información almacenada en la base de datos.
- **Bajo acoplamiento:** En el modelo de datos se definieron un conjunto de clases persistentes, entre las cuales se establecieron las relaciones necesarias de manera que fueran más independientes y reutilizables para reducir el impacto de los cambios, aumentando la oportunidad de una mayor productividad dentro del sistema.

- **Controlador:** La clase controladora: `EstadosfinancieroController`, es ejemplo de la aplicación de este patrón, pues la misma tendrá a cargo la responsabilidad de manejar todos los eventos dentro del componente.

En el diseño del componente además de hacer uso de los patrones antes mencionados se emplearon los siguientes patrones GoF:

- **Cadena de Responsabilidad:** Está concebido que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo sea manejado por el Modelo, creando una nueva excepción de tipo `ZendExt_Exception`. Dicha excepción debe ser propagada al Controlador, el cual será el encargado de capturarla y enviarla a la Vista ya traducida, siendo esta última la que mostrará un mensaje al usuario en un lenguaje entendible, notificando el error y sin especificar detalles del mismo. De esta manera se distribuyen las responsabilidades entre los diferentes componentes, evidenciándose por lo tanto el empleo de este patrón.
- **Fachada:** Brinda una interfaz unificada simple para el acceso de una o varias interfaces de un subsistema. Su aplicación se puede observar en el uso de los servicios, donde la relación que existe entre las clases controladoras y éstos, permite acceder a métodos que están implementados en otros componentes dentro y fuera del subsistema Contabilidad.
- **Singleton:** Su utilización se pone de manifiesto durante las conexiones a la base de datos donde si ya el objeto de conexión está creado no es necesario volver a crear uno nuevo sino que se establece la conexión desde un mismo punto de acceso.

2.5.4. Validación del diseño

En aras de validar el diseño de clases propuesto, se utilizaron las métricas TOC y RC y se determinó en qué medida se cumple con cada uno de los atributos de calidad definidos en éstas.

Métrica Tamaño Operacional de las Clases (TOC):

A continuación se muestran las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado, ver Tabla 12.

Clases	Operaciones	Responsabilidad	Complejidad de implementación	Reutilización
<code>EstadosfinancieroController</code>	21	Alta	Alta	Baja
<code>DatGrupoooperadoresModel</code>	4	Baja	Baja	Alta

DatColumnnaestadofModel	4	Baja	Baja	Alta
DatConceptoModel	4	Baja	Baja	Alta
DatDesgloseModel	4	Baja	Baja	Alta
DatEstadofconcepto columnaModel	4	Baja	Baja	Alta
NomColumnaModel	4	Baja	Baja	Alta
NomEstadofModel	4	Baja	Baja	Alta
DatGrupoconceptoModel	4	Baja	Baja	Alta
DatDesglose	2	Baja	Baja	Alta
DatColumnnaestadof	2	Baja	Baja	Alta
Datconcepto	2	Baja	Baja	Alta
DatEstadofconcepto columna	1	Baja	Baja	Alta
DatGrupoconcepto	7	Media	Media	Media
DatGrupooperadores	2	Baja	Baja	Alta
NomColumna	3	Baja	Baja	Alta
Nomestadof	3	Baja	Baja	Alta

Tabla 12. Resultados de la aplicación de la métrica TOC para cada clase del componente.

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

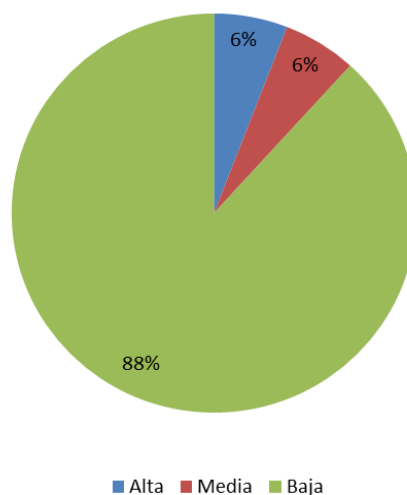


Figura 6. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

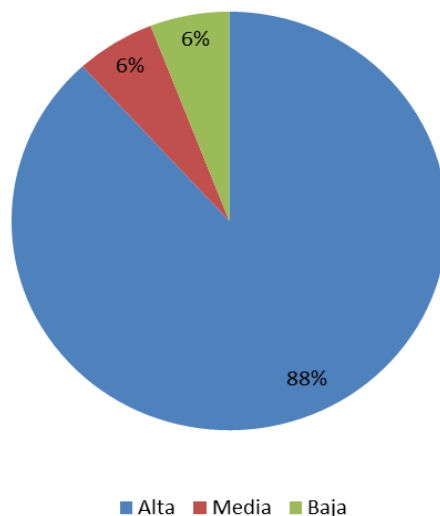


Figura 7. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

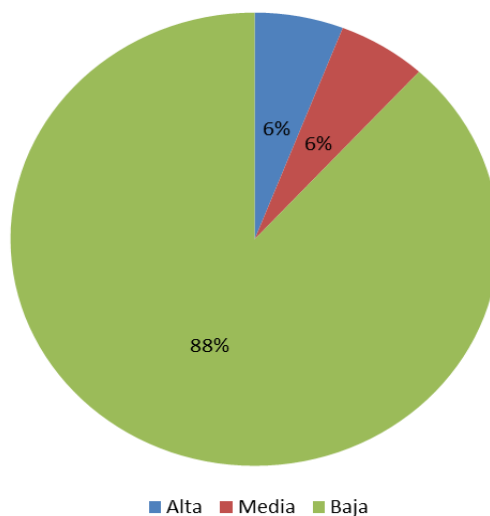


Figura 8. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.

Haciendo un análisis de los resultados obtenidos para los atributos de la métrica TOC en la evaluación del instrumento, se puede observar que la mayoría de las clases que conforman el sistema para los atributos responsabilidad y complejidad están dentro de la categoría Media y Baja para un 94% del total, mientras que el atributo reutilización cuenta con igual por ciento en las categorías Alta y Media mostrando así que el componente cuenta con una elevada reutilización, baja complejidad y responsabilidad en el diseño propuesto. Por lo que se concluye que los resultados obtenidos según esta métrica son positivos.

Métrica Relaciones entre Clases (RC):

A continuación se muestran las clases del componente a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado, ver Tabla 13.

Clases	Relaciones de uso	Acoplamiento	Complejidad del mantenimiento	Reutilización	Cantidad de pruebas
Estadosfinanciero Controller	8	Alta	Alta	Baja	Alta
DatGrupoooperadores Model	1	Baja	Baja	Alta	Baja
DatColumnaestadof Model	1	Baja	Baja	Alta	Baja
DatConcepto Model	1	Baja	Baja	Alta	Baja
DatDesglose Model	1	Baja	Baja	Alta	Baja
DatEstadofconcepto columnaModel	1	Baja	Baja	Alta	Baja
NomColumna Model	1	Baja	Baja	Alta	Baja
NomEstadof Model	1	Baja	Baja	Alta	Baja
DatGrupoconcepto Model	1	Baja	Baja	Alta	Baja
DatDesglose	1	Baja	Baja	Alta	Baja
DatColumnaestadof	1	Baja	Baja	Alta	Baja
Datconcepto	1	Baja	Baja	Alta	Baja
DatEstadofconcepto columna	1	Baja	Baja	Alta	Baja
DatGrupoconcepto	1	Baja	Baja	Alta	Baja
DatGrupoooperadores	1	Baja	Baja	Alta	Baja
NomColumna	1	Baja	Baja	Alta	Baja
Nomestadof	1	Baja	Baja	Alta	Baja

Tabla 13. Resultados de la aplicación de la métrica RC para cada clase del componente.

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

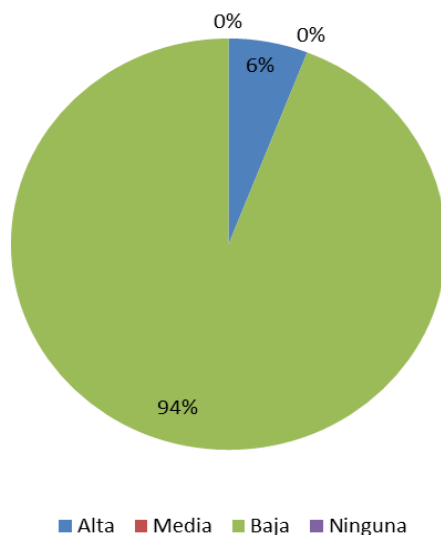


Figura 9. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

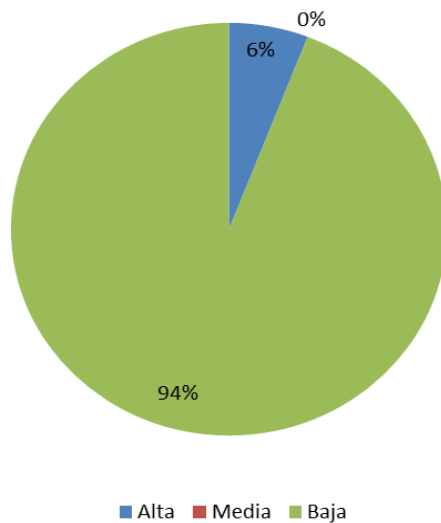


Figura 10. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.

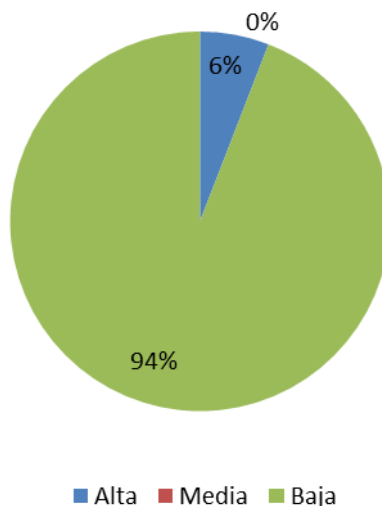


Figura 11. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Mantenimiento.

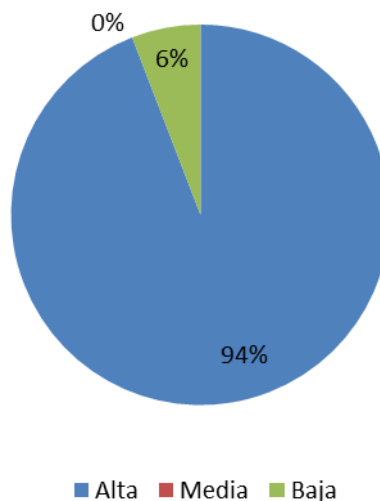


Figura 12. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica RC demuestran que las clases del diseño poseen un bajo acoplamiento, ya que para este atributo la categoría baja representa un 94% del total, mostrando igual por ciento en la categoría alta del atributo reutilización. Para el caso de los atributos mantenimiento y cantidad de pruebas ambos presentaron un 94% para la categoría baja, lo que demuestra que no es necesario un elevado esfuerzo en el momento de realizar cambios, rectificaciones y pruebas al software.

2.6. Conclusiones parciales

Una vez concluido el presente capítulo vale resaltar que:

- El mapa de proceso y el diagrama de procesos del negocio junto a la descripción del mismo, posibilitaron comprender cómo funciona el proceso que es necesario informatizar.
- Mediante la realización del levantamiento de requisitos, se identificaron las funcionalidades que el componente debe brindar.
- Los artefactos generados como parte del diseño representan la base sobre la cual se llevará a cabo la implementación de la solución.
- Posterior a la validación del diseño mediante las métricas TOC y RC se determinó que el mismo se realizó de manera correcta debido a la obtención de criterios de calidad positivos.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

3.1. Introducción

A continuación se muestran aspectos relacionados con la disciplina implementación. Donde se incluyen el diagrama de componentes y los estándares de codificación empleados para la implementación de la nueva versión del componente.

3.2. Implementación

3.2.1. Diagrama de componentes

En el diagrama de componente se representan los componentes y las interacciones entre éstos. A continuación se muestra cómo se integra el componente Estados financieros con los demás componentes del sistema. Dicho componente recibe la información referente a la entidad que está configurando el reporte del subsistema Estructura y composición. Por otra parte, la información concerniente a la fecha, el período y el ejercicio la recibe del subsistema Configuración general. Además, a través del componente Nomenclador de cuentas recibe las cuentas a asociar a los distintos conceptos configurados. Por otra parte, el componente provee servicios a los componentes Análisis financiero y Agregación. Ver Figura 13.

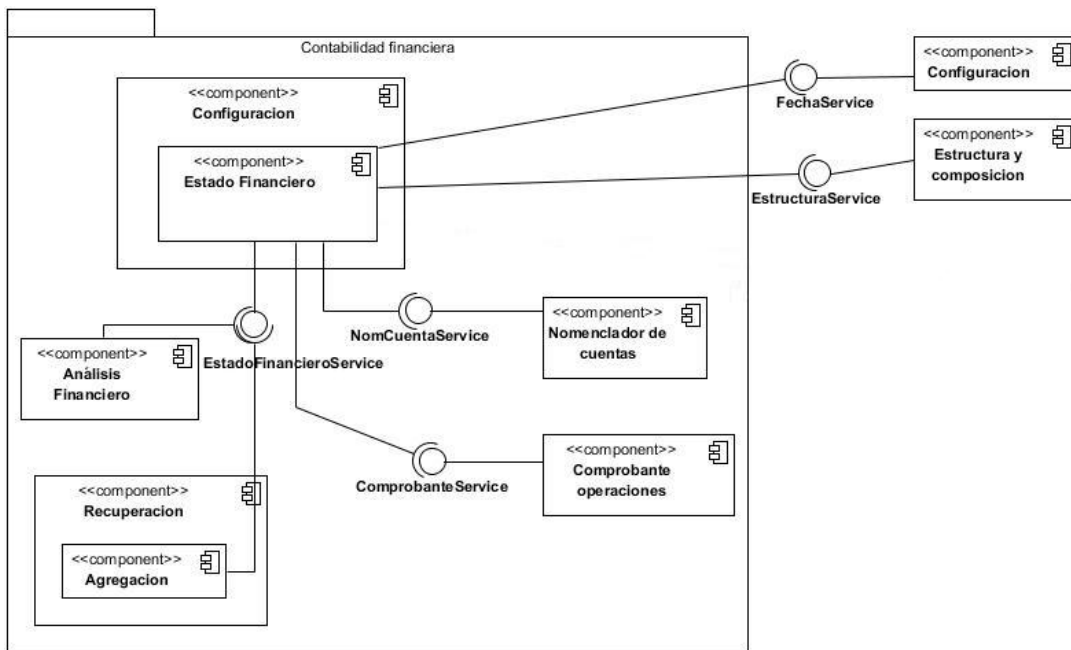


Figura 13. Diagrama de componentes.

3.2.2. Estándares de codificación

Los estándares de codificación establecerán las pautas que conlleven a lograr un código más legible y reutilizable, de tal forma que pueda aumentar su mantenibilidad a lo largo del tiempo. A continuación se muestran los estándares aplicados durante la implementación de la solución.

Estándares de Nomenclatura:

➤ **Nomenclatura de las clases:**

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing**. Con sólo leerlo se reconoce el propósito de la misma.

➤ **Nomenclatura según el tipo de clase**

- Clases controladoras

La clase controladora después del nombre lleva la palabra **Controller**. Ver Figura 14.

```
<?php  
  
class EstadosfinancieroController extends ZendExt_Controller_Secure {
```

Figura 14. Notación *PascalCasing** aplicada en el nombre de la clase EstadosFinancieroController.

- Clases Modelos:

Business:

A las clases que se encuentran dentro de la carpeta Business se les incorpora después del nombre la palabra **Model**. Ver Figura 15.

```
<?php  
  
class NomEstadofModel extends ZendExt_Model {
```

Figura 15. Notación *PascalCasing** aplicada en el nombre de la clase NomEstadofModel.

Domain:

Las clases que se encuentran dentro de la carpeta Domain reciben su nombre de acuerdo a la tabla de la base datos que representan. Ver Figura 16.

```

<?php

/**
 * This class has been auto-generated by the Doctrine ORM Framework
 */
class NomEstadof extends BaseNomEstadof {

```

Figura 16. Notación PascalCasing* aplicada en el nombre de la clase NomEstadof.

Generated:

Las clases que se encuentran dentro del directorio Domain/generated reciben su nombre de acuerdo a la tabla en la base de datos que representan y a la vez se les agrega la palabra Base al inicio. Ver Figura 17.

```

<?php

/**
 * This class has been auto-generated by the Doctrine ORM Framework
 */
abstract class BaseNomEstadof extends Doctrine_Record {

```

Figura 17. Notación PascalCasing* aplicada en el nombre de la clase BaseNomEstadof.

➤ Nomenclatura de las funciones

El nombre de las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing**, y con sólo leerlo se reconoce el propósito de la misma. Ver Figura 18.

```

public function insertarEstadoF($arrDatos) {

```

Figura 18. Notación CamelCasing* aplicada en el nombre de la función insertar estado financiero.

En caso de ser una función de la clase controladora se le pone un nombre seguido de la palabra Action. Ver Figura 19.

```

*/
public function adicionarConceptoAction() {

```

Figura 19. Notación CamelCasing* aplicada en el nombre de la función adicionar concepto.

➤ Nomenclatura de variables

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing**, y comenzando con un prefijo según el tipo de datos. Ver Figura 20.

```
$intIdEstadoF = $this->_request->getPost('idestadof');

$intIdGrupo = $this->_request->getPost('idgrupococepto');

$arrNewColumns = $arrConcepto[0]->columns;
```

Figura 20. Notación *CamelCasing** aplicada en los nombres de las variables.

3.2.3. Normas de comentarios

➤ Nomenclatura de los comentarios

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando.

- Clases

Antes de la declaración de una clase se escribe una breve descripción donde se explique el propósito de la misma. Ver Figura 21.

```
/**
 * NomEstadofModel
 * Clase modelo del negocio para gestionar los estados financieros
 * @author Alberto Mesa Martínez
 * @package Contabilidad
 * @subpackage Contabilidad Financiera
 * @copyright UICID-ERP Cuba
 * @version 1.1-0 */
```

Figura 21. Comentarios de la clase *NomEstadofModel*.

- Funciones

Antes de la declaración de la función se escribe una breve descripción donde se explique el propósito de la misma. Ver Figura 22.

```

/**
 * insertarEstadoF
 * Inserta un estado financiero
 * @param array arrDatos - Arreglo con todos los datos del estado financiero
 * @return *(se pone lo que devuelve la función y un comentario)
 */

public function InsertarEstadoF($arrDatos) {

```

Figura 22. Comentarios de la función insertar estado financiero.

3.3. Pruebas

3.3.1. Diseño de casos de prueba

Unas de las formas para comprobar los requisitos de software es utilizando los diseños de casos de prueba, durante la realización de las pruebas de caja negra. Para la realización de las mismas se empleó la técnica partición de equivalencia, a través de la cual se divide el dominio de entrada del componente en clases de datos válidos y no válidos. A continuación se representa el diseño de caso de prueba para el requisito funcional Adicionar grupo (del cual se muestra **el requisito a probar y la descripción de las variables, la tabla juego de datos a probar** se encuentra en el Anexo 2), los 25 diseños de casos de pruebas restantes se encuentran en el expediente de proyecto del centro CEIGE, además de los restantes artefactos entregados del componente.

Requisito a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar grupo.	Se adiciona un nuevo grupo con los atributos: Nombre, Operación, Fila, Subtotal y Visible.	EP 1.1: Adicionar un nuevo grupo, presionando el botón Aceptar e introduciendo datos válidos.	<ul style="list-style-type: none"> – Se presiona el botón Nuevo. – Se introducen los datos del grupo que se desea adicionar. – Se presiona el botón Aceptar de la ventana y esta se cierra. – Se muestra un mensaje de información, informando que se realizó la acción satisfactoriamente. – Se presiona el botón Aceptar del mensaje de información.
		EP 1.2: Adicionar un nuevo grupo, presionando el botón Aplicar e introduciendo datos	<ul style="list-style-type: none"> – Se presiona el botón Nuevo. – Se introducen los datos del grupo que se desea adicionar. – Se presiona el botón Aplicar de

		válidos.	<p>la ventana y esta se mantiene visible.</p> <ul style="list-style-type: none"> – Se muestra un mensaje de información, informando que se realizó la acción satisfactoriamente. – Se presiona el botón Aceptar del mensaje de información.
		EP1.3: Adicionar un nuevo grupo, presionando el botón Aceptar y dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se presiona el botón nuevo. – Se introducen los datos dejando campos requeridos en blanco. – Se presiona el botón Aceptar de la ventana. – El sistema muestra un mensaje de error y señala los campos requeridos en blanco. – Se presiona el botón Aceptar del mensaje de error.
		EP1.4: Adicionar un nuevo grupo, presionando el botón Aplicar y dejando campos requeridos en blanco.	<ul style="list-style-type: none"> – Se presiona el botón nuevo. – Se introducen los datos dejando campos requeridos en blanco. – Se presiona el botón Aplicar de la ventana. – El sistema muestra un mensaje de error y señala los campos requeridos en blanco. – Se presiona el botón Aceptar del mensaje de error.
		EP1.5: Adicionar un nuevo grupo introduciendo errores en los datos.	<ul style="list-style-type: none"> – Se presiona el botón Nuevo. – Se introducen los datos del grupo que se desea adicionar introduciendo errores en los mismos. – El sistema no permite que se introduzcan caracteres no válidos.
		EP 1.6: Cancelar	<ul style="list-style-type: none"> – Se presiona el botón Nuevo. – Se introducen o no los datos en el formulario. – Se presiona el botón Cancelar.

Descripción de variables

No	Nombre de campo	Tipo	Válido	Inválido	Inválido
----	-----------------	------	--------	----------	----------

1	Nombre	Alfanumérico	Activos	Vacío	(/*-i".\$%&.)
2	Operación	Lista desplegable	Suma	Vacío	(/*i".\$%&)
3	Fila	Numérico	12	(/*-i".\$%&)	Primera fila
4	Subtotal	Boolean	false	N/A	N/A
5	Visible	Boolean	True	N/A	N/A

3.3.2. Resultados de las pruebas

Una vez realizadas las pruebas al componente, se detectaron 5 no conformidades (NC) en la primera iteración que fueron resueltas para la segunda iteración de las pruebas, en la cual no fueron detectadas otras NC. Cumpliéndose con el objetivo principal de las pruebas al asegurar que el software cumpliera con las especificaciones requeridas y la eliminación de los posibles defectos que éste pudiera tener. Una vez resueltas las NC detectadas se obtiene el Acta de Liberación por parte del Departamento de Calidad del Centro CEIGE.

3.4. Validación de la solución

Con el objetivo de validar la propuesta de solución se pretende reflejar a través de un preexperimento la existencia de personalización de los estados financieros, definidos por el MFP, y en la medida en que ésta se manifiesta en la nueva versión del componente.

Posterior a un estudio de las funciones básicas de la versión actual del componente se determinó que éstas eran insuficientes a la hora de configurar los estados financieros establecidos en las resoluciones vigentes. A partir de lo antes mencionado y del análisis de los estados financieros establecidos, se determinaron una serie de opciones de configuración que garantizan en cierta medida la personalización de los mismos, de acuerdo a las necesidades de las entidades cubanas. Dichas opciones se determinaron a partir de los distintos elementos (columnas, cuentas, elementos de gasto, conceptos y grupos) que conforman a los estados financieros y las distintas asociaciones que se establecen entre ellos. Dichas opciones se menciona y definen a continuación:

- Configuración de grupos: el componente debe permitir la creación y modificación de grupos permitiendo configurar sus datos.
- Definición jerárquica de grupos: el componente de ser capaz de crear grupos de manera jerárquica hasta un segundo nivel.

- Configuración de conceptos: el componente debe permitir la creación y modificación de conceptos permitiendo la configuración de sus datos.
- Configuración dinámica de columnas: el componente debe permitir la posibilidad de asignar columnas y configurar su valor en dependencia del estado financiero.
- Asociación entre grupos: el componente debe permitir la posibilidad de establecer grupos subtotales que contendrían el resultado final de las operaciones llevadas a cabo entre otros grupos.
- Asociación grupo-concepto: el componente debe ser capaz de poder asignar a un grupo un concepto o conjunto de conceptos.
- Asociación concepto-cuenta: el componente debe permitir establecer cierta relación entre un concepto y una o varias cuentas.
- Asociación concepto-elemento de gasto: el componente debe permitir establecer cierta relación entre un concepto y uno o varios elementos de gasto.
- Asociación concepto-concepto: el componente debe permitir establecer cierta relación entre un concepto y uno o varios conceptos del estado financiero que se está configurando o de otro estado financiero previamente insertado en el sistema.
- Asociación exclusiva: el componente debe permitir en el caso de los grupos la asociación con otros grupos (en esta caso la asociación padre e hijo de la jerarquía) o con conceptos, no ambas.

Se determinó como personalización de estados financieros: a la configuración de éstos según las características y necesidades de las entidades. Aparejado a ello se estableció un índice de personalización, el cual estará dado por el porcentaje de estados financieros configurados por una u otra versión del componente. Por lo que se hace necesario resaltar cuáles son las opciones incluidas en ambas versiones (donde 1 significa que la opción está presente y 0 lo contrario). Ver Tabla 14.

Opciones de configuración	versión 1.0	versión 1.1
Configuración de grupos	1	1
Definición jerárquica de grupos	0	1

Configuración de conceptos	1	1
Configuración dinámica de columnas	0	1
Asociación entre grupos	1	1
Asociación grupo-conceptos	1	1
Asociación concepto-cuentas	1	1
Asociación concepto-elemento de gasto	0	1
Asociación concepto-concepto	0	1
Asociación exclusiva	0	1

Tabla 14. Opciones de configuración por componente.

Para realizar el experimento planteado se tomaron como muestra 2 de las entidades económicas registradas en el sistema, una entidad presupuestada y una entidad patrimonial. Por cada entidad se intentaron configurar los estados financieros establecidos en las resoluciones del MFP usando la versión 1.0 y posteriormente usando la versión 1.1 del componente.

Valoración de los resultados obtenidos

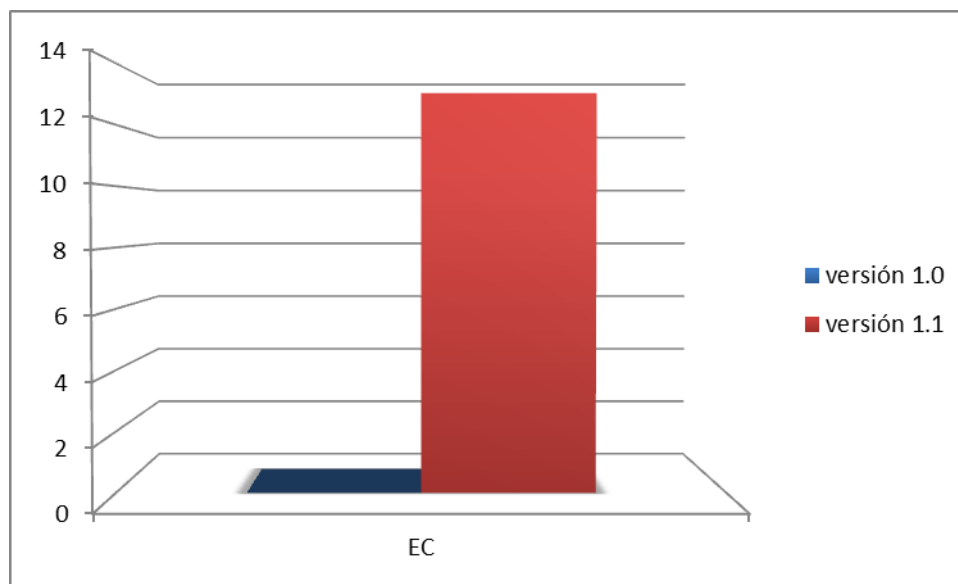


Figura 23. Índice de personalización de estados financieros por versión del componente.

Posterior a la ejecución del experimento se obtuvo para el caso de la versión 1.0 un 0% de personalización, debido a que impide la configuración de los estados financieros comprendidos en las resoluciones vigentes. Para el caso de la versión 1.1 se obtuvo un 86.67% de

personalización, permitiendo la obtención de 13 de los 15 estados financieros establecidos por el MFP.

3.5. Conclusiones parciales

Al finalizar el presente capítulo se arribó a las siguientes conclusiones parciales:

- Con la implementación de la nueva versión del componente se garantiza la personalización de los estados financieros establecidos en las resoluciones vigentes.
- Los resultados obtenidos durante las pruebas de caja negra y el preexperimento realizado permitieron validar la propuesta de solución.

CONCLUSIONES

Una vez finalizado el presente Trabajo de Diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- La elaboración del marco teórico del trabajo, permitió la selección de las herramientas, lenguajes y el modelo de desarrollo utilizado en la implementación del componente.
- La modelación del diseño y su posterior validación, permitió el desarrollo de una solución robusta y acorde a las necesidades de los clientes.
- El componente desarrollado contribuye a la personalización de los estados financieros durante el proceso de obtención de los mismos.
- Las pruebas realizadas permitieron verificar el correcto funcionamiento del componente así como el ajuste a las necesidades y requisitos del cliente.

RECOMENDACIONES

- Implementar los servicios necesarios en el subsistema costos y procesos que permitan el trabajo con las partidas, los elementos asociados a éstas y sus respectivos saldos.
- Implementar las funcionalidades necesarias para la configuración de los estados financieros Estado de Gastos por Partidas tanto para la actividad presupuestada como autofinanciada, en el componente Estados financieros.

BIBLIOGRAFÍA

Acosta Romero, Carolina y Porras Medina, Carlos Hernán. 2007. *OpenBravo*. Bogota : s.n., 2007.

Acosta Sigler, Lillian , Fuentes Suárez, Jesús y González de las Casas, Yanelis. 2007. *Sistema Informático para la consolidación de los Estados Financieros de asignación de presupuesto en el MINFAR*. Ciudad de La Habana : s.n., 2007.

Ávila, Emilio Aviles: Diagramas de iteración (Secuencia y Comunicación/Colaboración). 2009. Slideshare. [En línea] 07 de 06 de 2009. [Citado el: 20 de 2 de 2013.] <http://www.slideshare.net/techmi/curso-uml-23-diagramas-de-interaccin>.

Baryolo, Oiner Gómez, Cabrera, Marianela Tenrero y Silega, Nemuris Martínez. 2008. *Plantilla Registro de la Propiedad intelectual(Sauxe)*. La Habana : UCI, 2008.

Cabrera Gonzáles, Miguel P., y otros. 2006. *Sistema Económico Integrado, Versat Sarasola. Documento técnico*. 2006.

CEIGE. 2013. *Modelo de desarrollo de software*. La Habana : s.n., 2013.

CENDITEL. 2011. *Plataforma de Desarrollo de Software Libre (PDSL). Manual del Usuario del Sistema de Control de Versiones (SVN)*. [En línea] CENDITEL, 2011. [Citado el: 20 de 11 de 2012.] <http://plataforma.cenditel.gob.ve/wiki/ManualUsuarioSvn>.

Chronotech, el futuro está en nuestras manos. *Modelos De Datos*. [En línea] [Citado el: 05 de mayo de 2012.] <http://sites.google.com/site/jalexiscv/modelosdedatos>.

CITMATEL. 2002. *Rodas XXI. Sistema Integral Administrativo*. [En línea] 2002. <http://www.rodasxxi.cu/>.

Colectivo de autores: UCID. 2009. *Proceso de Desarrollo y Gestión de Proyectos de Software*. Ciudad de la Habana : s.n., 2009.

Colectivo de JSON. 2011. *JSON*. [En línea] JSON, 2011. [Citado el: 30 de Noviembre de 2011.] <http://www.json.org/json-es.html>.

Colectivo editorial W3C Consortium. 2008. *W3C Consortium*. [En línea] 9 de 1 de 2008. [Citado el: 19 de 11 de 2012.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.

Doctrine-Project.org. 2010. *Doctrine*. [En línea] 2010. [Citado el: 23 de 11 de 2012.] <http://www.doctrine-project.org>.

Durá, Javier Ruiz. 2011. *0120_Arquitectura de Software*. La Habana : s.n., 2011.

2010. *Ecured*. [En línea] 14 de diciembre de 2010. [Citado el: 8 de Diciembre de 2013.] http://www.ecured.cu/index.php/RODAS_XXI.

2008. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. *DISAC Casa Consultora*. [En línea] 2008. [Citado el: 8 de diciembre de 2013.] <http://www.disaic.cu/modules.php?name=content&pa=showpage&pid=818>.

Escalona, María José y Koch, Nora. 2002. *Ingeniería de Requisitos en Aplicaciones para la Web*. Sevilla : s.n., 2002.

Gamma, Erich, y otros. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995.

García Ramírez, Christian y Pavón Reyes, Raydel. 2013. *Componente dinámico para la interoperabilidad de procesos*. La Habana : s.n., 2013.

Gerencie.com. *Gerencie.com*. [En línea] [Citado el: 8 de 11 de 2013.]

Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y García Tejo, Darien. 2012. *Arquitectura tecnológica para el desarrollo de software*. 2012.

González, Mairelys Fernández y Rivera, Osley Zorrilla. 2010. Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. Ciudad de la Habana : UCI, 2010.

Gamma, Erich, y otros. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995.

HTML.net. 2012. HTML.net. [En línea] 2012. [Citado el: 19 de 11 de 2012.] <http://es.html.net/tutorials/html/lesson2.php>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : s.n., 2000.

Larman, Craig. 1999. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN 0-13-748880-7.

Larman, Craig. 2001. *Patterns for Effective Use Cases*. 2001.

Layola, William. 2006. *Maestría en Sistemas de Información Gerencial*. 2006.

Lic. Miguel P. Cabrera González, Msc. Guillermo Obregón Rodríguez, Msc. Margarita Cárdenas Negrin, Lic. Luis Mario Carralero Silva. Sistema Económico Integrado Versat sarasola. [En línea] [Citado el: 8 de diciembre de 2013.] <http://www.forum.villaclara.cu/index.php/component/phocadownload/category/1-forumponencias/>.

López, Lic. Rafael Antonio Almaguer. 2010. *Consultor electrónico del contador y el auditor*. La Habana : DISAIC, 2010.

- López, Martín E. 2007.** Boletín N°8 Rama de Estudiantes de IEEE-UNED. [En línea] 2007. http://www.ieec.uned.es/ieee/investigacion/ieee_dieec/sb/boletin/boletin_8_Octubre_2007.pdf.
- Martínez, Rafael. 2009.** PostgreSQL-es. [En línea] 6 de 6 de 2009. [Citado el: 20 de 11 de 2012.] <http://www.postgresql.org.es/node/297>.
- MFP. 2006.** *NORMA CUBANA DE CONTABILIDAD No. 1 PRESENTACIÓN DE ESTADOS FINANCIEROS*. La Habana : s.n., 2006.
- . 2012.** *Resolución No. 434/2012*. La Habana : s.n., 2012.
- Microsoft. 2014.** Microsoft. *Developer Network*. [En línea] 2014. [Citado el: 8 de 12 de 2014.]
- Moreno, Adarlis Fernández y Fontanills, Edwing Robert Odelín. 2009.** *Modelado de negocio y Levantamiento de requisitos del subsistema Activos Fijos Tangibles del sistema Cedrux*. La Habana : s.n., 2009.
- Mozilla Developer Network. 2011.** Mozilla Developer Network. [En línea] 09 de 11 de 2011. [Citado el: 5 de 22 de 2013.] https://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript.
- Mozilla Firefox. 2009.** GetFirefox. [En línea] 2009. [Citado el: 22 de 11 de 2012.] <http://www.getfirefox.es/firefox-features..>
- Observatorio Tecnológico. 2008.** Observatorio Tecnológico. [En línea] 2008. [Citado el: 22 de 5 de 2013.] <http://recursostic.educacion.es/observatorio/web/es/software/servidores/580-elvira-mifsud>.
- OMG. 2014.** OMG. *Object Management Group*. [En línea] 2014. [Citado el: 20 de Febrero de 2014.] <http://www.omg.org/bpmn/Documents/FAQ.htm>.
- 2009.** OpenBravo. *Accounting Report*. [En línea] 2 de Diciembre de 2009. [Citado el: 8 de Marzo de 2014.] http://wiki.openbravo.com/wiki/Accounting_Report.
- OpenBravo. 2007.** OpenBravo wiki. [En línea] 2007. [Citado el: 08 de 12 de 2013.]
- 2011.** OpenBravo wiki. *Balance sheet and PL structure*. [En línea] 22 de Marzo de 2011. [Citado el: 8 de Marzo de 2014.] http://wiki.openbravo.com/wiki/General_Accounting_Report.
- Oracle Corporation. 2012.** NetBeans. [En línea] 2012. [Citado el: 21 de 11 de 2012.] <http://netbeans.org/community/releases/70/>.
- Pressman, Roger S. 2005.** *Ingeniería del Software. Un enfoque práctico*. 2005.
- ProgramacionWeb.net. 2005.** ProgramacionWeb.net. [En línea] ProgramacionWeb.net, 4 de 10 de 2005. [Citado el: 15 de 11 de 2012.] <http://www.programacionweb.net/cursos/curso.php?num=2>.

- Ramírez, Jaime. 2012.** Métodos de Prueba del Software. Unidad de Programación. [En línea] 2012. [Citado el: 28 de mayo de 2012.] <http://lml.ls.fi.upm.es/>.
- 2002.** RODAS XXI Sistema Integral Económico Administrativo. [En línea] 2002. [Citado el: 8 de diciembre de 2013.] <http://www.rodasxxi.cu/index.php>.
- S. Pressman, Roger. 2001.** *Ingeniería de Software un enfoque práctico*. Madrid : s.n., 2001.
- SAP AG. 1999.** *Reporting made easy*. Palo Alto, California : s.n., 1999.
- Sommerville, Ian. 2005.** *Ingeniería de software*. Madrid : Pearson educación, S.A., 2005.
- The FreeBSD Foundation. 2012.** The FreeBSD Project. [En línea] 2012. [Citado el: 23 de 11 de 2012.] <http://www.freebsd.org/doc/es/articles/explaining-bsd/article.html>.
- The PHP Group. 2011.** PHP: Hipertext Preprocesor. [En línea] php.net, 4 de 2 de 2011. [Citado el: 19 de 11 de 2012.] <http://php.net/manual/es/intro-what-is.php>.
- Toro, Amador Durán. 2000.** *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Sevilla : s.n., 2000.
- Tovar, C.P. Juan Carlos Torres. 1979.** *Contabilidad I. Introducción a la contabilidad*. México : Diana, 1979.
- Visual Paradigm. 2007.** freedownloadmanager.org. [En línea] 5 de 3 de 2007. [Citado el: 20 de 11 de 2012.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
- White, Stephen A. 2014.** OMG. *Object Managment Group*. [En línea] 2014. [Citado el: 20 de Febrero de 2014.] http://www.omg.org/bpmn/Documents/Notations_and_Workflow_Patterns.pdf.
- Zamora, Yordanis Milanes. 2006.** *Sistema Automatizado para la consolidación de los Estados Financieros en el MINFAR*. Ciudad de La Habana. : s.n., 2006.

GLOSARIO DE TÉRMINOS

Ant: Es una herramienta Open-Source utilizada en la compilación y creación de programas Java, Está escrita en XML y Java.

BSD: La licencia BSD (Berkeley Software Distribution) fue creada inicialmente para los sistemas operativos de la Universidad de Berkeley. Se califica como una licencia mucho más libre que la GPL y más rápida.

Doctrine: Es un mapeador objetos relacional para PHP. Permite trabajar con los datos persistidos como si fueran parte de una base de datos orientada a objetos posibilitando escribir consultas a la base de datos en un dialecto orientado a objetos de propiedad SQL.

Elementos de gasto: El elemento es un concepto económico asociado al gasto que permite la cuantificación de los recursos materiales, laborales y monetarios en los cuales se expresan los gastos de trabajo vivo y pretérito para un período en el conjunto de la actividad empresarial.

ERP (por sus siglas en inglés, Enterprise Resource Planning) o sistemas de planificación de recursos empresariales, es un paquete de software que permite administrar todos los procesos operativos de una empresa, integrando varias funciones de gestión en un único sistema.

ExtJS: Es una librería construida con JavaScript para el diseño de interfaces de usuario del lado del cliente haciendo uso extensivo de Ajax. Emplea una arquitectura flexible que permite construir aplicaciones complejas utilizando componentes predefinidos.

Log: Registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre: quién, qué, cuándo, dónde y por qué: un evento ocurre. Por lo tanto se puede decir que un log es una evidencia digital.

Posición Económica: Capacidad que tiene una empresa de obtener resultados, a través de la comparación de todos los ingresos con los gastos, dicho resultados pueden ser positivos o negativos (ganancia o pérdida).

Posición Financiera: Capacidad que tiene la empresa para afrontar sus deudas a sus respectivos plazos de vencimientos (deudas a cortos y largo plazo).

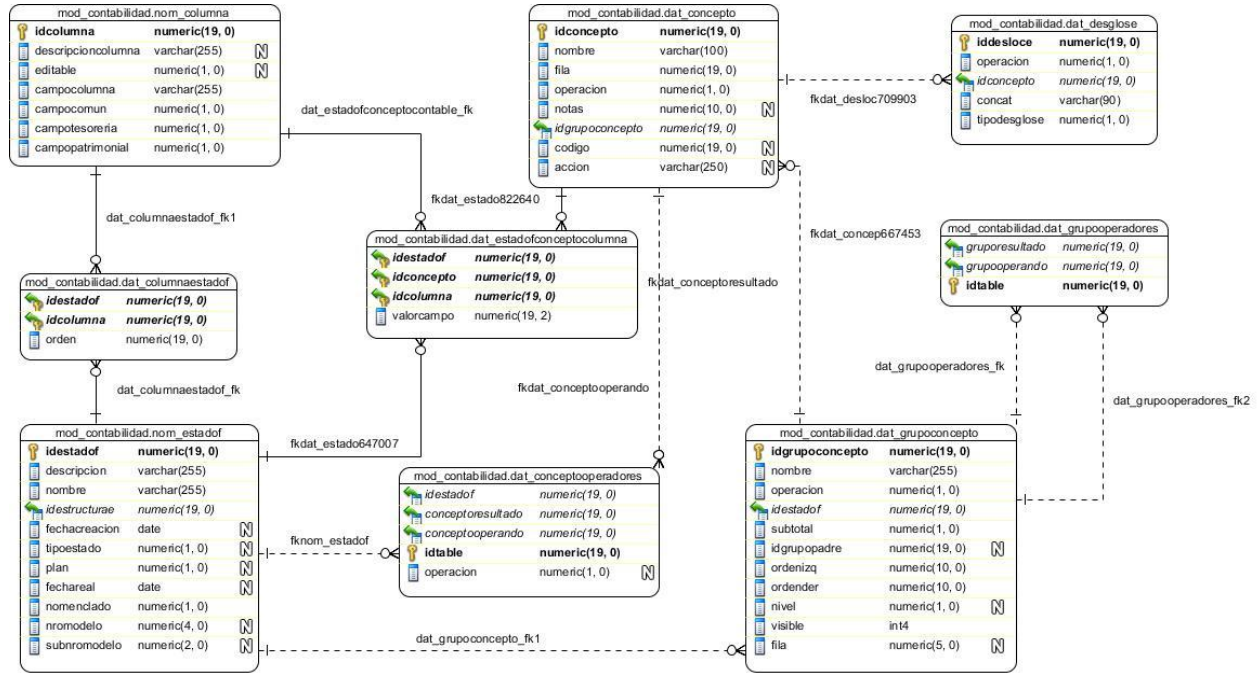
Refactoring: Es una técnica para la reestructuración de un cuerpo existente de código, alterando su estructura interna sin cambiar su comportamiento externo.

Secure Sockets Layer: protocolo criptográfico que proporciona comunicación segura en Internet.

Zend Framework: Es un Framework para el desarrollo de aplicaciones y servicios Web con PHP. Es de código abierto y emplea PHP 5 en sus más recientes versiones.

ANEXOS

Anexo 1. Modelo de datos



Anexo 2. Juegos de datos a probar

Escenario	Nombre	Operación	Fila	Subtotal	Visible	Respuesta del sistema
Adicionar un nuevo grupo, presionando el botón aceptar e introduciendo datos válidos.	V (Activos)	V(Suma)	V(1)	V(False)	V(True)	Se adiciona el grupo. Se oculta la ventana y se muestra un mensaje informando que “El grupo fue adicionado satisfactoriamente”
	V (Total de Activos)	V(Suma)	V(2)	V(True)	V(False)	
	V(Para agrupar)	V(Resta)	V(Vacío)	V(False)	V(False)	
	V(Total)	V(Suma)	V(3)	V(True)	V(True)	
Adicionar un nuevo grupo,	V (Activos)	V(Suma)	V(1)	V(False)	V(True)	Se adiciona el grupo. Se

presionando el botón aplicar e introduciendo datos válidos.	V (Total de Activos)	V(Suma)	V(2)	V(True)	V(False)	mantiene la ventana visible y se muestra un mensaje informando que “El grupo fue adicionado satisfactoriamente”
	V(Para agrupar)	V(Resta)	V(Vacío)	V(False)	V(False)	
	V(Total)	V(Suma)	V(3)	V(True)	V(True)	
Adicionar un nuevo grupo, presionando el botón aceptar y dejando campos requeridos en blanco.	I(Vacío)	V(Suma)	V(Vacío)	V(False)	V(False)	Se muestra los campos de texto en rojo. El sistema muestra el mensaje de error “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)”. La ventana Adicionar Grupo se mantiene visible.
	V (Patrimonio neto.)	I(Vacío)	V(4)	V(True)	V(True)	
Adicionar un nuevo grupo, presionando el botón aplicar y dejando campos requeridos en blanco.	I(Vacío)	V(Suma)	V(Vacío)	V(False)	V(False)	Se muestra los campos de texto en rojo. El sistema muestra el mensaje de error “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)”. La ventana Adicionar Grupo se mantiene visible.
	V (Patrimonio neto.)	I(Vacío)	V(4)	V(True)	V(True)	
Adicionar un nuevo grupo introduciendo errores en los datos.	I(^*{ };\$)	V(Suma)	V(1)	V(True)	V(True)	El sistema no permite que se introduzcan caracteres especiales en el

						campo nombre.
	V (Elementos de gasto.)	V(Resta)	I(Primera Fila)	V(False)	V(False)	El sistema no permite que se introduzcan letras en el campo fila.
	V (Elementos de gasto.)	V(Resta)	I(1234)	V(False)	V(False)	El sistema señala que el número máximo de caracteres en el campo fila es 3.
	V (Elementos de gasto.)	V(Resta)	I((^{ };\$))	V(False)	V(False)	El sistema no permite que se introduzcan caracteres especiales en el campo fila.
Cancelar	NA					Se cancela la operación. Se cierra la ventana: Adicionar grupo.