

# Universidad de las Ciencias Informáticas

## FACULTAD 4



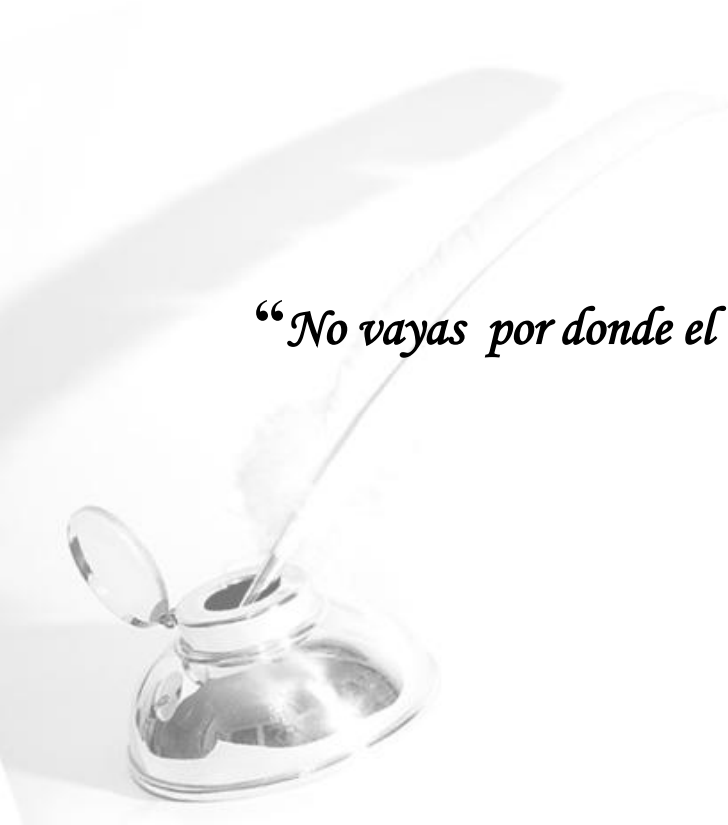
### **Título: Análisis y diseño del subsistema Multimoneda del ERP-Cuba.**

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

**Autor(es):** Lilian Cid Escalona  
Ismel Sarduy Sánchez

**Tutor(es):** Ing. César Lage Codorníu  
Ing. Orlando Palacios Martínez

Mayo, 2009



*“No vayas por donde el camino te lleve, ve por donde no haya  
camino y deja un sendero”*

*Ralph Waldo Emerson*

## DATOS DEL CONTACTO



**Ing. César Lage Codorníu.**

Ingeniero recién graduado que se incorpora al proyecto ERP-Cuba, donde se desempeña como uno de los responsables de la línea de arquitectura.

*e-mail:* clage@uci.cu



**Ing. Orlando Palacios Martínez.**

Durante el 4 cursos consecutivos (3 de estudiante y uno de graduado), cumple tareas como jefe de línea de implementación en el Módulo de Inventario del Proyecto MINFAR, donde desarrolló un generador de capa de acceso a datos como herramienta complementaria para el desarrollo, realiza su tesis sobre el tema Inventario, específicamente Entrega de medios materiales, sistema que está siendo usado en las principales unidades militares del Ministerio en cuestión. Luego pasa a formar parte del proyecto ERP-Cuba, donde se desempeña como arquitecto de la línea de configuración.

*e-mail:* opalacios@uci.cu

## AGRADECIMIENTOS



*Agradecemos a todas las personas que han colaborado para con la realización de este trabajo. A todos aquellos que de manera directa o aún sin saberlo han hecho su aporte a nuestra formación tanto personal como profesional. Un aparte especial en este espacio para nuestros tutores, a quienes queremos agradecerles por su ayuda y apoyo incondicional en todo momento.*

*A mis padres que han sido el mayor símbolo de perseverancia, mi motor impulsor y los orientadores de mi vida.*

*A mi familia.*

*A mis amigos (as) y compañeros(as), todos esos que han marcado mi vida y han hecho que mi estancia en esta universidad sea un tanto, trascendental. Obligado para mí mencionar nombres como: Daríel Chirino, Daylín Real, Aliankis y mi entrañable y ocioso compañero de tesis, Ismel.*

*A Gisselle, Yesenia, Mileidys y Yaneska quienes se han convertido en mis amigas y de las que siento no haber tenido la posibilidad de conocerlas con anterioridad.*

*A Adolfo y su familia, en especial a Nancy por su preocupación, apoyo y cariño.*

*A mi entrenador del equipo UCI de Fútbol sala femenino, Ariel.*

*A Yaniselis Sánchez, que fue más que mi profesora y me animó a seguir cuando creía que el cielo se desplomaba sobre mí.*

*A mi gente de Puerto Padre que siempre me apoyó y me animó a seguir adelante.*

*Lilian Cid Escalona.*

*A mis padres y a mi familia que me han sabido llevar siempre por el mejor camino, que me dieron las fuerzas para seguir adelante cuando pensé abandonar.*

*A mi tío y a Yami por haberme recibido y atendido durante estos 5 años como un hijo más.*

*A mis profesores, principalmente a todos los que fueron nuestros amigos.*

*A todo mi grupo que estuvo desde primer año y a los que se incorporaron después. A Sergio, Lesniel, Alexey, Edwing, Miguel Angel, Yenisel, Reinier, Noel, Mailyn, y Lilian, mi compañera de tesis.*

*A René por ser más que el jefe de proyecto, por ser nuestro amigo.*

*A todos con los que de una manera u otra he conocido y compartido durante estos 5 años.*

*Ismel Sarduy Sánchez.*

## DEDICATORIA



*A la Revolución y al Comandante en Jefe Fidel Castro, artífices de este genial proyecto que es la Universidad de las Ciencias Informáticas; institución que nos brinda la posibilidad de optar por el título que persigue el presente trabajo...*

*Los autores.*

## RESUMEN

Las tendencias actuales en el ambiente empresarial han supuesto para Cuba la necesidad de integrar la información y automatizar los procesos en las entidades. La Universidad de las Ciencias Informáticas ha asumido la construcción de un sistema de Planificación de Recursos Empresariales, que permita gestionar las operaciones de las empresas cubanas dedicadas a la producción de bienes o servicios.

La intención fundamental de este trabajo es lograr una solución para el tratamiento de la multimonedada y sus particularidades, adecuada al sistema que se desarrolla. Para esto se pretende elaborar el análisis y diseño del subsistema Multimonedada en el ERP-Cuba. Incluye un compendio de las características multimonedada de varios sistemas ERP existentes, las cuales son el cimiento de esta propuesta de solución que exhibe, entre sus principales virtudes, la centralización de todas las funciones identificadas dentro de un único componente.

Su impacto se concentra en el establecimiento de las bases para la posterior implementación de este componente, teniendo en cuenta los resultados que aquí se reseñan. Su misión es brindar soporte funcional y optimizar los procesos dentro de los restantes subsistemas que integran el producto Cedrux, siendo un recurso que se ha concebido para satisfacer las necesidades del cliente en el mayor grado posible.

**Palabras claves:** ERP, subsistema, multimonedada, dualidad monetaria, requerimientos, análisis y diseño.

## TABLA DE CONTENIDOS

DATOS DEL CONTACTO .....	I
AGRADECIMIENTOS .....	I
DEDICATORIA .....	II
RESUMEN .....	III
TABLA DE CONTENIDOS .....	IV
INTRODUCCIÓN .....	8
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>12</b>
1.1. <i>Introducción</i> .....	12
1.2. <i>Puntos importantes sobre el tratamiento de la multimonedas</i> .....	12
1.2.1. <i>Ventajas</i> .....	12
1.3. <i>Estudio de sistemas ERP que incluyen multimonedas</i> .....	13
1.3.1. <i>Descripción del tratamiento a la multimonedas en los sistemas estudiados</i> .....	13
1.4. <i>Temas a considerar, en CedruX, para el tratamiento de la multimonedas y sus especificaciones</i> .....	18
1.4.1. <i>Especificaciones generales para multimonedas</i> .....	18
1.4.1.1. <i>Particularidades para la implementación de la multimonedas en la solución</i> .....	18
1.4.2. <i>Dualidad monetaria</i> .....	22
1.4.2.1. <i>Breve reseña</i> .....	22
1.4.2.2. <i>Especificaciones generales para dualidad monetaria (4)</i> .....	23
1.4.2.3. <i>Implicaciones de la dualidad monetaria en la solución</i> .....	25
1.4.3. <i>Reevaluación de cuentas</i> .....	27
1.4.3.1. <i>Consideraciones para la reevaluación de cuentas</i> .....	27
1.4.4. <i>¿Qué se propone para el tratamiento de la multimonedas en CedruX?</i> .....	30
1.5. <i>Ingeniería de requerimientos</i> .....	31
1.5.1. <i>Requerimientos</i> .....	32
1.5.2. <i>Obtención de los requerimientos</i> .....	33
1.5.3. <i>Especificación de requerimientos</i> .....	36
1.5.4. <i>Validación de requerimientos</i> .....	37
1.6. <i>Diseño de software</i> .....	38

1.6.1.	<i>Modelo de Diseño</i> .....	39
1.6.1.1.	Principios básicos del diseño .....	39
1.7.	<i>Tendencias tecnológicas actuales a considerar en la solución.</i> .....	40
1.7.1.	<i>Ingeniería del software.</i> .....	40
1.7.2.	<i>Metodología, notación y lenguaje de modelado</i> .....	40
1.7.2.1.	<i>Metodología de desarrollo.</i> .....	41
1.7.2.2.	<i>Business Process Modeling Notation (BPMN)</i> .....	41
1.7.2.3.	<i>Unified Modeling Language (UML)</i> .....	42
1.7.3.	<i>Herramientas CASE.</i> .....	43
1.7.3.1.	Visual Paradigm .....	44
1.8.	<i>Conclusiones.</i> .....	44
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.</b> .....		<b>46</b>
2.1.	INTRODUCCIÓN .....	46
2.2.	ANÁLISIS DEL ENTORNO DEL PROBLEMA .....	46
2.2.1.	PROCESO REEVALUACIÓN DE CUENTAS .....	46
2.2.2.	MODELO CONCEPTUAL .....	48
2.2.3.	ESTRATEGIA PARA LA IDENTIFICACIÓN DE CONCEPTOS (19) .....	48
2.2.4.	DEFINICIÓN Y ESPECIFICACIÓN DE LOS CONCEPTOS IDÓNEOS PARA EL DOMINIO DEL PROBLEMA.....	49
2.2.5.	REPRESENTACIÓN DEL MODELO CONCEPTUAL .....	49
2.2.5.1.	PROPUESTA DE MODELO CONCEPTUAL .....	50
2.2.5.2.	DESCRIPCIÓN DEL MODELO CONCEPTUAL .....	50
2.3.	INGENIERÍA DE REQUERIMIENTOS APLICADA AL ENTORNO DEL PROBLEMA.....	51
2.3.1.	<i>Obtención</i> .....	51
2.3.2.	<i>Especificación</i> .....	52
2.3.3.	<i>Validación</i> .....	53
2.4.	ESPECIFICACIÓN DE REQUERIMIENTOS PARA EL SUBSISTEMA MULTIMONEDA .....	54
2.4.1.	FUNCIONALIDADES IDENTIFICADAS .....	54
2.4.2.	REQUISITOS FUNCIONALES .....	55
2.4.3.	ESPECIFICACIÓN DE REPORTES .....	88
2.4.4.	VALIDACIÓN DE REQUERIMIENTOS.....	89
2.5.	CONCLUSIONES .....	90



<b>CAPÍTULO 3: DISEÑO DEL SISTEMA .....</b>	<b>91</b>
3.1. INTRODUCCIÓN .....	91
3.2. UTILIZACIÓN DEL SUBSISTEMA MULTIMONEDA DENTRO DEL SISTEMA INTEGRAL DE GESTIÓN CEDRUX .....	91
3.3. ACTIVIDADES DEL DISEÑO .....	93
3.4. ARQUITECTURA DEL SISTEMA .....	94
3.4.1. <i>Decisiones arquitectónicas</i> .....	95
3.5. PATRONES DE DISEÑO.....	95
3.6. DIAGRAMAS DE CLASES DEL DISEÑO.....	98
3.6.1. <i>Diagrama de clases con estereotipos WEB</i> .....	99
3.6.1.1. <i>Clases controladoras (controller)</i> .....	104
3.6.1.2. <i>Descripción de las clases más significativas</i> .....	105
3.7. MÉTRICAS PARA LA EVALUACIÓN DEL DISEÑO .....	115
3.7.1. <i>Resultados de la evaluación del diseño según las métricas aplicadas</i> .....	116
3.8. CONCLUSIONES .....	121
<b>CONCLUSIONES .....</b>	<b>122</b>
<b>RECOMENDACIONES .....</b>	<b>123</b>
<b>BIBLIOGRAFÍA.....</b>	<b>124</b>
<b>ANEXOS .....</b>	<b>128</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>138</b>

**INDICE DE IMÁGENES**

ANEXO 1- FLUJO DE ACTIVIDADES DEL MODELO DE DESARROLLO ..... 128

ANEXO 2- TABLA COMPARATIVA ENTRE TÉCNICAS PARA OBTENCIÓN DE REQUERIMIENTOS ..... 128

ANEXO 3- EJEMPLO DEL REPORTE HISTORIAL DE TASAS DE UNA MONEDA EN UN PERÍODO DEFINIDO ..... 130

ANEXO 4- EJEMPLO DEL REPORTE DIARIO DE TASAS ..... 131

ANEXO 5- REPRESENTACIÓN DEL PATRÓN ARQUITECTÓNICO ..... 131

ANEXO 6- TABLA DE RELACIÓN DE LOS PATRONES GRASP. .... 132

ANEXO 7 – ESTEREOTIPOS PARA EL DISEÑO DE APLICACIONES WEB ..... 134

ANEXO 8 – VALORES DE LA APLICACIÓN DE LA MÉTRICA TOC ..... 135

ANEXO 9 – VALORES DE LA APLICACIÓN DE LA MÉTRICA RC..... 136

## INTRODUCCIÓN

El inexorable paso del tiempo ha provocado un incremento en la necesidad de contar con información confiable, íntegra y oportuna para el cumplimiento de los objetivos estratégicos de las organizaciones. En los últimos años, la Tecnología de Información (TI<sup>1</sup>) ha evolucionado para dejar de ser un componente básico del negocio y convertirse en un elemento crítico requerido para la ejecución de las estrategias en este ámbito. Una de las respuestas más sólidas para esta tendencia está reflejada en los Sistemas de Planeación de Recursos Empresariales (ERP<sup>2</sup>), que se concretan como solución luego de un intenso período de transformación que tuvo sus inicios en la década del '60.

Los ERP pueden definirse como sistemas que integran y manejan muchas de las prácticas de los negocios asociados con las operaciones de producción y los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios. (1) Es un sistema estructurado para satisfacer la demanda de soluciones de gestión empresarial, basado en el ofrecimiento de una solución completa que permite a las empresas evaluar, implementar y administrar con mayor facilidad su negocio. (2)

La implantación de un sistema de gestión sirve de soporte para una administración eficiente. En los últimos tiempos han adquirido un auge considerable dentro del mercado empresarial pues las compañías están obligadas a ser más competitivas, para lo que se hace necesario la optimización e integración de sus flujos internos de información y sus relaciones comerciales externas. Lo anterior es vital para el cumplimiento de objetivos básicos tales como: mejorar la productividad, la calidad, el servicio al cliente y la reducción de costos. Las tecnologías de la información han permitido, en gran medida, el logro de dichos propósitos. Entre las razones por las que una compañía decide implantar un sistema ERP se puede mencionar, por ejemplo, la necesidad de integrar la información financiera, normalizar y acelerar los procesos de manufactura, reducir el inventario así como estandarizar la información de los recursos humanos. La variedad del software de tipo empresarial en el mundo es amplia, sin embargo la autenticidad de un ERP está determinada por tres características esenciales; son sistemas integrales, adaptables y modulares. (3)

---

<sup>1</sup> TI: Tecnología de la Información.

<sup>2</sup> ERP: Enterprise resource planning.

**Integrales:** Porque tratan a las diferentes áreas o departamentos de una organización como elementos unidos por la información que generan.

**Adaptables:** Porque son sistemas capaces de adaptarse a cualquier empresa, independiente del sector al que pertenezcan y de las particularidades de los procesos de negocio.

**Modulares:** Porque estos sistemas están formados por un número específico de módulos, independientes entre sí, pero que a la vez están comunicados, lo que permite una gran adaptabilidad a las empresas de acuerdo a su tamaño y disponibilidad de recursos. Esta última es una de las mayores ventajas que exhiben estos sistemas. Entre los módulos más comunes de los software ERP se encuentran: Gestión financiera, Gestión de ventas, Gestión de compras, Gestión de la distribución y logística, Gestión y Planificación de la producción, Gestión de proyectos, Gestión de recursos humanos y otros, que varían en función de las necesidades del cliente.

Cuba no está ajena al desarrollo de las TIC<sup>3</sup> y el reto que hoy enfrentan las empresas es creciente, por ello se ha decidido emprender un proyecto para la construcción de un ERP propio. Se precisa de un sistema que posibilite estandarizar la comunicación entre las empresas nacionales, que pueda servir como herramienta para la gestión económica de los recursos, a todos los niveles de organización. Un software capaz de acelerar la gestión de información, lo que es esencial para apoyar la toma de decisiones y de alta utilidad para luchar contra la corrupción y las ilegalidades. Es importante, también, porque significa no tener que importar sistemas extranjeros lo cual, en la actualidad, se traduce en un gasto por concepto de licencias y consultoría ascendente a varios millones de CUC<sup>4</sup>.

El proyecto de construcción de un ERP nacional ha sido asumido por la Universidad de las Ciencias Informáticas (UCI) y constituye un desafío para el equipo de desarrollo vinculado a este software. La implementación de este producto, denominado ERP- Cuba, es una tarea novedosa pues es la primera vez que se pretende liberar una aplicación de esta gran magnitud, ya que sus antecedentes más cercanos habían estado orientados a un sector o rama de la economía, como es el caso de VERSAT, SISCONT y RODAS, por mencionar algunos, sin embargo, este se ha concebido para todas las áreas de la economía e incluso de la sociedad. Destaca, de igual forma, porque las herramientas que se utilizarán para su desarrollo pertenecen al mundo del software libre, lo que constituye un paso decisivo para llevar a un nuevo nivel estas tecnologías dentro de la sociedad cubana.

---

<sup>3</sup> TIC: tecnologías de la información y la comunicación.

<sup>4</sup> CUC: Pesos Cubanos Convertibles.

El ERP cubano toma experiencias positivas de otras soluciones, entre las que se debe mencionar el sistema VERSAT Sarasola que se emplea en el MINAZ<sup>5</sup> desde hace algunos años pero busca proporcionar un producto completo, adaptable a todas las esferas de la economía y la sociedad cubana. Los subsistemas generales que se han definidos son: Contabilidad, Inventario, Activos fijos tangibles e intangibles, Facturación, Cobros y pagos, Caja, Banco, Estructura y composición, Capital humano, Costos y procesos, Planificación, Configuración, Multimoneda y Auditoría, los cuales están en correspondencia con las soluciones de ERP que se comercializan en el mercado pero que han sido personalizadas de acuerdo a las condiciones cubanas.

Una de las premisas fundamentales para esta solución es garantizar el flujo monetario en cada empresa del país. La misma, deviene en la interrogante de cómo manejar la multimoneda, que no es más que la utilización de varias monedas en las transacciones económicas, la convertibilidad de varias monedas con respecto a la moneda contable<sup>6</sup> y todas las particularidades que implica el tema de la dualidad monetaria, consecuencia de la existencia de dos monedas, que comparten legalmente las mismas funciones del dinero en la economía nacional. De igual manera, hay que enfrentarse a las consecuencias directas de la descentralización que existe para el manejo de esta información, lo que se traduce en: falta de conocimiento y descontrol sobre el uso de las monedas, poca o escasa accesibilidad a estos temas en un lugar único y la existencia de saldos desactualizados en cuentas que registran operaciones en moneda diferente a la que guía la contabilidad en la entidad, por mencionar algunos efectos.

Por lo expuesto anteriormente, el **problema científico** que motiva esta investigación se centra en: ¿Cómo lograr una solución para el tratamiento de la multimoneda, adecuada al sistema ERP-Cuba?

Se define como **objeto de estudio** las soluciones multimoneda en los sistemas ERP y se especifica como **campo de acción** el Análisis y diseño de la solución multimoneda del ERP-Cuba.

Para guiar la investigación, la **idea a defender** trazada es que: la centralización de los elementos de multimoneda en un subsistema, puede facilitar la disponibilidad de los mismos en el sistema ERP-Cuba.

En general, el **objetivo** del presente trabajo de diploma es elaborar el análisis y diseño de la solución multimoneda del ERP-Cuba, el mismo se ha desglosado en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.

<sup>5</sup> MINAZ: Ministerio del Azúcar (cubano).

<sup>6</sup> Moneda contable: Término que define la moneda base de las transacciones económicas. Ver Glosario de Términos para más información.

- Identificar y especificar requisitos funcionales para los elementos generales de multimonedas del proyecto ERP-Cuba.
- Confeccionar los prototipos de interfaz de usuario.
- Realizar el diseño adecuado de cada una de las funcionalidades identificadas para el subsistema Multimonedas.
- Evaluar la efectividad del diseño realizado a través de la aplicación de métricas.

Los métodos teóricos de investigación utilizados son:

**Analítico – Sintético:** Se divide el problema a estudiar en sus principales componentes, en este caso, la solución se nutre de multimonedas, dualidad monetaria y reevaluación de cuentas, para comprender las implicaciones de cada uno por separado. El resultado de este análisis se complementa con la selección de las características principales de los fenómenos anteriores, las que se integran para buscar la mayor completitud en la solución que se desarrolla.

**Histórico – Lógico – Tendencial:** Este trabajo exhibe una propuesta que se basa en el estudio de las soluciones históricas que ha tenido el tratamiento de la multimonedas en sistemas ERP, las condiciones que impusieron su surgimiento y las principales consecuencias que tiene el mismo.

El contenido de esta tesis está estructurado en tres capítulos:

**Capítulo 1. Fundamentación teórica:** Recoge el estudio de los sistemas que existen vinculados al campo de acción, además de la investigación realizada en torno a las herramientas que se utilizarán para la confección del software que se quiere desarrollar.

**Capítulo 2. Características del sistema:** En este capítulo se realiza una caracterización del sistema; identificando los conceptos vinculados al dominio definido para la solución del problema. Además se definen y especifican los requerimientos funcionales identificados para el subsistema Multimonedas del producto CedruX<sup>7</sup>.

**Capítulo 3. Diseño del sistema:** Se muestran los elementos de diseño concebidos para la solución que se propone para el subsistema Multimonedas.

---

<sup>7</sup> CedruX: Sistema Integral de gestión desarrollado por el proyecto ERP- Cuba en la Universidad de las Ciencias Informáticas. Es conocido, también, como ERP Cuba, ERP cubano.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

### 1.1. Introducción

Este capítulo engloba aspectos relacionados con el objeto de estudio definido para el presente trabajo. Se analizan algunos sistemas automatizados existentes en los que se tienen en cuenta, los conceptos asociados al campo de acción del problema propuesto. Se presenta, además, una reseña sobre determinados temas a tener en cuenta para el tratamiento de la dualidad monetaria y su posible impacto en la solución. Igualmente se mencionan las herramientas, técnicas, metodologías y notación utilizadas en el desarrollo de la solución que se propone.

### 1.2. Puntos importantes sobre el tratamiento de la multimonedas.

En el ámbito general, multimonedas es un término referido a la diversidad de monedas. A los efectos de un sistema de gestión, representa la posibilidad de utilizar varias monedas en la ejecución de transacciones económicas y su registro. Esta propiedad supone la existencia de una moneda base<sup>8</sup> o contable y de otras monedas que tienen convertibilidad con relación a la moneda base, la misma se expresa en tasas de cambios.

Una característica de las operaciones en multimonedas es que las transacciones originales se conservan en la moneda en que se realizaron, siendo en los procesos de registro contable, donde se homogeniza la moneda aplicando la tasa de cambio establecida para tales fines. (4)

#### 1.2.1. Ventajas

La implementación de funcionalidades referentes al tratamiento de múltiples monedas en un sistema ERP proporciona los siguientes beneficios:

- *Competitividad* ya que con la internacionalización de las empresas este se ha hecho un requerimiento obligatorio para estos sistemas.
- *Flexibilidad* permitiendo registrar las operaciones o transacciones en varias divisas.
- *Incremento* en la productividad.
- *Escalabilidad* del sistema al propiciar fácil adaptación a cambios en la moneda base o en alguna de las monedas definidas.

---

<sup>8</sup> Moneda base: Análogo a moneda contable. (Ver Glosario de Términos).

- *Convertibilidad* propiciando trabajar con tasas de cambio, las que se expresan en función de la moneda contable.
- *Contabilización* de los saldos reales que poseen las cuentas.

### 1.3. Estudio de sistemas ERP que incluyen multimoneda.

Esta investigación centra sus esfuerzos en estudiar y comprender el tratamiento que recibe el tema de la multimoneda en sistemas existentes. Hoy en día, el empleo de este término se ha convertido en propiedad obligatoria para cualquier ERP, ya que la inmensa mayoría de las empresas del mundo trabajan con varias monedas, lo que viene dado por la internacionalización de las mismas.

#### 1.3.1. Descripción del tratamiento a la multimoneda en los sistemas estudiados.

El estudio efectuado para comprender el dominio del problema mostró que casi la totalidad de los sistemas ERP que se conciben en el mundo son software que facilitan el trabajo con varias monedas. A continuación, se presenta una reseña de la multimoneda en los sistemas que se observó mayor solidez y organización al respecto:

**Microsoft Dynamics:** Producto de la compañía Microsoft que constituye una línea de soluciones tecnológicas integradas para la gestión empresarial. Adaptable a todo tipo y escala de negocio, que permite optimizar el desempeño y tomar decisiones con un mayor grado de certeza. En Dynamics, existen tres áreas claves donde se trata la utilización de varias monedas: Gestión de las relaciones con el cliente, Gestión de la cadena de suministro y Gestión financiera.

El área de Gestión financiera cuenta con un módulo denominado Administración multimoneda, cuyas capacidades están estrechamente vinculadas a otros módulos como Contabilidad general, Cuentas por Cobrar, Cuentas por pagar, Compras, Facturación, Activos fijos y Transacciones intercompañía. (5)

La Administración multimoneda permite manejar un número ilimitado de monedas así como, de tasas de cambio por cada una de ellas. Posibilita administrar en múltiples monedas los servicios de: ventas, recibos de caja, extractos y las ganancias o pérdidas por diferencia en cambio. Ofrece otras funcionalidades como:

- Transacciones en la moneda original o en la moneda funcional con su equivalente reflejado en la misma pantalla.



- Registrar tasas de cambio con la hora y la fecha permitiendo el uso de múltiples tasas de cambio por día.
- Generar reportes financieros en varias monedas para compartirlos con sus asociados alrededor del mundo. (5)

Entre los beneficios identificados con la utilización de este módulo se pueden mencionar:

- Administración de operaciones multinacionales con facilidad: Cumple fácilmente con las normas internacionales de divisas y se puede trabajar con clientes y proveedores internacionales mientras realiza transacciones financieras en múltiples divisas.
- Definir la moneda de acuerdo a las necesidades: Aumenta y mantiene los tipos de cambio actualizados y se pueden definir nuevos tipos de cambio para clientes o contratos específicos según sea necesario.
- Mejora el control y la flexibilidad: Facilita a los empleados la administración de estadísticas financieras y operaciones de contabilidad mientras se utilizan múltiples divisas y maneja tipos de cambio de variación constante.
- Administración de eficiencia financiera: A través de una completa gestión de administración de tipos de cambio de múltiples divisas, se puede controlar la forma en que se hacen negocios a escala global y definir tantas divisas o tipos de cambio según sus necesidades. (6)

**SAP R/3:** Creado por la empresa SAP<sup>9</sup> AG, se define como un software abierto, basado en la tecnología cliente/servidor, diseñado para manejar las necesidades de información de una empresa. Es un sistema integrado de gestión que permite controlar todos los procesos que se realizan en una organización mediante módulos como: Finanzas, Ventas y distribución, Almacenes e inventarios, Producción, Recursos humanos y otros. Provee a los clientes de poderosas herramientas para el desarrollo y personalización del sistema en relación con los requerimientos individuales de la empresa.

El sistema SAP R/3 es capaz de diferenciar entre la moneda local (moneda contable) y la moneda de transacción (moneda de entrada). Esto fue puesto en práctica durante la fase de moneda dual<sup>10</sup> donde las

---

<sup>9</sup> SAP: Sistemas, Aplicaciones y Productos en proceso de datos.

<sup>10</sup> Fase moneda dual: Periodo de tiempo en el cual dos monedas comparten las mismas funciones. Ej. Migración al Euro que experimentaron los países de la Unión Europea (UE) que cambiaron de moneda entre el 1 de enero de 1999 hasta el 30 de junio del 2002.

empresas que lo utilizaban pudieron procesar sus operaciones en la moneda de transacción Euro, aún cuando la moneda local era distinta. Igualmente, brinda la posibilidad de convertir informes individuales como balances o cuenta de pérdidas y ganancias en Euros, aún en el caso de que continúen llevando su contabilidad en moneda nacional.

**Openbravo:** Es un sistema de gestión de código abierto basado en la web, realizado por la compañía Openbravo S.L fundada en agosto del 2001. Openbravo ERP consta de dos versiones: Openbravo Community Edition (libre y gratuita) y Openbravo Network Edition (con elementos privativos y comerciales).

Está dirigido a empresas de mediano y pequeño tamaño. Incluye una amplia área de funcionalidades propias de las soluciones ERP como: Gestión de compras y almacenes, Gestión de proyectos y servicios, de la producción, Comercial y Gestión económico-financiera.

En lo referido a la presente investigación, Openbravo permite escoger la moneda en la que se van a gestionar los registros, procesos o informes. Admite definir Rangos de conversión, que son los valores de conversión que se utilizarán para calcular los importes de los documentos en diferentes divisas, donde se precisará el período de tiempo en el cual será válida la conversión.

Los temas de gestión de monedas y de tasas de cambio de monedas se tramitan en el módulo Configuración general que incluye este ERP.

En Open Bravo se contabiliza en base a la moneda establecida en el esquema contable, utilizando los Rangos de conversión de moneda que el usuario introduzca en el sistema. Cualquier tipo de ajuste debido a un cambio en la conversión se refleja en las cuentas:

- (B\_Revaluationloss\_Acct) - Pérdida por minusvalías.
- (B\_Revaluationgain\_Acct) - Beneficio por plusvalías.

Las que se localizan en (C\_ACCTSCHEMA\_DEFAULT), lo que representa el Esquema contable - Valores por defecto.

Las operaciones de reevaluación no están automatizadas lo que implica que la realización de estos cálculos y el registro de los mismos se hacen de modo manual.

**Compiere ERP:** Es una solución de negocios de código abierto desarrollada por la empresa Compiere Inc. Provee de un recurso de fácil utilización, categorizado como de primer nivel para empresas de mediano tamaño. No está organizado en módulos, sino en procesos de negocios. Este diseño permite manejar los procesos, en lugar de los departamentos tradicionales lo que encaja perfectamente en las

tendencias actuales, donde, en las empresas medianas los empleados realizan habitualmente el proceso de negocio entero o procesos interrelacionados. Entre los principales procesos de negocio de Compiere están: Cotización Ingresos, Requisición a pago, Administración de Clientes, Administración de Socios, Abastecimientos y Análisis de Resultados. Compiere permite trabajar en muchos países, debido a que está diseñado para operar con múltiples monedas, funcionalidad que interviene cuando el sistema se usa para varias organizaciones. Los beneficios de la inclusión de los temas de Multimonedas dan pie a las siguientes funcionalidades:

- *Transacciones multimonetarias*: Esto se traduce en la posibilidad de efectuar transacciones en una moneda distinta a la moneda contable. Posibilidad de revalorizar las transacciones y tener cuentas bancarias en una moneda diferente a la contable.
- *Reportes multimonetarios*: Posibilidad de traducir transacciones o balances con la intención de generar reportes. Esto es útil cuando los socios de negocios están en otros países.
- *Contabilidad Multimonedas*: Posibilidad de contabilizar transacciones paralelas en diferentes monedas.

Compiere soporta todos los aspectos de trabajar con múltiples monedas (desde listas de precios, hasta moneda preferida por el cliente) sin necesidad de copiar o replicar las transacciones. Una transacción puede tener una o varias monedas. Sin embargo, todas las monedas están al mismo nivel, no hay una moneda primaria, esto, a simple vista resulta inconveniente pero a los efectos de este ERP permite agregar, cambiar o detener operativas en distintas monedas sin dificultad. A esta funcionalidad se le atribuyen dos puntos de vital importancia, el turismo y el comercio internacional. (7)

**SINAPSIS ERP**: Desarrollado por SINAPSIS TECNOLOGÍAS, empresa de construcción de software empresarial que se enfoca en el área de las tecnologías de la información y ofrecer soluciones globales para la optimización de los procesos empresariales al servicio de la PYME<sup>11</sup>.

Es un sistema organizativo completo que permite gestionar a través de una única interfaz todos los recursos de la empresa, proporcionando mejoras en la productividad y el rendimiento de la misma. Entre sus funcionalidades se pueden observar; la gestión de almacenes, compras, ventas, contabilidad, impuestos, tesorería y gestión de la producción.

---

<sup>11</sup> PYME: Pequeña y mediana empresa (conocida también por su acrónimo PyME o pyme), es un negocio con características distintivas. Son agentes con lógicas, culturas, intereses y espíritu emprendedor específico.

Es un sistema multimonedado porque permite comprar y vender en cualquier moneda, así como contabilizar en la que se designe como moneda contable.

La multimonedado en sinapsis ERP está concebida como una característica fundamental. Esta recibe un tratamiento individual en cada uno de los módulos que deben considerar las especificaciones del tema.

En Contabilidad, los apartados que tienen relación son: Borrador de asientos multimonedado, Extractos multimonedado por períodos, Mayor multimonedado, Sumas y Saldos multimonedado.

En Inmovilizado, que se encarga de gestionar la amortización del inmovilizado de la empresa. Permite llevar las amortizaciones y de esta forma obtener unos balances de situación al día, se incluye el tema, desde el punto de vista que se pueden introducir amortizaciones de bienes comprados en cualquier moneda.

**VERSAT-Sarasola:** Es un sistema cubano que automatiza casi todas las actividades de planificación, control y análisis económico de cualquier tipo de entidad. El mismo está conformado por 12 subsistemas. Este sistema se puede adecuar a las características de cada entidad y tiene como objetivo brindarle al usuario la oportunidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo.

En lo referente al tema de investigación este sistema permite trabajar con diferentes monedas para revalorizar los estados financieros en una fecha y con una tasa determinada. También permite procesar los documentos primarios de los subsistemas en doble moneda. Sin embargo, se debe resaltar que esto se trata a nivel de las necesidades individuales de cada uno de sus módulos, es decir, que no posee una centralización de estos aspectos, de igual forma, estas consideraciones se incorporaron en las últimas versiones del producto ya que en su concepción inicial no se valoró esta característica.

De manera conclusiva, con lo estudiado en este epígrafe, es posible resumir que los sistemas ERP que se producen en el mundo exhiben entre sus características funcionales el hecho de ser multimonedado. Esta particularidad se resume a un tema de configuración, es decir, que el software permita que el cliente opere con varias monedas, según su necesidad. En términos generales, resulta ventajoso tener la posibilidad de efectuar transacciones en múltiples divisas. Los saldos pueden ser expresados en la moneda base, la cual es definida por los usuarios. Como elemento negativo, se puede mencionar que en la mayoría de estos sistemas las operaciones se realizan por separado en cada uno de los subsistemas, aunque hay algunos sistemas ERP como el Dynamics que centralizan todos los aspectos relacionados con la Multimonedado en un módulo. Se puede decir que después de hacer un profundo análisis de los ERP expuestos

anteriormente se han logrado comprender las principales características y funcionalidades que permiten a estos sistemas ser multimonedados. Se ha delimitado el conocimiento básico que respalda la utilización de diversas monedas en un sistema de gestión, permitiendo identificar la moneda contable, moneda original, tasas de cambio y la reevaluación de cuentas, como algunos de los aspectos importantes que deben considerarse para la propuesta de solución.

#### 1.4. Temas a considerar, en CedruX, para el tratamiento de la multimonedada y sus especificaciones.

En este epígrafe se resumen las características de la multimonedada, así como las especificaciones que supone su tratamiento dentro de la solución a desarrollar.

##### 1.4.1. Especificaciones generales para multimonedada.

1. Las transacciones quedarán registradas en la moneda original y en la moneda contable, utilizando para ello el factor de conversión y la tasa de cambio de cada moneda. Para aplicar este postulado se solicitará la moneda original y se aplicará la siguiente fórmula:

$$\frac{\text{Importe moneda original}}{\text{Factor de conversión}} * \text{Tasa de cambio} = \text{Importe moneda contable}$$

2. En este mismo sentido, las recuperaciones que se realicen deberán imprimir información en moneda contable, por defecto y permitir recuperar en moneda original en todos los casos.
3. Una alternativa que no se puede descartar es la recuperación en moneda contable de todas las transacciones que fueron realizados en otras monedas.
4. Las configuraciones y transacciones sobre multimonedada y doble moneda deben quedar registradas a nivel de unidad contable. (8)

##### 1.4.1.1. Particularidades para la implementación de la multimonedada en la solución.

Se evalúa, desde el punto de vista funcional la repercusión de la multimonedada dentro del producto que se construye.

###### Multimonedada

- Define las monedas a utilizar en el sistema.
- De las monedas definidas se especifica cuál es la moneda contable.
- Gestionará las tasas de cambio y los factores de conversión de las monedas con relación a la moneda contable definida.

- Revaluación de partidas monetarias de cuentas.
- Expresar la división de las monedas y billetes de manera que puedan ser contabilizados en cantidades físicas de cada uno.

### Contabilidad

- Define las cuentas de Ingresos y Gastos que servirán para registrar las variaciones por tipo de cambio y las diferencias por aproximación de las conversiones de moneda original a moneda contable.
- Permite la captación de los comprobantes de operaciones en moneda original, aplicando la fórmula expuesta en el epígrafe postulado general de este documento.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes (Comprobantes de Operaciones, Balance de Comprobación de Saldos, Utilidad Acumulada, Mayor y Submayores).
- Emite comprobante de operaciones en moneda contable y original.

### Caja y Banco

- Permite la captación de las operaciones de Caja y Banco en moneda original, aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Elaboración de los flujos de efectivo en moneda contable o en moneda original.
- Emite comprobante de operaciones en moneda contable y original.

### Cobros y Pagos

- Permite la captación de los derechos y obligaciones en moneda original, aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Elaboración de los análisis en moneda contable o en moneda original.
- Emite comprobante de operaciones en moneda contable y original.

### Inventario

- Permite la captación de los movimientos de inventarios en moneda original (Recepción por compras), aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento. Si se ha definido moneda alternativa, se deberá mostrar dos campos de importe identificados por tipos de monedas.
- El precio unitario de los productos se formará por el importe en moneda contable, que deberá agrupar la parte de la moneda contable y el equivalente en moneda contable de la moneda original, si la hubiera.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Los derechos y obligaciones se traspasarán a Cobros y Pagos con la misma filosofía del punto no. 1 de este módulo.
- Elaboración de los análisis en moneda contable o en moneda original.
- Emite comprobante de operaciones en moneda contable y original.

#### Costos y Procesos

- Las operaciones de traspaso y ajuste al costo deben aplicarse tanto a los importes en moneda contable como a los importes en moneda original.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Elaboración de los análisis en moneda contable o en moneda original.
- Emite comprobante de operaciones en moneda contable y original.

#### Activos Fijos

- Permite la captación de los movimientos que generan entradas de Activos Fijos (compras) en moneda original aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento. Si se ha definido moneda alternativa, se deberá mostrar dos campos de importe identificados por tipos de monedas en la compra y venta de estos medios. El resto de los movimientos se realizará a moneda contable.
- El precio unitario de los productos se formará por el importe en moneda contable, que deberá agrupar la parte de la moneda contable y el equivalente en moneda contable de la moneda original, si la hubiera.
- La depreciación o amortización, según proceda, se calculará sobre el importe en moneda contable.

- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Emite comprobante de operaciones en moneda contable y original.
- Los derechos y obligaciones se traspasarán a Cobros y Pagos con la misma filosofía de lo planteado en la primera consideración de este módulo.
- Elaboración de los análisis en moneda contable o en moneda original.

### Facturación

- Define en la matriz de precios, los valores por tipo de moneda a utilizar. Si se decide aplicar dualidad monetaria, en la matriz de precio deberá definirse el componente en la moneda alternativa, aplicando el factor de conversión y la tasa de cambio indirecta.
- Permite la captación de los elementos necesarios para facturar inventarios, Activos Fijos y Servicios aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento. Si se ha definido moneda alternativa, se deberá mostrar dos campos de importe identificados por tipos de monedas en la venta de estos medios.
- La facturación mostrará los importes por tipos de moneda.
- Emite comprobante de operaciones en moneda contable y original.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.
- Los derechos y obligaciones se traspasarán a Cobros y Pagos aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento.
- Elaboración de los análisis en moneda contable o en moneda original.

### Capital humano

- Trabaja en moneda contable, cuando ésta coincide con la moneda funcional utilizada para el pago de salario.
- Cuando se paga estimulación en moneda diferente a la contable, deberá especificarse y se aplicará la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento.
- Emite comprobante de operaciones en moneda contable y original.



- Los derechos y obligaciones se traspasarán a Cobros y Pagos aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento, si se usa una moneda diferente a la contable.
- Elaboración de los análisis en moneda contable o en moneda original.

#### Planificación Empresarial y Presupuestaria

- Permite la elaboración de los indicadores del Plan que son importes en moneda original, aplicando la fórmula *Importe moneda contable* que se especifica en el punto No. 1 del epígrafe “Especificaciones generales” (1.4.1) de este documento. Si se ha definido moneda alternativa, se deberá mostrar dos campos de importe identificados por tipos de monedas.
- El plan se formará a partir de la moneda contable, que deberá agrupar la parte de la moneda contable y el equivalente en moneda contable de la moneda original, si la hubiera.
- Permite la recuperación de la información en moneda contable o en moneda original de las transacciones y demás reportes.

#### **1.4.2. Dualidad monetaria.**

La dualidad monetaria está asociada a la existencia de dos monedas que coexisten y comparten legalmente las funciones del dinero en la economía nacional. Se utilizan como medio de pago, medida de valor (expresión de los precios de los bienes y servicios que se comercializan, las deudas y los registros de valores económicos) y como medio de atesoramiento (depósitos en los bancos y efectivo).<sup>12</sup>

En Cuba, el peso cubano (CUP) es la moneda oficial y la que se reconoce como moneda base de los registros económicos y el peso convertible (CUC) es la que comparte la situación de dualidad monetaria.

##### **1.4.2.1. Breve reseña.**

Como una consecuencia de la crisis económica de los años ´90, el comienzo de la dualidad monetaria en la economía cubana puede ubicarse en el año 1993. En agosto de este año se despenalizó la tenencia de divisas para las personas naturales y se le otorgó la posibilidad de tener cuentas bancarias en dólares americanos. También se iniciaron las ventas minoristas en dólares de entidades estatales a la población. Alrededor de esta fecha, la dolarización empezó igualmente a extenderse en el sector empresarial. La

---

<sup>12</sup> Banco Central de Cuba y Departamento Económico del CCPC. (abril - junio de 2008). Material de Estudio. *La Dualidad Monetaria*. La Habana: Editora Política.

dolarización fue parcial. Los salarios, las pensiones y una parte mayoritaria del ahorro se mantuvieron en pesos cubanos. Muchas empresas e instituciones siguieron operando en pesos cubanos.

Diez años después, en julio de 2003, se convirtieron a pesos convertibles (CUC) las cuentas bancarias de las empresas e instituciones del estado que se habían dolarizado. En octubre de 2004 las tiendas minoristas que habían estado fijando sus precios y comercializando en dólares empezaron a hacerlo en pesos convertibles. Al año siguiente, con la revaluación de las tasas de cambio de las monedas nacionales, se incentivó la desdolarización también de las cuentas bancarias de la población.

Durante diez años la dualidad monetaria estuvo definida por la circulación del peso cubano y el dólar estadounidense. Actualmente, las dos monedas que ocupan la casi totalidad de las transacciones internas son el peso cubano y el peso convertibles, ambas emitidas por el Banco Central de Cuba. (9)

#### 1.4.2.2. Especificaciones generales para dualidad monetaria (4)

El registro contable de esta situación se comporta de dos formas:

1. Cuando el hecho económico que se registra sólo utiliza el peso convertible (CUC).
2. Cuando el hecho económico que se registra utiliza ambas monedas, pesos cubanos (CUP) y convertibles (CUC) y que generalmente está asociada a la compra venta de bienes de consumo o duraderos.

En el *primer caso*, el registro se realiza bajo el principio de expresar la transacción en pesos cubanos (CUP) utilizando la tasa de cambio vigente entre ellos. Esta operación es típica de la funcionalidad de multimoneda.

*La NCC<sup>13</sup> No. 6 Operaciones con moneda extranjera o pesos convertibles, en los párrafos del 13 al 15, establece:*

“Transacciones en moneda extranjera o pesos convertibles”

Las transacciones en moneda extranjera o en Pesos Convertibles deben registrarse, al reconocerlas, en la moneda para el registro contable, aplicando al importe en moneda extranjera o Peso Convertible el tipo de cambio vigente entre la moneda para el registro contable y la moneda extranjera o Peso Convertible en la fecha de la transacción.

---

<sup>13</sup> NCC: Norma Cubana de Contabilidad. En el caso citado se refiere a la No. 6 “Operaciones con Moneda Extranjera o Pesos Convertibles”.

- Si la entidad decide conservar la moneda origen de la transacción y utiliza cuentas de conversión, el saldo de esta cuenta se registrará como ingresos o gastos por operaciones con moneda extranjera, según proceda, al emitir los Estados Financieros.

*En cada fecha de cierre del balance:*

- Las partidas monetarias en moneda extranjera o Pesos Convertibles deben anotarse aplicando el cambio de cierre o vigente al final del día.
- Las partidas no monetarias que se asientan según su costo histórico expresado en una moneda extranjera o Peso Convertible deben anotarse aplicando el tipo de cambio en la fecha de la transacción.
- Las partidas no monetarias que se asientan según su valor razonable expresado en una moneda extranjera o Pesos Convertibles deben anotarse aplicando el tipo de cambio vigente en el momento que se determinaron los valores.

Las diferencias de cambio que se derivan de la liquidación de partidas monetarias o de presentar las partidas monetarias de la entidad utilizando tipos de cambio distintos de los tipos a los que esas partidas se anotaron inicialmente durante el ejercicio, o se presentaron en Estados Financieros anteriores, deben reconocerse como ingresos o gastos del ejercicio en que se producen. “

En el *segundo caso*, el registro se realiza bajo el principio de expresar la transacción en pesos cubanos (CUP) utilizando la tasa de cambio vigente entre ellos, pero demanda un proceso de control y seguimiento en ambas monedas lo que produce, de manera general, los efectos siguientes:

- La emisión y tratamiento de documentos primarios con dos tipos de monedas, cuando lo común es que en cada documento sólo aparezca una sola moneda. De esta forma, al procesar una factura o un informe de recepción, aparecerán estos elementos, introduciendo un proceso de multimoneda en el procesamiento de cada documento que presente estas características.
- Control de dos precios para cada inventario, cuando lo clásico es que los productos adquiridos en una moneda distinta a la de registro, su precio quede expresado en la moneda base o de registro utilizando la tasa de cambio.
- Recuperación y tratamiento de la información de manera diferenciada por tipo de monedas que intervienen en el proceso de dualidad monetaria. Esta incidencia acentuada por las peculiaridades del proceso de planificación en el país.

- Incidencia en la gestión comercial a partir de las disposiciones vigentes lo que provoca que un artículo que tenga su precio compuesto por ambas monedas pueda ser comercializado en una de ellas o en ambas.
- En el caso de los bienes que se insumen y que pasan a formar parte de los costos de producción y o servicios se necesita un tratamiento de manera diferenciada por tipo de monedas que intervienen en el proceso de dualidad monetaria.
- Necesidad de identificar en la solución la presencia de procesos asociados a la dualidad monetaria. La experiencia que se tienen con este tema a partir de su adición a sistemas nacionales certificados es que la implementación de esta funcionalidad afecta el diseño de la BD, las interfaces de usuarios y la lógica del negocio de varios módulos del sistema.

### 1.4.2.3. Implicaciones de la dualidad monetaria en la solución.

A partir de estas consideraciones técnicas y de la experiencia de conocer estas funcionalidades en soluciones nacionales certificadas, las operaciones de dualidad monetaria se manifiestan en los módulos de CedruX de la manera siguiente:

#### Multimoneda

Debe precisarse la moneda base de registro, el resto de las monedas y sus reglas de conversión. Deberá precisarse la existencia de dualidad monetaria y la segunda moneda que interviene en este proceso.

#### Contabilidad (Incluye Estados Financieros)

Las cuentas que representan activos y pasivos monetarios<sup>14</sup> deben aparecer identificadas por tipos de monedas en algún nivel y contendrán el saldo de las operaciones en la moneda original y en la moneda de registro para posibilitar los procesos de revaluación en el momento que se decida.

En el caso de las cuentas que representan activos y pasivos no monetarios, será necesaria su separación en los componentes por monedas, en los elementos de gastos para facilitar el cálculo de los costos.

#### Caja y Banco

Por ser operaciones que identifican activos monetarios deben aparecer identificadas por tipos de monedas en algún nivel y contendrán el saldo de las operaciones en la moneda original y en la moneda de registro para posibilitar los procesos de revaluación y conciliación en el momento que se decida.

---

<sup>14</sup> Activos y Pasivos monetarios, conocidos también como Partidas monetarias son unidades monetarias mantenidas en efectivo, así como activos y pasivos que se van a recibir o pagar, mediante una cantidad fija o determinable de unidades monetarias.

### Cobros y Pagos

Por ser operaciones que identifican activos y pasivos monetarios deben aparecer identificadas por tipos de monedas en algún nivel y contendrán el saldo de las operaciones en la moneda original y en la moneda de registro para posibilitar los procesos de revaluación y conciliación en el momento que se decida.

### Inventario

Son operaciones de activos no monetarios que en presencia de dualidad monetaria demandan un registro y seguimiento de los precios en ambas monedas y los procesos de recuperación que no viole la hipótesis del precio de adquisición.

### Facturación

Son operaciones de activos no monetarios que en presencia de dualidad monetaria demandan un registro y seguimiento de los precios en ambas monedas y de la gestión comercial con una o ambas monedas, según sea el caso.

Ejemplo: Compras centralizadas de la Educación y la Salud se realizan en ambas monedas y su comercialización a las unidades presupuestadas se realiza en pesos cubanos (CUP).

### Activos Fijos Tangibles e Intangibles

Son operaciones de activos no monetarios que en presencia de dualidad monetaria pueden o no demandar un registro y seguimiento de los precios en ambas monedas y de la gestión comercial con una o ambas monedas, según sea el caso. La decisión puede impactar en el cálculo de la depreciación y amortización de éstos medios.

### Costo y procesos

Son operaciones de acumulación que en presencia de dualidad monetaria demandan un registro y seguimiento de los gastos en ambas monedas y de la determinación del costo con una o ambas monedas, según sea el caso. Depende de lo que se logre en el resto de los módulos.

### Capital Humano (Incluye Nóminas)

En el caso de la nómina, hoy sólo se relaciona con los sistemas de estimulación en pesos convertibles.

### Planificación Empresarial y Presupuestaria

En presencia de dualidad monetaria demandan un registro y seguimiento de la gestión comercial con una o ambas monedas, según sea el caso.

### Generador de reportes

Debe permitir la recuperación de la información bajo la presencia de dualidad monetaria y de multimonedas.

### **1.4.3. Reevaluación de cuentas**

La reevaluación de cuentas surge como consecuencia directa de la existencia de monedas distintas a la contable durante el proceso de registro de las operaciones económicas, por esta razón se incluye dentro de los aspectos analizados para concebir la solución que recibirá la multimoneda en CedruX.

#### **1.4.3.1. Consideraciones para la reevaluación de cuentas**

En la NCC No. 6 Operaciones con moneda extranjera o pesos convertibles, en su modificación No. 1 se plantea que, en lo referido a:

##### Las transacciones en moneda extranjera o pesos convertibles:

1. Las transacciones en moneda extranjera o en Pesos Convertibles deben registrarse, al reconocerlas, en la moneda para el registro contable, aplicando al importe en moneda extranjera o Peso Convertible el tipo de cambio vigente entre la moneda para el registro contable y la moneda extranjera o Peso Convertible en la fecha de la transacción.
  - a) Si la entidad decide conservar la moneda origen de la transacción y utiliza cuentas de conversión, el saldo de esta cuenta se registrará como ingresos o gastos por operaciones con moneda extranjera, según proceda, al emitir los Estados Financieros.
2. En cada fecha de cierre del balance:
  - a) Las partidas monetarias en moneda extranjera o Pesos Convertibles deben anotarse aplicando el cambio de cierre o vigente al final del día.
  - b) Las partidas no monetarias que se asientan según su costo histórico expresado en una moneda extranjera o Peso Convertible deben anotarse aplicando el tipo de cambio en la fecha de la transacción y
  - c) Las partidas no monetarias que se asientan según su valor razonable expresado en una moneda extranjera o Pesos Convertibles deben anotarse aplicando el tipo de cambio vigente en el momento que se determinaron los valores.

3. Las diferencias de cambio que se derivan de la liquidación de partidas monetarias o de presentar las partidas monetarias de la entidad utilizando tipos de cambio distintos de los tipos a los que esas partidas se anotaron inicialmente durante el ejercicio, o se presentaron en Estados Financieros anteriores, deben reconocerse como ingresos o gastos del ejercicio en que se producen.

### **Utilización de una moneda de presentación <sup>15</sup>distinta de la moneda para el registro contable**

#### Conversión a la moneda de presentación:

4. La entidad presenta sus Estados Financieros en la moneda de presentación. Si la moneda de presentación difiere de la moneda para el registro contable de la entidad, ésta deberá convertir sus resultados y situación financiera a la moneda de presentación exigida.
5. Para convertir sus resultados y situación financiera a la moneda de presentación exigida, la entidad utilizará cualquiera de los siguientes procedimientos:
  - I. Procedimiento No. 1
    - a) Los activos y pasivos se convertirán a la tasa de cambio de cierre en la fecha de presentación de los estados financieros.
    - b) Los ingresos y gastos se convertirán a la tasa de cambio de la fecha de cada transacción, puede utilizarse un tipo de cambio aproximado de los cambios existentes en las fechas de las transacciones.
    - c) Todas las diferencias de cambio que se produzcan como resultado de lo anterior, se reconocerán como un componente separado del patrimonio neto (también denominado diferencias de conversión).
  - II. Procedimiento No. 2
    - a) Todos los importes (es decir, activos, pasivos, partidas del patrimonio neto, gastos e ingresos) se convertirán a la tasa de cambio de cierre correspondiente a la fecha del balance.

#### Información a revelar:

6. La entidad revelará la siguiente información:
  - a) El importe de las diferencias de cambio reconocidas en el resultado del ejercicio, con excepción de las procedentes de los instrumentos financieros que se valoren al valor razonable con cambios en el resultado del ejercicio, y

---

<sup>15</sup> Moneda de presentación: Moneda en que se presentan los Estados Financieros.

- b) Las diferencias netas de cambio clasificadas como un componente separado del patrimonio neto, así como una conciliación entre los importes de estas diferencias al principio y al final del ejercicio.
7. Cuando la moneda de presentación sea diferente de la moneda para el registro contable, este hecho se pondrá de manifiesto, revelando además la identidad de la moneda para el registro contable, así como la razón de utilizar una moneda de presentación diferente.
8. Cuando se haya producido un cambio en la moneda para el registro contable, ya sea de la entidad que informa o de algún negocio significativo en el extranjero, se revelará este hecho, así como la razón de dicho cambio.

Cuando la entidad presente sus Estados Financieros en una moneda que sea diferente de la moneda de presentación, sólo podrá calificar a sus Estados Financieros como conformes con las Normas Cubanas de Información Financiera, si cumplen con todos los requerimientos de cada Norma que sea de aplicación y con cada interpretación de esas Normas que sea aplicable, incluyendo el método de conversión establecido en esta norma.

**Al cerrar el ejercicio económico, hay que tener en cuenta las siguientes consideraciones:**

Efectivo en Caja y banco.

- La moneda nacional se valora a su valor nominal y la moneda extranjera, al tipo o tasa de cambio bancario que esté en vigor al momento de cada transacción y al cierre de cada ejercicio económico, por lo que es preciso ajustar los saldos existentes al valor de cotización o tipo de cambio vigente en dicha fecha.
- Los beneficios o las pérdidas resultantes, de lo establecido en el párrafo anterior, forman parte del resultado del ejercicio económico.

Inversiones a Corto Plazo o Temporales

- En el momento de formularse los estados financieros se valuarán por el valor neto de realización o cotización, según corresponda. si se trataran de títulos y valores en moneda extranjera, se aplicará el criterio de conversión de la moneda de origen a la moneda nacional o de curso legal, afectando los ingresos o pérdidas resultantes, el resultado del ejercicio económico.
- Los depósitos a plazo fijo se valoran a su valor nominal. Al cierre del ejercicio los no vencidos se valoran por su valor nominal más los intereses devengados hasta esa fecha y de tratarse de



depósitos a plazo fijo en moneda extranjera deberán convertirse a la moneda de curso legal, según el criterio expresado anteriormente.

#### Efectos y Cuentas por Cobrar

- Las Cuentas y Efectos por Cobrar en moneda extranjera se valuarán al tipo de cambio vigente al momento de su origen y al vigente en la fecha de cierre del ejercicio. Cuando esto signifique una disminución del valor pactado originalmente, la pérdida afectará el resultado del ejercicio económico.

#### Pasivos Circulantes y a Largo Plazo

- Los Pasivos en moneda extranjera (Efectos y Cuentas por Pagar) se valúan de acuerdo a la cotización de la moneda de que se trate, al tipo de cambio vigente en la fecha del ingreso de los fondos. Si al cierre del período contable la deuda permaneciera impagada, la misma se ajustará a la cotización o tasa de cambio vigente en la fecha de cierre del ejercicio. Cuando esto signifique un aumento del valor pactado originalmente se considerará como pérdida, afectándose el resultado del ejercicio económico.

#### **1.4.4. ¿Qué se propone para el tratamiento de la multimonedas en CedruX?**

El estudio y análisis de lo planteado hasta este momento muestra que la solución que se propone es revolucionaria y novedosa. Capaz de flexibilizar las características multientidad y multimonedas hasta el punto de que el cliente que utilice CedruX puede configurar el nomenclador de monedas de acuerdo a sus requerimientos específicos. Esto implica que cada entidad trabajará solo con las monedas que son de su interés.

En lo referente a funcionalidades, clasifica como un sistema que permite el trabajo con la multimonedas, lo que se traduce como la posibilidad de utilización y registro de importes y consumo en cualquiera de las monedas conocidas y que deben ser definidas en el sistema. De igual manera, se debe permitir:

- Definir con cuales de todas las monedas existentes en el mundo puede trabajar una entidad determinada.
- Especificar la moneda base (contable) con la que trabajara la entidad. Igualmente se puede especificar si existe moneda alternativa, en consonancia con las especificaciones que incorpora la dualidad monetaria.

- Brinda un alto nivel de gestión para las tasas de cambio establecidas para cada moneda, además de las asociadas a cada cliente.
- Desglosar los tipos de billetes y monedas en los que se puede desglosar físicamente una moneda.
- Reevaluar las cuentas con saldo monedas extranjeras, en función de la tasa que se encuentre en uso.

### 1.5. Ingeniería de requerimientos

La Ingeniería de Requerimientos está enfocada en la definición de lo que se desea producir, de ahí que cumpla un papel esencial en el proceso de producción de software. Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones que deben ser cumplidas al construir un producto, siempre teniendo en cuenta la información suministrada por los clientes.

Esta ingeniería comprende, básicamente, todo el proceso y tareas involucradas en la Captura, Definición y Validación de requisitos o requerimientos. Al decir de Pressman la: “Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”. (10) En consecuencia de lo que se puede definir como el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar la información que se necesita para desarrollar el sistema. La meta de la ingeniería de requerimientos es entregar una especificación de requisitos de software completa y correcta.

Entre las principales ventajas de esta práctica se pueden mencionar:

- Permite gestionar las necesidades del proyecto en forma estructurada.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados.
- Disminuye los costos y retrasos del proyecto
- Mejora la calidad del software
- Mejora la comunicación entre equipos
- Evita rechazos de usuarios finales

Como se menciona al inicio de este epígrafe, el proceso de especificación de requisitos se puede dividir en tres grandes actividades (11):

1. Elicitación de requisitos.<sup>16</sup>
2. Definición de requisitos.
3. Validación de requisitos.

La captura y gestión de requisitos tiene un impacto decisivo en el ciclo de vida del proyecto. Aunque no es una ciencia exacta, existen técnicas y métodos a seguir tanto para la captura de los requisitos como para su representación (12), algunas de las cuales se abordarán más adelante. El resultado de esta actividad es un “contrato” entre el cliente, los usuarios y el equipo que desarrollará el proyecto.

La realización del documento de requisitos (ERS<sup>17</sup>) es un paso fundamental para facilitar el éxito del proyecto.

### 1.5.1. Requerimientos

Según la IEEE<sup>18</sup>, un requerimiento es: una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. (13)

Los requerimientos representan las necesidades de los usuarios y los objetivos del sistema. Estos varían con frecuencia dentro del ciclo de vida del proyecto, por diferentes razones. Entre las que se pueden mencionar; no captar correctamente la información, cambios que surgen en las empresas o en las perspectivas del cliente y cambios en el mercado. Es necesario destacar que las variaciones en los requisitos repercuten en el tiempo de desarrollo y así, sucesivamente en todas las actividades que se encuentran relacionadas, incluso pueden llegar a perjudicar la calidad del software.

Los requerimientos se pueden clasificar en: funcionales y no funcionales. (14)

---

<sup>16</sup> Elicitación: Definido como la acción de obtener de manera provocada información de una fuente. En el ámbito de Ingeniería de Requerimientos es análogo a la captura de requisitos.

<sup>17</sup> ERS: Especificación de Requisitos de Software

<sup>18</sup> IEEE: Siglas del Instituto de Ingenieros Eléctricos y Electrónicos, (Institute of Electrical and Electronics Engineers). Fundada en 1984, es una asociación técnico-profesional mundial dedicada a la estandarización. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática e ingenieros en telecomunicación.

*Requerimientos funcionales:* Condición o capacidad que el sistema debe cumplir. Definen las funciones que el sistema será capaz de realizar. Estos deben definir y describir los procesos y actividades que componen el sistema completo.

*Requerimientos no funcionales:* Propiedades o cualidades que el producto debe tener. Son aquellos que no están relacionados directamente con los procesos que definen el sistema, se enfocan en las características adicionales, tales como seguridad, hardware, software, soporte, portabilidad, rendimiento, interfaz externa.

### 1.5.2. Obtención de los requerimientos

Los requerimientos surgen como ideas o conceptos que pueden originarse para dar respuesta a la necesidad de reemplazar un sistema existente o crear uno nuevo. En el mundo se han experimentado diversas maneras de interactuar con los interesados para llegar a unos requerimientos completos y que satisfagan en la mayor magnitud posible al cliente. Hay muchas técnicas para identificar los requerimientos, incluyendo las siguientes:

- Talleres estructurados.
- Sesiones de tormentas de ideas.
- Entrevistas.
- Desarrollo de aplicaciones en grupo.
- Cuestionarios.
- Observación de campo.
- Mapa de conceptos.
- Revisión de la documentación técnica.
- Análisis de mercado.
- Ingeniería inversa.
- Simulaciones.
- Prototipos.

Mientras se trabaja en las tareas de obtención de requisitos los analistas deben reunir la mayor cantidad de información posible, buscando que lo que se defina, satisfaga en el mayor grado al cliente. Durante este proceso de búsqueda y absorción de información se utilizaron, fundamentalmente, de las siguientes técnicas:

*Entrevistas y cuestionarios*

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado mientras que el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema.

Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él.

Las preguntas deben ser de alto nivel, para obtener información sobre aspectos globales del problema del usuario y soluciones potenciales. Con frecuencia, se utilizan preguntas abiertas para descubrir sentimientos, opiniones y experiencias generales, o para explorar un proceso o problema. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado y no están influenciadas por el conocimiento de la solución. Las mismas pueden ser enfocadas en un elemento del sistema, tales como usuarios, procesos y productos.

El éxito de esta técnica combinada, depende de la habilidad del entrevistador y de su preparación para la misma. Los analistas necesitan ser sensibles las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales.

#### JAD (Joint Application Development/Desarrollo conjunto de aplicaciones)

Es una técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes (15). Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas, llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. (12)

Esta técnica se torna ventajosa frente a las entrevistas tradicionales, ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados. (12)

#### Brainstorm (Lluvia o tormenta de Ideas)

Técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. (15) Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. (12)

Su objetivo es buscar que los involucrados en un proyecto desarrollen su creatividad. Entre los principios de la lluvia de ideas se pueden referenciar:

- Aplazar el juicio y no realizar críticas, hasta que no agoten las ideas, ya que actuaría como un inhibidor. Se ha de crear una atmósfera de trabajo en la que nadie se sienta amenazado.
- Cuantas más ideas se sugieren, mejores resultados se conseguirán: “la cantidad produce la calidad”. Las mejores ideas aparecen tarde en el periodo de producción de ideas, será más fácil que encontremos las soluciones y tendremos más variedad sobre la que elegir.
- La producción de ideas en grupos puede ser más efectiva que la individual.
- Tampoco debemos olvidar que durante las sesiones, las ideas de una persona, serán asociadas de manera distinta por cada miembro, y hará que aparezcan otras por contacto.

Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como éste. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros. (12)

### Revisión de documentación

Esta técnica depende de la información almacenada por las entidades acerca de los procesos y términos que se manejan dentro de la misma. Las organizaciones tienen información referente a sus procesos, los modelos o informes necesarios para el desarrollo de la misma. Pueden ser además videoconferencias y otros, dependiendo del grado de automatización que posean. Para el desarrollo del proyecto ERP-Cuba fue necesario revisar documentación de la gran mayoría de las entidades cubanas, para comprender la mayor parte de los procesos que realizan. Se examinaron las soluciones existentes, utilizando toda la información real, entre la que pueden mencionarse artículos de promoción, descripciones del producto, manuales de usuario, entre otros. Los materiales anteriormente mencionados se revisaron antes de tener un contacto directo con los especialistas para adquirir una idea inicial de los procesos. Luego de las sesiones de entrevistas fueron leídos nuevamente y analizados para limar detalles y aclarar dudas.

Hay que destacar que la revisión de documentos no es del todo, segura, pues existe desactualización en la información e incluso, hay procesos que ocurren de manera distinta a lo que se registra en la bibliografía; por esta razón se recomienda aplicar esta técnica vinculada a otras para lograr un resultado efectivo.

### Prototipos

Los prototipos permiten al desarrollador crear un modelo del software que debe ser construido. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos definiendo las áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un “diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo, el cual es evaluado por cliente y usuarios y se utiliza para refinar los requerimientos del software a ser desarrollado.

Existen principalmente dos tipos de prototipos:

- Prototipo rápido (o concept prototipe): El prototipado rápido es un mecanismo para lograr la validación pre-compromiso. Se utiliza para validar requerimientos en una etapa previa al diseño específico. En este sentido, el prototipo puede ser visto como una aceptación tácita de que los requerimientos no son totalmente conocidos o entendidos antes del diseño y la implementación. El prototipo rápido puede ser usado como un medio para explorar nuevos requerimientos y así ayudar a “controlar” su constante evolución.
- Prototipo evolutivo: Desde una perspectiva diferente, todo el ciclo de vida de un producto puede ser visto como una serie incremental de detallados prototipos acumulativos.

Útiles para comunicar, discutir y definir las ideas entre los diseñadores y las partes responsables; puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos hasta un complejo software. Permiten llegar a niveles muy detallado en las especificaciones del futuro software, concretando sus elementos, centrándose en la representación de aquellos aspectos del software que serán visibles para el cliente o usuario final. Resulta oportuno puntualizar, que esta es una técnica que puede usarse durante la obtención de los requerimientos y también en las actividades de validación.

### **1.5.3. Especificación de requerimientos**

La especificación permite documentar los requisitos identificados. Obtener especificaciones de requerimientos con la calidad suficiente es fundamental para asegurar un software que se corresponda

con las necesidades del cliente, convirtiéndose en una tarea que requiere de mucha experiencia y habilidades. (16)

Las técnicas utilizadas durante la especificación de requisitos son las siguientes:

#### Glosarios de términos.

Consiste en registrar el conocimiento que se va adquiriendo sobre el dominio del problema y compartirlo con todos los participantes en el proyecto. Para ello se establece un vocabulario con términos propios. Estos se coleccionan y definen los conceptos más relevantes y críticos para el sistema.

#### Plantillas o patrones

Esta técnica tiene por objetivo describir los requisitos mediante el lenguaje natural pero de una forma organizada. Una plantilla es una tabla con una serie de campos predefinidos que el equipo de desarrollo va cumplimentando, usando para ello el lenguaje del usuario. Las plantillas eliminan parte de las confusiones propias del lenguaje natural al estructurar la información; cuanto más organizada esté, menos ambigüedad ofrece.

### **1.5.4. Validación de requerimientos**

Al caracterizar los requerimientos, siempre se enfatiza en la necesidad de que deben ser especificados por escrito y posibles de verificar o probar. La validación de requerimientos, es la etapa final de este proceso y se nutre de las fases anteriores, examinando las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos con la mayor precisión. Su preocupación es mostrar que los requisitos realmente reflejan las necesidades de los usuarios. Buscan identificar problemas de: omisión, inconsistencia y/o ambigüedad. (17)

Se verifica que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares y criterios de calidad establecidos para el proceso, el proyecto y el producto. El resultado de esta actividad es el Documento de Requerimientos, que será uno de los artefactos rectores en las restantes etapas de la Ingeniería de Software, por lo que se torna importante realizar de manera correcta esta actividad pues un error que se descubra en fases posteriores puede conducir a gastos excesivos.

Existen diversos tipos de enfoques que posibilitan la verificación de los requerimientos. De las estudiadas en (18), ha sido factible aplicar las que siguen:

#### Revisiones

Como su nombre lo indica, está basada en revisar las especificaciones de los requerimientos. Es una técnica manual, cuyo principio es analizar nuevamente los requerimientos para emitir una versión final de



los mismos. Su objetivo es detectar posibles conflictos (defectos y/o contradicciones) en el producto que se examina, de forma que puedan plantearse alternativas y los participantes aumenten su conocimiento sobre el producto en cuestión.

### Construcción de prototipos

Esta técnica consiste en crear un modelo en base a los requerimientos ya establecidos. Existen dos tipos de prototipos, los desechables y los evolutivos.

Los prototipos desechables concretan un modelo para demostrar el cumplimiento de los requerimientos definidos con lo cual se puede reducir los costos involucrados en el ciclo de vida del desarrollo. Una vez hecha esta evolución el prototipo se desecha.

Los prototipos evolutivos consisten en desarrollar una implementación inicial, que se presenta a los involucrados y se refina hasta llegar a un desarrollo de sistema adecuado. Su objetivo es desarrollar un prototipo rápido para que el cliente valore los resultados y recomiende cambios de una manera oportuna.

En el caso de este proyecto, fue importante su uso durante la obtención, pero se puso en práctica dentro de esta actividad ya que ayuda a la identificación de errores en los requisitos propuestos.

## **1.6. Diseño de software**

El diseño es la primera etapa técnica del proceso de Ingeniería del Software, consiste en producir un modelo o representación técnica del software que se va a desarrollar, es el proceso sobre el que se asienta la calidad del software.

El diseño de software es un proceso iterativo a través del cual se traducen los requisitos una representación del software, se representa a un alto nivel de abstracción, un nivel que se puede seguir hasta requisitos específicos de datos, funcionales y de comportamiento. En el contexto de la ingeniería del software el diseño se centra en cuatro áreas importantes de interés: datos, arquitectura, interfaces y componentes. (10)

- Diseño de datos:

Transforma el modelo del dominio de información que se crea durante el análisis en las estructuras de datos que se necesitarán para implementar el software.

- Diseño arquitectónico:

Define las relaciones entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema.

- Diseño componentes:

Transforma los elementos estructurales de la arquitectura del software en una descripción procedimental de los componentes del software.

- Diseño de interfaz

Describe cómo se comunica el software consigo mismo y con su entorno.

### 1.6.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos y restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Describe la jerarquía de subsistemas de diseño que contienen clases del diseño, realizaciones de casos de uso-diseño e interfaces. Sirve de abstracción de la implementación del sistema y es usado como una entrada fundamental de las actividades de implementación. (10)

#### 1.6.1.1. Principios básicos del diseño

Según lo dicho por M.A. Jackson<sup>19</sup>; "El comienzo de la sabiduría para un ingeniero de software es reconocer la diferencia entre hacer que un programa funcione y conseguir que lo haga del modo correcto". Los principios básicos de diseño hacen posible que el ingeniero del software navegue por el proceso de diseño. Cuando los principios de diseño se aplican adecuadamente, el ingeniero del software crea un diseño que muestra los factores de calidad tanto internos (los que tienen importancia para los ingenieros de software) como externos (velocidad, fiabilidad, grado de corrección, usabilidad). (10)

Entre los más importantes principios se encuentran: (20)

- El diseñador debe considerar enfoques alternativos juzgando a cada uno en base a los requisitos del problema, los resultados disponibles y los criterios de calidad interna.
- Se deberían poder seguir los pasos de diseño hasta el modelo de análisis.

---

<sup>19</sup> M. A. Jackson: Michael Anthony Jackson (nacido en 1936) es un científico Británico vinculado al mundo de las computadoras y consultor independiente de computación en Londres, Inglaterra. Ha sido parte de algunas investigaciones de la Compañía AT&T. En 1997 recibió el "Stevens Award" por el desarrollo de métodos de software. Ha desarrollado métodos como el "Jackson Structured Programming (JSP)", que se describe en su libro Principios del Diseño de Programas.

- El diseño no va a reinventar nada que ya esté inventado.
- El diseño debería presentar uniformidad e integración.
- Debe estructurarse para admitir cambios.
- El diseño no es escribir código y escribir código no es diseñar.
- Se debe valorar la calidad del diseño mientras se crea, no después de terminado.

De manera general, estos principios proporcionan las claves para desarrollar la actividad de diseño.

## **1.7. Tendencias tecnológicas actuales a considerar en la solución.**

Este epígrafe resume los conceptos fundamentales tratados durante la concepción del proceso de modelado del Módulo multimonedas de CedruX, en lo referente a tecnologías, metodología de desarrollo y herramientas.

### **1.7.1. Ingeniería del software.**

La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos a través de los que se indica cómo construir técnicamente el software, el proceso que es el fundamento de la Ingeniería de Software y el elemento de unión que mantiene juntas las capas de la tecnología, así como las herramientas, que constituye el soporte automático o semiautomático para el proceso y los métodos.

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación necesaria requerida para desarrollar, operar (funcionar) y mantener dichos programas.

### **1.7.2. Metodología, notación y lenguaje de modelado**

El desarrollo del proyecto ERP-Cuba se emprende bajo excepcionales condiciones, la necesidad de obtener un producto funcional en el menor tiempo posible fue uno de los alicientes para que la dirección del proyecto en coordinación con los arquitectos decidiera tomar experiencia positiva de varias metodologías y proponer un modelo de desarrollo propio, para guiar el proceso de desarrollo del software. A continuación se muestran las particularidades del mismo, así como las ventajas de la aplicación de la notación BPMN<sup>20</sup> para la etapa de modelamiento del negocio y el lenguaje UML<sup>21</sup> para el resto del desarrollo del software como principios básicos de la ingeniería aplicada al entorno del proyecto.

---

<sup>20</sup> BPMN: (Business Process Modeling Notation). Notación para el modelado de procesos de negocio.

### 1.7.2.1. Metodología de desarrollo.

El ERP- Cuba es un proyecto que por la complejidad de sus procesos de negocio, tamaño e integración entre los subsistemas que lo conforman y por la gran cantidad de información que estos almacenan y analizan, requiere de un largo período de elaboración, construcción y coordinación entre los equipos de desarrollo.

La construcción de CedruX ha sido asumida de manera excepcional, sobre todo por el corto tiempo del que se dispone para liberar la primera versión del producto. Este particular, condujo a que se propusiera la utilización de una metodología de desarrollo personalizada, definida por la dirección del proyecto. Esta se nutre de los aspectos positivos y ágiles de varias de las metodologías conocidas dentro del mundo del desarrollo del software (RUP<sup>22</sup>, XP<sup>23</sup>, Orientada a Componentes). El resultado de dicho proceso está definido en el documento *Modelo de Desarrollo orientado a componentes del proyecto ERP Cuba*, del que se resumen los principales aspectos en el anexo # 1 de este trabajo.

En dicho documento se establecen las responsabilidades por rol, la notación, técnica y herramientas a utilizar en las diferentes fases del desarrollo, así como los principales artefactos que deben ser construidos por cada trabajador, en dependencia del momento en que se encuentre el ciclo de vida del software.

En el marco de este trabajo de diploma, teniendo en cuenta lo definido por la metodología de desarrollo, se deben construir los siguientes artefactos: Descripción de procesos de negocio, Modelo conceptual, Registro de requisitos, Especificación de requisitos, Especificación de reportes, Diagrama de clases y Descripción de clases.

### 1.7.2.2. Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (BPMN) es una notación para la gestión de procesos del negocio. Es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo (workflow). BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (Object Management Group). (20)

---

<sup>21</sup> UML: (Unified Modeling Language). Lenguaje unificado de modelado

<sup>22</sup> RUP: Proceso Unificado de Rational (*Rational Unified Process*)

<sup>23</sup> XP: Extreme Programming

El principal objetivo de BPMN es proveer una notación estándar que sea entendible para todos los involucrados e interesados del negocio (stakeholders), entre los que están los analistas de negocio, los desarrolladores técnicos, los gerentes, administradores del negocio, entre otros.

Define un Diagrama de Procesos de Negocio (BPD<sup>24</sup>), basado en la técnica de “flowcharting” (diagramado de flujos) que ajusta modelos gráficos de operación de procesos de negocio. El modelado de procesos de negocio a menudo comienza capturando las actividades de más alto nivel, para luego profundizar unos niveles más bajos de detalle dentro de diagramas separados. (20)

Este tipo de modelado percibe un aumento en las organizaciones que lo adoptan como estándar pues ayuda a unificar la manera de expresar tanto los conceptos básicos de procesos de negocio como conceptos avanzados de modelado, además de reducir, considerablemente el tiempo en que se realiza la actividad. Algunas de sus ventajas son:

- Ofrece un único diagrama, que cubre todas las etapas del diseño de procesos, desde la concepción general hasta la definición de los modos de operar, es decir el lenguaje de ejecución del proceso (el ejecutable).
- Permite dar una visión detallada del proceso: a través del uso de *hiperlinks* se puede ir especificando desde una visión general hasta el máximo de detalles requerido y viceversa.
- Posibilita diseñar transacciones genéricas, incluyendo el tratamiento de excepciones y las compensaciones necesarias para mantener la consistencia del proceso.

En síntesis, BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

### 1.7.2.3. Unified Modeling Language (UML)

UML, del inglés (Unified Modeling Language) surge debido a la necesidad de estandarizar la manera de representar gráficamente los modelos y posibilitar que los diseños gráficos realizados se pudieran compartir fácilmente entre los involucrados en el proceso de desarrollo.

Es un lenguaje para comunicar las ideas a otros desarrolladores y que sirve de apoyo en los procesos de análisis de un problema. (19)

La definición más universal del término es la que expresa que: UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Este indica

---

<sup>24</sup> BPD: Business Process Diagram.

cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (19)

Es un lenguaje de modelado, independiente de los procesos, por lo que no se considera como una metodología. Define la notación que es utilizada por los métodos para representar los diseños. Puede ser usado con cualquier proceso, a lo largo del desarrollo del ciclo de vida y separado de la tecnología de implementación. Está constituido por un conjunto de diagramas que proporciona un estándar a analistas, clientes y desarrolladores que contribuye al desarrollo del sistema ya que cada modelo está orientado a algún involucrado en específico.

Desde finales de 1997 este lenguaje de modelado orientado a objetos ha sido reconocido como estándar por la OMG <sup>25</sup>(Object Management Group). Es utilizado por las empresas de mayores resultados en el mundo de la construcción del software.

### 1.7.3. Herramientas CASE<sup>26</sup>.

Las herramientas CASE constituyen un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo del sistema de información, completamente o en algunas fases.

---

<sup>25</sup> OMG: La OMG (Object Management Group) es una asociación sin fines de lucro formada por grandes corporaciones, muchas de ellas de la industria del software, como IBM, Apple, Sun Microsystems y Microsoft. Se encarga de la definición y el mantenimiento de estándares para aplicaciones de la industria de la computación. Algunos de los estándares definidos por la OMG son el UML y el CORBA, que ofrece interoperabilidad multiplataforma a nivel objetos de negocios.

<sup>26</sup> CASE: Ingeniería de Software Asistida por Ordenador. (Computer Aided Software Engineering).

Las herramientas CASE ayudan a los gestores y practicantes de la ingeniería del software en todas las actividades asociadas a los procesos de software. Automatizan las actividades de gestión de proyectos, gestionan todos los productos de los trabajos elaborados a través del proceso y ayudan a los ingenieros en el trabajo de análisis, diseño y codificación. Contribuyen a que los resultados del trabajo de un ingeniero de software sean de alta calidad. Al disponer de automatización, contribuye a que el usuario de CASE elabore resultados adicionales y personalizados que no serán fáciles ni prácticos de producir sin el soporte de estas herramientas. Constituyen un complemento importante de las prácticas de ingeniería del software.

La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados lo que garantiza el aumento de la productividad.

### 1.7.3.1. Visual Paradigm

Se utiliza Visual Paradigm en su versión 3.1 como herramienta UML CASE visual, la misma ayuda a construir aplicaciones de manera rápida, económica y con mayor calidad.

Visual Paradigm usa un lenguaje estándar para todo el equipo de desarrollo que facilita la comunicación. Tiene capacidad para la ingeniería directa e inversa en varios lenguajes de programación, de igual manera, existe disponibilidad de múltiples versiones, para cada necesidad.

Es multiplataforma lo que favorece los principios de independencia tecnológica que se fomentan actualmente en el país. De gran utilidad para la generación de código fuente en PHP<sup>27</sup>, tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML, de manera instantánea, creando de forma simple toda la documentación.

Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros de equipo. (20)

## 1.8. Conclusiones

Una vez finalizado el estudio que se presenta en este capítulo es posible concluir que el tema multimonedado en Cedux constituye una solución que está acorde a las tendencias internacionales en torno al desarrollo de este tipo de software. En su concepción se tienen en cuenta las particularidades de la

---

<sup>27</sup> PHP: Hypertext Pre-processor (pre-procesador PHP de hipertexto). Lenguaje de programación de libre licencia empleado comúnmente para la creación de páginas web dinámicas.

economía cubana en lo referido al tratamiento de la multimonedas y los requerimientos de la dualidad monetaria, identificando el impacto de ambos conceptos en la solución a desarrollar. La valoración crítica de estos factores hace que se presente un producto donde se centralizan las funciones en un subsistema para brindar servicios al resto de los subsistemas. En general, se propone una solución compacta, organizada, adaptable y funcional sobre los temas referidos a la multimonedas, teniendo en cuenta las especificaciones que, para el caso cubano, genera la existencia de la dualidad monetaria. Los beneficios de esta concepción se proyectan, sobre todo, a la eficiencia y economía (de tiempo y recursos) que reporta integrar en un paquete, conceptos que son de uso colectivo dentro de la solución.

De igual manera, comprende un bosquejo sobre la utilización de metodologías, lenguajes, notación, herramientas CASE para la obtención de una solución eficaz que gestione las funcionalidades de la multimonedas.



## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.**

### **2.1. Introducción**

Este capítulo recoge los principales conceptos tratados dentro del dominio del problema propuesto. Se realiza una descripción de los requerimientos funcionales que debe cumplir el sistema que se desarrollará, los cuales son el resultado del estudio realizado en relación al tema de la multimonedas y su presencia en la solución de CedruX. Se muestran también, las relaciones de los conceptos más importantes a través de un modelo conceptual, que junto a la especificación de requisitos que se genera como parte de este proceso, constituirán las principales guías para el posterior diseño del sistema.

### **2.2. Análisis del entorno del problema**

A los efectos ingenieriles, en el proceso de desarrollo de software debe definirse si es posible modelar el negocio o si hay que remitirse a la realización de un modelo de dominio.

El subsistema Multimonedas, engloba operaciones, que en su mayor parte, se realizan de forma manual y para el momento en que se necesitan. Son tareas que se ejecutan, pero no están compuestas por un flujo identificado de acciones a cumplir, sino que complementan la realización de otros procesos. Sirven de soporte funcional y simplifican la ejecución de otros eventos. En la propuesta de solución, se centraliza este conjunto de operaciones en un subsistema, para organizar y optimizar el trabajo del sistema contable en general, brindando la posibilidad de tener la información concentrada en pos de gestionar estos temas de una manera más eficiente y capaz de agilizar el trabajo de quienes precisen de estos datos.

Si se parte de que un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para la organización, sus inversores o sus clientes. El análisis anterior indica que no es posible identificar los procesos a automatizar, así como los involucrados en cada una de las operaciones. Esto muestra la necesidad de realizar modelo conceptual, con excepción de reevaluación de cuentas. Esta es una operación que se incluye en este subsistema porque es consecuencia directa de la presencia de varias monedas y es un proceso claramente identificado, con una secuencia definida de acciones para su realización. Su descripción y modelado se muestra a continuación.

#### **2.2.1. Proceso reevaluación de cuentas**

La reevaluación de cuentas es una operación que permite conocer el saldo de las cuentas que operan en moneda diferente a la contable, en un momento determinado. Esta actividad se realiza de forma manual y

supone una serie de pasos que se describen en la Figura 2.2. Consta de dos acciones principales: Calcular reevaluación y revertir reevaluación. La primera referida a la obtención del saldo actual en base a la tasa vigente, obteniendo las diferencias en relación con el saldo original para registrarlas en cuentas de ingreso o gastos, según corresponda. Por su parte, revertir reevaluación es el proceso inverso del mencionado con anterioridad. Una descripción mas detallada de este proceso de negocio puede encontrarse en los documentos *ERP-ARQ-MULT -Descripción proceso Reevaluación de cuentas* y *ERP-ARQ-MULT -Descripción proceso Revertir reevaluación*, que se encuentran entre los entregables generados como parte de las tareas cumplimentadas para la confección de este trabajo de diploma, así como en la descripción de requisitos de software, cuyo compendio se expone en este documento.

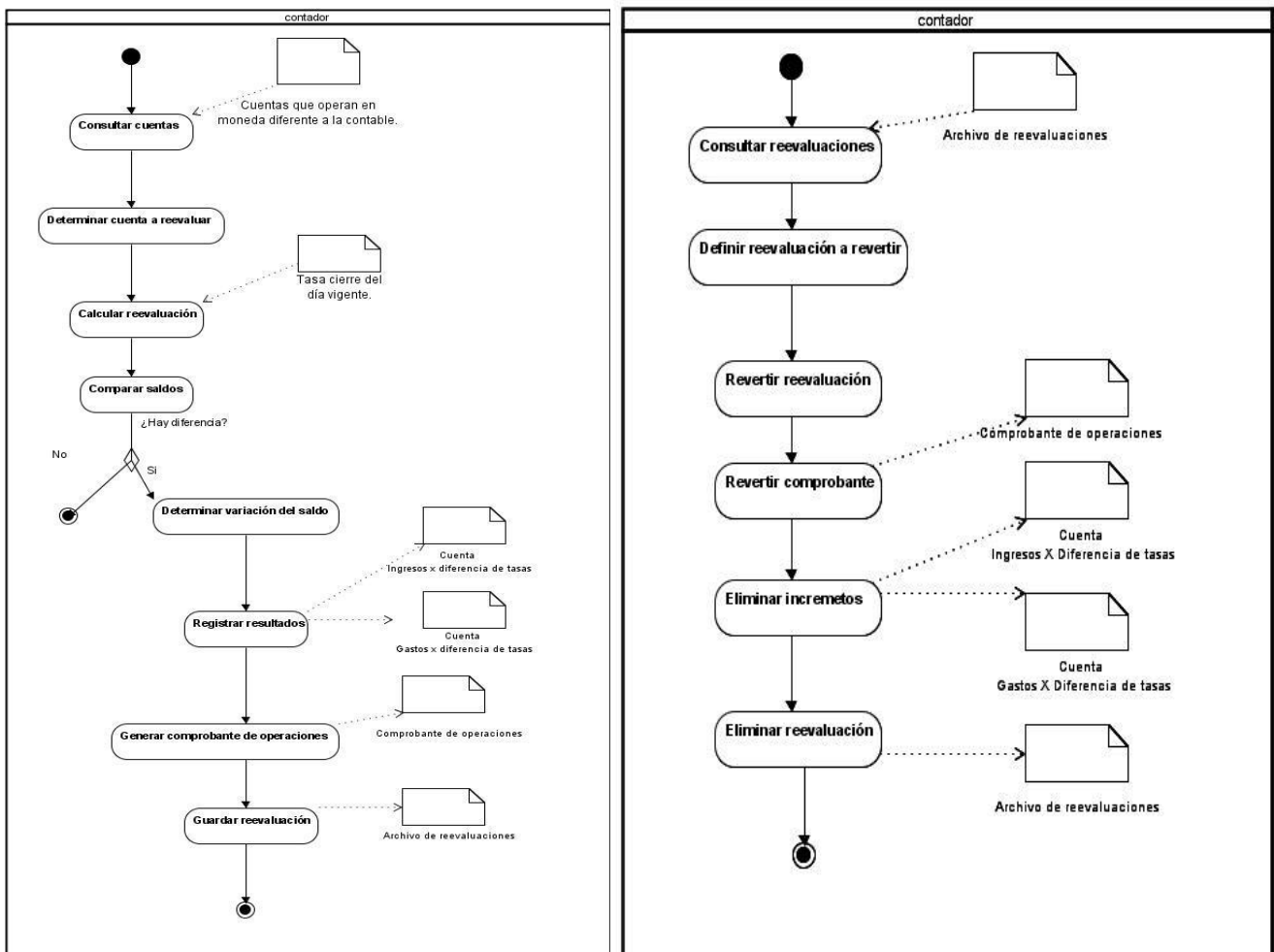


Figura 2.1 -Descripción del proceso de reevaluación de cuentas: calcular y revertir reevaluación

### 2.2.2. Modelo conceptual

El Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. No representa términos relacionados con los componentes de software sino que se enfoca en clases conceptuales del dominio del problema y conceptos del mundo real.

Una *clase conceptual* puede ser una idea o un objeto físico (símbolo, definición y extensión).

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. (21)

La finalidad de esta actividad es la de estipular una especificación del dominio del problema y los requerimientos desde la perspectiva de la clasificación por objetos y desde el punto de vista de entender los términos empleados en el dominio. Para descomponer el dominio del problema hay que identificar los conceptos, los atributos y las asociaciones del dominio que se consideran importantes. El resultado puede expresarse en un modelo conceptual, el cual se muestra gráficamente en un grupo de diagramas que describen los conceptos (objetos).

### 2.2.3. Estrategia para la identificación de conceptos (19)

La meta es crear un modelo conceptual de conceptos interesantes o significativos del dominio en cuestión.

Una directriz interesante a la hora de identificar los conceptos es:

“Es mejor exagerar y especificar un modelo conceptual con muchos conceptos refinados que no especificarlo cabalmente”

Un modelo conceptual no es más adecuado si tiene menos conceptos; generalmente sucede lo contrario. Es frecuente omitir conceptos durante la fase inicial de identificación, muchos son descubiertos más tarde, cuando se examinan los atributos o asociaciones e incluso, durante la fase de diseño. Cuando se detecten, habrá que incorporarlos al modelo conceptual.

Un concepto no debe excluirse del modelo, simplemente, porque los requerimientos no indiquen una necesidad evidente o porque carezca de atributos. Es perfectamente válido tener conceptos sin atributos o conceptos con un papel de comportamiento en el dominio en vez de un papel informacional.

Para la obtención de un modelo conceptual, deben tenerse en cuenta todos los términos que suceden dentro del dominio del problema, luego refinar estos términos para identificar los que como conceptos tendrán influencia en el modelo final.

#### 2.2.4. Definición y especificación de los conceptos idóneos para el dominio del problema

En todo problema, existen términos que son importantes dentro del dominio que este representa. En el caso específico en el que se basa esta tesis se han identificado los siguientes:

*Moneda:* Representa el contenedor de todas las monedas.

*Moneda contable:* Expresa las monedas que se definan como contable y/o alternativa. Sobre las cuales se expresa el valor de la mayoría de las transacciones que se realicen.

*País:* Representa el conjunto de todos los países reconocidos por la ONU<sup>28</sup>.

*Tasa:* Refleja todos los datos a tener en cuenta a la hora de definir una tasa de cambio para una moneda en relación a la moneda contable.

*Nomenclador tipo tasa:* Representa el conjunto de tipos de tasa definidos con los que se operan las tasas. Ejemplo: Contratada, que define una asociación con un cliente determinado, para el cual, la tasa de cambio es un valor que no varía durante el período en que está vigente el contrato.

*Cliente-proveedor:* Relativo a los clientes y/o proveedores con los que la organización se relaciona.

*Moneda entidad:* Agrupa las monedas específicas con las que trabaja la entidad. Define una manera más viable de manipular los elementos del Nomenclador de monedas, al definirse solamente las que son de interés primario para cada entidad u organización.

*Desglose:* Referido a la división numérica de cada moneda, expresada en los diferentes tipos de billetes y de monedas que tenga definidos.

*Tipos de billetes y monedas:* Define las diferentes divisiones que tenga asociada cada moneda.

*Cuentas reevaluadas:* Utilizado para actualizar el saldo actual de las cuentas, esto varía en consonancia con la tasa que esté vigente en el momento que se efectúe la revaluación.

*Reevaluación:* Registra los datos de cada reevaluación que se efectúe.

#### 2.2.5. Representación del Modelo conceptual

El modelo conceptual no es una descripción de los componentes del software; representa los conceptos en el dominio del problema en el mundo real. (19)

Entre los elementos que componen este diagrama pueden existir asociaciones, que no es más que una relación entre dos conceptos que indica alguna conexión significativa e interesante entre ellos. (19)

---

<sup>28</sup> ONU: Organización de Naciones Unidas.

Posterior a la definición de los conceptos, resulta necesario identificar las asociaciones que deben existir entre ellos de manera tal que, satisfagan los requerimientos, además de que contribuyen a una mejor comprensión del modelo conceptual.

### 2.2.5.1. Propuesta de modelo conceptual

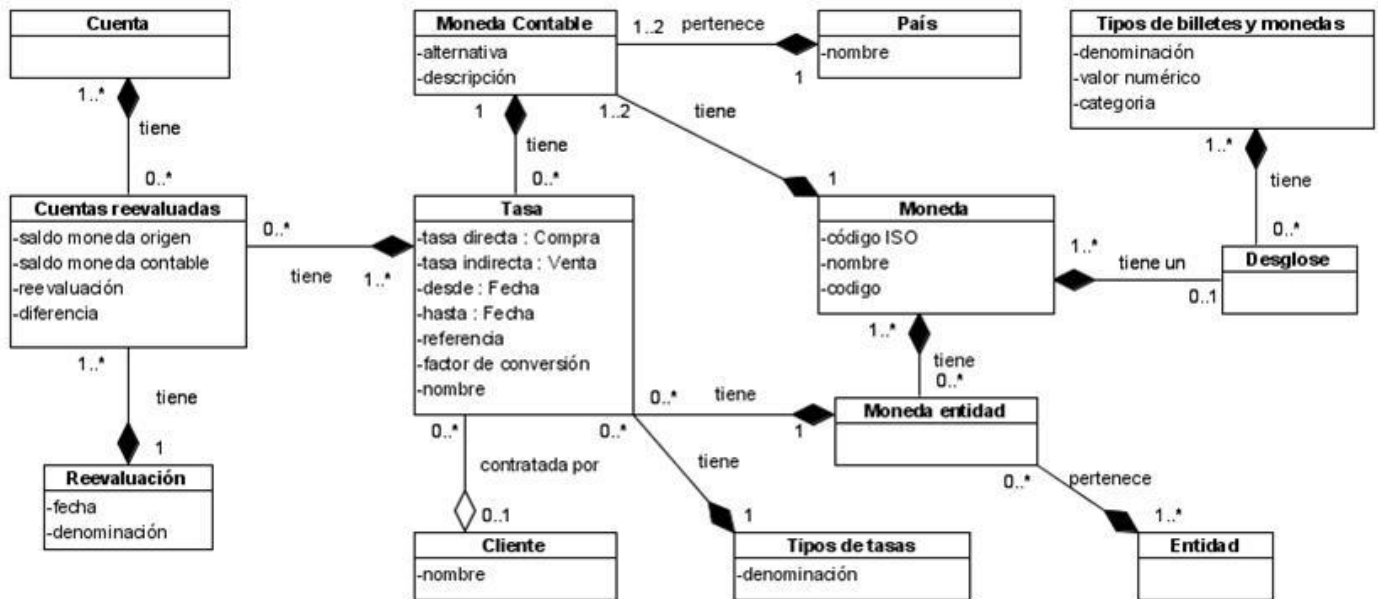


Figura 2.2 - Modelo conceptual: subsistema Multimoneda

### 2.2.5.2. Descripción del modelo conceptual

La descripción del modelo conceptual es importante ya que brinda la información necesaria para el mejor entendimiento del mismo. Uno de los principales aspectos a tener en cuenta dentro de este apartado son las asociaciones que se establecen entre los conceptos que se representan, ellas muestran la forma en que se interactúan y se relaciona en el dominio donde se enmarca el problema. El anterior modelo, describe las siguientes relaciones entre conceptos:

*Moneda contable tiene Moneda:* Relación que se establece entre los conceptos moneda contable y moneda que posibilita conocer qué moneda será la moneda contable y/o la moneda alternativa.

*Moneda pertenece País:* Relación que se establece entre los conceptos moneda contable y país que contiene la información de cuál es el país de la moneda contable y/o alternativa.

*Moneda entidad pertenece entidad:* Relación que se establece entre los conceptos moneda entidad y entidad. Esta permite asociarles a la entidad las monedas con las que trabajará.

*Moneda entidad tiene moneda:* Relación que se establece entre los conceptos moneda entidad y moneda. Esta permite saber cuáles son las monedas con las que trabajara la entidad.

*Tasa tiene Moneda contable:* Relación entre los conceptos tasa y moneda contable que los asocia para definir en base a qué moneda se expresan las tasas de cambio.

*Tasa tiene Tipos de tasas:* Relación entre los conceptos tasa y tipos de tasa que indica de que tipo es la tasa a definir.

*Tasa tiene Moneda por entidad:* Se relacionan los conceptos tasa y moneda entidad con el fin de seleccionar la moneda que se desea convertir en moneda contable.

*Tasa contratada por cliente:* En caso de que el tipo de tasa sea contratada, se relacionan los conceptos tasa y cliente para conocer con quien se realizó el contrato.

*Cuenta reevaluada tiene tasa:* Las cuentas reevaluadas utilizan una tasa en base a la que se efectuó la reevaluación.

*Cuenta reevaluada tiene reevaluación:* Cada cuenta que se haya sido reevaluada debe guardar los datos de la reevaluación a la que pertenece. En este caso la denominación y la fecha en que se realizó la misma.

*Cuenta reevaluada tiene cuenta:* Permite conocer la cuenta original a la que se realizó la reevaluación y en qué cuenta se guardaron los resultados de la misma.

*Desglose tiene una moneda:* Asociación que posibilita seleccionar la moneda que se va a desglosar.

*Desglose tiene tipos de billetes y monedas:* Para asignarle a una moneda, los tipos de billetes y monedas en los que se divide.

### **2.3. Ingeniería de requerimientos aplicada al entorno del problema.**

La evaluación y comprensión del estudio realizado sobre los temas de la Ingeniería de Requisitos, que se presenta en el primer capítulo posibilita identificar las técnicas más factibles a usar durante el proceso de especificación de los requisitos funcionales que deben ser cumplidos con la construcción del subsistema Multimoneda. A continuación se presenta un resumen de las mismas aplicadas al escenario de este trabajo.

#### **2.3.1. Obtención**

Durante la primera fase, los analistas deben reunir la mayor cantidad de información posible, buscando que lo que se defina, satisfaga en el mayor grado al cliente. Este proceso de búsqueda y absorción de información se sirvió, fundamentalmente, de las siguientes técnicas:

- Tormenta de ideas

En su fácil aplicación, descansa la amplia gama de ideas que genera su puesta en práctica. Constituyó una de las técnicas más usadas ya que en el entorno de CedruX, prácticamente a diario se reúne un conjunto de personas relacionadas al producto para socializar y discutir sus ideas libremente. Las dudas generadas después de la tormenta de ideas, fueron aclaradas con la puesta en práctica de otras técnicas de las que se explican.

- JAD (Joint Application Development)

Su traducción al español se refiere a Desarrollo conjunto de aplicaciones; es una técnica que consiste en realizar sesiones conjuntas entre los analistas del sistema y los expertos en el dominio. Fue aplicada debido a que su utilización favorece la obtención de sistemas más enfocados a la realidad, además de la necesidad agilizar la concepción del sistema.

- Revisión de documentos

Técnica que depende de la información almacenada por las entidades. Se realizó un estudio de la documentación existente para identificar estructuras, ideas, hechos y un lenguaje similar que favoreciera el proceso de entendimiento de todos los conceptos para determinar los requisitos asociados a estos.

- Entrevistas

Muy utilizada durante el proceso de obtención de los requisitos debido a lo natural que resulta para los seres humanos el hecho de intercambiar criterios e ideas. Se aconseja planificar la entrevista con anterioridad, así como evaluar los resultados una vez realizada la misma; no pueden ser improvisadas porque estaría expensa a olvidarse detalles importantes.

Tras agrupar la información, se deben especificar por escrito los requerimientos, para este proceso también se han elaborado métodos a modo de guía para facilitar el mismo.

### 2.3.2. Especificación

La especificación permite documentar los requisitos identificados. Obtener especificaciones de requerimientos con la calidad suficiente es fundamental para asegurar un software que se corresponda con las necesidades del cliente, convirtiéndose en una tarea que requiere de mucha experiencia y habilidades. (18)

Las técnicas utilizadas durante la especificación de requisitos son las siguientes:

- Glosarios de términos.

Mientras se van determinando las principales características de lo que se desea construir, van apareciendo términos de obligada referencia para los involucrados en el desarrollo del software. Esto propicia el establecimiento de un vocabulario con términos propios, los que se coleccionan ya que definen los conceptos más relevantes y críticos para el sistema.

- Plantillas o patrones

Esta técnica tiene por objetivo describir los requisitos mediante el lenguaje natural pero de una forma organizada. Una plantilla es una tabla con una serie de campos predefinidos que el equipo de desarrollo va cumplimentando, usando para ello el lenguaje del usuario. Las plantillas eliminan parte de las confusiones propias del lenguaje natural al estructurar la información; cuanto más organizada esté, menos ambigüedad ofrece.

### 2.3.3. Validación

Al caracterizar los requerimientos, siempre se enfatiza en la necesidad de que deben ser especificados por escrito y posibles de verificar o probar. La validación de requerimientos, es la etapa final de este proceso y se nutre de las fases anteriores, examinando las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos con la mayor precisión. Su preocupación es mostrar que los requisitos realmente reflejan las necesidades de los usuarios. Buscan identificar problemas de: omisión, inconsistencia y/o ambigüedad. (17)

Se verifica que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares y criterios de calidad establecidos para el proceso, el proyecto y el producto. El resultado de esta actividad es el Documento de Requerimientos, que será uno de los artefactos rectores en las restantes etapas de la Ingeniería de Software.

Las técnicas puestas en prácticas se caracterizan a continuación:

- Prototipos



De gran utilidad debido a la facilidad que propicia a la hora de comunicar, discutir y definir las ideas entre los involucrados. Permiten detallar las particularidades del futuro software y propicia una visión de lo que se implementará. En CedruX, la elaboración de los prototipos de interfaz de usuarios fue un factor importante para la aprobación final de las especificaciones propuestas, haciendo uso del prototipado de tipo rápido o desechable que es el que se usa comúnmente para las tareas de validación ya que permite mostrar el primer acercamiento visual a lo que se va a construir.

- Revisiones

Con la finalidad de eliminar posibles errores en las descripciones de los requerimientos, se utilizó esta técnica. En el caso de las especificaciones de las funcionalidades implicadas para con la multimonedada fueron sometidas a una Revisión técnica formal, donde participaron las partes involucradas y que concluyó con la firma y aprobación de las mismas.

## 2.4. Especificación de requerimientos para el subsistema Multimonedada

A partir del modelo conceptual propuesto (ver Figura 2.1) y con el apoyo de las técnicas mencionadas en los procesos de obtención, especificación y validación de requerimientos, los que se describen en el epígrafe 2.3.2 son los requerimientos identificados para dar solución a las necesidades enmarcadas en el dominio del problema que soporta el presente trabajo.

### 2.4.1. Funcionalidades identificadas

La primera versión de CedruX incluye un subsistema Multimonedada que es el encargado de controlar y gestionar las operaciones que se realicen en torno a temas monetarios y referentes a la dualidad monetaria. Estas funcionalidades son imprescindibles para el trabajo del resto de los subsistemas. La información que se integra en este subsistema está estructurada de la siguiente manera:

- *Gestionar moneda contable*: Esta función permite definir la moneda contable con la que trabajará el sistema, siempre respetando la concepción de que solo puede ser declarada una única vez y una vez realizada esta operación la misma será invariable. Por su parte, esta funcionalidad también permite realizar operaciones sobre la moneda alternativa con la salvedad de que esta es plenamente configurable aunque solo puede existir como máximo una moneda alternativa en el sistema.
- *Gestionar tasas*: Esta función posibilita gestionar toda la información correspondiente a una tasa de cambio.

- *Gestionar moneda por entidad*: Posibilita gestionar todas las monedas con que una entidad puede hacer transacciones u operaciones.
- *Gestionar reevaluación de cuentas*: Esta funcionalidad posibilita hacer la reevaluación de una o varias cuentas, las cuales se seleccionan del árbol de cuentas, que es la estructura que contiene todas las cuentas registradas en el sistema. La reevaluación es una actualización del saldo de las cuentas que registran operaciones en moneda diferente a la contable, en relación a la tasa vigente.
- *Gestionar monedas*: Es un nomenclador que contiene un grupo de monedas internacionalmente conocidas que serán utilizadas para efectuar revalorizaciones a los estados financieros y llevar el control de los recursos materiales y financieros en diferentes monedas. El sistema tiene predefinidas las monedas que actualmente existen en el mundo pero además permite Adicionar una nueva moneda, Modificar o Eliminar las existentes así como realizar búsquedas dentro del listado de monedas.
- *Gestionar tipo de tasas*: Nomenclador que contiene los tipos de tasa con los que se puede operar en el sistema, estos están ligados a distintas operaciones del sistema y podrán ser utilizados por cualquiera de los subsistemas. El sistema brinda la posibilidad de gestionar y manipular toda la información que se almacene en este nomenclador.
- *Gestionar desglose*: Funcionalidad utilizada para expresar las distintas divisiones internas en las que puede expresar una moneda.
- *Gestionar tipos de billetes y monedas*: Funcionalidad que representa un nomenclador para definir los tipos de billetes y monedas existentes en cuanto a su valor.

#### 2.4.2. Requisitos funcionales

Para nombrar los requerimientos funcionales se tomó como referencia la notación **RF #**. Que responde al tipo de requisito y el número que le corresponde. La especificación de cada uno de los RF<sup>29</sup> se realiza sobre una plantilla definida por la dirección del proyecto, en la que, a modo de tabla se recogen los principales datos implicados.

##### **RF1. Gestionar moneda**

---

<sup>29</sup> RF: Requisito funcional.

Para su mejor comprensión se han agrupado varias funcionalidades relacionadas en un solo requerimiento. Este requisito engloba las operaciones de Adicionar, Eliminar, Modificar y Buscar, siendo el gestor del nomenclador de monedas del sistema.

- RF1.1 Adicionar moneda.

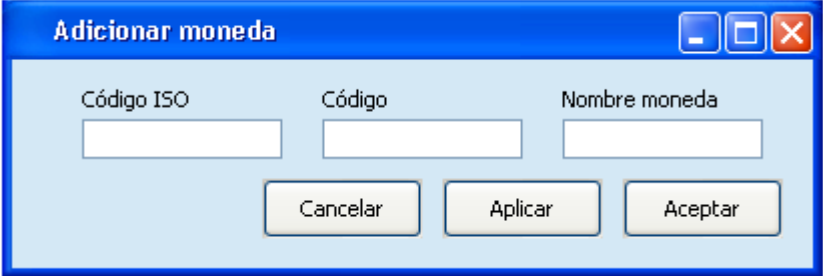
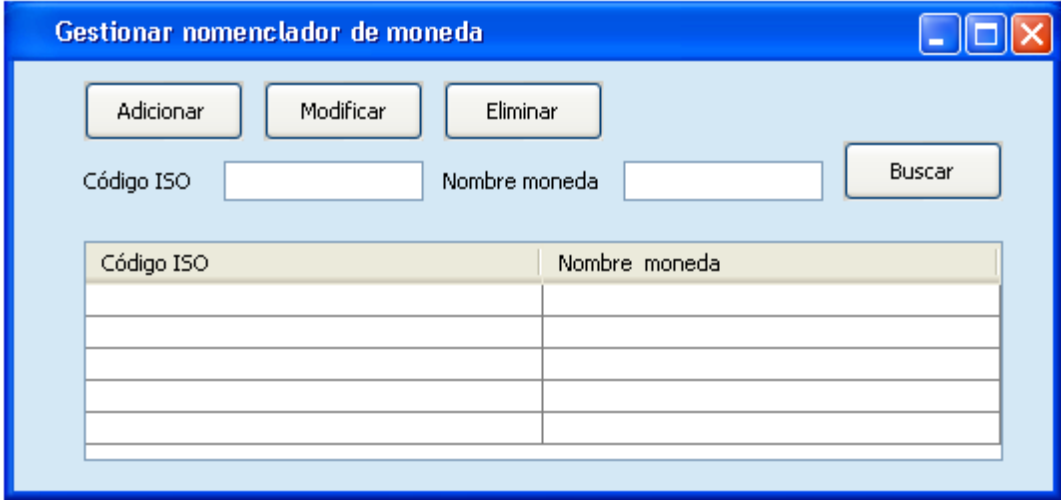
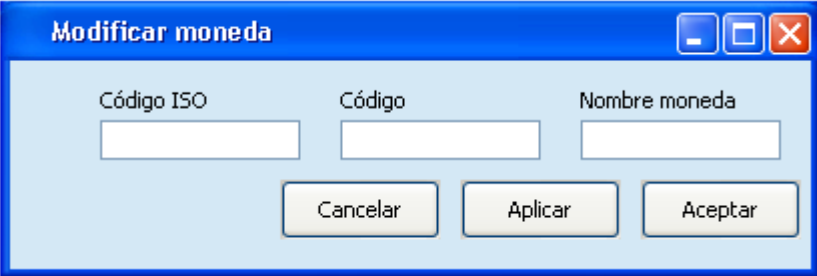
Conceptos tratados	Conceptos	Atributos
		Moneda
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	
Descripción	Permite ingresar datos de la Moneda (código ISO, nombre). Se validan los datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	La moneda no puede existir en el sistema.	
Post-condiciones	Se ha adicionado una nueva Moneda.	
Post-requisito	No procede.	
Prototipo de Interfaz		

Figura 2.3 - Interfaz de adicionar moneda

- RF1.2 Eliminar moneda.

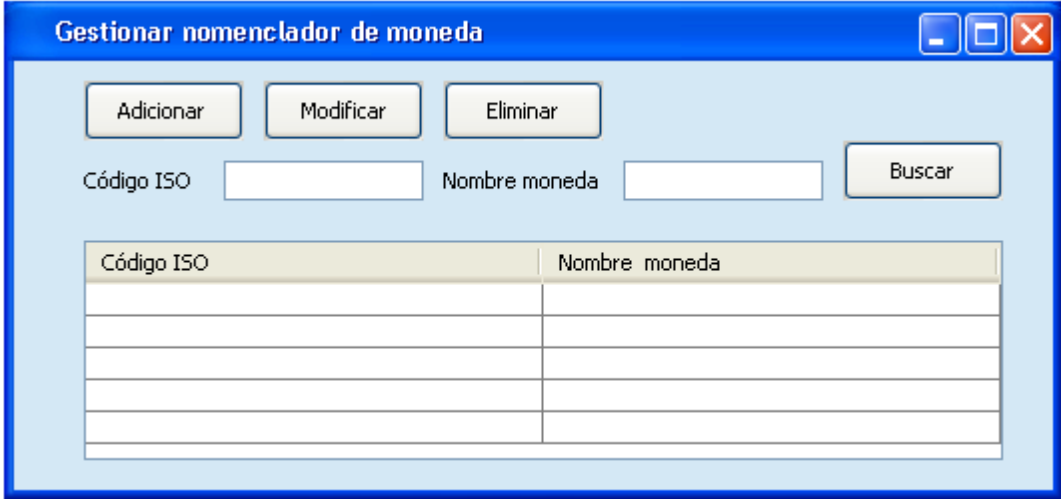
Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO, nombre
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. La moneda debe existir en el sistema.	RF1.1 Adicionar moneda
Descripción	Permite seleccionar la Moneda y eliminarla. Si no hay inconvenientes, se solicita confirmación al usuario. Si se confirma, elimina la moneda o cancela la acción	
Validaciones	Que no haya operaciones registradas con la Moneda que se desea eliminar.	
Post-condiciones	Se ha eliminado la Moneda.	
Post-requisito	No procede.	
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.4 - Interfaz de eliminar moneda</b></p>	

- RF1.3 Modificar moneda.

Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO, nombre
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. La moneda debe existir en el sistema.	RF1.1 Adicionar moneda
Descripción	Permite modificar los atributos de la moneda. Se validan los nuevos datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	La moneda debe existir en el sistema. Los datos modificados deben ser correctos.	
Post-condiciones	Se han modificado los atributos de la Moneda.	
Post-requisito	No procede.	
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.5 - Interfaz de modificar moneda</b></p>	

- RF1.4 Buscar moneda.

<b>Conceptos</b>	Conceptos	Atributos
------------------	-----------	-----------

<b>tratados</b>	Moneda	código ISO, nombre
<b>Precondiciones</b>	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene permisos para realizar la operación. Se listan todas las monedas que existen en el sistema.	RF1.1 Adicionar moneda
<b>Descripción</b>	Mostrar listado de las monedas registradas en el sistema que cumplen con un criterio de búsqueda suministrado por el usuario. Si no se puede realizar la acción notificar al usuario.	
<b>Validaciones</b>	No procede.	
<b>Post-condiciones</b>	Listar las monedas que cumplen con el criterio de búsqueda dado.	
<b>Post-requisito</b>	No procede.	
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.6 - Interfaz de buscar moneda</b></p>	

**RF2. Gestionar moneda contable y/o alternativa.**

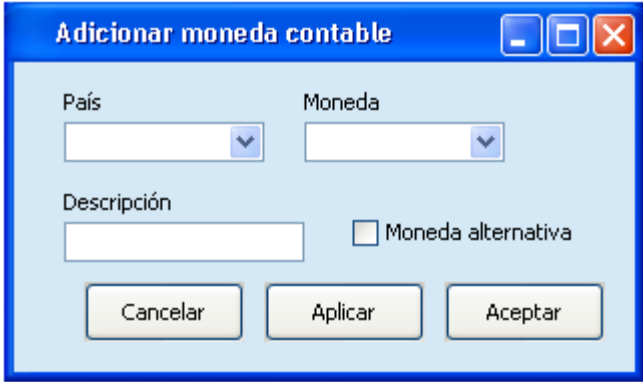
Este RF sustenta las funcionalidades que se pueden realizar sobre la moneda contable y sobre la moneda alternativa del sistema. Para el mismo, existen algunas restricciones que se han definido en relación a las características del producto en su primera versión:

- La moneda contable se puede adicionar una única vez. En caso de error en cuanto a esto, las modificaciones deben realizarse a nivel de datos.
- El sistema puede tener hasta una moneda alternativa.
- La moneda definida como contable para el sistema, es invariable y sus atributos no se pueden modificar.

En equilibrio con estas disposiciones, los requerimientos identificados son:

- RF2.1 Adicionar moneda contable y/o alternativa.

Conceptos tratados	Conceptos	Atributos
	Moneda País Moneda contable	Moneda País Descripción, Moneda alternativa.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	RF1.1 Adicionar moneda
Descripción	Permite escoger la moneda contable o alternativa con la que trabajará la entidad. Se validan los datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	Solamente se podrá adicionar una moneda contable y una moneda alternativa. Aunque esta última puede no existir declarada en el sistema.	
Post-condiciones	Se ha seleccionado la moneda contable o alternativa.	
Post-requisito	No procede.	

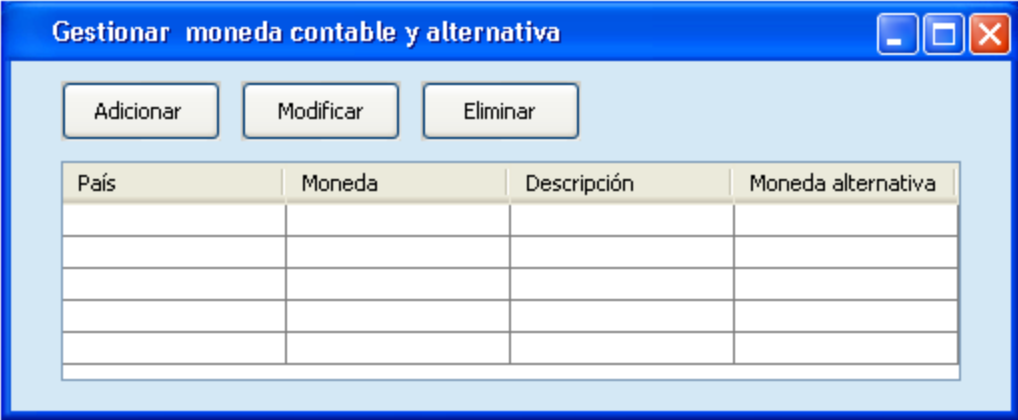
<p>Prototipo de Interfaz</p>	
------------------------------	--

**Figura 2.7 - Adicionar moneda contable.**

- RF2.2 Eliminar moneda alternativa.

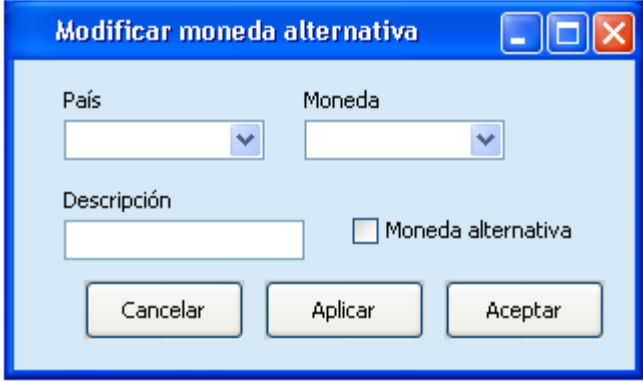
Conceptos tratados	Conceptos	Atributos
	Moneda País Moneda contable	Moneda País Descripción, Moneda alternativa.
Precondiciones	Precondiciones  El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.  La moneda alternativa debe existir en el sistema.	Pre-requisito  RF2.1 Adicionar moneda contable y/o alternativa.
Descripción	Permite seleccionar la moneda alternativa y eliminarla. Si no hay inconvenientes, se solicita confirmación al usuario. Si se confirma , se elimina la moneda alternativa o se cancela la acción	
Validaciones	Que no haya operaciones registradas con la moneda alternativa. La moneda contable no se puede eliminar. Los datos se validan según lo especificado en el modelo conceptual.	
Post-condiciones	Se ha eliminado la moneda alternativa.	



Post-requisito	No procede.
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.8 - Eliminar moneda alternativa.</b></p>

- RF2.3 Modificar moneda alternativa.

Conceptos tratados	Conceptos	Atributos
	Moneda País Moneda contable	Moneda País Descripción, Moneda alternativa.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. La moneda alternativa debe existir en el sistema.	RF2.1 Adicionar moneda contable y/o alternativa.
Descripción	Permite modificar los atributos de la moneda alternativa. Se validan los nuevos datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.	

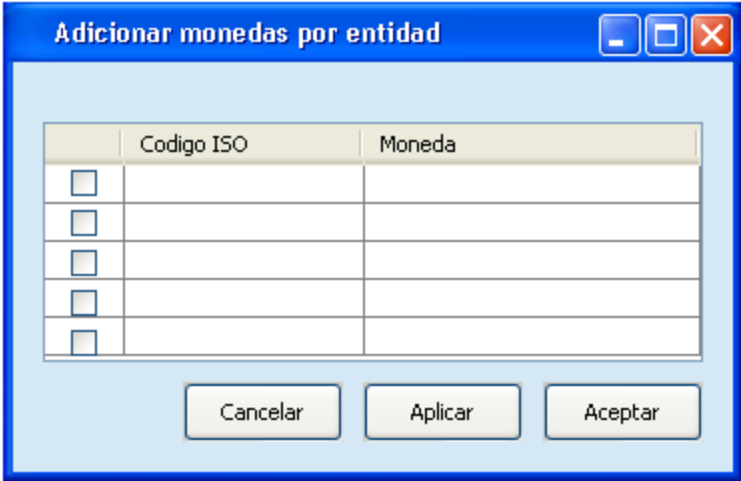
	La moneda contable no se puede modificar.
Post-condiciones	Se han modificado los atributos de la moneda alternativa.
Post-requisito	No procede.
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.9 - Modificar moneda alternativa.</b></p>

**RF3. Gestionar moneda por entidad.**

Requerimiento que responde a la necesidad de especificar las monedas con las que operará cada entidad.

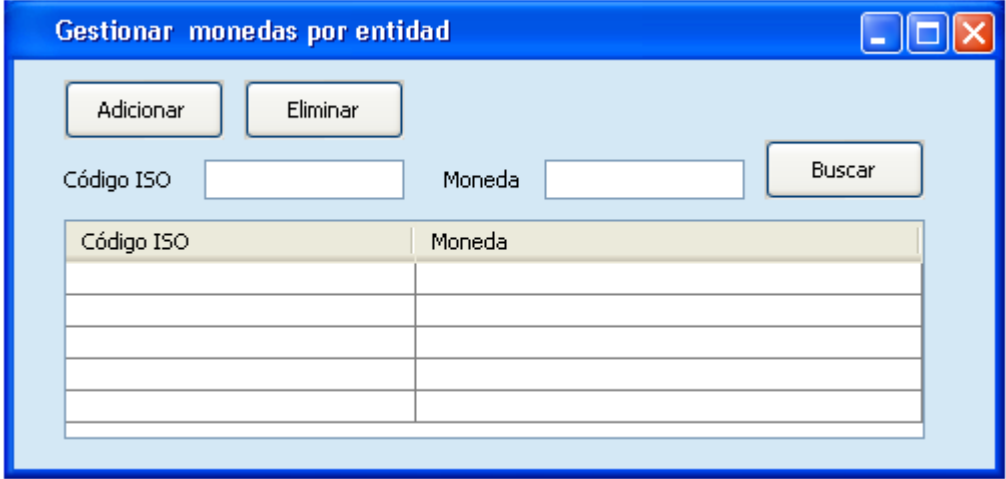
- RF3.1 Adicionar moneda por entidad.

Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO, nombre
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. Deben existir monedas en el Nomenclador.	RF1.1 Adicionar moneda.
Descripción	Lista las monedas disponibles en el Nomenclador de monedas del sistema, permite seleccionar as monedas asociadas a la entidad.	

	Se validan los datos, si son correctos se registran. Si no se le notifican al usuario los errores y se permite corregirlos.
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.
Post-condiciones	Se han asociado las monedas deseadas a la entidad.
Post-requisito	No procede.
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.10 - Adicionar moneda por entidad.</b></p>

- RF3.2 Eliminar moneda por entidad.

Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO, nombre
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. El elemento debe existir en el sistema.	RF3.1 Adicionar moneda por entidad.

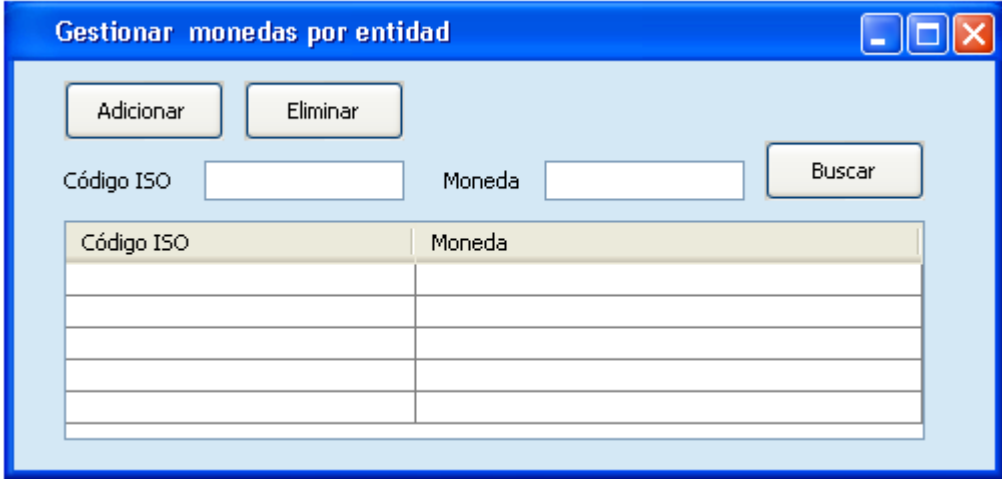
<p>Descripción</p>	<p>Permite seleccionar una moneda y eliminar su asociación con la entidad. Si no existen inconvenientes, el sistema solicita confirmación al usuario. Si se confirma, se elimina el elemento, de lo contrario, se cancela la acción. En cualquiera de los casos, se notifica al usuario.</p>
<p>Validaciones</p>	<p>No se puede eliminar un elemento que este siendo usado. Los datos se validan según lo especificado en el modelo conceptual.</p>
<p>Post-condiciones</p>	<p>Se ha eliminado una moneda de las asociadas a la entidad.</p>
<p>Post-requisito</p>	<p>No procede.</p>
<p>Prototipo de Interfaz</p>	 <p style="text-align: center;"><b>Figura 2.11 - Eliminar moneda por entidad.</b></p>

- RF3.3 Buscar moneda por entidad.

<p>Conceptos tratados</p>	<p>Conceptos</p>	<p>Atributos</p>
	<p>Moneda</p>	<p>código ISO, nombre.</p>
<p>Precondiciones</p>	<p>Precondiciones</p>	<p>Pre-requisito</p>
	<p>El usuario se ha identificado y autenticado ante el sistema y tiene permisos para realizar la operación.</p>	<p>RF3.1 Adicionar moneda por entidad.</p>

	Se listan todos los elementos que existen en el sistema.
Descripción	Mostrar listado las monedas registradas para la entidad, en el sistema que cumplen con un criterio de búsqueda suministrado por el usuario. Si no se puede realizar la acción notificar al usuario.
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.
Post-condiciones	Listar los elementos que cumplen con el criterio de búsqueda dado.
Post-requisito	No procede.

Prototipo de Interfaz



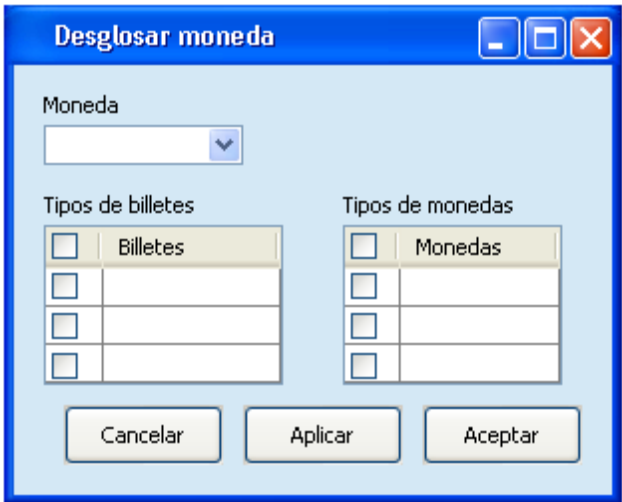
**Figura 2.12 - Buscar moneda por entidad.**

**RF4. Gestionar desglose de moneda.**

Requerimiento que responde a la necesidad de especificar las divisiones en las que puede expresarse cada moneda.

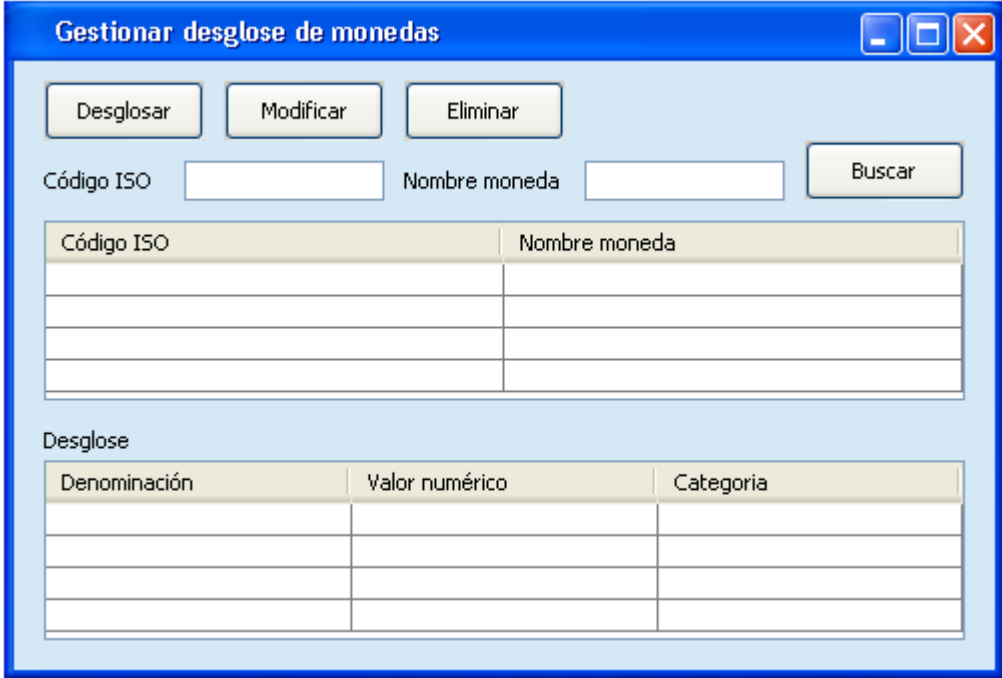
- RF4.1 Desglosar moneda.

Conceptos tratados	Conceptos	Atributos
	Nomenclador de moneda, Tipos de billetes y monedas.	Moneda, tipos de billetes, tipos de monedas
Precondiciones	Precondiciones	Pre-requisito

	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	RF1.1 Adicionar moneda. RF5.1 Adicionar tipo de billete o moneda.
Descripción	Permite asignarle a una moneda los tipos de billetes y monedas en los que esta se desglosará. Se validan los datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.	
Post-condiciones	Se ha desglosado la moneda seleccionada por sus tipos de billetes y monedas.	
Post-requisito	No procede.	
Prototipo de Interfaz	 <p style="text-align: center;"><b>Figura 2.13 - Desglosar moneda.</b></p>	

- RF4.2 Eliminar desglose.

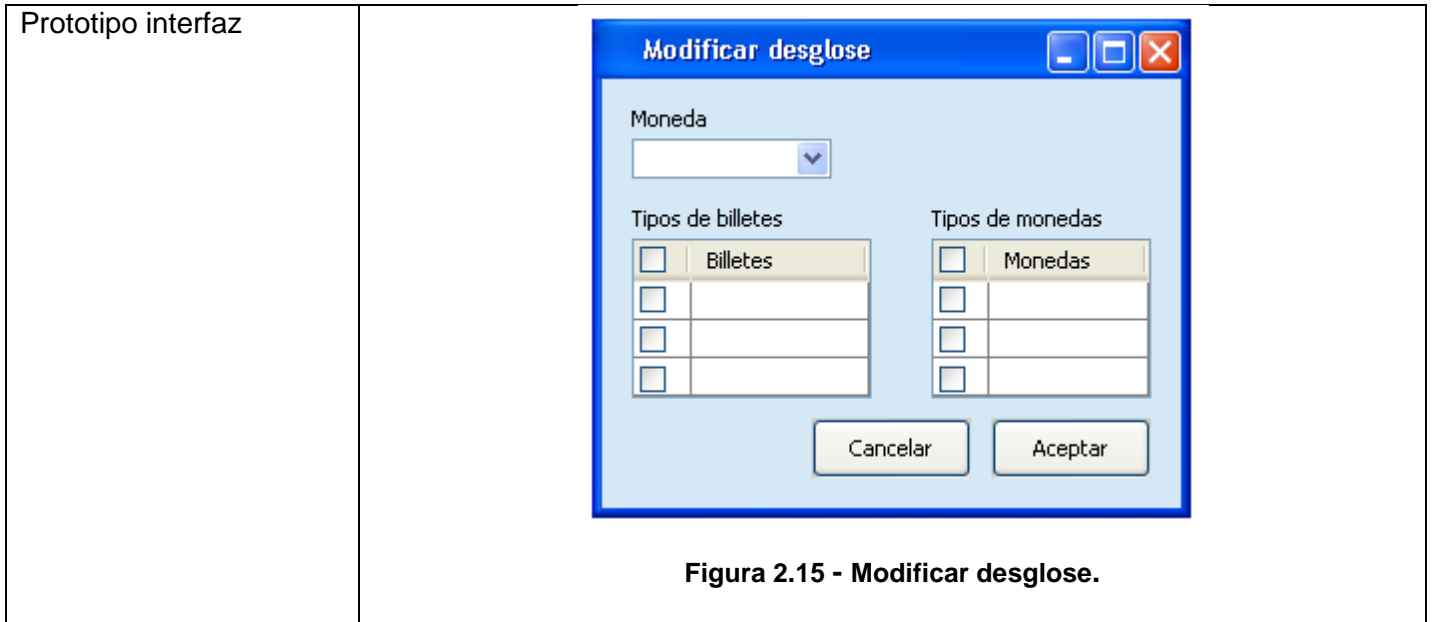
Conceptos tratados	Conceptos	Atributos
	Nomenclador de moneda, Tipos de billetes y monedas.	Moneda, tipos de billetes, tipos de monedas
Precondiciones	Precondiciones	Pre-requisito

	<p>El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.</p> <p>El desglose debe existir en el sistema.</p>	<p>RF4.1 Desglosar moneda</p>
<p>Descripción</p>	<p>Permite seleccionar la moneda y eliminar el desglose.</p> <p>Si no hay inconvenientes, se solicita confirmación al usuario.</p> <p>Si se confirma, se elimina el desglose de la moneda o se cancela la acción.</p>	
<p>Validaciones</p>	<p>Que no haya operaciones registradas con el desglose que se desea eliminar.</p>	
<p>Post-condiciones</p>	<p>Se ha eliminado el desglose.</p>	
<p>Post-requisito</p>	<p>No procede.</p>	
<p>Prototipo interfaz</p>	 <p style="text-align: center;"><b>Figura 2.14 - Eliminar desglose.</b></p>	

- RF4.3 Modificar desglose.

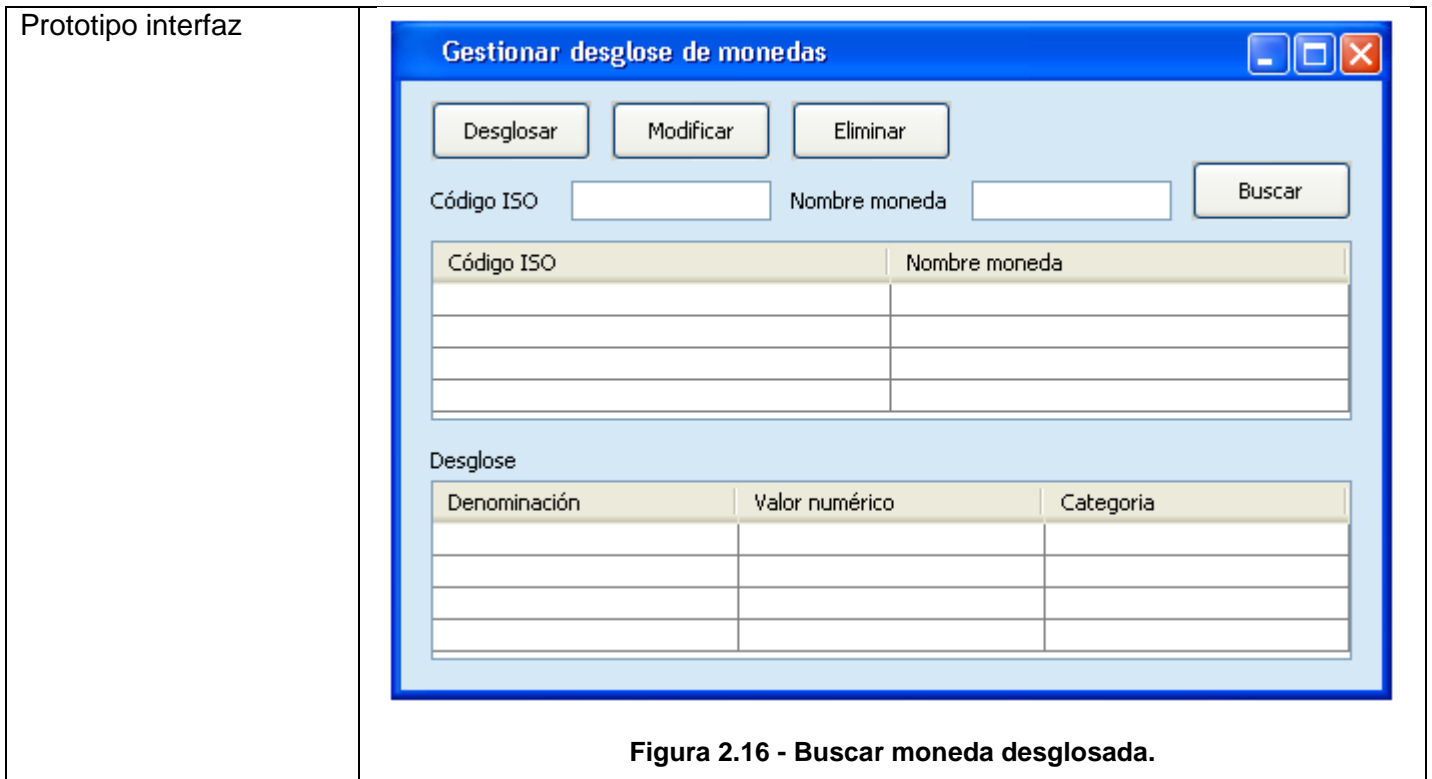
Conceptos tratados	Conceptos	Atributos
	Nomenclador de moneda, Tipos de billetes y monedas.	Moneda, tipos de billetes, tipos de monedas
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. El desglose debe existir en el sistema.	RF4.1 Desglosar moneda
Descripción	Permite modificar los atributos del desglose. Se validan los nuevos datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.	
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.	
Post-condiciones	Se han modificado los atributos del desglose de la moneda.	
Post-requisito	No procede.	





- RF4.4 Buscar moneda desglosada.

Conceptos tratados	Conceptos	Atributos
	Nomenclador de moneda.	Código ISO, Nombre.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene permisos para realizar la operación. Se listan todas las monedas que tengan desglose.	RF4.1 Desglosar moneda
Descripción	Mostrar listado de las monedas desglosadas en el sistema que cumplen con un criterio de búsqueda suministrado por el usuario. Si no se puede realizar la acción notificar al usuario.	
Validaciones	No procede.	
Post-condiciones	Listar las monedas que cumplen con el criterio de búsqueda dado.	
Post-requisito	No procede.	

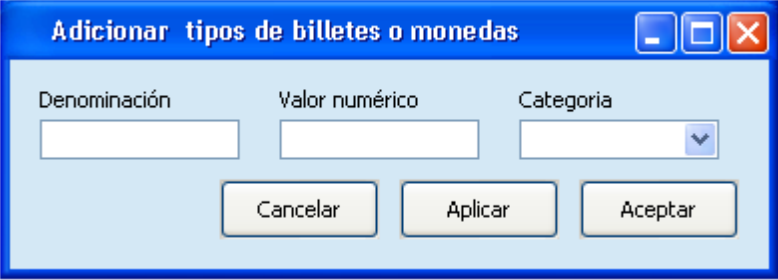


**RF5. Gestionar tipos de billetes y monedas.**

Actúa a modo de nomenclador, gestionando las diferentes divisiones que se definen para cada moneda que se someta al desglose.

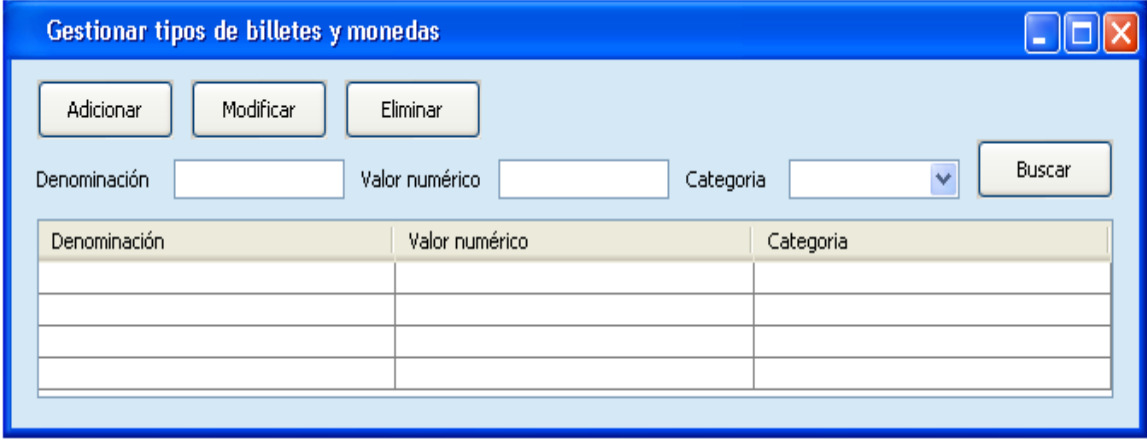
- RF5.1 Adicionar tipo de billete o moneda.

Conceptos tratados	Conceptos	Atributos
	Tipos de billetes y monedas.	Denominación, Valor numérico, Categoría
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	

Descripción	Permite adicionar tipos de billetes y monedas. Se validan los datos. Si son correctos se registran. Si no se notifica al usuario y se permite corregirlos.
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.
Post-condiciones	Se ha adicionado un nuevo tipo de billete o moneda
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.17 - Adicionar tipo de billete o moneda.</b></p>

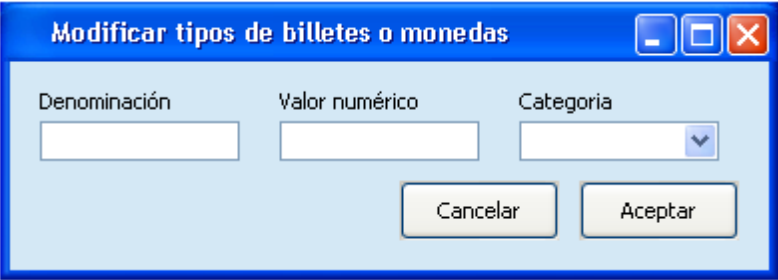
- RF5.2 Eliminar tipo de billete o moneda.

Conceptos tratados	Conceptos	Atributos
	Tipos de billetes y monedas.	Denominación, Valor numérico, Categoría.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. El tipo de billete o moneda debe existir en el sistema.	RF5.1 Adicionar tipo de billete o moneda.
Descripción	Permite eliminar el tipo de billete o moneda que se ha seleccionado. Si no hay inconvenientes, se solicita confirmación al usuario. Si se confirma, se elimina el tipo de billete o moneda o se cancela la acción	
Validaciones	Que no haya operaciones registradas con el tipo de billete o moneda que se desea eliminar.	
Post-	Se ha eliminado el tipo de billete o moneda.	

condiciones	
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.18 - Eliminar tipo de billete o moneda</b></p>

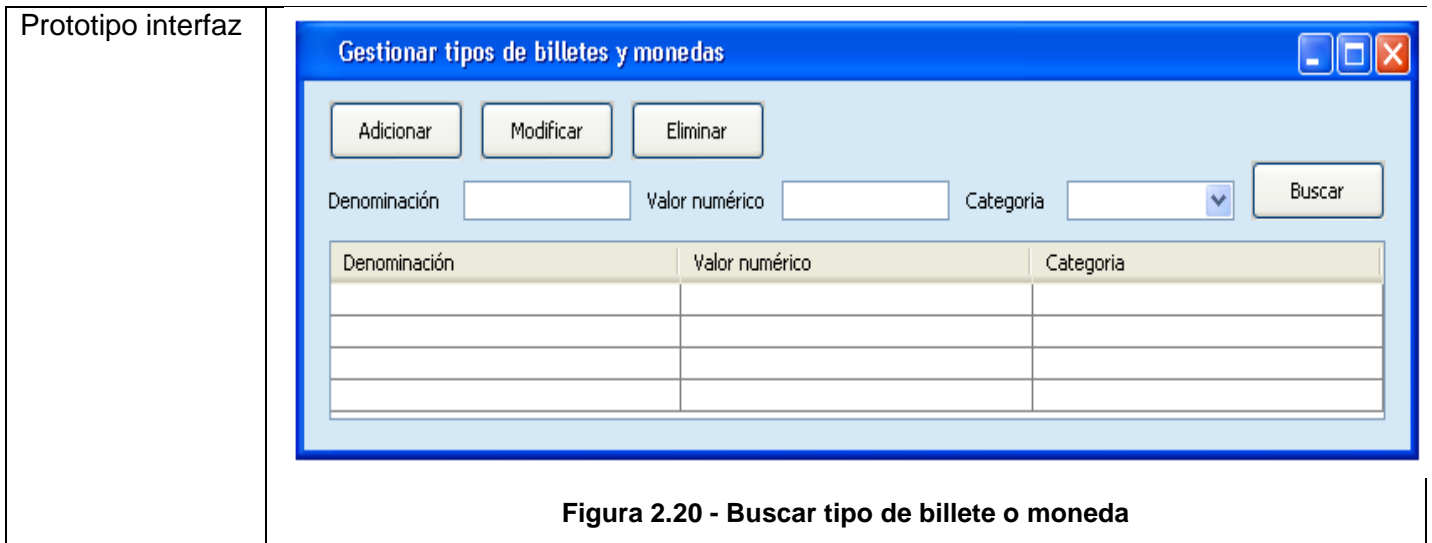
- RF5.3 Modificar tipo de billete o moneda.

Conceptos tratados	Conceptos	Atributos
	Tipos de billetes y monedas.	Denominación, Valor numérico, Categoría.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. El tipo de billete o moneda debe existir en el sistema.	RF5.1 Adicionar tipo de billete o moneda.
Descripción	Permite modificar los atributos de los tipos billetes o monedas Se validan los nuevos datos. Si son correctos se registran.	

	Si no se notifica al usuario y se permite corregirlos.
Validaciones	Los datos se validan según lo especificado en el modelo conceptual.
Post-condiciones	Se han modificado los atributos de los tipos de billetes y monedas.
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.19 - Modificar tipo de billete o moneda</b></p>

- RF5.4 Buscar tipo de billete o moneda.

Conceptos tratados	Conceptos	Atributos
	Tipos de billetes y monedas.	Denominación, Valor numérico, Categoría
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene permisos para realizar la operación. Se listan todas los tipos de billetes y monedas.	RF5.1 Adicionar tipo de billete o moneda.
Descripción	Mostrar listado de los billetes o monedas del sistema que cumplen con un criterio de búsqueda suministrado por el usuario. Si no se puede realizar la acción notificar al usuario.	
Validaciones	Los criterios de búsqueda deben ser válidos.	
Post-condiciones	Listar los tipos de billetes o monedas que cumplen con el criterio de búsqueda dado.	
Post-requisito	No procede.	

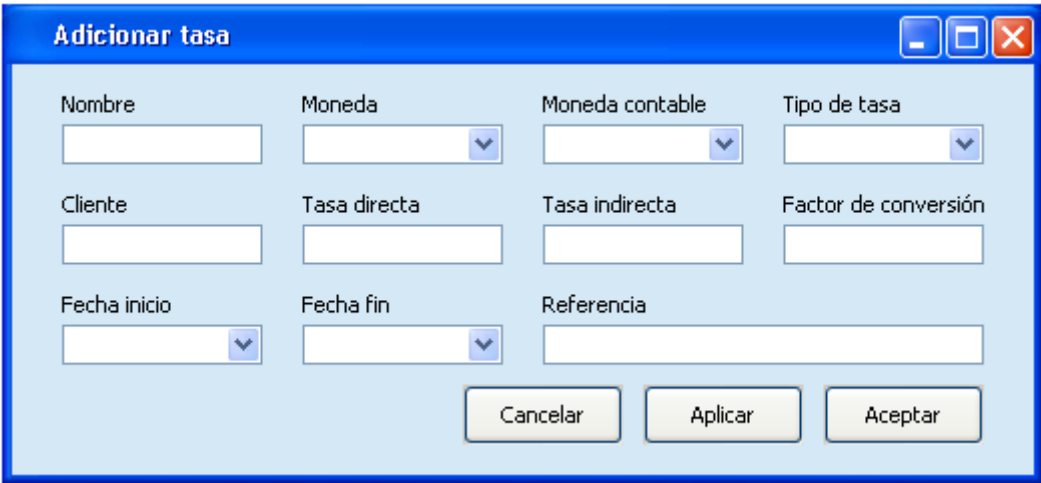


**RF6. Gestionar tasa de cambio.**

Requerimiento que permite gestionar las tasas de cambio.

- RF6.1 Adicionar tasa.

Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO
	Moneda contable	código ISO
	Tasa de cambio	Nombre, factor de conversión, tasa directa, tasa indirecta, fecha Inicio, fecha Fin, fuente o referencia.
	Cliente/Proveedor	Nombre.
	Tipo de tasa	Denominación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	RF7.1 Adicionar tipo de tasa
Descripción	Permite seleccionar una Moneda e indicar su tasa de cambio respecto a la moneda contable.	

	<p>Permite registrar los datos de la tasa de cambio.</p> <p>Validar datos.</p> <p>Si los datos son correctos se registran.</p> <p>Si no son correctos se notifica al usuario y se permite corregirlos.</p>
Validaciones	<p>Los datos se validan según lo especificado en el modelo conceptual.</p> <p>La tasa de cambio de la Moneda contable es 1 x 1, lo cual esta dado por el hecho de que las tasas de cambio (compra y venta) se definen con respecto a ella misma.</p>
Post-condiciones	<p>Se ha creado una nueva tasa de cambio.</p> <p>Se ha asociado la tasa de cambio a las Monedas correspondientes.</p>
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.21 - Adicionar tasa de cambio</b></p>

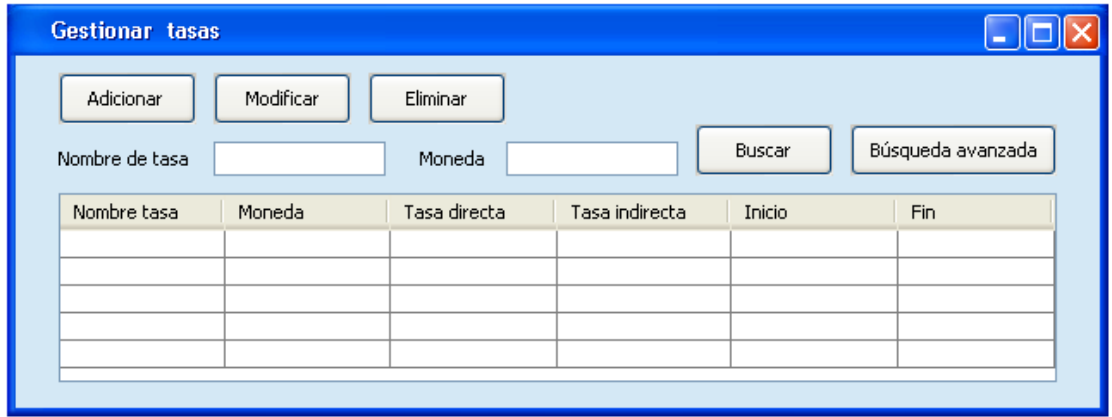
- RF6.2 Eliminar tasa.

Conceptos	Conceptos	Atributos
-----------	-----------	-----------

tratados	Moneda Moneda contable Tasa de cambio  Cliente/Proveedor Tipo de tasa	código ISO código ISO Nombre, factor de conversión, tasa directa, tasa indirecta, fecha Inicio, fecha Fin, fuente o referencia. Nombre. Denominación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.  La tasa debe existir con anterioridad en el sistema.	RF6.1 Adicionar tasa
Descripción	Permite seleccionar una tasa y eliminarla. Solicitar confirmación al usuario. Si se confirma, la tasa se elimina en caso de no existir inconvenientes, de lo contrario, se informa al usuario del suceso y se cancela la acción.	
Validaciones	Solo se puede eliminar la tasa si no se han registrado operaciones respecto a ella.	
Post-condiciones	Se ha eliminado la tasa.  Se ha eliminado la relación con Moneda.	
Post-requisito	No procede.	



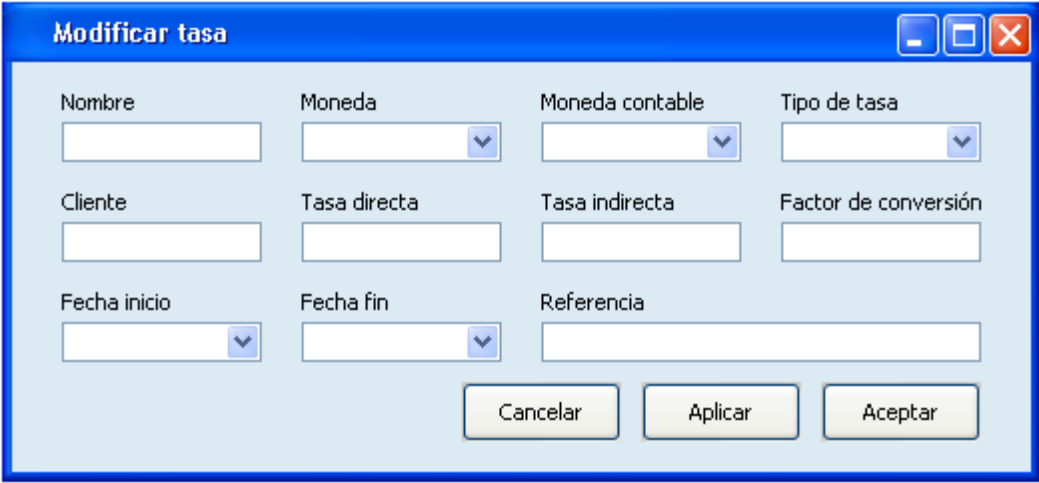
Prototipo interfaz



**Figura 2.22 - Eliminar tasa de cambio**

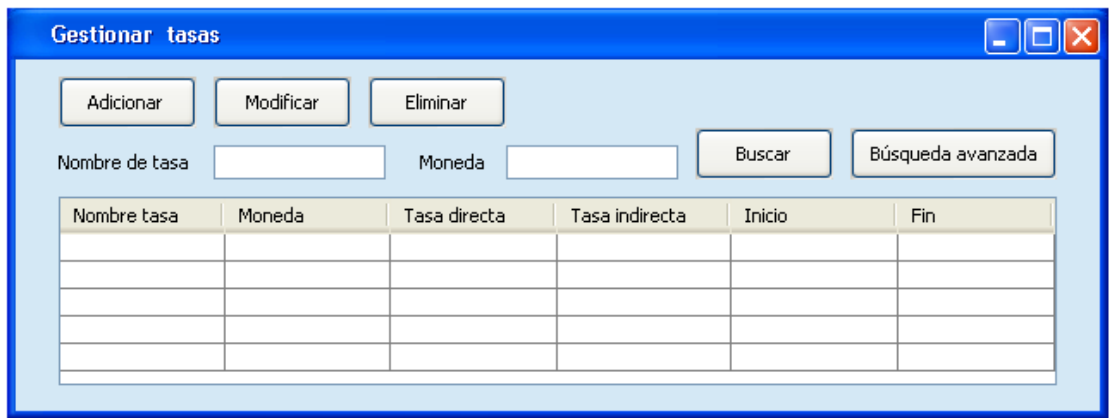
- RF6.3 Modificar tasa.

Conceptos tratados	Conceptos	Atributos
	Moneda Moneda contable Tasa de cambio Cliente/Proveedor Tipo de tasa	código ISO. código ISO. Nombre, factor de conversión, tasa directa, tasa indirecta, fecha Inicio, fecha Fin, fuente o referencia. Denominación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. La tasa debe existir en el sistema.	RF6.1 Adicionar tasa
Descripción	Permite seleccionar una tasa y modificar sus atributos. Validar datos. Si los datos son correctos se registran.	

	Sino son correctos se notifica al usuario y se permite corregirlos.
Validaciones	Los datos se validan según lo definido en el modelo conceptual. La tasa de cambio de la Moneda contable es 1 x 1, lo cual está dado por el hecho de que las tasas de cambio (compra y venta) se definen con respecto a ella misma.
Post-condiciones	Se han modificado los atributos de la tasa seleccionada.
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.23 - Modificar tasa de cambio</b></p>

- RF6.4 Buscar tasa.

Conceptos tratados	Conceptos	Atributos
	Moneda	código ISO.
	Tasa de cambio	Nombre, fecha Inicio, fecha Fin.
	Cliente/Proveedor	Nombre.
	Tipo de tasa	Denominación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los	RF6.1 Adicionar tasa

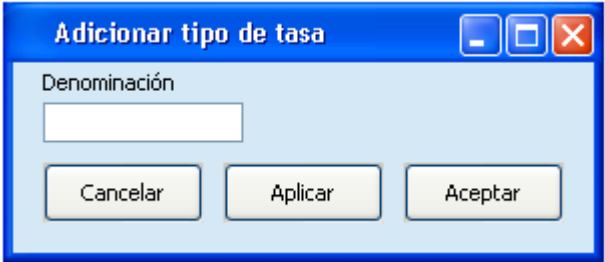
	<p>permisos para realizar esta acción.</p> <p>Se listan todos los tipos de tasa que existen en el sistema</p>
Descripción	<p>Permite buscar tasas que coincidan con los criterios de búsqueda suministrados por el usuario.</p> <p>Lista las tasas que coinciden con los criterios introducidos.</p> <p>Si no se encuentran coincidencias con los criterios o no se puede realizar la búsqueda se notifica al respecto al usuario.</p>
Validaciones	Debe introducirse al menos un criterio de búsqueda.
Post-condiciones	No procede
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.24 - Buscar tasa de cambio</b></p>

**RF7. Gestionar tipo de tasa.**

Requerimiento que permite gestionar los tipos de tasas.

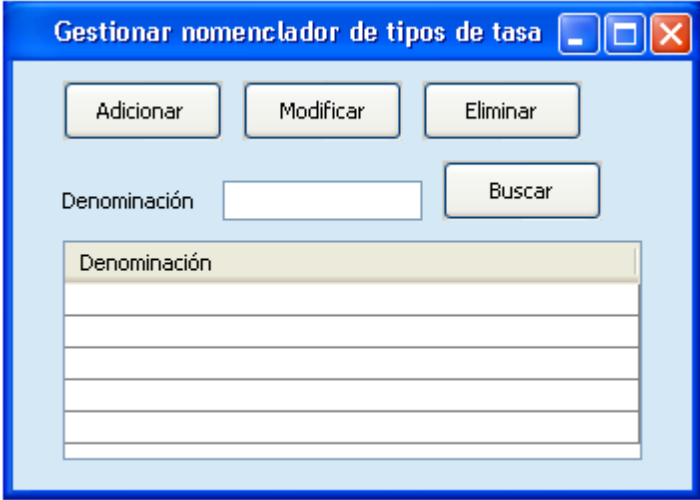
- RF7.1 Adicionar tipo de tasa.

Conceptos tratados	Conceptos	Atributos
	Tipo de Tasa	Denominación.

Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.	
Descripción	Permite registrar los datos del tipo de tasa (denominación). Se validan los datos, si son correctos se registran. Si no se le notifican al usuario los errores y se permite corregirlos.	
Validaciones	El sistema valida los datos según lo descrito en el Modelo Conceptual.	
Post-condiciones	Se ha creado un nuevo tipo de tasa.	
Post-requisito	No procede.	
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.25 -Adicionar tipo de tasa.</b></p>	

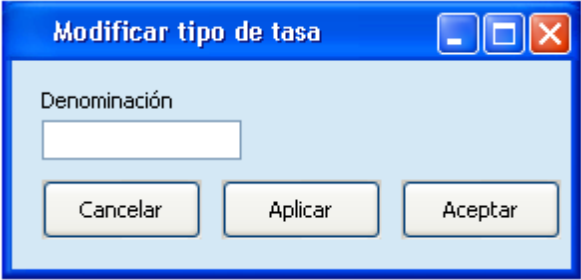
- RF7.2 Eliminar tipo de tasa.

Conceptos tratados	Conceptos	Atributos
	Tipo de Tasa	Denominación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar	RF7.1 Adicionar tipo de tasa

	<p>esta acción.</p> <p>El tipo de tasa debe existir en el sistema.</p>
Descripción	<p>Permite seleccionar un tipo de tasa y eliminarlo. Si no existen inconvenientes, el sistema solicita confirmación al usuario.</p> <p>Si se confirma, se elimina tipo de tasa, de lo contrario, se cancela la acción.</p>
Validaciones	Si el elemento está siendo usado, no se elimina del sistema.
Post-condiciones	<p>Se ha eliminado el tipo de tasa.</p> <p>Se ha eliminado la relación con tasa de cambio.</p>
Post-requisito	No procede.
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.26 -Eliminar tipo de tasa.</b></p>

- RF7.3 Modificar tipo de tasa.

Conceptos tratados	Conceptos	Atributos
	Tipo de Tasa	Denominación
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado	RF7.1 Adicionar tipo de tasa

	<p>ante el sistema y tiene los permisos para realizar esta acción.</p> <p>El tipo de tasa debe existir en el sistema.</p>
Descripción	<p>Permite seleccionar un tipo de tasa y modificar sus atributos (denominación).</p> <p>Validar datos.</p> <p>Si los datos son correctos se registran.</p> <p>Si no son correctos se notifica al usuario y se permite corregirlos.</p>
Validaciones	<p>El sistema valida los datos según lo descrito en el modelo conceptual.</p>
Post-condiciones	<p>Se han modificado los atributos del tipo de tasa.</p> <p>Se ha modificado la relación del tipo de tasa si lo ha indicado el usuario.</p>
Post-requisito	<p>No procede.</p>
Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.27 -Modificar tipo de tasa.</b></p>

- RF7.4 Buscar tipo de tasa.

Conceptos tratados	Conceptos	Atributos
	Tipo de Tasa	Denominación
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha identificado y autenticado ante el sistema y tiene permisos para realizar la	RF7.1 Adicionar tipo de tasa

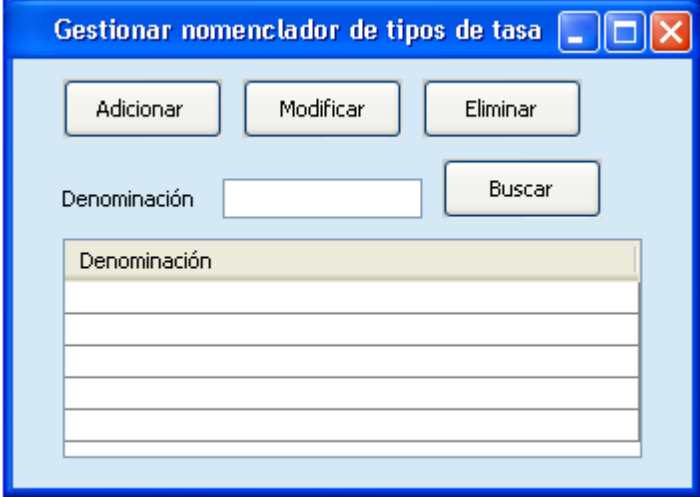
	operación. Se listan todos los tipos de tasa que existen en el sistema.	
Descripción	Mostrar listado de los tipos de tasa registrados en el sistema que cumplen con un criterio de búsqueda suministrado por el usuario. Si no se puede realizar la acción notificar al usuario.	
Validaciones	No procede.	
Post-condiciones	Listar los elementos que cumplen con el criterio de búsqueda dado.	
Post-requisito	No procede.	
Prototipo interfaz		

Figura 2.28 -Buscar tipo de tasa.

**RF8. Gestionar reevaluación de cuentas.**

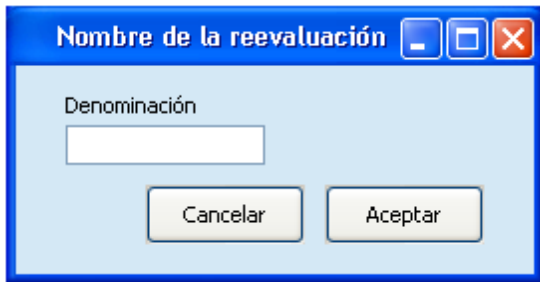
Requerimiento que permite calcular y revertir la reevaluación de las cuentas. Esto significa contabilizar la diferencia de saldo de las cuentas seleccionadas en función de la tasa vigente en el momento que se efectúa la reevaluación.

- RF7.1 Calcular reevaluación.

Conceptos tratados	Conceptos	Atributos
--------------------	-----------	-----------

	<p>Árbol cuenta</p> <p>Tipo de Tasa</p> <p>Tasa de cambio</p>	<p>Cuenta</p> <p>Denominación</p> <p>Factor de conversión.</p>
Precondiciones	Precondiciones	
	<p>El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción.</p> <p>Deben existir elementos en el árbol de cuentas.</p> <p>El subsistema al que pertenezca la cuenta debe hacer efectuado cierre.</p>	
Descripción	<p>Permite conocer el estado actual de las cuentas indicadas.</p> <p>Se seleccionan las cuentas a reevaluar.</p> <p>El sistema calcula el saldo actual de la cuenta seleccionada, mediante su reevaluación en función de la tasa vigente, para ello se basa en las siguientes disposiciones:</p> <p>En el caso de que la cuenta sea acreedora si luego de la reevaluación:</p> <p>Incrementa su saldo, la diferencia de saldos se registra en la cuenta de Gastos x diferencia de tasa.</p> <p>Disminuye su saldo, la diferencia de saldos se registra en la cuenta de Ingresos x diferencia de tasa.</p> <p>En el caso de que la cuenta sea deudora si luego de la reevaluación:</p> <p>Incrementa su saldo, la diferencia de saldos se registra en la cuenta de Ingresos x diferencia de tasa.</p> <p>Disminuye su saldo, la diferencia de saldos se registra en la cuenta de Gastos x diferencia de tasa.</p> <p>Se validan los datos, si no existen inconvenientes, se muestra una interfaz que permite guardar la operación. De lo contrario, se cancela la acción. En cualquiera de las acciones, se advierte al usuario y se solicita su confirmación.</p>	
Validaciones	<p>El sistema valida los datos según lo descrito en el modelo conceptual.</p> <p>La cuenta o grupo de cuentas a reevaluar deben representar operaciones en una moneda diferente a la contable.</p>	

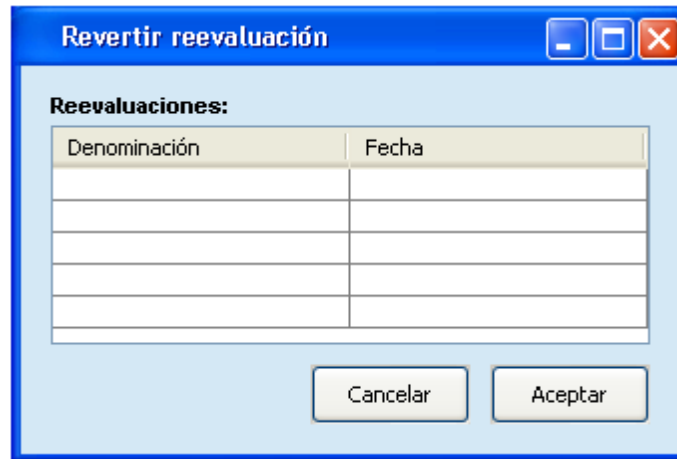


Prototipo interfaz	 <p style="text-align: center;"><b>Figura 2.29 -Calcular reevaluación.</b></p>
--------------------	--

- RF7.1 Revertir reevaluación.

Conceptos tratados	Conceptos	Atributos
	Árbol cuenta Tipo de Tasa Cuentas reevaluadas	Cuenta Denominación Denominación
Precondiciones	Precondiciones	
	El usuario se ha identificado y autenticado ante el sistema y tiene los permisos para realizar esta acción. Debe existir, al menos, una reevaluación de cuentas almacenada.	
Descripción	Lista todas las reevaluaciones almacenadas en el sistema. Permite seleccionar una reevaluación y decidir, de la misma, que cuenta desea revertir. Se validan los datos, si no existen inconvenientes, el sistema solicita confirmación al usuario. Si se confirma, se revierte la operación en la cuenta seleccionada, efectuando el proceso contrario a lo expuesto en el requisito Calcular reevaluación. En caso contrario, se cancela la acción.	
Validaciones	No procede	

Prototipo interfaz



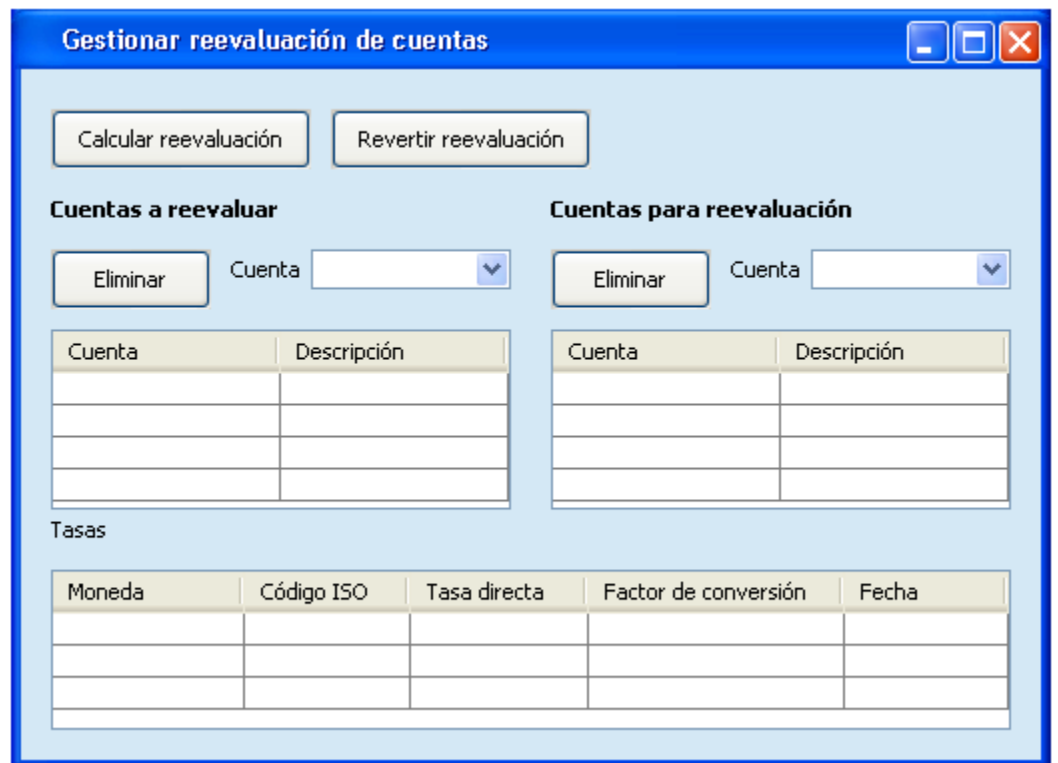
**Revertir reevaluación**

**Reevaluaciones:**

Denominación	Fecha

Cancelar      Aceptar

Figura 2.30 -Revertir reevaluación.



**Gestionar reevaluación de cuentas**

Calcular reevaluación      Revertir reevaluación

**Cuentas a reevaluar**      **Cuentas para reevaluación**

Eliminar    Cuenta       Eliminar    Cuenta

Cuenta	Descripción

Cuenta	Descripción

**Tasas**

Moneda	Código ISO	Tasa directa	Factor de conversión	Fecha

Figura 2.31 –Gestionar reevaluación.

### 2.4.3. Especificación de reportes

Para enumerar los reportes se tomó como referencia la notación **R #**. Donde la “R” responde a la denominación reporte y el número es el correspondiente. La especificación de cada uno de los reportes se realiza sobre una plantilla definida por la dirección del proyecto, en la que, a modo de tabla se recogen los principales datos a tener en cuenta.

- R1 Historial de tasas

<b>Objetivo</b>	Conocer el historial de tasas de una moneda para un periodo dado por el cliente.													
<b>Parámetros</b>	\$idmoneda ; \$desde; \$hasta													
<b>FORMATO</b>														
Encabezado	<b>Reporte del Historial de Tasas de una Moneda durante un período</b>													
Encabezado de página	Entidad: _____		Fecha: <u>DD/MM/AA</u>											
Pie de página														
Resumen	<b>Observaciones:</b>													
Origen de los datos	<i>Consulta que se realiza en la base de datos para obtener los valores requeridos para la emisión del reporte.</i>													
Ejemplo de formato (Ver Anexo 3)	<p><b>Reporte del Historial de Tasas de una Moneda durante un período</b></p> <p>Entidad: _____ Fecha: <u>DD/MM/AA</u></p> <p>Periodo Inicio: _____ Fin: _____</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Siglas</th> <th style="text-align: left;">Moneda</th> <th style="text-align: left;">Tasa directa</th> <th style="text-align: left;">Tasa indirecta</th> <th style="text-align: left;">Fecha inicio</th> <th style="text-align: left;">Fecha fin</th> <th style="text-align: left;">Tipo de tasa</th> </tr> </thead> </table>							Siglas	Moneda	Tasa directa	Tasa indirecta	Fecha inicio	Fecha fin	Tipo de tasa
Siglas	Moneda	Tasa directa	Tasa indirecta	Fecha inicio	Fecha fin	Tipo de tasa								
<b>ELEMENTOS A MOSTRAR</b>														

- R1 Reporte diario de tasas

<b>Objetivo</b>	Tener el control de las Tasas que están vigentes para el día		
<b>Parámetros</b>	\$fecha		
<b>FORMATO</b>			
Encabezado	<b>Reporte diario de tasas</b>		
Encabezado de página	<b>Entidad:</b> _____	<b>Fecha:</b> <u>DD/MM/AA</u>	
Pie de página			
Resumen	<b>Observaciones:</b>		
Origen de los datos	<i>Consulta que se realiza en la base de datos para obtener los valores requeridos para la emisión del reporte.</i>		
Ejemplo de formato (Ver Anexo 4)	<b>Reporte diario de tasas</b>  <b>Entidad:</b> _____ <b>Fecha:</b> <u>DD/MM/AA</u>  <b>Siglas      Moneda      Tasa directa      Tasa indirecta</b>		
<b>ELEMENTOS A MOSTRAR</b>			

La Gestión de requerimientos, es una actividad que, aunque no se hace mención directa de la misma, es válido aclarar que estuvo presente durante toda esta fase del desarrollo. Esta es una actividad que se ejecuta en paralelo a las mencionadas anteriormente, pues los cambios en los requisitos ocurren con periodicidad y por tanto es algo que no se puede descuidar.

#### 2.4.4. Validación de requerimientos

La validación de los requerimientos identificados para el subsistema Multimoneda se apoyó en los métodos expuestos en el Capítulo 1. La aplicación de las técnicas de revisiones y la confección de prototipos no funcionales de interfaz de usuario, tal y como se describe en el epígrafe 2.3.3 de este documento fue la vía para que una vez obtenidas las especificaciones de los requerimientos, se realizara

una revisión profunda por parte de los analistas de la línea, el grupo de analistas principales del proyecto, funcionales y otras personas involucradas , donde se rectificaron errores y se mejoraron las soluciones propuestas originalmente. Este proceso concluyó con la firma de las especificaciones, quedando confirmada su validez.

### **2.5. Conclusiones**

Con la confección de este capítulo fue posible identificar y describir las funcionalidades que presentará CedruX en el subsistema Multimoneda. Se ha realizado la especificación de los requisitos funcionales, la cual obtuvo de la aplicación de las actividades expuestas dentro de la Ingeniería de Requisitos, partiendo del modelo conceptual identificado en el dominio del problema. Fue posible entender qué requiere el cliente y utilizar técnicas que posibilitan su especificación de forma sencilla y a la vez que facilite su integración con las demás fases de desarrollo de software. Este proceso se tornó satisfactorio e importante ya que se interactuó con el ambiente del problema, adquiriendo mayor conocimiento de las operaciones que se incluyen en la solución propuesta.

Este contenido es la principal entrada de las tareas de diseño, las que se documentarán en el próximo capítulo de esta tesis.

## CAPÍTULO 3: DISEÑO DEL SISTEMA

### 3.1. Introducción

Este capítulo recoge los elementos considerados para elaborar el diseño del subsistema Multimoneda de CedruX. En el mismo se explican los aspectos importantes dentro del modelo de diseño propuesto, sentando las bases para la implementación de un sistema que satisfaga las necesidades del cliente y responda, eficientemente a los requerimientos identificados. Comprende el modelado del diseño del subsistema. Para ello, se definen las clases y patrones utilizados durante el desarrollo de los modelos. Igualmente se hace referencia las decisiones arquitectónicas tomadas por la línea de arquitectura que tienen impacto en la tareas de diseño, las cuales constituyen el objetivo a desarrollar en esta parte del trabajo.

Incluye los diagramas que sustentan la solución que será implementada en plataformas de desarrollo web y se hace una referencia a la relación del subsistema Multimoneda, como componente, dentro del producto CedruX en cuanto a servicios y la utilización de sus funcionalidades.

### 3.2. Utilización del subsistema Multimoneda dentro del Sistema Integral de Gestión CedruX

Lo expuesto hasta este epígrafe detalla las operaciones que se han identificado como funcionalidades del subsistema Multimoneda. Funcionalidades que conforman un componente ampliamente utilizado por el resto de los subsistemas de CedruX debido a su principal concepción de brindar soporte y facilitar la reutilización de operaciones comunes dentro de la solución de gestión. Para brindar una idea general de la interacción de multimoneda con CedruX se muestra una representación de misma en las Figura 3.32 y 3.33, la cual puntualiza interacción entre las funcionalidades de Multimoneda y el resto de la solución. Estas se incluyen en una clase denominada MultimonedaServices, que implementa los servicios que proporciona el componente para el trabajo de los restantes subsistemas.

Para un mejor entendimiento del flujo, se debe tener en cuenta que se han dividido las funcionalidades del componente Multimoneda de manera tal en que cada una aparece representada por colores, según lo que se plantea a continuación:



- Reevaluación de cuentas.



- Moneda alternativa.



- Moneda de la entidad.

- Moneda contable.
- Tasa.
- Desglose.

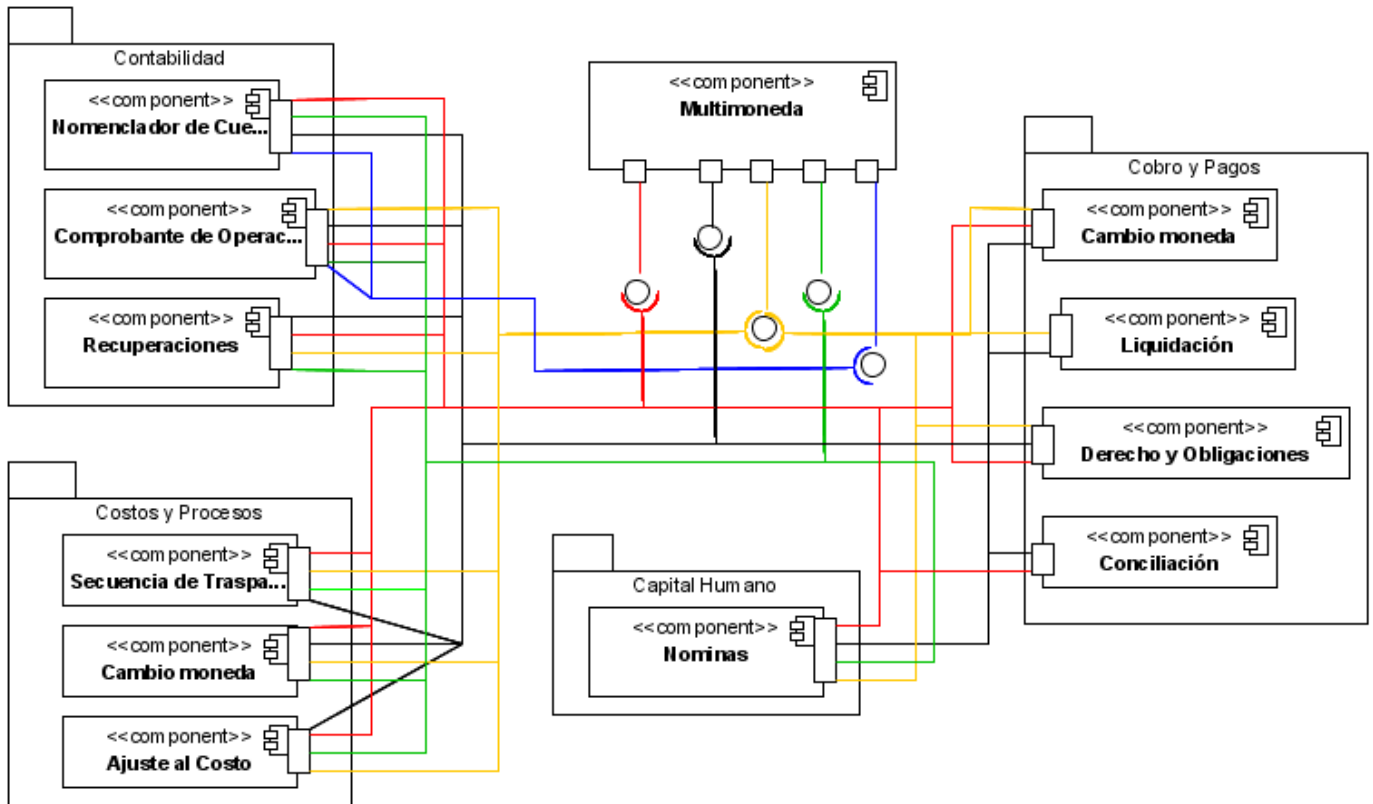
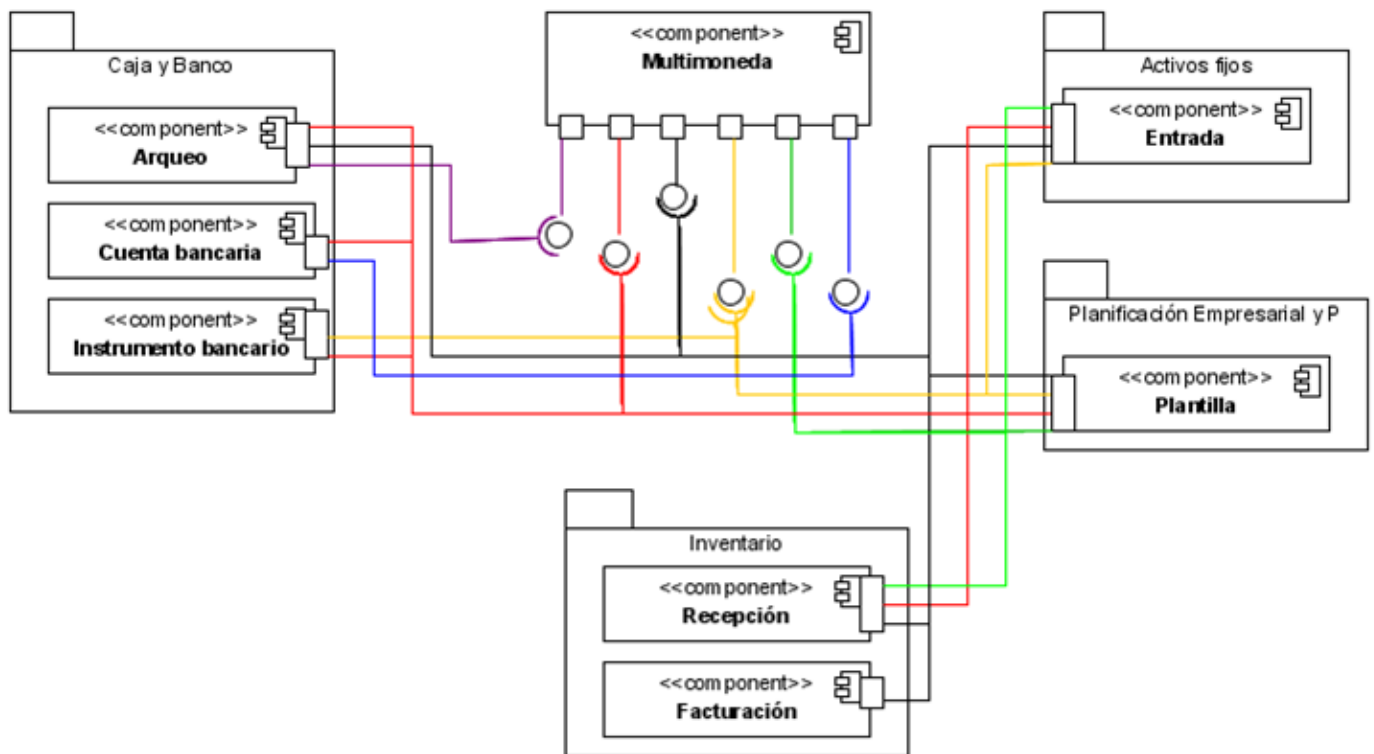


Figura 3.32 - Interacción entre el componente Multimonedas y los restantes componentes de CedruX (primera parte)



**Figura 3.33 - Interacción entre el componente Multimonedada y los restantes componentes de CedruX (segunda parte)**

### 3.3. Actividades del diseño

De acuerdo a lo establecido en el documento que rige la metodología a seguir para el desarrollo del software, se realiza un estudio en el que se determina que el subsistema no presenta gran complejidad en los requerimientos definidos por lo que durante el taller de análisis<sup>30</sup> se decide proceder con la realización del modelo de diseño. Entrar en la especificación de aspectos como el lenguaje de programación, la

<sup>30</sup> Taller de análisis: Actividad definida en el Modelo de desarrollo del ERP-Cuba que tiene la finalidad de discutir las tareas a cumplir durante esta fase del desarrollo. En la misma, se evalúan los requerimientos y procesos identificados y validados, a partir de los cuales se desarrollan el mapa de procesos que permitirá:

- Agrupar los requerimientos y procesos por componentes.
- Identificar las dependencia entre los componentes, así como los contratos de los mismo en el mismo modulo.

En general, se establecen las prioridades de desarrollo de cada uno de los componentes, dependiendo de su complejidad.



plataforma de desarrollo, tecnologías de distribución, sistemas operativos y demás elementos necesarios que sí se tienen en cuenta en el diseño, lo que ayuda a gestionar mejor el tiempo y el avance del proceso de desarrollo.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos y restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. (14)

A tono con las disposiciones referidas a las tareas del diseño, que se exponen en el modelo de desarrollo que guía la evolución de este proyecto se identifica la realización de los diagramas de clases y la descripción de las clases como parte del diseño del sistema.

La importancia del diseño del software se puede describir con una sola palabra -calidad-. El diseño es el lugar en donde se fomenta la calidad en la ingeniería del software. El diseño es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado. (10)

### 3.4. Arquitectura del sistema

La arquitectura proporciona una visión global del sistema a construir. Describe la estructura y la organización de los componentes del software, sus propiedades y las conexiones entre ellos. Los componentes del software incluyen módulos de programas y varias representaciones de datos que son manipulados por el programa. (10)

La arquitectura de software es quien le da forma al software para que soporte todos los requisitos. Establece los fundamentos para que todos los involucrados en el equipo de desarrollo trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema. Se necesita una arquitectura robusta, que guíe el proceso de desarrollo, y que defina de manera abstracta los componentes que llevan a cabo alguna tarea, sus interfaces y la comunicación entre ellos. (10)

En CedruX, la arquitectura está basada, fundamentalmente en el sub-estilo arquitectónico Model-View-Controller (MVC).

#### Modelo Vista Controlador

El modelo arquitectónico predominante es el Modelo-Vista-Controlador (MVC<sup>31</sup>). En MVC un elemento tiene tres partes (Ver anexo 5) y contiene:

---

<sup>31</sup> MVC: Patrón arquitectónico Modelo-Vista-Controlador.

- Un modelo que se encarga de gestionar los datos.
- Una vista que gestiona como se muestran esos datos
- Un controlador que determina que modificaciones hay que hacer cuando se interacciona con el elemento.

Los componentes del MVC contienen los siguientes elementos:

*Modelo:* Incluye la información con la que el sistema opera.

- Domain: Están los dominios de datos y las clases de acceso a los datos.
- Business: Clases que implementan la lógica de negocio a la cual responden.

*Vista:* Maneja la interacción con la interfaz de usuario.

- CSS: Define los estilos que se le van a aplicar a las interfaces.
- Idioma: Contiene los ficheros de configuración para el multilinguaje.
- JS: Representan los ficheros javascript de las interfaces.
- Script: Páginas phtml que hacen los llamados a los elementos que se mencionan anteriormente.
- Phtml: Páginas web que llevan código PHP para ser generadas.

*Controlador:* Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y en la vista.

- Controller: Recepciona las acciones y se redireccionan hacia los modelos.

### 3.4.1. Decisiones arquitectónicas

En el equipo de desarrollo de CedruX incluye un equipo de arquitectura, el cual tiene a su cargo la definición de los componentes tecnológicos que soportarán la solución. Entre las determinaciones de los arquitectos se encuentran el uso de PHP 5.2.4 como herramienta de desarrollo. La utilización del Zendframework en su versión 1.5 como marco de trabajo, mientras que Doctrine y PostgreSql se encargan de las tareas referentes al acceso a datos.

### 3.5. Patrones de diseño.

El uso de patrones en las tareas del diseño es una actividad que aporta beneficios a los equipos de desarrollo porque facilita el trabajo en gran medida. Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva y reusable.

Los patrones son soluciones de sentido común que deberían formar parte del conocimiento de un diseñador experto. Además facilitan la comunicación entre diseñadores, pues establecen un marco de referencia (terminología, justificación). (22)

Durante el diseño de sistemas es factible usar patrones de los denominados GRASP<sup>32</sup> pues la asignación correctamente las responsabilidades es muy importante en el diseño. Dentro de este grupo de patrones es posible encontrar los siguientes: Experto, Creador, Bajo acoplamiento, Alta cohesión, Controlador, Fabricación pura, Indirección, Variaciones protegidas, No hables con extraños y Polimorfismo. (20) Para una mayor información sobre el tema puede remitirse al Anexo #6.

Su aplicación se evidencia en la realización de los diagramas de clases del diseño, estos son principios de vital importancia que constituyen una base necesaria para la realización de un diseño eficiente. A continuación se reseñan algunos ejemplos de su utilización:

#### Experto

Está enfocado en resolver el problema de ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos? La solución de basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. (19).

En la solución propuesta, se presenta en las clases modeladoras, por ejemplo, en la clase Monedamodel, en la Gestión de moneda contable, es la responsable de manejar información y efectuar las operaciones que conciernen a su función (insertar, eliminar y modificar moneda), asumiendo toda la lógica para cada una de ellas. Sobre este mismo principio se realiza el diseño en las restantes funcionalidades.

#### Creador

Teóricamente responde la interrogante de ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?, asignando a una clase la responsabilidad de crear una instancia de otra clase en casos diferentes. (19)

En la práctica, se evidencia en las clases del paquete de dominio, quienes son las encargadas de crear los objetos de tipo query, que permiten el acceso a la información almacenada a nivel de datos.

---

<sup>32</sup> GRASP: Acrónimo de General Responsibility Assignment Software Patterns (Patrones de Software para la asignación General de Responsabilidad).

Un ejemplo específico puede ser, el que se referencia en el siguiente fragmento de código de la clase Dattasa, donde se crea un objeto que debe permitir el acceso a los datos que se encuentran en la base de datos del sistema y a través del que se obtiene la información requerida para la operación Cantidad, a través de la que se deben obtener la cantidad de tasas que existan registradas en el sistema.

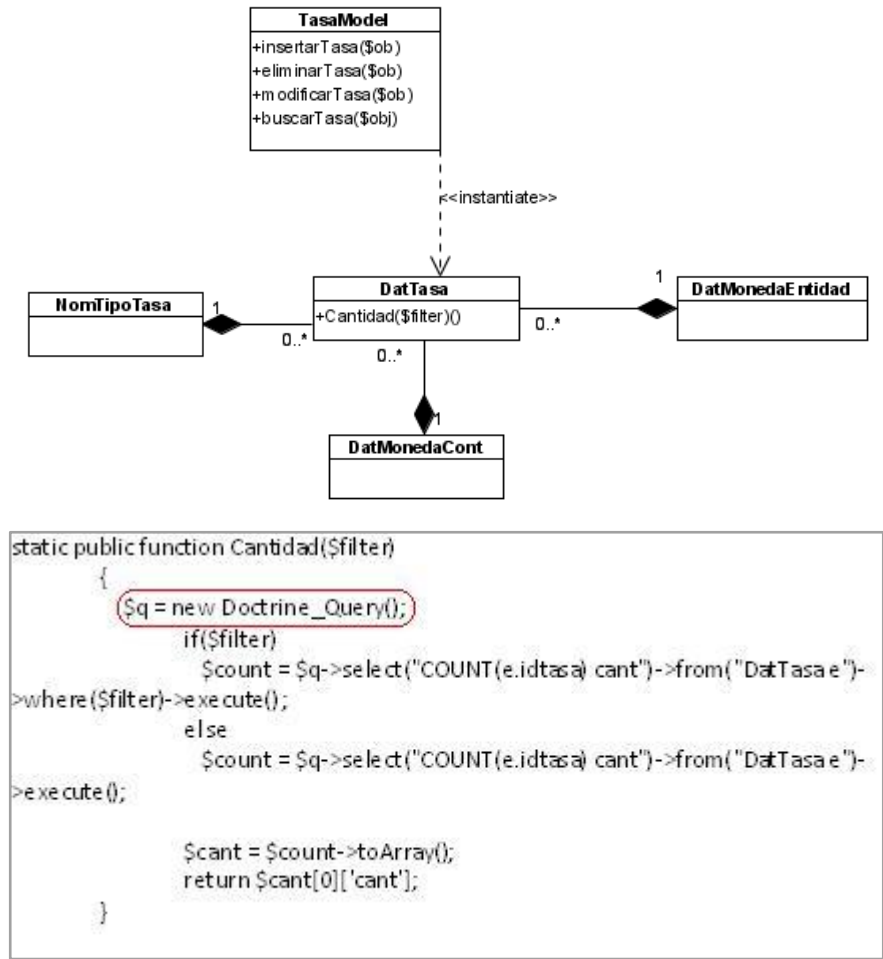


Figura 3.34 – Ejemplo de la aplicación del patrón creador

### Controlador

En la asignación de responsabilidades, es contradictorio definir ¿Quién debería encargarse de atender un evento del sistema?, sin embargo, el uso de este patrón elimina la incógnita anterior sugiriendo asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una de las clases.

Lo descrito a priori significa asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, entre otras cosas.) Si se aplica estos principios, el controlador no realiza las actividades mencionadas sino que las delega en otras clases con las que mantiene un modelo de alta cohesión.

Esto tiene solución a través del uso de una clase controladora que es la que administra el flujo de acciones invocadas desde la interfaz de usuario y en cada una de ellas asocia la lógica del negocio al modelo responsable. En el modelo, donde está implementada la lógica del negocio, las clases modeladoras puede instanciar otras clases en dependencia de la información que se requiera.

En el diagrama de clases de la funcionalidad Reevaluación de cuentas, la clase ContabilizarReevaluacionModel tiene la responsabilidad de crear objetos de otras clases modeladoras (ReevaluacionModel y CuentasReevaluadasmodel) pues estas son las expertas en la información que ella necesita para ejecutar la operación contabilizar, por ejemplo. Esto se muestra en la figura que aparece a continuación:

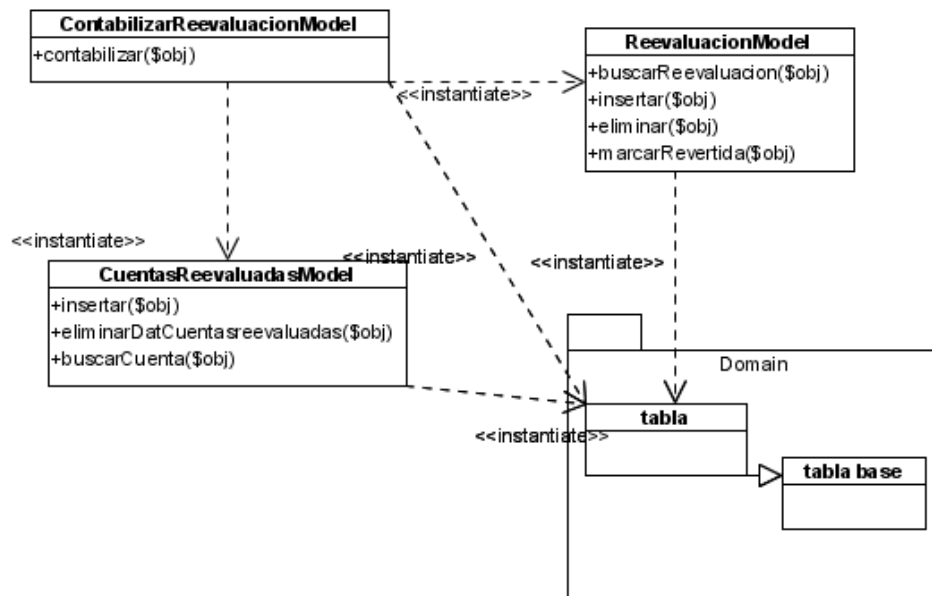


Figura 3.35 – Ejemplo de la aplicación del patrón controlador

### 3.6. Diagramas de clases del diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contiene la siguiente información:

- Clases, asociaciones y atributos
- Interfaces, con sus operaciones y constantes
- Métodos
- Información sobre los tipos de los atributos
- Navegabilidad
- Dependencias

A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. (23)

### 3.6.1. Diagrama de clases con estereotipos WEB

La aplicación a desarrollar es Web, es decir, que tiene como elemento significativo de su arquitectura un navegador y un protocolo de comunicación HTTP. Para el modelado de este tipo de aplicaciones se aprovecha una de las características que le da flexibilidad a la notación de UML. La misma consiste en un conjunto de mecanismos de extensión. El uso de estereotipos permite enriquecer el significado de los elementos clásicos de su notación. Las particularidades de esta, aplicadas a la web incluyen símbolos representativos de las clases en esta arquitectura así como las relaciones entre ellas, las especificaciones de este tema se describen en el Anexo 7; los mismos se muestran a continuación:

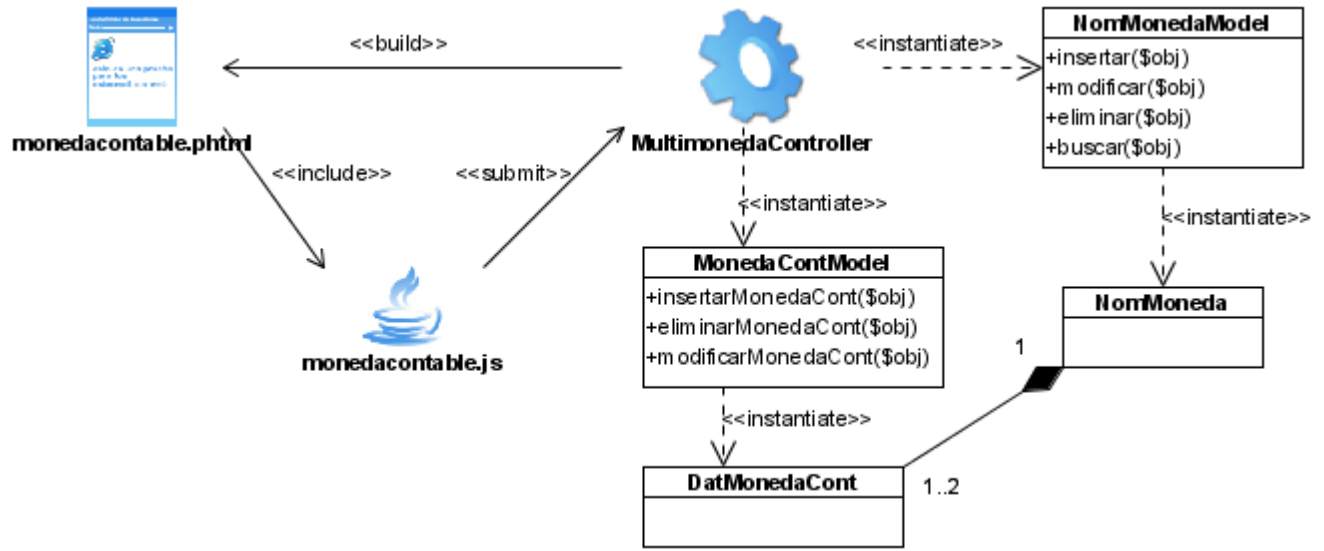


Figura 3.36 Gestionar moneda contable

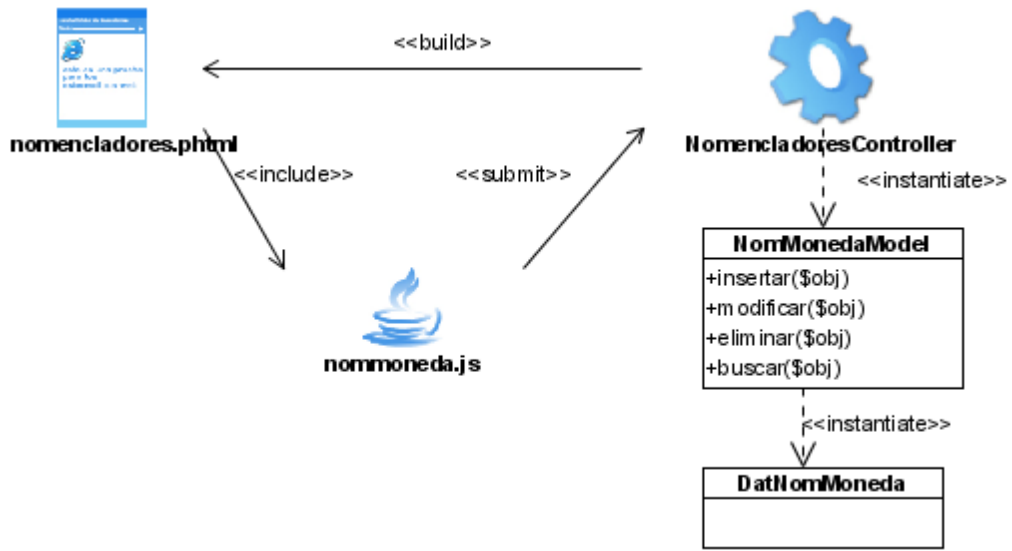


Figura 2.37 - Gestionar nomenclador de moneda

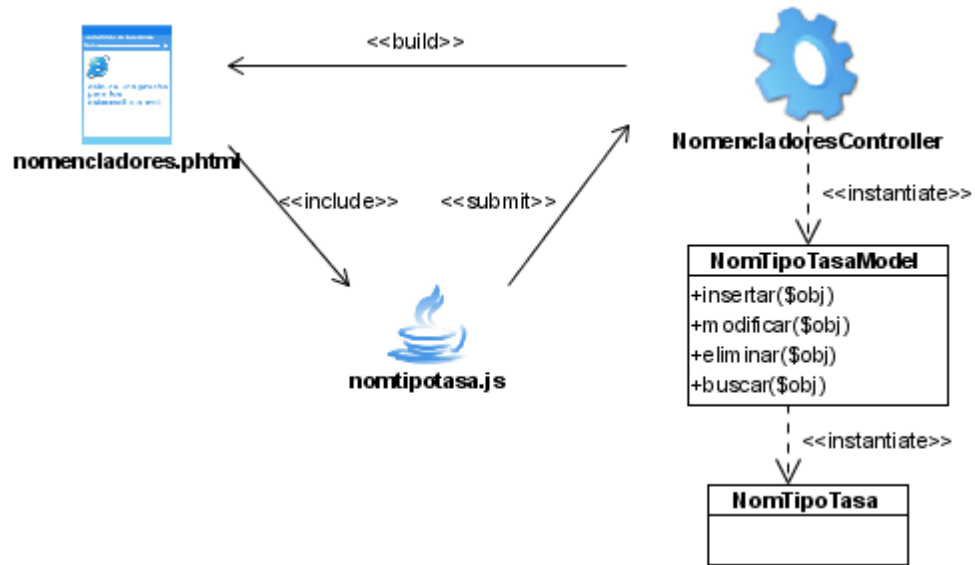


Figura 3.38 - Gestionar nomenclador de tipos de tasas



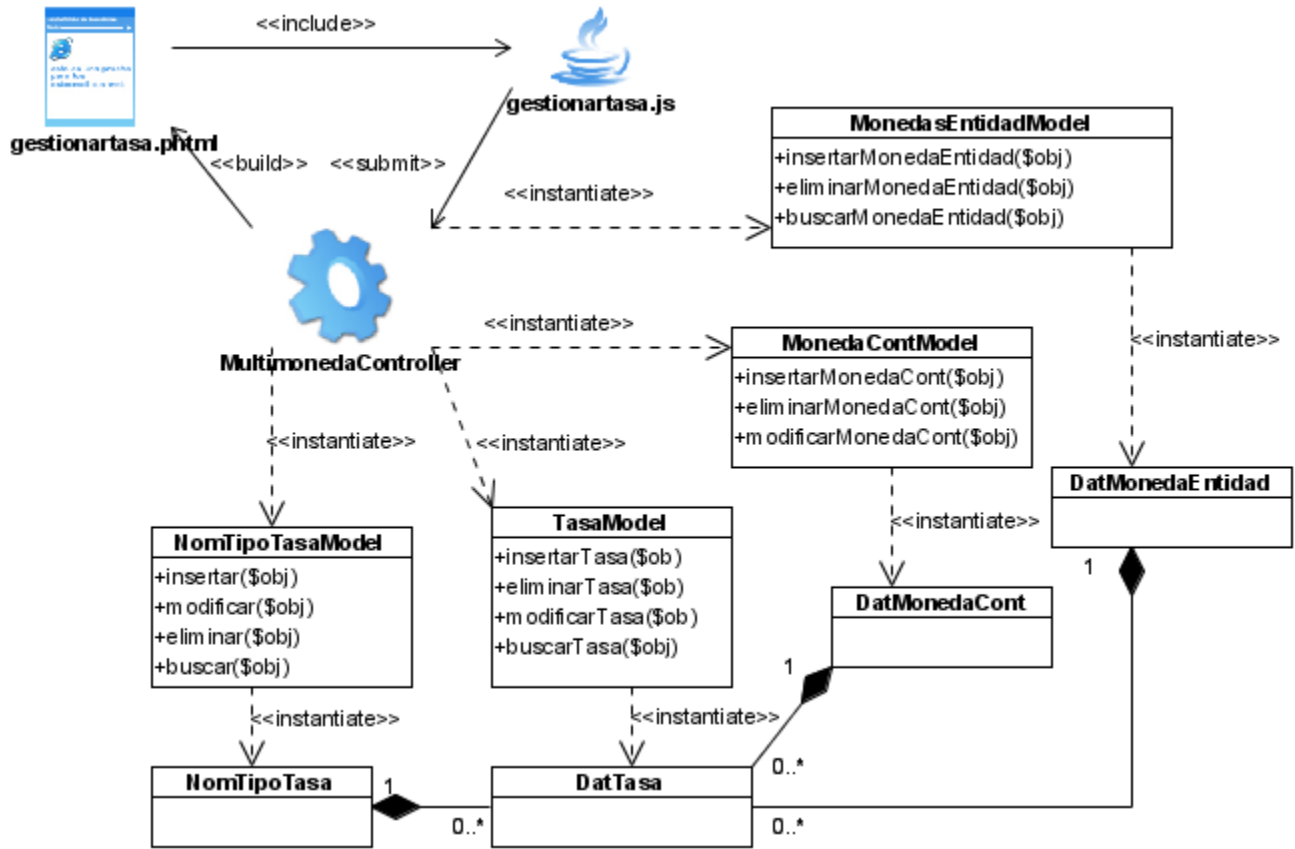


Figura 3.39 Gestionar tasa

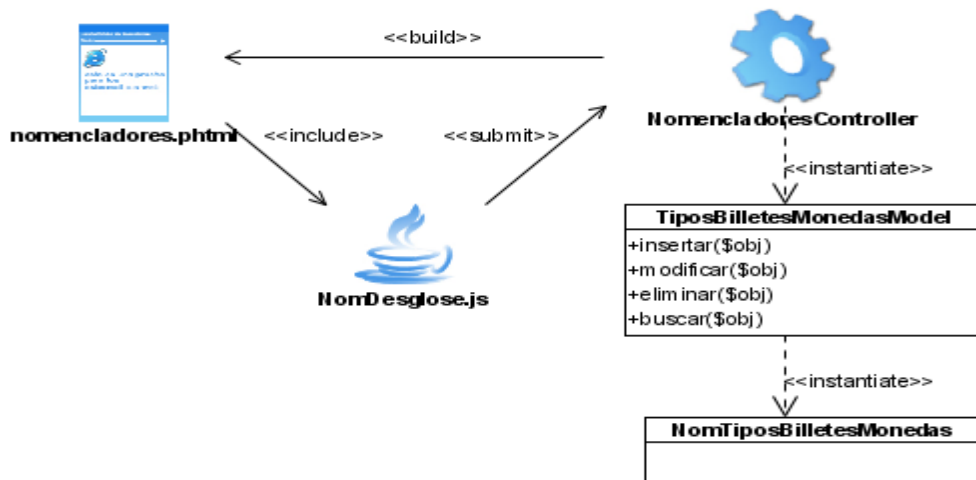


Figura 3.40 -Gestionar billetes y monedas

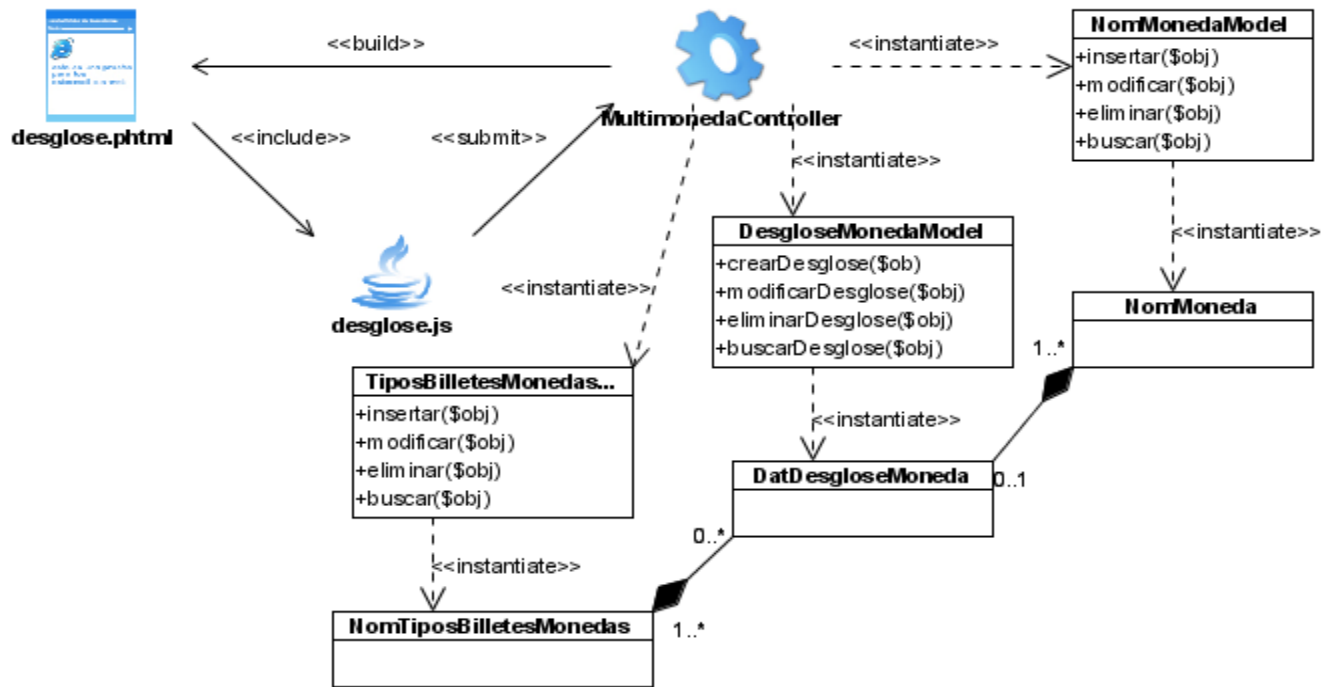


Figura 3.41- Gestionar desglose

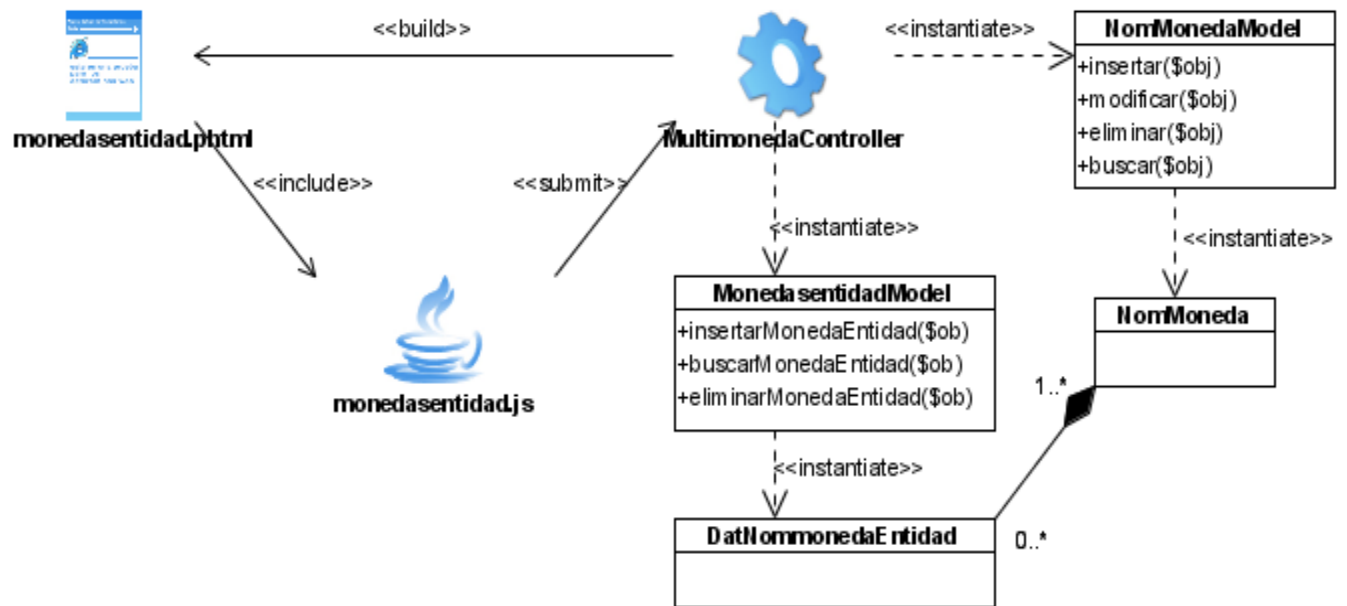


Figura 3.42- Gestionar moneda por entidad

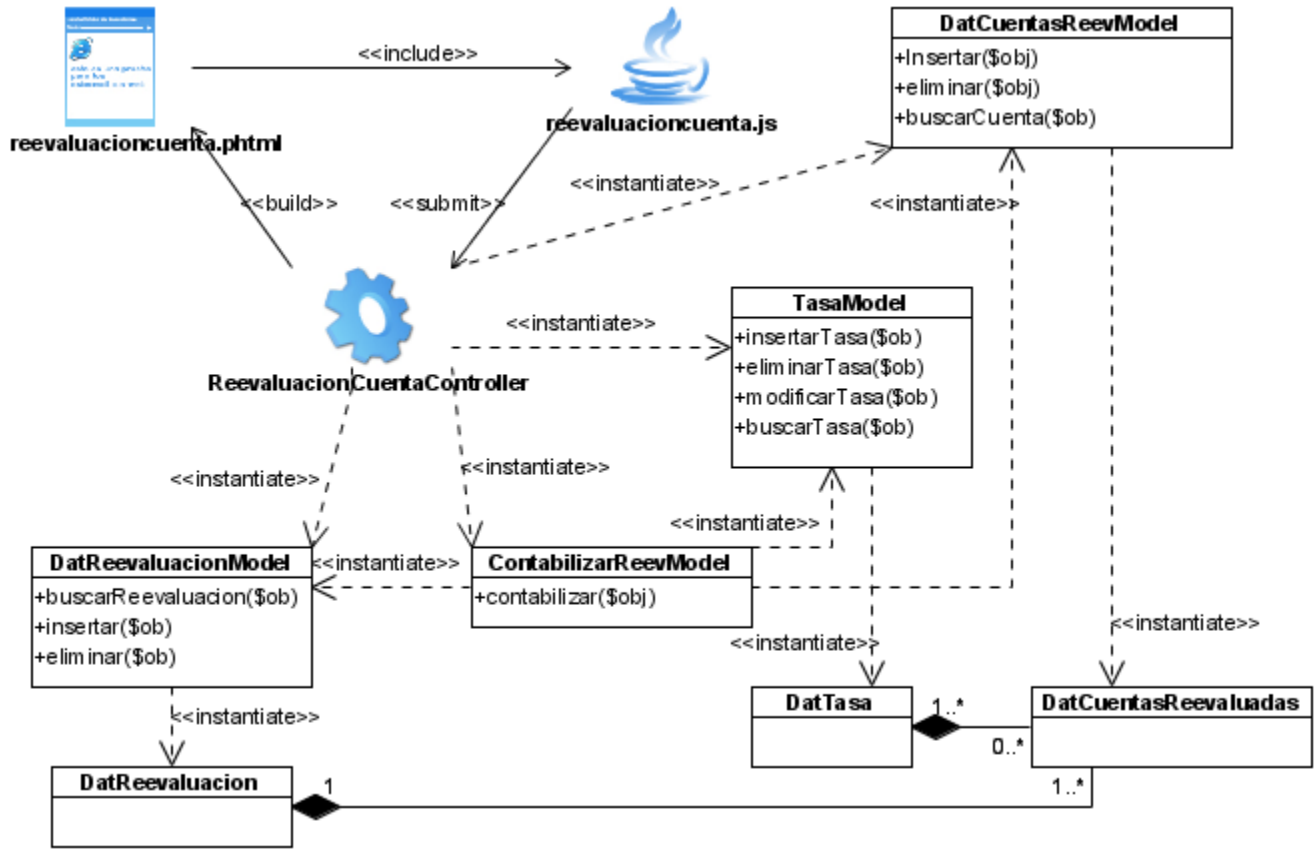


Figura 3.43 Gestionar Reevaluación de cuentas

### 3.6.1.1. Clases controladoras (controller)

Se exponen las clases controladoras que intervienen en el diseño de las funcionalidades del subsistema Multimoneda. (Figuras 3.42, 3.43, 3.44).

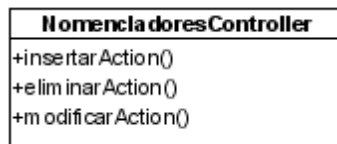


Figura 3.44- Clase controladora nomencladores

MultimonedaController
+insertarTasaAction()
+insertarMonedaContAction()
+insertarMonedaentidadAction()
+crearDesgloseAction()
+modificarTasaAction()
+modificarMonedaAlterAction()
+modificarDesgloseAction()
+eliminarTasaAction()
+eliminarMonedaAlterAction()
+eliminarMonedaEntidadAction()
+eliminarDesgloseAction()

Figura 3.45 - Clase controladora multimoneda

ReevaluacionCuentaController
+reevaluarCuentaAction()
+revertirReevaluacionAction()

Figura 3.46- Clase controladora reevaluación de cuenta

### 3.6.1.2. Descripción de las clases más significativas

Con este epígrafe se ofrece la información detallada de la estructura y funcionalidades que se muestran en las clases identificadas durante el diseño. Se toman como referencia las clases que controlan las operaciones y gestionan el intercambio de información entre el usuario y la base de datos del sistema. La descripción se ha hecho sobre la base de un modelo elaborado por la dirección del proyecto para estandarizar esta actividad que es uno de los artefactos que se generan como parte del modelo de diseño.

- Controladora de multimoneda

<b>Nombre:</b>	MultimonedaCotroller
<b>Tipo de clase:</b>	Controladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insTasaAccion()	Permite obtener los datos por POST para crear un objeto de la clase DatTasa y enviarlo a la clase del negocio TasaModel.
insertarMonedaContAccion()	Permite obtener los datos por POST para crear un objeto de la

	clase <code>DatMonedaCont</code> y enviarlo a la clase del negocio <code>MonedaModel</code> .
<code>insertarMonedaEntidadAccion()</code>	Permite obtener los parámetros por POST para crear un objeto de la clase <code>DatMonedaEntidad</code> y enviarlo para la clase del negocio <code>MonedasEntidadModel</code> .
<code>creardesgloseAccion()</code>	Permite obtener los parámetros por POST para crear un objeto de la clase <code>Desglose</code> y enviarlo para la clase del negocio <code>DesgloseMonedaModel</code> .
<code>modificarTasaAccion()</code>	Permite obtener los parámetros por POST para cambiar los datos de en un objeto de <code>DatTasa</code> y enviarlo para la clase del negocio <code>TasaModel</code> .
<code>modificarMonedaAlterAccion()</code>	Permite obtener los parámetros por POST para cambiar los datos de en un objeto de <code>DatMonedaContable</code> y enviarlo para la clase del negocio <code>MonedaModel</code> .
<code>modificarDesgloseAccion()</code>	Permite obtener los parámetros por POST para cambiar los datos de en un objeto de <code>Desglose</code> y enviarlo para la clase del negocio <code>DesgloseMonedaModel</code> .
<code>eliminarTasaAccion()</code>	Permite obtener el id de la tasa a eliminar, mediante POST, se verifica que la tasa no este asociada a otras operaciones y se envía el id de esta a la clase <code>TasaModel</code> para ser eliminada.
<code>eliminarMonedaAlterAccion()</code>	Permite obtener el id de la moneda alternativa a eliminar, mediante POST, se verifica que la moneda no este asociada a otras operaciones y se envía el id a la clase <code>MonedaModel</code> para ser eliminada.
<code>eliminarMonedaEntidadAccion()</code>	Permite obtener el id de la moneda perteneciente a la entidad que se desea eliminar, mediante POST, se verifica que la moneda no este asociada a otras operaciones y se envía el id a la clase <code>MonedasEntidadModel</code> para ser eliminada.
<code>eliminarDesglose()</code>	Permite obtener el id del desglose a eliminar, mediante POST, se verifica que el desglose no este asociado a otras operaciones y se

	envía el id de este a la clase DesgloseMonedaModel para ser eliminado.
--	--

- Controladora de nomencladores

<b>Nombre:</b>	NomencladoresController
<b>Tipo de clase:</b>	Controladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insertarAction()	Permite obtener los datos y el tipo de nomenclador por POST para crear un objeto de este nomenclador y enviarlo a la clase del negocio relacionada a este para ser guardados.
eliminarAction()	Permite obtener el id del objeto a eliminar y el nomenclador al que pertenece, mediante POST, se verifica que el objeto no este asociado a otras operaciones y se envía el id de este a la clase modeladora perteneciente al nomenclador del objeto para ser eliminado.
modificarAction()	Permite obtener los parámetros y el tipo de nomenclador por POST para cambiar los datos del objeto perteneciente al nomenclador y enviarlo para la clase del negocio relacionada a este.

- Controladora de reevaluación de cuentas

<b>Nombre:</b>	ReevaluacionCuentaController
<b>Tipo de clase:</b>	Controladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
reevaluarCuentaAction()	Permite calcular y obtener los datos necesarios para realizar la reevaluación, los que serán enviados a las clases del negocio ReevaluacionModel y CuentasReevaluadasModel para que estas los guarden y le envía un mensaje a la clase del negocio

	ContabilizarReevaluacionModel para que esta realice la contabilización.
revertirReevaluacionAction()	Permite obtener el id de la reevaluación a revertir, mediante POST, se le envía el id a la clase del negocio CuentasReevaluadasModel para revertir la reevaluación.

- Modeladora de moneda contable

<b>Nombre:</b>	MonedaModel
<b>Tipo de clase:</b>	Modeladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insertarMonedaCont(\$obj)	Permite adicionar una moneda contable o alternativa recibiendo como parámetro el objeto a guardar en la base de datos.
eliminarMonedaCont(\$obj)	Permite eliminar la moneda alternativa recibiendo como parámetro un objeto para ser eliminado de la base de datos
modificarMonedaCont(\$obj)	Permite modificar la moneda alternativa recibiendo como parámetro el objeto a guardar en la base de datos.

- Modeladora del nomenclador de monedas

<b>Nombre:</b>	NomMonedaModel
<b>Tipo de clase:</b>	Modeladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insertar(\$obj)	Permite adicionar monedas recibiendo como parámetro el objeto a guardar en la base de datos.
eliminar(\$obj)	Permite eliminar monedas recibiendo como parámetro un objeto para ser eliminado de la base de datos
modificar(\$obj)	Permite modificar monedas recibiendo como parámetro el objeto a guardar en la base de datos.

buscar(\$obj)	Permite buscar monedas recibiendo como parámetro el objeto a buscar en la base de datos.
---------------	--

- Modeladora de tasa

<b>Nombre:</b>	TasaModel
<b>Tipo de clase:</b>	Modeladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insertarTasa(\$obj)	Permite adicionar una tasa recibiendo como parámetro el objeto a guardar en la base de datos.
eliminarTasa(\$obj)	Permite eliminar una tasa recibiendo como parámetro un objeto para ser eliminado de la base de datos
modificarTasa(\$obj)	Permite modificar una tasa recibiendo como parámetro el objeto a guardar en la base de datos.
buscarTasa(\$obj)	Permite buscar tasas recibiendo como parámetro el objeto a buscar en la base de datos.

- Modeladora del nomenclador de tipos de tasa

<b>Nombre:</b>	NomTipoTasaModel
<b>Tipo de clase:</b>	Modeladora
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción</b>
insertar(\$obj)	Permite adicionar un tipo de tasa recibiendo como parámetro el objeto a guardar en la base de datos.
eliminar(\$obj)	Permite eliminar un tipo de tasa recibiendo como parámetro un objeto para ser eliminado de la base de datos
modificar(\$obj)	Permite modificar un tipo de tasa recibiendo pasando como parámetro el objeto a guardar en la base de datos.
buscar(\$obj)	Permite buscar tipos de tasas recibiendo como parámetro el



	objeto a buscar en la base de datos.
--	--------------------------------------

- Modeladora de moneda entidad

<b>Nombre:</b>	MonedaEntidadModel	
<b>Tipo de clase:</b>	Modeladora	
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	<b>Descripción</b>	
insertarMonedaEntidad(\$obj)	Permite adicionar monedas a la entidad recibiendo como parámetro el objeto a guardar en la base de datos.	
eliminarMonedaEntidad(\$obj)	Permite eliminar monedas de la entidad recibiendo como parámetro un objeto para ser eliminado de la base de datos	
buscarMonedaEntidad(\$obj)	Permite buscar las monedas relacionadas a la entidad recibiendo como parámetro el objeto a buscar en la base de datos.	

- Modeladora del nomenclador de tipos de billetes y monedas

<b>Nombre:</b>	NomTiposBilletesMonedasModel	
<b>Tipo de clase:</b>	Modeladora	
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	<b>Descripción</b>	
insertar(\$obj)	Permite adicionar tipos de billetes o monedas recibiendo como parámetro el objeto a guardar en la base de datos.	
eliminar(\$obj)	Permite eliminar tipos de billetes o monedas recibiendo como parámetro un objeto para ser eliminado de la base de datos	
modificar(\$obj)	Permite modificar tipos de billetes o monedas recibiendo como parámetro el objeto a guardar en la base de datos.	
buscar(\$obj)	Permite buscar los tipos de billetes y monedas recibiendo como parámetro el objeto a buscar en la base de datos.	

- Modeladora de desglose de monedas

<b>Nombre:</b>	DatDesgloseMonedaModel	
<b>Tipo de clase:</b>	Modeladora	
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	<b>Descripción</b>	
crearDesglose(\$obj)	Permite adicionarle a una moneda los tipos de billetes o monedas por los que esta se desglosara recibiendo como parámetro el objeto a guardar en la base de datos.	
eliminarDesglose(\$obj)	Permite eliminar un desglose realizado a una moneda, recibiendo como parámetro un objeto para ser eliminado de la base de datos	
modificarDesglose(\$obj)	Permite modificar un desglose realizado a una moneda, recibiendo como parámetro el objeto a guardar en la base de datos.	
buscarDesglose(\$obj)	Permite buscar los desgloses realizados recibiendo como parámetro el objeto a buscar en la base de datos.	

- Modeladora de contabilizar reevaluación

<b>Nombre:</b>	ContabilizarReevdeModel	
<b>Tipo de clase:</b>	Modeladora	
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	<b>Descripción</b>	
Contabilizar	Organiza los datos de la forma adecuada para ser enviados al componente Reglas Contables mediante un servicio que generará el comprobante final.	

- .Modeladora de cuentas reevaluadas

<b>Nombre:</b>	DatCuentasReevModel	
<b>Tipo de clase:</b>	Modeladora	

Para cada responsabilidad:	
Nombre:	Descripción
Insertar (\$obj)	Permite guardar los datos de las cuentas reevaluadas recibiendo como parámetros el objeto a guardar en la base de datos.
eliminar (\$obj)	Permite eliminar los datos de una cuenta reevaluada recibiendo el objeto a ser eliminado en la base de datos.
buscarCuenta(\$obj)	Permite buscar los datos las cuentas reevaluadas recibiendo como parámetro el objeto a buscar en la base de datos.

- Modeladora de reevaluaciones

<b>Nombre:</b>	DatReevaluacionModel
<b>Tipo de clase:</b>	Modeladora
Para cada responsabilidad:	
Nombre:	Descripción
Insertar (\$obj)	Permite adicionar los datos de la reevaluación recibiendo como parámetros el objeto a guardar en la base de datos.
Eliminar(\$obj)	Permite eliminar una reevaluación recibiendo como parámetro el objeto a ser eliminado en la base de datos.
BuscarReevaluacion(\$obj)	Permite buscar todas las reevaluaciones realizadas recibiendo como parámetro el objeto a buscar en la base de datos.
MarcarRevertida(\$obj)	Permite cambiar el estado de una reevaluación a revertida recibiendo como parámetro el objeto a cambiar en la base de datos.

- Clase del dominio de moneda contable

<b>Nombre:</b>	DatMoneda
<b>Tipo de clase:</b>	Dominio

Descripción	
Permite el acceso a la tabla de la base de datos DatMoneda, que contiene la información referente a la moneda contable y alternativa.	

- Clase del dominio del nomenclador de monedas

<b>Nombre:</b>	NomMoneda
<b>Tipo de clase:</b>	Dominio
Descripción	
Permite el acceso a la tabla de la base de datos NomMoneda, que es un nomenclador que contiene todas las monedas existentes y la información de cada una de ellas.	

- Clase del dominio de tasa

<b>Nombre:</b>	DatTasa
<b>Tipo de clase:</b>	Dominio
Descripción	
Permite el acceso a la tabla de la base de datos DatTasa, que contiene todas las tasas existentes y la información referente a ellas.	

- Clase del dominio del nomenclador de tipos de tasa

<b>Nombre:</b>	NomTipoTasa
<b>Tipo de clase:</b>	Dominio
Descripción	
Permite el acceso a la tabla de la base de datos NomTipoTasa, que es un nomenclador que contiene todos los tipos de tasas existentes.	

- Clase del dominio de moneda entidad

<b>Nombre:</b>	DatMonedaEntidad
<b>Tipo de clase:</b>	Dominio
Descripción	

Permite el acceso a la tabla de la base de datos DatMonedaEntidad, que contiene las referencias de las monedas que utilizará la entidad.

- Clase del dominio del nomenclador de tipos de billetes y monedas

<b>Nombre:</b>	NomTiposBilletesMonedas
<b>Tipo de clase:</b>	Dominio
<b>Descripción</b>	
Permite el acceso a la tabla de la base de datos NomTiposBilletesMonedas, que es un nomenclador que contiene todos los tipos de billetes y monedas, así como la información referente a cada tipo.	

- Clase del dominio de desglose de monedas

<b>Nombre:</b>	DatDesgloseMoneda
<b>Tipo de clase:</b>	Dominio
<b>Descripción</b>	
Permite el acceso a la tabla de la base de datos DatDesgloseMoneda, que contiene las referencias de los tipos de billetes y monedas por los que se han desglosado las monedas.	

- Clase del dominio de cuentas reevaluadas

<b>Nombre:</b>	DatCuentasReevaluadas
<b>Tipo de clase:</b>	Dominio
<b>Descripción</b>	
Permite el acceso a la tabla de la base de datos DatCuentasReevaluadas, que contiene los datos de las cuentas reevaluadas.	

- Clase del dominio de reevaluaciones

<b>Nombre:</b>	DatReevaluacion
<b>Tipo de clase:</b>	Dominio
<b>Descripción</b>	
Permite el acceso a la tabla de la base de datos DatReevaluacion, que contiene los datos de las	

reevaluaciones realizadas.

### 3.7. Métricas para la evaluación del diseño

Cuando los modelos de diseño aumentan en tamaño y complejidad resulta beneficioso aplicar métricas para verificar la calidad y objetividad del mismo. La visión objetiva de un modelo de diseño Orientado a Objetos (OO) se obtiene a través de la adecuada aplicación de métricas, cuyo resultado aporta el componente cualitativo en la evaluación que se hace. Dentro del diseño OO, se definen nueve características medibles: tamaño, complejidad, acoplamiento, suficiencia, integridad, cohesión, originalidad, similitud y volatilidad. (24)

Las métricas buscan comprobar, de manera cuantitativa, en qué grado el software posee las distintas características que definen la calidad de un producto software, de ahí que su utilización sea una fase importante dentro de la evaluación del diseño. Entre las referenciadas por Pressman (10), se han seleccionado las que se aplican con mayor facilidad dentro del sistema en cuestión, considerando que este estudio brinda un modelo sencillo de implementar y que a su vez cubre los principales atributos de calidad de software. Lo referido al aseguramiento de la calidad es la principal razón por la que se conciben estas métricas. Los atributos de calidad que se consideran en las evaluaciones realizadas son:

1. *Responsabilidad*. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
2. *Complejidad de implementación*. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
3. *Reutilización*. Consiste en el grado de reutilización de presente en una clase o estructura de clase, dentro de un diseño de software.
4. *Acoplamiento*. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, esta muy ligada a la característica de Reutilización.
5. *Complejidad del mantenimiento*. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
6. *Cantidad de pruebas*. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, modulo, clase, conjunto de clases, etc.) diseñado.

La valoración de las actividades realizadas dentro del modelo de diseño se ha realizado con la aplicación de las métricas siguientes:

- **Tamaño operacional de clase (TOC).**

Esta métrica expresa el número de métodos asignados a una clase. El impacto que tiene su aplicación en los atributos de calidad mencionados se evidencia con mayormente en responsabilidad, complejidad y reutilización. La relación está expresada porque el aumento de la cantidad de métodos de la clase es directamente proporcional a la responsabilidad y complejidad de la clase, lo que supone un aumento de los mencionados indicadores. De igual manera, se traduce en una reducción en la reutilización que pueda tener la misma.

- **Relaciones entre clases (RC)**

Métrica que está dada por el número de relaciones de uso de una clase con otras. El aumento de las RC se traduce en un aumento del acoplamiento, complejidad de mantenimiento, cantidad de pruebas que requiere la clase; al tiempo, que, de la misma forma que la métrica anterior, implica una reducción en la reutilización de la clase.

### 3.7.1. Resultados de la evaluación del diseño según las métricas aplicadas.

En este aspecto se muestra un resumen de la aplicación de las métricas, para una información más profunda sobre estos temas deben consultarse los anexos correspondientes (Anexo 8 y Anexo 9), que aparecen al final de este documento,

#### Instrumento de evaluación de la Métrica Tamaño operacional de clase (TOC)

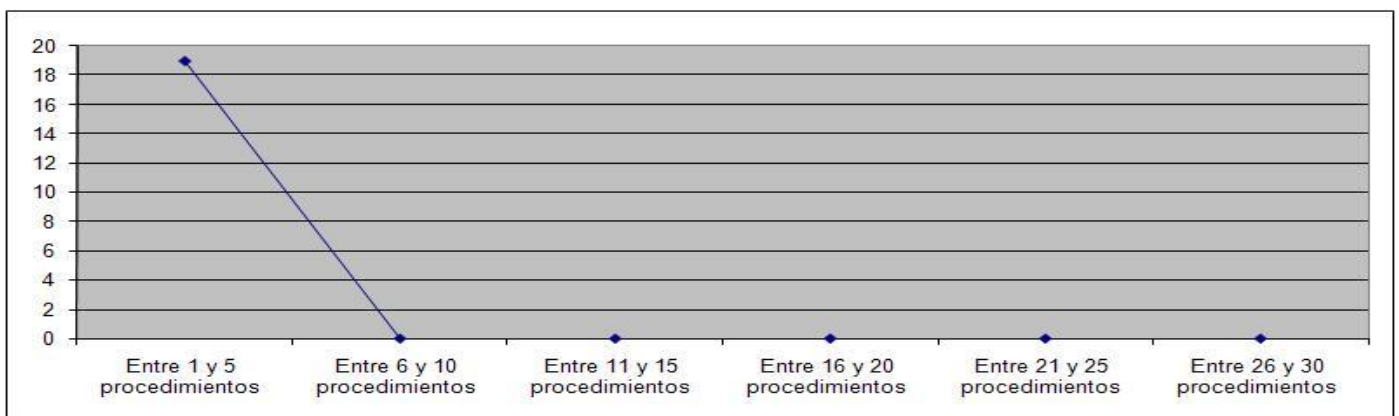


Figura 3.47- Representación de los resultados obtenidos (agrupados por de intervalos)

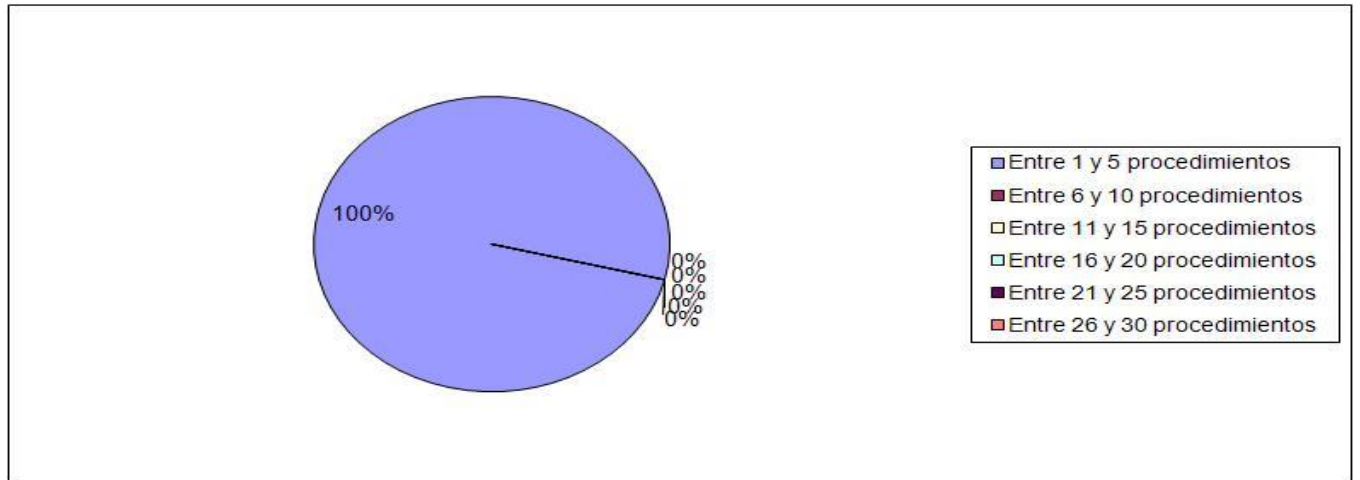


Figura 3.48- Representación del % que representan los intervalos definidos.

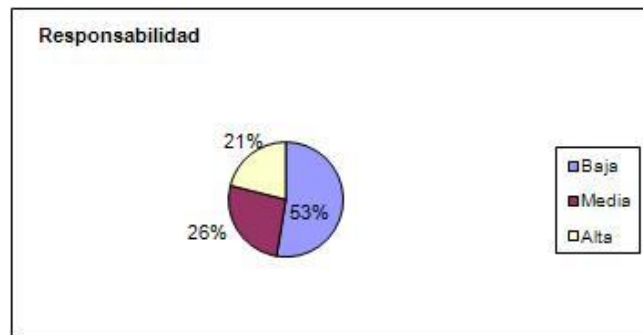


Figura 3.49- Repercusión de la métrica TOC en el atributo de responsabilidad

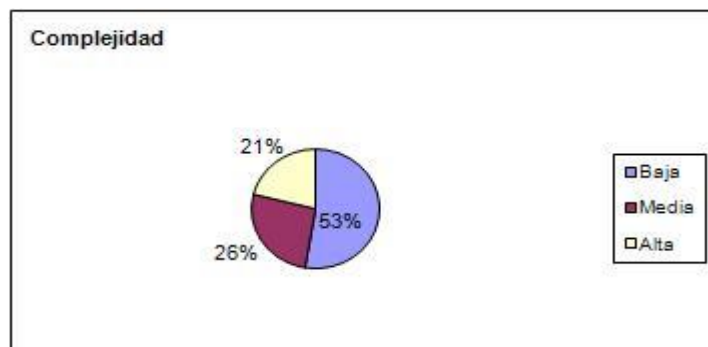


Figura 3.50- Repercusión de la métrica TOC en el atributo de complejidad



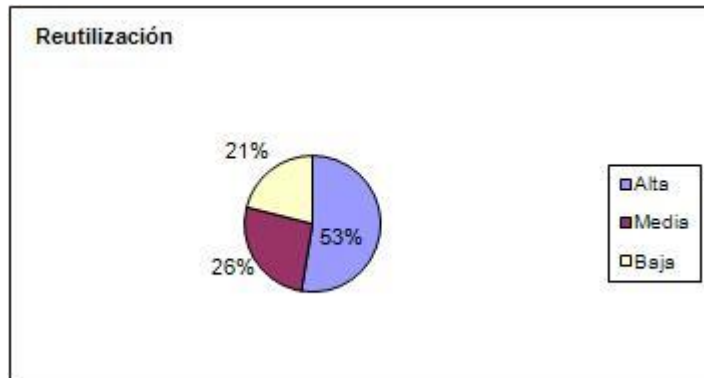


Figura 3.51- Repercusión de la métrica TOC en el atributo de reutilización

Luego de la interpretación de los resultados arrojados por la evaluación de la métrica TOC se puede decir que los mismos son satisfactorios y se incluyen dentro del rango calidad aceptable. El 79% de las clases se encuentra en el rango de las evaluaciones positivas en los atributos sobre los que incide esta métrica (responsabilidad, complejidad y reutilización).

**Instrumento de evaluación de la Métrica Relaciones entre clases (RC)**

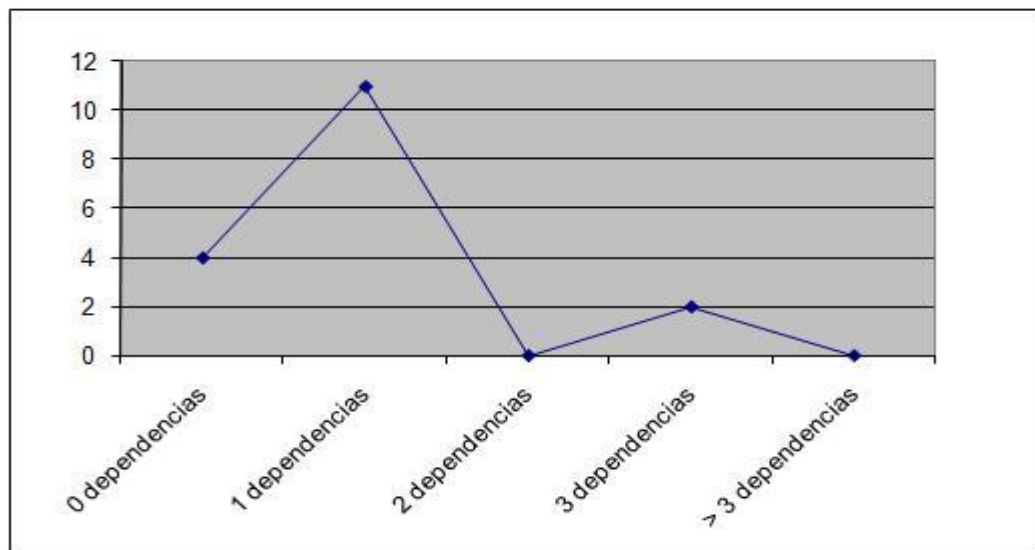


Figura 3.52- Representación de los resultados obtenidos (agrupados por de intervalos)

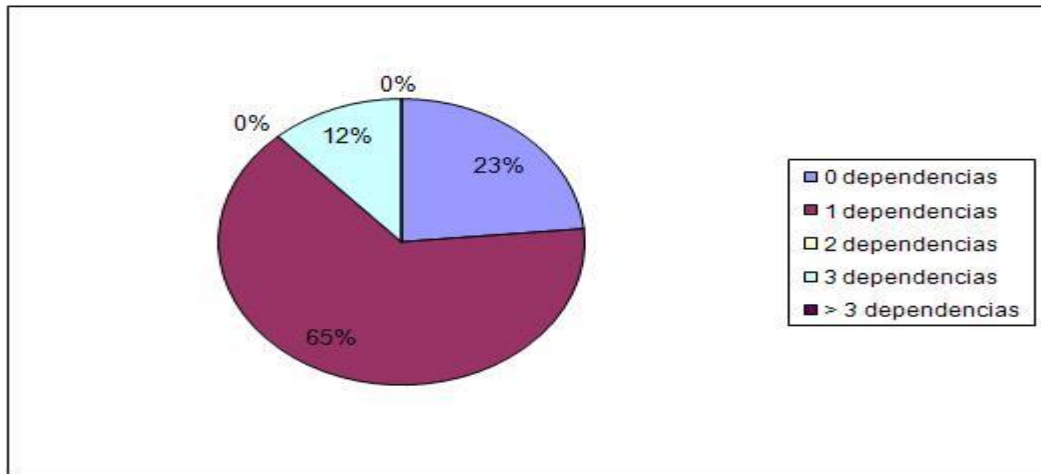


Figura 3.53- Representación de los resultados obtenidos (agrupados por de intervalos)

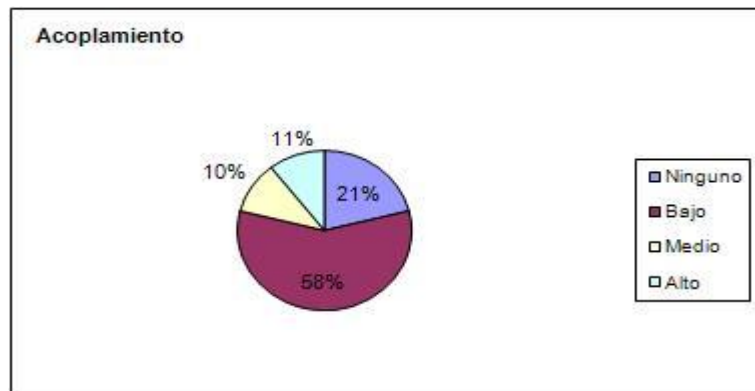


Figura 3.54- Repercusión de la métrica TOC en el atributo de acoplamiento

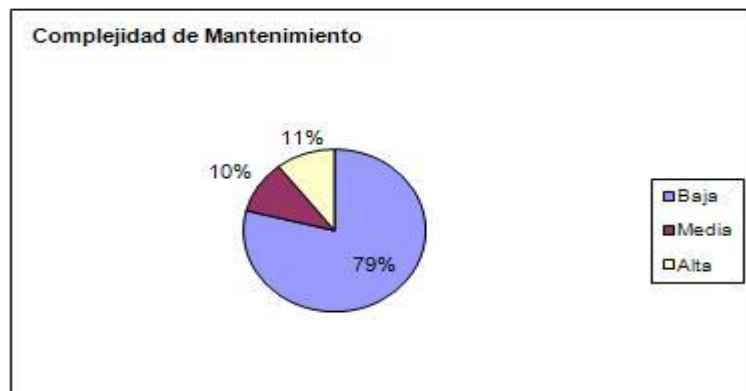


Figura 3.55- Repercusión de la métrica TOC en el atributo de complejidad de mantenimiento

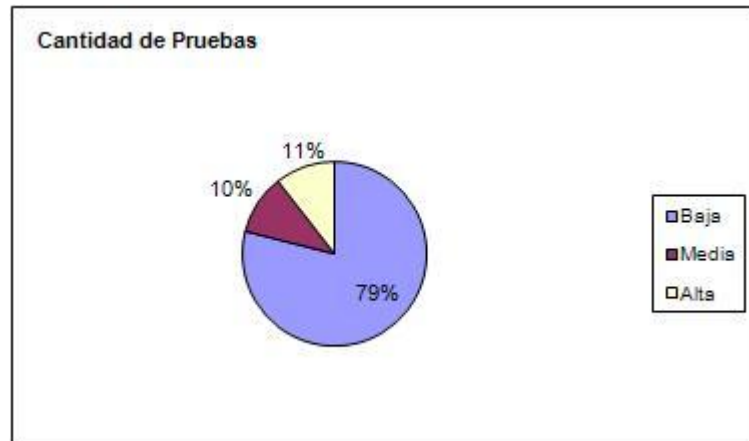


Figura 3.56- Repercusión de la métrica TOC en el atributo de cantidad de pruebas

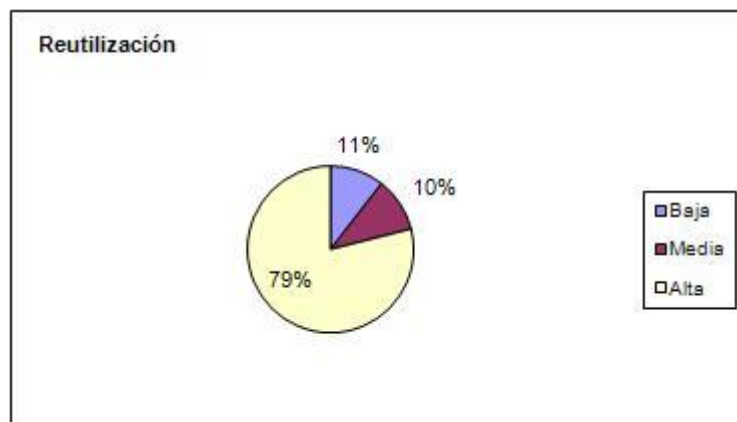


Figura 3.57- Repercusión de la métrica TOC en el atributo de reutilización

El análisis de los resultados obtenidos en la evaluación la métrica RC evidencia que es posible concluir que el diseño del subsistema Multimoneda tiene una calidad admisible. Para fundamentar lo anterior, se puede mencionar que el 88 % de las clases posee menos de 3 dependencias de otras clases. A este resultado se le puede anexar que: el 21% de las clases no tienen acoplamiento con otras y el 68% de las clases tienen un bajo o medio acoplamiento con otras, por lo que se puede decir que este es un índice factible para a este atributo. Igualmente los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 89 % de las clases.

Para resumir los resultados que se desglosan con anterioridad se puede decir que los atributos afectados por cada una de las métricas tuvieron resultados favorables que superan el 75% en todos los casos.

### 3.8. Conclusiones

La terminación de este capítulo es un paso importante a la implementación del sistema, ya que se cuenta con el modelo del diseño que sirve como esquema para la implementación del sistema el cual se esfuerza en conservar la estructura del sistema definida en esta actividad. El diseño del software sirve como fundamento para continuar con el desarrollo del software y de la ingeniería del software. Sin un diseño, se corre el riesgo de construir un sistema inestable y cuya calidad no puede evaluarse hasta muy avanzado el proceso, sin tiempo suficiente y con mucho dinero gastado en él. Se muestran, además, otros elementos significativos que conforman la base para el desarrollo del subsistema.

## CONCLUSIONES



Una vez terminado este trabajo de diploma se puede concluir que se desarrollaron todas las tareas, a fin, de cumplir los objetivos propuestos. Esto fue posible por la integración de los conocimientos adquiridos durante la investigación que precede este trabajo.

Los resultados de esta tesis describen los principales pasos a seguir durante el proceso de identificación de funcionalidades así como el diseño de las mismas, encaminadas a la obtención de una solución adecuada para tratamiento de la multimoneda en el sistema ERP-Cuba. El mismo tiene un alto impacto en el desarrollo de software de este tipo pues incorpora mejoras que facilitarán el trabajo de los usuarios ya que los requerimientos identificados cumplen con las exigencias del cliente.

Esta propuesta exhibe valor técnico, dentro de lo que es factible mencionar que se han incorporado los principios por los que se mide la factibilidad de un diseño de software, por ejemplo la utilización de patrones que posibilita la reutilización, garantizando la sostenibilidad y mantenimiento del sistema.

En general, se propone una solución novedosa, donde su importancia radica en la centralización de los elementos de multimoneda en un subsistema que proporcionó una mayor reutilización de sus funcionalidades dentro de la solución. Esto fue posible ya que se integraron todos los conceptos que inciden sobre una gestión multimoneda dentro de un único componente, permitiendo el ahorro (tiempo y recursos). De igual forma, la solución propuesta incluye en su concepción todas las consideraciones que, en este sentido, impone la economía cubana.

Es importante agregar que, la solución Multimoneda del sistema CedruX implementada a partir de lo que se refleja en este documento, se encuentra en fase de prueba en seis entidades cubanas y durante poco más de dos meses que lleva desplegada a reportado un número bajo de incidencias. Esto sirve aval para la solución que se propone a través del presente trabajo de diploma.

## RECOMENDACIONES



Al concluir el presente trabajo de diploma, amén de considerar cumplidos los objetivos trazados en el mismo, se recomienda:

- Considerar la propuesta de concentrar todas las funcionalidades en una solución compacta que brinde servicios para su aplicación en problemas similares.

## BIBLIOGRAFÍA

1. **Brito, Ing. Henry Raúl González.** Informática-Habana. [En línea] Octubre de 2006. [Citado el: 8 de Diciembre de 2008.] [http://www.informaticahabana.com/evento\\_virtual/files/SWL11.pdf](http://www.informaticahabana.com/evento_virtual/files/SWL11.pdf).
2. **Bailey, Cristian E. R.** Monografías.com. *Metodología para la selección de sistemas empresariales y su implementación.* [En línea] [Citado el: 17 de Diciembre de 2008.] <http://www.monografias.com/trabajos59/metodologia-seleccion-sistemas-empresariales/metodologia-seleccion-sistemas-empresariales.shtml>.
3. **JS Development.** JS, SOLUCIONES INFORMATICAS PARA LA EMPRESA. *sitio Web JS Development.* [En línea] [Citado el: 29 de Abril de 2009.] <http://www.jsd.es/software-erp/3-47-48-47.htm>.
4. **Ríos, José Carlos del Toro.** *Impacto de la dualidad monetaria en la solución.* Ciudad de La Habana : s.n., 2008.
5. **Microsoft Corporation.** Microsoft Dynamics. *sitio Web de Microsoft Dynamics.* [En línea] [Citado el: 20 de Enero de 2009.] <http://www.microsoft.com/latam/dynamics/introduccion.mspx>.
6. —. Sinca. *Sinca.* [En línea] 2007. [Citado el: 4 de Marzo de 2009.] <http://www.sinca.com/LinkClick.aspx?fileticket=MB9ZD9%2f1zgU%3d&tabid=88&mid=454>.
7. **Daniel Caraballo, Mario Madera, Marcelo Odin.** *ESTUDIO DEL OPEN/FREE (GNU/LINUX) COMO PLATAFORMA DE SERVICIOS DE RED EN ENTORNOS EMPRESARIALES.* Facultad de Ingeniería en Computación, Universidad de la República. Montevideo, Uruguay : s.n., 2004. Proyecto de Grado.
8. **Ríos, José Carlos del Toro.** *Especificaciones para la implementación de la Multimoneda y la moneda alternativa en la solución.* Ciudad de La Habana : s.n., 2008.
9. *Redimensionando la dualidad monetaria.* **Alejandro, Pavel Vidal.** Ciudad de La Habana : s.n., agosto de 2007, BOLETÍN CUATRIMESTRAL:ECONOMÍA Y GERENCIA EN CUBA: AVANCES DE INVESTIGACIÓN. Sitio web: <http://www.uh.cu/centros/ceec>.
10. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.* Quinta edición. s.l. : McGraw-Hill Companies, 2002. pág. 640. ISBN: 8448132149.
11. **Lowe, D. Hall, W.** *Hypermedia and the web: an engineering approach.* s.l. : John Wiley, 1999.

12. **María José Escalona, Nora Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web -Un estudio comparativo-*. Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla. Sevilla : s.n., 2002.
13. *Guest Editors Introduction: RE 02--A Major Step toward a Mature Requirements Engineering Community.* **Eric Dubois, Klaus Pohl.** 1, s.l. : I E E E C o m p u t e r S o c i e t y, Enero de 2003, Software, IEEE, Vol. 20, págs. 14-15. ISSN: 0740-7459.
14. **Ivar Jacobson, James Rumbaugh, Grady Booch.** *El proceso unificado de desarrollo de software.* [ed.] Addison-Wesley. Madrid : Pearson, 2000. ISBN 84-7829-036-2.
15. **IBM OOTC.** *Developing Object Oriented Software.* s.l. : Prentice-Hall, 1997.
16. *Lectures Notes of Requirements Elicitation.* **Raghavan, S., Zelesnik, Ford, G.** 1994. Material educativo.
17. **Andriano, Ing. Natalia Valeria.** *Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación.* Facultad de Informática, Universidad Nacional de La Plata. La Plata, Argentina : s.n., 2006. pág. 151, Tesis de Maestría.
18. **Carrasco, Luis de Salvador.** Luis de Salvador:Página personal. [En línea] [Citado el: 13 de Marzo de 2009.] [http://www.luisdesalvador.com/Oposicion/T020\\_023\\_RequisitosSoftware.pdf](http://www.luisdesalvador.com/Oposicion/T020_023_RequisitosSoftware.pdf).
19. **Sommerville, Ian.** *Ingeniería de Software.* Sexta edición. s.l. : Addison-Wesley, 2002. ISBN: 970-26-0206-8.
20. **MILESTONE, C.** [En línea] [Citado el: 20 de Enero de 2009.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
21. **Larman, Craig.** *UML y Patrones, introducción al análisis orientado a objetos.* México DF : Prentice Hall Hispanoamericana, S.A, 1999. ISBN 970-1 7-0261-1.
22. **Visual Paradigm.** Increase productivity, communication, and collaboration using UMLvisual modeling platform. *Visual Paradigm.* [En línea] <http://www.visual-paradigm.com/>.
23. **Instituto de Ingeniería Eléctrica (IIE).** Instituto de Ingeniería Eléctrica. *sitio Web Instituto de Ingeniería Eléctrica de la Universidad de la República.* [En línea] 2007. [Citado el: 20 de Febrero de 2009.] <http://iie.fing.edu.uy>.

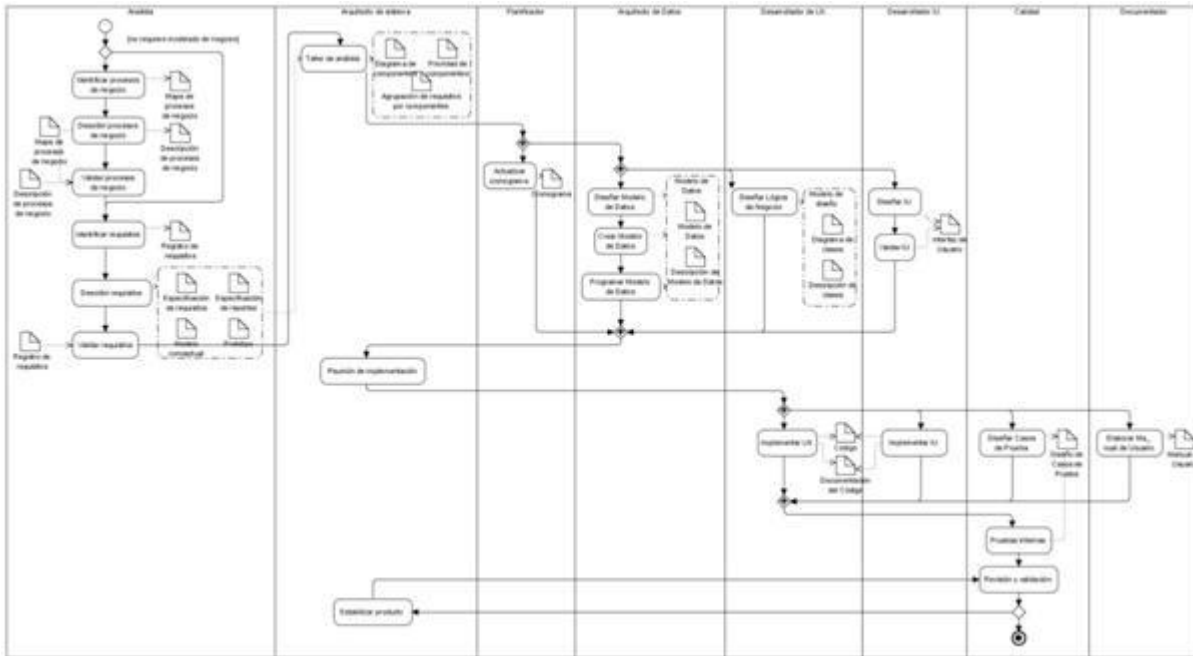


24. **Torres, Manuel Lagos.** El Rincón del Programador. *sitio Web El Rincón del Programador*. [En línea] 28 de Diciembre de 2002. [Citado el: 10 de Abril de 2009.]  
<http://www.elrincondelprogramador.com/default.asp?pag=infoAutor.asp>.
25. **Marcello Visconti, Hernán Astudillo.** Departamento de Informática UTFSM. *Sitio Web de la Universidad Técnica Federico Santa María*. [En línea] [Citado el: 28 de Abril de 2009.]  
<http://portal.inf.utfsm.cl/>.
26. **Whitmire, Scott.** *Object-Oriented Design Measurement*. s.l. : John Wiley & Sons, 1997.
27. **Aguilar, Ing. Juan Carlos Egoavil.** ACITSI-Soluciones Informáticas. [En línea] [Citado el: 17 de Diciembre de 2008.] <http://www.acitsi.com>.
28. **Molina, Lic. Gloria Cabrero.** Instituto Nacional de Estadística y Geografía (INEGI). [En línea] Agosto de 2001. [Citado el: 7 de Enero de 2009.]  
<http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/fabrica/erp/index.html>.
29. **Vera, Ángelo Benvenuto.** *Implementación de Sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC*. Departamento de Contabilidad y Auditoría, Facultad De Ciencias Económicas y Administrativas, Universidad de Concepción. 2006. ISSN 0718-4662.
30. *La encrucijada de la dualidad monetaria.* **Alejandro, Pavel Vidal.** 216, Ciudad de La Habana : s.n., Julio-Agosto de 2008, Nueva Sociedad. Sitio web <[www.nuso.org](http://www.nuso.org)>. ISSN: 0251-3552.
31. **Dina Yaksilik Torres Saquipova, Carlos de la Rosa Hernández.** *Análisis y Diseño de los módulos Inventario y Administración del proyecto ERP Cubano*. Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2008. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Disponible en: [http://bibliodoc.uci.cu/TD/TD\\_1385\\_08.pdf](http://bibliodoc.uci.cu/TD/TD_1385_08.pdf) .
32. *Del ERP al comercio electrónico: rentabilidad y progreso.* **Delgado, Jesús Megal.** 204, Estrategia financiera.
33. **Sinca Soluciones Integrales.** Sinca. *Sinca*. [En línea] 2009. [Citado el: 4 de Marzo de 2009.]  
<http://www.sinca.com/Soluciones/ERP/Financiero/tabid/88/Default.aspx>.
34. **J., Lizka Johany Herrera.** Monografías. *Monografías*. [En línea] [Citado el: 5 de Marzo de 2009.] <http://www.monografias.com/trabajos6/resof/resof2.shtml#teec>.

35. **Rojas, M.C. Juan Carlos Olivares.** Instituto Tecnológico de Morelia. *Instituto Tecnológico de Morelia*. [En línea] [Citado el: 5 de Marzo de 2009.]  
[http://antares.itmorelia.edu.mx/~jcolivar/documents/dp\\_requerimientos.ppt..](http://antares.itmorelia.edu.mx/~jcolivar/documents/dp_requerimientos.ppt..)
36. **Pan, D., Zhu, D., Johnson, K.** *Requirements Engineering Techniques*. Department of Computer Science, University of Calgary. 2001. Internal Report.
37. **Open Bravo.** Open Bravo wiki. *sitio Web de la comunidad open bravo*. [En línea] [Citado el: 13 de Febrero de 2009.]  
[http://wiki.openbravo.com/wiki/User\\_Manual\\_2.40/es#Configuraci.C3.B3n\\_General](http://wiki.openbravo.com/wiki/User_Manual_2.40/es#Configuraci.C3.B3n_General).
38. **González, Ing. Guillermo.** *Tecnologías Web Modelo Vista Controlador (MVC)*. Facultad Politécnica, Universidad Nacional de Asunción. Asunción, Paraguay : s.n., 2008.
39. **Sinapsis tecnologías.** Sinapsis Tecnologías. *sitio Web de Sinapsis tecnologías*. [En línea] [Citado el: 16 de Abril de 2009.] <http://www.sinapsistecnologias.es/files/inf-sinapsiserp.pdf>.
40. **Microsoft Corporation.** Microsoft. *Microsoft*. [En línea] 2009. [Citado el: 4 de Marzo de 2009.] <http://www.microsoft.com/latam/dynamics/introduccion.msp.x>.

## ANEXOS

### Anexo 1- Flujo de Actividades del Modelo de Desarrollo



### Anexo 2- Tabla comparativa entre técnicas para obtención de requerimientos

Técnica	Ventajas	Desventajas
Entrevistas y Cuestionarios	<p>Mediante ellas se obtiene una gran cantidad de información correcta a través del usuario.</p> <p>Pueden ser usadas para obtener un pantallazo del dominio del problema.</p> <p>Son flexibles.</p> <p>Permiten combinarse con otras técnicas.</p>	<ul style="list-style-type: none"> <li>- La información obtenida al principio puede ser redundante o incompleta.</li> <li>- Si el volumen de información manejado es alto, requiere mucha organización de parte del analista, así como la habilidad para tratar y comprender el comportamiento de todos los involucrados.</li> </ul>

<b>Lluvia de Ideas</b>	<p>Los diferentes puntos de vista y las confusiones en cuanto a terminología, son aclaradas por expertos.</p> <p>Ayuda a desarrollar ideas unificadas basadas en la experiencia de un experto.</p>	<ul style="list-style-type: none"> <li>- Es necesaria una buena compenetración del grupo participante.</li> </ul>
<b>Prototipos</b>	<p>Ayudan a validar y desarrollar nuevos requerimientos.</p> <p>Permite comprender aquellos requerimientos que no estén muy claros y que sean de alta volatilidad.</p>	<ul style="list-style-type: none"> <li>- El cliente puede llegar a pensar que el prototipo es una versión del software que será desarrollado.</li> <li>- A menudo, el desarrollador hace compromisos de implementación con el objetivo de acelerar la puesta en funcionamiento del prototipo</li> </ul>
<b>Análisis Jerárquico</b>	<p>Permite determinar el grado de importancia de cada requerimiento.</p> <p>Ayuda a identificar conflictos en los requerimientos.</p> <p>Muestra el orden en que deben ser implementados los requerimientos.</p>	<ul style="list-style-type: none"> <li>- Debe construirse un estándar claro de evaluación, que incluya la participación del cliente.</li> </ul>
<b>Casos de Uso</b>	<p>Representan los requerimientos desde el punto de vista del usuario.</p> <p>Permiten representar más de un rol para cada afectado.</p> <p>Identifica requerimientos estancados, dentro de un conjunto de requerimientos.</p>	<ul style="list-style-type: none"> <li>- En sistemas grandes, toma mucho tiempo definir todos los casos de uso.</li> <li>- El análisis de calidad depende de la calidad con que se haya hecho la descripción inicial.</li> </ul>

### Anexo 3- Ejemplo del reporte Historial de tasas de una moneda en un período definido

#### Reporte del historial de tasas de una moneda durante un periodo

Entidad: ICIDFecha: 27/04/09Periodo Inicio: 10/04/09Fin: 26/04/09

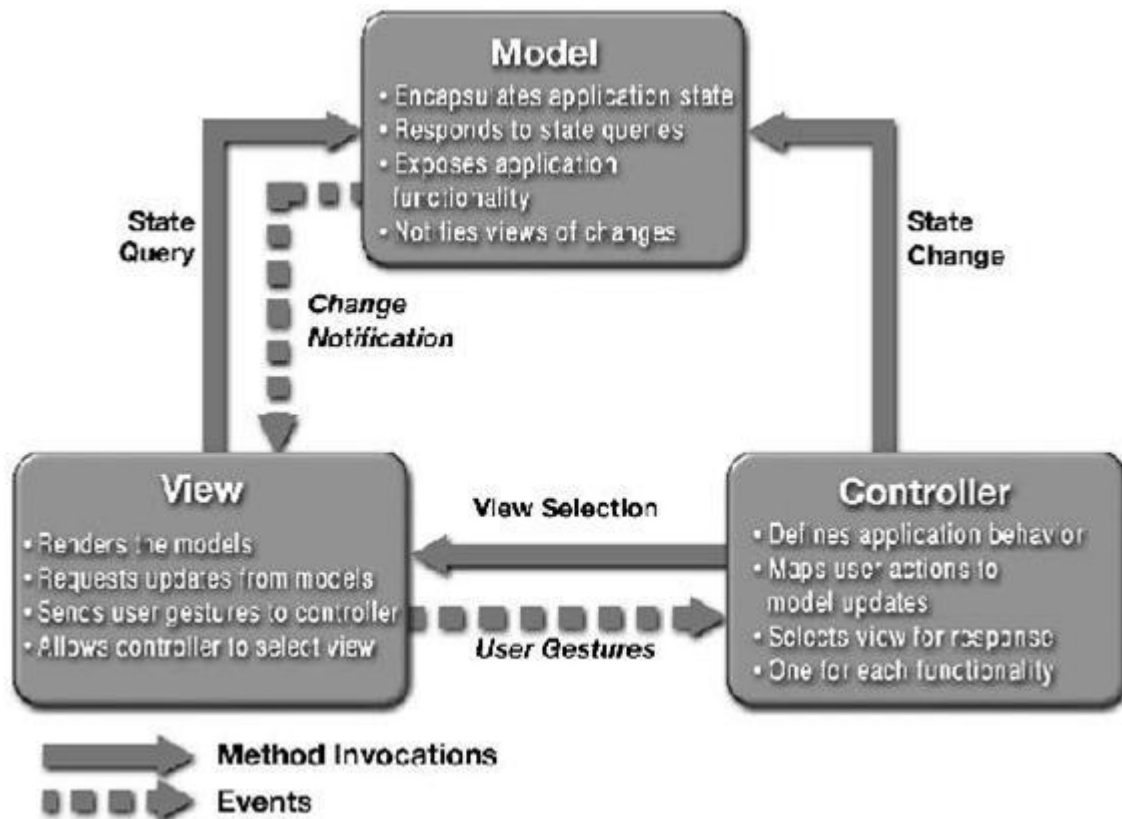
Siglas	Moneda	Tasa directa	Tasa indirecta	Fecha inicio	Fecha fin	Tipo de tasa
EUR	Euro	1.33950	1.16135	10/04/2009	10/04/2009	Cierre del día
EUR	Euro	1.33950	1.16136	11/04/2009	11/04/2009	Cierre del día
EUR	Euro	1.23960	1.16137	12/04/2009	12/04/2009	Cierre del día
EUR	Euro	1.23961	1.16138	13/04/2009	13/04/2009	Cierre del día
EUR	Euro	1.23962	1.16139	14/04/2009	14/04/2009	Cierre del día
EUR	Euro	1.23963	1.16140	15/04/2009	15/04/2009	Cierre del día
EUR	Euro	1.23964	1.16141	16/04/2009	16/04/2009	Cierre del día
EUR	Euro	1.23965	1.16142	17/04/2009	17/04/2009	Cierre del día
EUR	Euro	1.23966	1.16143	18/04/2009	18/04/2009	Cierre del día
EUR	Euro	1.23967	1.16144	19/04/2009	19/04/2009	Cierre del día
EUR	Euro	1.23968	1.16145	20/04/2009	20/04/2009	Cierre del día
EUR	Euro	1.23969	1.16146	21/04/2009	21/04/2009	Cierre del día
EUR	Euro	1,15554	1.16147	22/04/2009	22/04/2009	Cierre del día
EUR	Euro	1.23971	1.16148	23/04/2009	23/04/2009	Cierre del día
EUR	Euro	1.23970	1.16149	24/04/2009	24/04/2009	Cierre del día
EUR	Euro	1.23972	1.16150	25/04/2009	25/04/2009	Cierre del día
EUR	Euro	1.16135	1.23958	26/04/2009	26/04/2009	Cierre del día

## Anexo 4- Ejemplo del Reporte diario de tasas

**Reporte diario de Tasas**Entidad: Empresa de envases Rafael TrejoFecha: 06/05/09

<b>Siglas</b>	<b>Moneda</b>	<b>Tasa directa</b>	<b>Tasa indirecta</b>
CAD	Dólar Canadiense	1, 30713	1, 30713
GBP	Libra Esterlina	1,35145	1, 35145
MXN	Peso Mexicano	1,581.975	1, 581975
CHF	Franco Suizo	1,25962	1, 25962
EUR	Euro	1,19898	1,19898
DKK	Corona Danesa	6,20743	6,20743
NOK	Corona Noruega	7,24827	7,24827
SEK	Corona Sueca	8,81442	8,81442
JPY	Yen Japonés	110,33915	110,33915
CUP	Peso Cubano	1, 00000	1,00000
USD	Dólar Estadounidense	1, 11628	1,11628

## Anexo 5- Representación del patrón arquitectónico



### Anexo 6- Tabla de relación de los patrones GRASP.

Nombre	Solución	Problema que resuelve
Experto	Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.	¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?
Creador	Asignarle a una clase la responsabilidad de crear una instancia de otra clase en casos diferentes.	¿Quién debería ser responsable de crear una nueva instancia de alguna clase?




Bajo acoplamiento	Asignar una responsabilidad para mantener bajo acoplamiento.	¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?
Alta cohesión	Asignar una responsabilidad de modo que la cohesión siga siendo alta.	¿Cómo mantener la complejidad dentro de límites manejables?
Controlador	Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una de las clases.	¿Quién debería encargarse de atender un evento del sistema?
Polimorfismo	Cuando por el tipo varían las alternativas o comportamientos afines, las responsabilidades del comportamiento se asignarán mediante operaciones polimórficas a los tipos en que el comportamiento presenta variantes.	¿Cómo manejar las alternativas basadas en el tipo? ¿De qué manera crear componentes de software conectables?
Fabricación Pura	Asignar un conjunto altamente cohesivo de responsabilidades a una clase artificial que no representa nada en el dominio del problema.	¿A quién asignar la responsabilidad cuando se está desesperado y no se quiere violar los patrones Alta Cohesión y Bajo Acoplamiento?
Indirección	Se asigna la responsabilidad a un objeto intermedio para que los componentes se terminen directamente acoplados.	¿A quién se asignarán las responsabilidades a fin de que se cumpla el acoplamiento directo? ¿De qué manera se desacoplarán los objetos de modo que se obtenga un Bajo Acoplamiento y se obtenga un alto potencial de reutilización?



No Hables con Extraños	Se asigna la responsabilidad a un objeto directo del cliente para que colabore con un objeto indirecto, de modo que el cliente no necesite saber nada del objeto indirecto	¿A quién asignar las responsabilidades para evitar conocer las estructuras de los objetos indirectos?
------------------------	--	---

## Anexo 7 – Estereotipos para el diseño de aplicaciones Web

### Estereotipos para las Clases en diagramas para aplicaciones WEB (Herramienta: visual Paradigm)

Estereotipo	Descripción
 Server Page	Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor
 Client Page	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, scripts del lado del cliente, islas de datos, etc.
 Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page). Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.

### Estereotipos para las Relaciones entre las Clases

<b>Link</b>	Representa un apuntador desde una “client page” hacia una “client page” o “server page”. Corresponde directamente con una etiqueta <a> (ancla) de HTML
<b>Submit</b>	Esta relación siempre se da entre una “form” y una “server page”, por supuesto, la “server page” procesa los datos que la “form” le envía (submits)
<b>Build</b>	Sirve para identificar cuales “server page” son responsables de de la creación de una “client page”. Una “server page” puede crear varias “client page”, pero una “client page”

sólo puede ser creada por una sola “server page”. Esta relación siempre es unidireccional

### Redirect

Esta es también una relación unidireccional que indica que una página Web redirige hacia otra. En caso de que la página origen sea una “client page” esta asociación corresponderá con la “META” etiqueta y valor HTTP-EQUIV de “Refresh”\*.

## Anexo 8 – Valores de la aplicación de la Métrica TOC

<b>Total de clases</b>	<b>19</b>
<b>Promedio de procedimientos</b>	<b>1,684210526</b>

### Rango de valores para la evaluación de la métrica TOC

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

### Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización)

No	Subsistema	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Multimoneda	TiposBilletesMonedasModel	4	Alta	Alta	Baja
2	Multimoneda	MonedaModel	3	Media	Media	Media
3	Multimoneda	NomMonedaModel	3	Media	Media	Media
4	Multimoneda	MonedasEntidadModel	3	Media	Media	Media
5	Multimoneda	TasaModel	3	Media	Media	Media
6	Multimoneda	NomTipoTasaModel	4	Alta	Alta	Baja
7	Multimoneda	ContabilizarReevaluaciondeModel	1	Baja	Baja	Alta
8	Multimoneda	ReevaluacionModel	4	Alta	Alta	Baja
9	Multimoneda	CuentasReevaluadasModel	3	Media	Media	Media
10	Multimoneda	DesgloseMonedaModel	4	Alta	Alta	Baja
11	Multimoneda	NomTipoTasa	0	Baja	Baja	Alta

12	Multimoneda	NomMoneda	0	Baja	Baja	Alta
13	Multimoneda	NomTiposBilletesMonedas	0	Baja	Baja	Alta
14	Multimoneda	DatTasa	0	Baja	Baja	Alta
15	Multimoneda	DatMonedaEntidad	0	Baja	Baja	Alta
16	Multimoneda	DatMoneda	0	Baja	Baja	Alta
17	Multimoneda	DatDesgloseMoneda	0	Baja	Baja	Alta
18	Multimoneda	DatReevaluacion	0	Baja	Baja	Alta
19	Multimoneda	DatCuentasReevaluadas	0	Baja	Baja	Alta

## Anexo 9 - Valores de la aplicación de la Métrica RC

<b>Total de clases</b>	<b>19</b>
<b>Promedio de asociaciones de uso</b>	<b>1,105263158</b>

### Rango de valores para la evaluación de la métrica RC

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
	Categoría	Criterio
Complejidad Mant.	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$> 2 * \text{Prom.}$
	Categoría	Criterio
Reutilización	Baja	$> 2 * \text{Prom.}$
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$\leq$ Prom.
	Categoría	Criterio
Cantidad de Pruebas	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$> 2 * \text{Prom.}$

Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de mantenimiento, Cantidad de pruebas y Reutilización)

No	Subsistema	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad De Pruebas
1	Multimoneda	TiposBilletesMonedasModel	1	Bajo	Baja	Alta	Baja
2	Multimoneda	MonedaModel	1	Bajo	Baja	Alta	Baja
3	Multimoneda	NomMonedaModel	1	Bajo	Baja	Alta	Baja
4	Multimoneda	MonedasEntidadModel	1	Bajo	Baja	Alta	Baja
5	Multimoneda	TasaModel	1	Bajo	Baja	Alta	Baja
6	Multimoneda	NomTipoTasaModel	1	Bajo	Baja	Alta	Baja
7	Multimoneda	ContabilizarReevaluaciondeModel	3	Alto	Alta	Baja	Alta
8	Multimoneda	ReevaluacionModel	1	Bajo	Baja	Alta	Baja
9	Multimoneda	CuentasReevaluadasModel	1	Bajo	Baja	Alta	Baja
10	Multimoneda	DesgloseMonedaModel	1	Bajo	Baja	Alta	Baja
11	Multimoneda	NomTipoTasa	0	Ninguno	Baja	Alta	Baja
12	Multimoneda	NomMoneda	0	Ninguno	Baja	Alta	Baja
13	Multimoneda	NomTiposBilletesMonedas	0	Ninguno	Baja	Alta	Baja
14	Multimoneda	DatTasa	3	Alto	Alta	Baja	Alta
15	Multimoneda	DatmonedaEntidad	1	Bajo	Media	Alta	Baja
16	Multimoneda	DatMoneda	1	Bajo	Media	Alta	Baja
17	Multimoneda	DatDesgloseMoneda	2	Medio	Media	Media	Media
18	Multimoneda	DatReevaluacion	0	Ninguno	Baja	Alta	Baja
19	Multimoneda	DatCuentasReevaluadas	2	Medio	Media	Media	Media

## GLOSARIO DE TÉRMINOS

**Arquitectura de llamada y de retorno:** Es una estructura de programa clásica que descompone la función en una jerarquía de control en donde el programa principal invoca un número de componentes de programa que a su vez pueden invocar aún a otros componentes. (10). Caracterizada porque: Refleja la estructura del lenguaje de programación. Los componentes de este estilo arquitectónico son los objetos también considerados originalmente como instancias de tipos abstractos de datos, aunque ahora se tratan como clases. Los objetos interactúan a través de invocaciones de funciones y procedimientos.

**Cliente:** Cualquier persona que tiene un interés en el éxito del resultado del proyecto.

**Código ISO:** Es un denominador único estandarizado internacionalmente para cada moneda.

**Diferencia de cambio:** Diferencia que surge al convertir un determinado número de unidades de una moneda a otra moneda, utilizando tipos de cambio diferentes.

**Doctrine:** Es un potente y completo sistema ORM (object relational mapper) para PHP 5.2+ que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientado a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL que mantiene un máximo de flexibilidad sin requerir la duplicación del código innecesario.

**Factor de conversión:** Referido al factor que se utiliza para ajustar el importe de las moneda, cuando la tasa de cambio está expresada en relaciones diferentes a la unidad.

**Herramienta:** Proporcionan el soporte automatizado para el proceso y los métodos.

**HTTP (*Protocolo de Transferencia de Hipertexto*):** Define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web para comunicarse; protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

**HTTPS (*Protocolo Seguro de Transferencia de Hipertexto*):** Es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP. Utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado más apropiado para el tráfico de información sensible que el protocolo HTTP.

**Ingeniería de software:** La aplicación de un enfoque sistemático disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

**Inversión de control:** Del inglés Inversion of Control (IOC), es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de

programación tradicionales. En ellos la interacción se expresa de forma imperativa, haciendo llamadas a procedimientos (procedure calls) o funciones.

**Método:** Forma en que se realiza un proceso.

**Moneda alternativa:** Moneda definida en el nomenclador de monedas que se utiliza para operaciones funcionales con igual poder de compra que la moneda contable. Es característico del tema de dualidad monetaria.

**Moneda contable:** Moneda definida en el nomenclador de monedas que se utiliza para el registro contable de las operaciones, según la legislación vigente.

**Moneda original:** Es una moneda diferente a las monedas definidas como contable y alternativa, en la que está expresada una operación determinada.

**Moneda de presentación:** Es la moneda en que se presentan los Estados Financieros.

**Moneda extranjera (o divisa):** Es cualquier moneda de otro país y que tienen convertibilidad aprobada por el Banco Central de Cuba.

**Moneda funcional:** Es la moneda del entorno económico principal en el que opera la entidad.

**Moneda para el registro contable:** Moneda en que se expresarán los hechos económicos en el registro contable.

**Partidas monetarias:** Unidades monetarias mantenidas en efectivo, así como activos y pasivos que se van a recibir o pagar, mediante una cantidad fija o determinable de unidades monetarias.

**Proceso:** Acción a realizar en el desarrollo del software.

**REST ( *Representational State Transfer* ):** Es un estilo de arquitectura que ofrece un buen desempeño, escalabilidad y abstracción de recursos, donde cada petición HTTP contiene toda la información necesaria para responder a la petición, sin necesidad que el cliente ni el servidor tenga que recordar el estado de su comunicación. Ofrece recursos y representaciones variadas por ejemplo, xml, html, text, pdf, binarios, etc. Depende directamente del protocolo HTTP en un sentido de petición- respuesta utilizando un conjunto simple y pequeño de operaciones bien definidas: POST, GET, PUT y DELETE.

**SGIS:** Sistema de Gestión Integral de Seguridad. Autorización, autenticación y auditoria a las aplicaciones. Este sistema gestiona la seguridad de otros sistemas que se suscriban a el. Controla la información de cada uno de los sistemas que son provistos de seguridad o auditados en un momento determinado. La seguridad del sistema se prevé en 3 dimensiones fundamentales: Cliente Ligero, Servidor de Aplicación y Servidor de Datos.

**SOAP (*Protocolo Simple de Acceso a Datos*):** Es uno de los protocolos utilizados en los servicios Web. Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí. SOAP funciona sobre cualquier protocolo de Internet proporciona un mecanismo estándar de empaquetar mensajes generalmente HTTP.

**Software de Gestión:** El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (p. Ej.: nominas, cuentas de haberes/débitos, inventarios, etc.), han evolucionado hacia el software de sistemas de información de gestión (SIG), que accede a una o más bases de datos grandes que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (p. Ej.: el procesamiento de transacciones en puntos de ventas).

**Software:** conjunto de instrucciones que al ser ejecutadas proporcionan las características, funciones y el grado de desempeño deseados.

**Tasa directa:** Equivale a la tasa de compra que está vigente para la moneda.

**Tasa indirecta:** Equivale a la tasa de venta que está vigente para la moneda.

**Tipo de cambio de cierre:** Tipo de cambio de contado existente en la fecha del balance.

**Tipo de cambio de contado:** Tipo de cambio utilizado en las transacciones con entrega inmediata.

**Tipo de cambio:** Relación de cambio entre dos monedas.

**Valor razonable:** Importe por el cual puede ser intercambiado un activo, o cancelado un pasivo, entre partes interesadas y debidamente informadas, en una transacción realizada en condiciones de independencia mutua.